Available online at www.sciencedirect.com

Chemical Engineering Research and Design

journal homepage: www.elsevier.com/locate/cherd

Real-time adaptive sparse-identification-based predictive control of nonlinear processes

Fahim Abdullah[a], *Panagiotis D. Christofides*[a,b,*]

[a] *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA*

[b] *Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA*

ARTICLE INFO

Article history:
Received 29 June 2023
Received in revised form 9 July 2023
Accepted 10 July 2023
Available online 13 July 2023

Keywords:
Nonlinear processes
Sparse identification
Adaptive control
Model predictive control

ABSTRACT

This study introduces a sparse identification-based model predictive control (MPC) framework that incorporates on-line updates of the sparse-identified model to account for nonlinear dynamics and model uncertainty in process systems. The methodology involves obtaining a nonlinear first-order ordinary differential equation model using sparse identification for nonlinear dynamics (SINDy), which is integrated into two control schemes: Lyapunov-based MPC (LMPC) for achieving steady-state operation and Lyapunov-based economic MPC (LEMPC) for achieving both closed-loop stability and optimal economic performance. To improve prediction accuracy, an on-line model update scheme is proposed for the SINDy models. Specifically, an error-trigger mechanism that utilizes prediction errors and then uses the most recent process data to update the parameters of the SINDy model in real-time is designed. By incorporating the error-triggered on-line model updates in the SINDy-based LMPC and LEMPC, the dynamic performance of the process is enhanced, ensuring closed-loop stability, optimality, and smooth control actions. Following theoretical results on the boundedness of the closed-loop states and detailed discussions on the selection criteria for parameters of the error-triggered SINDy update scheme, the effectiveness of the proposed methodology is demonstrated through a chemical process example with time-varying disturbances under the LEMPC framework.

1. Introduction

Advanced process control techniques, such as model predictive control (MPC) play a crucial role in industrial applications and can leverage the recent and ongoing revolution in data-driven approaches in the science and engineering ecosystem. MPC is widely used due to its ability to handle strongly nonlinear processes with constraints, which are challenging for traditional linear control methods (Garcia et al., 1989; Mayne, 2014). MPC offers advantages such as straightforward tuning, control of systems with time delays and instability, incorporation of known constraints and multiple operating conditions, compensation for dead time, and flexibility in defining control objectives. However, a major drawback is the requirement for a suitable model to predict the future states in the real-time calculations, which can be costly and time-consuming to develop for large-scale, complex nonlinear processes using existing system identification or model reduction techniques (Brunton and Noack, 2015). The quality of process models is influenced by various

[*] Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.cherd.2023.07.011

factors such as parameter estimation, model uncertainty, assumptions made during model development, dimensionality, model structure, and computational complexity for real-time implementation (Ge and Wang, 2004; Ge et al., 2007).

To combat the difficulty of developing process models for large-scale or poorly understood processes, over the last decade, there has been a paradigm shift from first-principles modeling to data-driven modeling. Machine learning techniques are a subset of data-driven modeling techniques that have seen increasing application in modeling chemical processes when traditional first-principle models are not available. For instance, in previous works (Wu et al., 2019a,b), recurrent neural networks were utilized to construct data-driven models for nonlinear processes, which were subsequently integrated into Lyapunov-based model predictive control (MPC) to ensure stability and performance. Although machine learning algorithms have demonstrated continuous success in modeling complex systems in the large-data limit due to their large number of hyperparameters and degrees of freedom of the model, their black-box nature can hinder their advancement to deployment in practical engineering systems, even more so in safety-sensitive fields such as chemical plants (Wu and Christofides, 2021). Such models are also usually strongly restricted to the domain of training data, and it is highly inadvisable to use such models for extrapolating the dynamics of the remainder of the state space. To better capture the physics of systems, several works focused on symbolic regression, which was a successful direction but computationally intractable for large-scale systems. Hence, this idea was developed further with the concept of compressive sensing (Ozolinš et al., 2013; 2014) into a relatively modern technique known as sparse identification for nonlinear dynamics (SINDy). Since its inception, SINDy has been applied to a broad range of systems (Proctor et al., 2014; Bai et al., 2015). In the field of process systems engineering, the goal of using SINDy for building process models is that SINDy enables the direct identification of models of explicit and closed-form nonlinear first-order ordinary differential equations (ODEs) from data. These identified equations can be readily incorporated into optimization problems, including MPC. The computational cost of integrating these explicit ODE models is typically low, particularly when the models are well-conditioned, thanks to the availability of efficient differential equation solvers that use well-established integration algorithms such as 4th/5th order Runge-Kutta methods.

In the recent literature, SINDy has been implemented successfully to develop models in chemical engineering, such as the identification of reaction networks (Hoffmann et al., 2019) and the development of reduced-order models for controlling hydraulic fracturing processes (Narasingam and Kwon, 2018) and nonlinear reactors (Abdullah et al., 2021a, 2021b). The practical challenge of handling noise in sensor measurements when using SINDy was also addressed in Abdullah et al. (2022a, 2022b), where subsampling, co-teaching, and ensemble learning were used to build accurate SINDy models that captured the original nonlinear system from noisy data sets. Despite the successes in initial model building using SINDy, in practical applications, process models undergo changes over time due to various factors, including external influences (such as aging equipment, disturbances, and deployment of new operational technology) and internal factors (such as equipment fouling or catalyst deactivation). As a result, the SINDy model trained on past normal operations may not accurately predict process states in the presence of disturbances. To address this challenge, researchers have explored adaptive, robust, and event-triggered control approaches within both classical (first-principles) modeling and data-driven modeling techniques like SINDy to mitigate the impact of model uncertainty.

Bhadriraju et al. (2020) proposed the operable adaptive sparse identification of systems (OASIS) framework where multiple SINDy models are constructed for the various regions of the state-space using single, short trajectories. However, due to the low data usage per model, the SINDy models were localized and could not extrapolate the entire state-space. Hence, a feedforward neural network was used to switch between the SINDy models based on partial state measurements and estimation of the remaining states via a Kalman filter. The authors demonstrated the effectiveness of the OASIS approach by building 100 different SINDy models for various sections of the operating region with relatively small data sets of single trajectories with 100 data points sampled every 0.01 hr. However, there was no update of the SINDy models themselves in real-time. Stanković et al. (2020) modified the original formulation of SINDy to handle output measurements and actuation in addition to state measurements and also proposed highly specific library terms that are relevant for power systems, demonstrating the superior performance of the proposed SINDy algorithm on synchronous generator models. In Sarić et al. (2020), the authors further combined SINDy with the manifold boundary approximation method to build models specifically favoring the identification of power systems and conducting their stability analyses. Specifically, the proposed algorithm had low data requirements and was ideal for updating models in real-time subject to changing dynamics, and this aspect was demonstrated via model reduction using limited data when applied to a number of synchronous generator models. However, much of the work was highly tailored to power systems, especially large power systems and their transient stability analysis. The structured online learning method was proposed in Farsi and Liu (2020) where a quadratic value function was used to yield equations that were a more general form of the linear quadratic regulator with certain advantages and improvements when operating at unstable steady states in a pendulum example. Due to the quadratic formulation, the parameters of the value function could be analytically computed with a low computational cost, while SINDy was used for the model identification part. However, the focus of Farsi and Liu (2020) was on the real-time update of the value function rather than the SINDy models. Similarly to Stanković et al. (2020), SINDy was generalized in Wang et al. (2022) to handle multi-input multi-output (MIMO) systems to model system outputs to sensor measurements, i.e., instead of obtaining ODEs in the states, the system output is obtained as functions of the state dynamics. Subsequently, a Kalman filter was used not to estimate the states but rather the model coefficients and update them in real-time using sensor measurements at every sampling time. Through two standard nonlinear examples, it was shown that this generalized SINDy with the Kalman filter approach could obtain better models than building a pure SINDy model using the sequential thresholded least-squares solver with similar or less amount of data.

The concept of real-time updates of SINDy models was first proposed in Quade et al. (2018), where a method for re-identification was introduced, allowing for the updating of model coefficients or the addition/removal of terms, or any combination thereof, as needed. The re-identification process was triggered by a noticeable deviation between the local Lyapunov exponent and the prediction horizon estimate, although the definition of “prediction horizon” in their study differs from its usage in the current manuscript. Although the findings of Quade et al. (2018) were mostly limited to the modeling context with no closed-loop data usage in the model updates, the methodology was recently adapted in Manzoor et al. (2022) in the modeling of ducted fan aerial vehicles (DFAVs) for closed-loop control under MPC. Specifically, an offline model was first built using deep physical knowledge of DFAVs, and the model parameters were then updated based on the proposed paradigm of Quade et al. (2018), using the same Lyapunov exponent-based trigger for the model update. The proposed methodology was shown to be able to control DFAVs, which are highly challenging to model due to their complex flow distribution, under the various cases and disturbances considered such as wind turbulence, which is of practical concern in such a setting. Although real-time update of SINDy models was also studied in Bhadriraju et al. (2019), the goal was not to update the SINDy model to process changes but rather to build the process model itself for an unchanging process step-by-step, improving the model with newer data as it becomes available. Hence, the work of Bhadriraju et al. (2019) can be considered an alternative to model building when there is no large data set from numerous simulations to build a high-fidelity SINDy model offline before deploying the controller. Finally, most recently, an MPC framework using SINDy models with updates to handle changing process conditions was investigated in Huang et al. (2023). The focus of the work was on model updates in the face of entirely unknown or first appearance process conditions and the operation of processes with multiple operating conditions. The discrete-time formulation of SINDy with actuation was used, and the greedy algorithm known as orthogonal matching pursuit was used to efficiently calculate only the matrix of changes to the model coefficients. During the model transition/update period, an elastic feedback correction method was used as a stopgap solution. The proposed error-triggered adaptive sparse identification for predictive control (ETASI4PC) method showed significant improvement over the state-of-the-art methods including SINDy without model updates as well as the aforementioned OASIS approach in the presence of large disturbances in the feed flow rate of a chemical reactor system. We note that the ETASI4PC method is a promising methodology for updating SINDy models in real-time in tracking MPC. At the moment, based on our survey of the literature, the operation of Lyapunov-based model predictive controllers, both Lyapunov-based tracking MPC (LMPC) and Lyapunov-based economic MPC (LEMPC) under SINDy models that are updated in real-time has not been investigated, which is the subject of the current manuscript. While a tracking MPC drives the state of a system to a desired set-point, economic MPC, a recent model-based control strategy, optimizes time-varying operation by considering future process states, economic objectives, and feedback. It incorporates economic factors and constraints to achieve improved process efficiency and desired closed-loop response characteristics. The potential benefits of economic MPC make it an attractive choice for industrial applications, as highlighted in studies such as Amrit et al. (2011); Huang et al. (2011); Heidarinejad et al. (2012); Ellis et al. (2014).

The rest of this manuscript is organized as follows: in Section 2, the notations, the class of nonlinear systems considered, and stability assumptions are provided. Section 3 provides a brief review of sparse identification and its implementation, followed by the design of Lyapunov-based tracking and economic MPCs using SINDy models as the predictive model. In Section 4, the error-triggering mechanism is introduced, the details of the SINDy model update procedure are given, the implementation strategy for adaptive SINDy models in LMPC and LEMPC is delineated, and rigorous closed-loop stability analyses are conducted for the two types of MPCs. Finally, a chemical reactor example is used in Section 5 to demonstrate the performance of the proposed adaptive SINDy-MPC methodology.

2. Preliminaries

2.1. Notation

If x is a vector, we denote its transpose as $x^{\top}$ and its weighted Euclidean norm as $|x|_Q$, where Q is a positive definite matrix. The standard Lie derivative $L_fV(x)$ is defined as $\frac{\partial V(x)}{\partial x}f(x)$. The operator “\” represents set subtraction, such that A\B is the set of elements $x \in \mathbb{R}^{n_x}$ that belong to A but not to B. A function $f(\cdot)$ is said to belong to the class $\mathscr{C}^1$ if it is continuously differentiable within its domain. A class $\mathscr{K}$ function is defined as a continuous function α: $[0, a) \to [0, \infty)$ that is strictly increasing and takes the value of zero only when it is evaluated at zero.

2.2. Class of systems

We examine a broad category of continuous-time nonlinear systems, characterized by the equation,

$$\dot{x} = F(x, u, w) := f(x) + g(x)u + h(x)w, \qquad x(t_0) = x_0 \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ denotes the state vector, $u \in \mathbb{R}^{n_u}$ represents the manipulated input vector, and $w \in W$ is the disturbance vector with $W := \{w \in \mathbb{R}^{n_w} \ | \ |w| \le w_m, \quad w_m \ge 0\}$. The functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are suitably smooth vector and matrix functions, respectively, with dimensions $n_x \times 1$, $n_x \times n_u$, and $n_x \times n_w$. Without loss of generality, we assume that the initial time t_0 and initial condition $f(0)$ are both equal to zero in this manuscript. Consequently, the steady-state of the nominal system of Eq. (1) is at the origin, specifically denoted as $(x_s^*, u_s^*) = (0, 0)$. Here, x_s^* and u_s^* represent the steady-state state and input vectors, respectively.

2.3. Stabilization via control Lyapunov function

Assuming noise-free state measurements and full state feedback for the nominal system described in Eq. (1), it is postulated that a stabilizing control law $u = \Phi(x) \in U$ exists, capable of exponentially stabilizing the origin of the closed-loop system mentioned in Eq. (1). According to converse Lyapunov theorems (Massera, 1956; Lin et al., 1996; Christofides and El-Farra, 2005), this implies the existence of a $\mathscr{C}^1$ control Lyapunov function $V(x)$, along with four positive constants c_1, c_2, c_3, c_4, satisfying the conditions,

$$c_1|x|^2 \le V(x) \le c_2|x|^2, \tag{2a}$$

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{2b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4|x| \tag{2c}$$

for all x in an open neighborhood D around the origin. In Eq. (2), $\dot{V}$ represents the time-derivative of the Lyapunov function, and $F(x, \Phi(x), 0)$ represents the nominal system from Eq. (1) under a candidate controller $\Phi(x)$, such as the universal Sontag control law (Lin and Sontag, 1991). Our first objective is to define a set of states ϕ_u, expressed as follows:

$$\phi_u = \left\{x \in \mathbb{R}^{n_x} \mid \dot{V}(x) = L_f V + L_g V u < -kV(x),\ u = \Phi(x) \in U,\ k > 0\right\} \cup \{0\}$$

under the controller $u = \Phi(x) \in U$, that satisfies the conditions described in Eq. (2). Subsequently, we define the closed-loop stability region Ω_ρ (Khalil, 2002) for the nominal system presented in Eq. (1) as a sublevel set of V within ϕ_u, denoted as $\Omega_\rho := \{x \in \phi_u | V(x) \leq \rho\}$, where $\rho > 0$ and $\Omega_\rho \subset \phi_u$. Furthermore, from the Lipschitz continuity property of $F(x, u, w)$ and the given bounds on u, it can be deduced that there exist positive constants N_w, L_x, L_x', L_w, and L_w' such that the following inequalities are satisfied for all x, $x' \in D$, $u \in U$, and $w \in W$:

$$|F(x, u, w)| \leq M \tag{3a}$$

$$|F(x, u, w) - F(x', u, 0)| \leq L_x|x - x'| + L_w|w| \tag{3b}$$

$$\left| \frac{\partial V(x)}{\partial x} F(x, u, w) - \frac{\partial V(x')}{\partial x} F(x', u, 0) \right| \leq L_x'|x - x'| + L_w'|w| \tag{3c}$$

3. Lyapunov-based MPC using sparse identification

In this section, the details of the sparse identification problem and its solution are provided, followed by the formulation of LMPC and LEMPC that utilize the SINDy model to predict the future states. Furthermore, closed-loop stability for the nonlinear system of Eq. (1) is discussed under the proposed LMPC and LEMPC.

3.1. *Sparse identification*

Sparse identification, a recent advancement in nonlinear system identification, has demonstrated its effectiveness in various engineering disciplines through numerous examples (Wang et al., 2011; Schaeffer et al., 2013; Ozolinš et al., 2013; Mackey et al., 2014; Brunton et al., 2014; Proctor et al., 2014; Bai et al., 2015). The objective of the Sparse Identification of Nonlinear Dynamics (SINDy) approach is to utilize discrete measurement data from a physical system to identify a first-order ordinary differential equation (ODE) of the following form:

$$\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u \tag{4}$$

where $\hat{x} \in \mathbb{R}^{n_x}$ represents the state vector of the model obtained through sparse identification, while $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are the vector fields that capture the underlying physical laws governing the system. We emphasize that the goal of using SINDy in this work is as a system identification tool, and the correct underlying physics need not necessarily be obtained exactly for the SINDy model to be accurate.

The fundamental assumption underlying SINDy is that the right-hand side of Eq. (4) typically comprises only a small number of nonlinear terms. As a result, when considering a large pool of potential nonlinear basis functions for $\hat{f}$ and $\hat{g}$, only a few terms will be active, with non-zero coefficients associated with them. This sparsity property of the candidate basis functions allows for efficient computation of the coefficients using convex optimization algorithms.

The application of SINDy begins with acquiring real-time data from the system of interest. This data can be collected through sensors in experimental or industrial setups, or generated from computer simulations based on theoretical models, such as first-principles or chemical process simulators. The collected data is then organized into two compact matrices: the data matrix X and the input matrix U,

$$X = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_{n_x}(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_{n_x}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_{n_x}(t_m) \end{bmatrix} \tag{5a}$$

$$U = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \cdots & u_{n_u}(t_1) \\ u_1(t_2) & u_2(t_2) & \cdots & u_{n_u}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \cdots & u_{n_u}(t_m) \end{bmatrix} \tag{5b}$$

where $x_i(t_\ell)$ represents the i^{th} state measurement at the ℓ^{th} sampling time, while $u_j(t_\ell)$ denotes the j^{th} input measurement at the ℓ^{th} sampling time. The indices i, j, and ℓ take values $i = 1, \ldots, n_x$, $j = 1, \ldots, n_u$, and $\ell = 1, \ldots, m$, respectively. The derivative of X, denoted as $\dot{X}$, is either directly measured or numerically estimated when direct measurement is not possible.

From the data matrices X and U, a function library matrix $\Theta(X, U)$ is constructed, containing p columns representing the nonlinear basis functions considered for the terms in $\hat{f}$ and $\hat{g}$. While polynomials and trigonometric functions are commonly used in engineering applications due to their universality, the basis set can be adapted based on performance and available knowledge of the system's structure. An example of a function library matrix that may be used as a starting point for system identification is as follows:

$$\Theta(X, U) = \begin{bmatrix} \mathbf{1} & X & \ln X & \sin(X) & e^X & U & UX^2 \end{bmatrix} \tag{6}$$

Each candidate basis function associated with a variable or row in Eq. (4) is assigned a coefficient, and these coefficients are stored in the matrix $\Xi \in \mathbb{R}^{p \times n_x}$. The sparse identification algorithm solves the equation,

$$\dot{X} = \Theta(X, U)\Xi \tag{7}$$

to calculate the coefficients, Ξ. A popular method for solving this equation is the sequential thresholded least squares (STLSQ), where a threshold value known as the sparsification knob λ is specified, and coefficients in Ξ below λ are set to zero. Specifically, the least-squares problem associated with Eq. (7) can be formulated in the following general form:

$$\Xi = \arg\min_{\Xi'} \|\dot{X} - \Theta(X, U)\Xi'\|_2 + \lambda\|\Xi'\|_1 \tag{8}$$

where Ξ' is a notational substitute for Ξ, and the second term is an L_1 regularization term that enforces sparsity of Ξ. To implement the above step, we start by defining the matrix Ξ'' to be the matrix Ξ' with all coefficients with magnitudes below λ set to zero, which is the practical implementation of the L_1 regularization term in Eq. (8). Subsequently, the problem is reduced to the form,

$$\Xi = \arg\min_{\Xi''} \|\dot{X} - \Theta(X, U)\Xi''\|_2 \tag{9}$$

which can be solved using, for example, MATLAB's built-in linear solver called with $A\backslash b$ where $A = \dot{X}$ and $b = \Theta(X, U)$. Eq. (9) is solved repeatedly until the large/nonzero coefficients (greater than λ in each iteration) converge. Due to the efficiency of linear solvers as well as the sparse structure of Θ, the coefficients converge rapidly in this step. The STLSQ method is used in the current manuscript due to not only its efficiency but also simplicity, as described above, which facilitates the necessary modifications required for real-time model updates.

Finally, the calculated coefficient values in Ξ are then used to construct the continuous-time ordinary differential equation

$$\dot{x} = \Xi^{\top}(\Theta(x^{\top}, u^{\top}))^{\top} \tag{10}$$

where $\Theta(x^{\top}, u^{\top})$ is not a matrix of data but a column vector of symbolic functions derived from the library of considered functions.

3.1.1. *Solving for individual variables*

The least-squares problem of Eq. (9) can also be solved not for the entire system but for a single variable of interest if only specific ODEs are required. From a programming perspective, it is identical whether the original full problem of Eq. (9) is solved to obtain all n_x ODEs in one computation or whether n_x problems are solved in a loop to identify the n_x ODEs. If Eq. (9) is solved individually for a variable we first define rows and columns of the relevant matrices as follows:

$$\dot{X} = [\dot{x}_1 \quad \dot{x}_2 \quad \cdots \quad \dot{x}_n] \tag{11a}$$

$$\Theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_p] \tag{11b}$$

$$\Xi = [\xi_1 \quad \xi_2 \quad \cdots \quad \xi_n] \tag{11c}$$

$$\Xi = \begin{bmatrix} \xi_1^{\top} \\ \xi_2^{\top} \\ \vdots \\ \xi_p^{\top} \end{bmatrix} \tag{11d}$$

where $\dot{x}_i \in \mathbb{R}^m$ and $\xi_i \in \mathbb{R}^p$ are the columns of $\dot{x}$ and Ξ, respectively, for $i = 1, \ldots, n_x$. The columns of Θ are denoted by $\theta_j \in \mathbb{R}^m$, while the rows of Ξ are represented by $\xi_j^{\top} \in \mathbb{R}^{n_x}$, for $j = 1, \ldots, p$. Using these notations, for the i^{th} variable, the sparse identification problem of Eq. (7) is of the form,

$$\dot{x}_i = \Theta(X, U)\xi_i \tag{12}$$

and the corresponding least-squares problem of Eq. (9) is

$$\xi_i = \arg\min_{\xi_i''} \|\dot{x}_i - \Theta(X, U)\xi_i''\|_2 \tag{13}$$

where ξ_i represents the coefficients in front of each library function for the i^{th} variable, and ξ_i'' is the notational substitute for the vector ξ_i with all coefficients with magnitudes below λ set to zero. Solving the minimization of Eq. (13) for $i = 1, \ldots, n_x$ is identical to solving the full-state problem of Eq. (9) but more advantageous in terms of formulation when performing partial SINDy modeling, i.e., model updates in real-time, as will be discussed in Section 4.2.

3.1.2. *Scaling of library functions*

In the case where the values of the functions in the library $\Theta(X, U)$ vary by orders of magnitudes for a data set, it may very likely be necessary to scale the library columns appropriately to yield a well-conditioned least-squares problem with a reasonable conditional number of $\Theta(X, U)$. When scaling the function library by a vector $\Lambda \in \mathbb{R}^p$, the sparse identification problem becomes

$$\dot{X} = \Theta(X, U)\Xi = \underbrace{\frac{\Theta(X, U)}{\Lambda}}_{\Theta_{\text{scaled}}} \underbrace{\Xi\Lambda}_{\Xi_{\text{scaled}}} \tag{14}$$

where Θ_{scaled} is the scaled library matrix with its i^{th} column divided by the i^{th} entry of Λ for $i = 1, \ldots, p$. Similarly, Ξ_{scaled} is the scaled coefficient matrix where each of its rows has been multiplied by the corresponding scalar entry of Λ. The STLSQ iteration step of Eq. (9) must now use the scaled library and coefficient matrices, i.e.,

$$\Xi_{\text{scaled}} = \arg\min_{\Xi''_{\text{scaled}}} \|\dot{X} - \Theta_{\text{scaled}}(X, U)\Xi''_{\text{scaled}}\|_2 \tag{15}$$

Since Ξ_{scaled} is used in the STLSQ step, the threshold λ does not need to be scaled as all the entries of the scaled matrix Ξ_{scaled} are already of similar orders of magnitudes. At the end of the STLSQ algorithm, once Ξ_{scaled} has been calculated, the original Ξ can be recovered by dividing every row of Ξ_{scaled} by the corresponding value in the scaling vector Λ.

3.2. *Lyapunov-based control using SINDy models*

In this section, we outline the formulation of Lyapunov-based Model Predictive Control (LMPC) and Lyapunov-based Economic Model Predictive Control (LEMPC) utilizing SINDy models for future state prediction. Initially, a SINDy model is constructed to approximate the nonlinear dynamics of the system described by Eq. (1) within the operating region Ω_ρ using data obtained from extensive open-loop simulations. Subsequently, LMPC and LEMPC are developed by leveraging SINDy models to ensure closed-loop stability for the nonlinear system represented by Eq. (1).

In this study, we update the SINDy model given by Eq. (10) to capture the nonlinear dynamics of the system described by Eq. (1) in the presence of time-varying bounded disturbances (i.e., $|w(t)| \le w_m$). Each SINDy model, denoted as $F_{si}^i(x, u)$ with $i = 1, 2, \ldots, N_T$, is updated using real-time data of closed-loop state trajectories and control actions. Here, N_T represents the total number of obtained SINDy models. We assume the existence of a set of stabilizing feedback controllers $u = \Phi_{si}^i(x) \in U$ that can render the origin of the SINDy models $F_{si}^i(x, u)$, with $i = 1, 2, \ldots, N_T$, of Eq. (10) exponentially stable within an open neighborhood $\hat{D}$ around the origin. Consequently, a $\mathscr{C}^1$ control Lyapunov function $\hat{V}(x)$ exists, satisfying the following inequalities for all x within $\hat{D}$:

$$\hat{c}_1^i|x|^2 \le \hat{V}(x) \le \hat{c}_2^i|x|^2, \tag{16a}$$

$$\frac{\partial \hat{V}(x)}{\partial x} F_{si}^i(x, \Phi_{si}^i(x)) \le -\hat{c}_3^i|x|^2, \tag{16b}$$

$$\left|\frac{\partial \hat{V}(x)}{\partial x}\right| \le \hat{c}_4^i|x| \tag{16c}$$

where $\hat{c}_1^i$, $\hat{c}_2^i$, $\hat{c}_3^i$, and $\hat{c}_4^i$ are positive constants, with $i = 1, 2, \ldots, N_T$. For simplicity, we will omit the superscript i in the symbols used to represent the SINDy models and controllers that satisfy Eq. (16) in the subsequent discussions. Similar to the approach used to characterize the closed-loop stability region Ω_ρ for the nonlinear system described by Eq. (1), we begin by characterizing a region denoted as $\hat{\phi}_u = \{x \in \mathbb{R}^n \ | \ \dot{\hat{V}}(x) < -\hat{c}_3|x|^2, u = \Phi_{si}(x) \in U\} \cup \{0\}$, from which the origin of the SINDy model given by Eq. (10) can be rendered exponentially stable under the controller $u = \Phi_{si}(x) \in U$.

The closed-loop stability region for the SINDy model given by Eq. (10) is defined as a level set of the Lyapunov function within $\hat{\phi}_u$: $\Omega_{\hat{\rho}} := \{x \in \hat{\phi}_u \ | \ \hat{V}(x) \le \hat{\rho}\}$, where $\hat{\rho} > 0$. It should be noted that $\Omega_{\hat{\rho}} \subseteq \Omega_\rho$ since the data set used to develop the SINDy model in Eq. (10) is generated from open-loop simulations with $x \in \Omega_\rho$ and $u \in U$. Additionally, there exist positive constants M_{si} and L_{si} such that the following inequalities hold for all $x, x' \in \Omega_{\hat{\rho}}$ and $u \in U$:

$$|F_{si}(x, u)| \le M_{si} \tag{17a}$$

$$\left| \frac{\partial \hat{V}(x)}{\partial x} F_{si}(x, u) - \frac{\partial \hat{V}(x')}{\partial x} F_{si}(x', u) \right| \le L_{si}|x - x'| \tag{17b}$$

Consider the existence of a bounded modeling error between the nominal system described by Eq. (1) and the SINDy model given by Eq. (10) (i.e., $|\nu| = |F(x, u, 0) - F_{si}(x, u)| \le \nu_m$, $\nu_m > 0$). The following proposition demonstrates that the feedback controller $u = \Phi_{si}(x) \in U$ can stabilize the nominal system of Eq. (1) if the modeling error is sufficiently small.

Proposition 1. Under the assumption that the origin of the closed-loop SINDy model described by Eq. (10) is rendered exponentially stable under the controller $u = \Phi_{si}(x) \in U$ for all $x \in \Omega_{\hat{\rho}}$, if there exists a positive real number $\Gamma < \hat{c}_3/\hat{c}_4$ that constrains the modeling error $|\nu| = |F(x, u, 0) - F_{si}(x, u)| \le \Gamma|x| \le \nu_m$ for all $x \in \Omega_{\hat{\rho}}$ and $u \in U$, then the origin of the nominal closed-loop system described by Eq. (1) under $u = \Phi_{si}(x) \in U$ is also exponentially stable for all $x \in \Omega_{\hat{\rho}}$.

Proof. To establish the exponential stability of the nominal system described by Eq. (1) under the controller based on the sparse-identified model from Eq. (10), we aim to demonstrate that the derivative of $\hat{V}$, which corresponds to the state of the nominal system, can be rendered negative for all x within the set $\Omega_{\hat{\rho}}$ under $u = \Phi_{si}(x) \in U$. By utilizing Eqs. (16b) and (16c), we can compute the time-derivative of $\hat{V}$ as follows:

$$\begin{aligned} \dot{\hat{V}} &= \frac{\partial \hat{V}(x)}{\partial x} F(x, \Phi_{si}(x), 0) \\ &= \frac{\partial \hat{V}(x)}{\partial x}(F_{si}(x, \Phi_{si}(x)) + F(x, \Phi_{si}(x), 0) - F_{si}(x, \Phi_{si}(x))) \\ &\le -\hat{c}_3|x|^2 + \hat{c}_4|x|(F(x, \Phi_{si}(x), 0) - F_{si}(x, \Phi_{si}(x))) \\ &\le -\hat{c}_3|x|^2 + \hat{c}_4|x|^2\nu_m \end{aligned} \tag{18}$$

By appropriately selecting ν_m, such that $\nu_m < \hat{c}_3/\hat{c}_4$, we can ensure that $\dot{\hat{V}} \le -\tilde{c}_3|x|^2 \le 0$, where $\tilde{c}_3 = -\hat{c}_3 + \hat{c}_4\nu_m > 0$. This implies that the closed-loop state of the nominal system converges to the origin under $u = \Phi_{si}(x) \in U$ for all $x_0 \in \Omega_{\hat{\rho}}$. □.

Upon integrating the SINDy model represented by Eq. (10) into the Lyapunov-based MPC designs, the control actions of the LMPC and LEMPC will be implemented using a sample-and-hold approach. Consequently, the subsequent propositions aim to establish the sample-and-hold characteristics of the Lyapunov-based controller $u = \Phi_{si}(x)$. Specifically, the next proposition derives an upper bound for the discrepancy between the states computed by the nominal system defined in Eq. (1) and the states predicted by the SINDy model given by Eq. (10).

Proposition 2. (c.f. proposition 3 in Wu et al. (2019a)) For the nonlinear system described by $\dot{x} = F(x, u, w)$ in Eq. (1) and the SINDy model given by $\dot{\hat{x}} = F_{si}(\hat{x}, u)$ in Eq. (10), assuming the same initial condition $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$, there exists a class $\mathcal{K}$ function $f_w(\cdot)$ and a positive constant κ such that the following inequalities hold for all x and $\hat{x}$ within $\Omega_{\hat{\rho}}$.

$$|x(t) - \hat{x}(t)| \le f_w(t) := \frac{L_w w_m + \nu_m}{L_x}(e^{L_x t} - 1) \tag{19a}$$

$$\hat{V}(x) \le \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{19b}$$

Proof. Let us denote the error vector between the solutions of the system $\dot{x} = F(x, u, w)$ and the SINDy model $\dot{\hat{x}} = F_{si}(\hat{x}, u)$ as $e(t) = x(t) - \hat{x}(t)$. By taking the time derivative of $e(t)$, we obtain:

$$\begin{aligned} |\dot{e}(t)| &= |F(x, u, w) - F_{si}(\hat{x}, u)| \\ &\le |F(x, u, w) - F(\hat{x}, u, 0)| + |F(\hat{x}, u, 0) - F_{si}(\hat{x}, u)| \end{aligned} \tag{20}$$

Using the Lipschitz condition from Eq. (3b), we have:

$$\begin{aligned} |F(x, u, w) - F(\hat{x}, u, 0)| &\le L_x|x(t) - \hat{x}(t)| + L_w|w(t)| \\ &\le L_x|x(t) - \hat{x}(t)| + L_w w_m \end{aligned} \tag{21}$$

The second term $|F(\hat{x}, u, 0) - F_{si}(\hat{x}, u)|$ in Eq. (20) represents the modeling error, which is bounded by $|\nu| \le \nu_m$ for all $\hat{x} \in \Omega_{\hat{\rho}}$. Hence, combining Eq. (21) and the bound on the modeling error, we can bound $\dot{e}(t)$ as follows:

$$\begin{aligned} |\dot{e}(t)| &\le L_x|x(t) - \hat{x}(t)| + L_w w_m + \nu_m \\ &\le L|e(t)| + L_w w_m + \nu_m \end{aligned} \tag{22}$$

With the zero initial condition ($e(0) = 0$), we can bound the norm of the error vector for all $x(t), \hat{x}(t) \in \Omega_{\hat{\rho}}$ and $w(t) \le w_m$:

$$|e(t)| = |x(t) - \hat{x}(t)| \le \frac{L_w w_m + \nu_m}{L_x}\left(e^{L_x t} - 1\right) \tag{23}$$

Next, to derive Eq. (19b) for all $x, \hat{x} \in \Omega_{\hat{\rho}}$, we expand $\hat{V}(x)$ using a Taylor series expansion around $\hat{x}$:

$$\hat{V}(x) \le \hat{V}(\hat{x}) + \frac{\partial \hat{V}(\hat{x})}{\partial x}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{24}$$

where κ is a positive real number. Using Eqs. (16a) and (16c), we can simplify Eq. (24) as follows:

$$\hat{V}(x) \le \hat{V}(\hat{x})\frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \tag{25}$$

This completes the proof of proposition 2. □.

3.2.1. LMPC using SINDy models

The formulation of a Lyapunov-based model predictive controller (LMPC) using a SINDy model can be expressed as follows (Abdullah et al., 2021b):

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), u(t))\, dt \tag{26a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{si}(\tilde{x}(t), u(t)) \tag{26b}$$

$$u(t) \in U, \quad \forall\ t \in [t_k, t_{k+N}) \tag{26c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{26d}$$

$$\dot{\hat{V}}(x(t_k), u) \le \dot{\hat{V}}(x(t_k), \Phi_{si}(x(t_k))), \quad \text{if} \quad x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}} \tag{26e}$$

$$\hat{V}(\tilde{x}(t)) \le \rho_{si}, \quad \forall\ t \in [t_k, t_{k+N}), \quad \text{if} \quad x(t_k) \in \Omega_{\rho_{si}} \tag{26 f}$$

where $\tilde{x}$ represents the predicted state trajectory. The set $S(\Delta)$ consists of piecewise constant functions with period Δ. N denotes the number of sampling periods in the prediction horizon. The term $\dot{\hat{V}}(x, u)$ denotes $\frac{\partial \hat{V}(x)}{\partial x} F_{si}(x, u)$. The optimal input trajectory computed by the LMPC, denoted as $u^*(t)$, is calculated over the entire prediction horizon $t \in [t_k, , t_{k+N})$. The control action computed for the first sampling period of the prediction horizon, $u^*(t_k)$, is applied during that period, and the LMPC is resolved at the next sampling time.

In the optimization problem defined by Eq. (26), the objective function in Eq. (26a) represents the integral of $L(\tilde{x}(t), u(t))$ over the prediction horizon. The constraint specified in Eq. (26b) describes the sparse-identified model of Eq. (10) used for state prediction in the closed-loop system. Eq. (26c) defines the input constraints applied throughout the prediction horizon, and Eq. (26d) defines the initial condition $\tilde{x}(t_k)$ based on the state measurement at $t = t_k$. The constraint expressed in Eq. (26e) ensures that, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}}$, the closed-loop state will move towards the origin. However, once $x(t_k)$ enters $\Omega_{\rho_{si}}$, the states predicted by the SINDy model from Eq. (26b) will remain within $\Omega_{\rho_{si}}$ for the entire prediction horizon.

The proposition below demonstrates that the closed-loop state of the nominal system described in Eq. (1) remains bounded within the region $\Omega_{\hat{\rho}}$ for all times and can ultimately be bounded in a smaller subset $\Omega_{\rho_{\min}}$ that includes the origin. This result is obtained under the sample-and-hold implementation of the Lyapunov-based controller $u = \Phi_{si}(x) \in U$.

Proposition 3. Consider the system described by Eq. (1) under the controller $u = \Phi_{si}(\hat{x}) \in U$. The controller is designed to stabilize the SINDy system represented by Eq. (10) and satisfies the conditions stated in Eq. (16). The controller operates in a sample-and-hold fashion, where $u(t) = \Phi_{si}(\hat{x}(t_k))\ \forall\ t$ within the interval $[t_k, t_{k+1})$, with $t_{k+1} := t_k + \Delta$. Let ϵ_s, ϵ_w, Δ, and $\hat{\rho}$ be positive values, and assume $\rho_{\min}$, ρ_{si}, and ρ_s satisfy the following conditions:

$$-\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{si} M_{si} \Delta \le -\epsilon_s \tag{27a}$$

$$-\frac{\tilde{c}_3}{\tilde{c}_2}\rho_s + L'_x M \Delta + L'_w w_m \le -\epsilon_w \tag{27b}$$

and

$$\rho_{si} := \max\left\{\hat{V}(\hat{x}(t + \Delta)) \quad | \quad \hat{x}(t) \in \Omega_{\rho_s}, u \in U\right\} \tag{28a}$$

$$\rho_{\min} \ge \rho_{si} + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) + \kappa(f_w(\Delta))^2 \tag{28b}$$

Then, the following inequality holds for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$:

$$\hat{V}(x(t)) \le \hat{V}(x(t_k)), \quad \forall\ t \in [t_k, t_{k+1}) \tag{29}$$

and the state $x(t)$ of the nonlinear system of Eq. (1) is bounded in $\Omega_{\hat{\rho}}$ for all times and ultimately bounded in $\Omega_{\rho_{\min}}$.

Proof. *Part 1*: Let's assume that $x(t_k) = \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$. We will now demonstrate that, under the controller $u(t) = \Phi_{si}(x(t_k)) \in U$, the value of $\hat{V}(\hat{x})$ decreases for $t \in [t_k, t_{k+1})$, where $x(t)$ and $\hat{x}(t)$ represent the solutions of the nonlinear system described by Eq. (1) in the presence of bounded disturbances and the SINDy system described by Eq. (10), respectively. We obtain the time-derivative of $\hat{V}(\hat{x})$ along the trajectory $\hat{x}(t)$ of the SINDy model within the interval $t \in [t_k, t_{k+1})$ as follows:

$$\begin{aligned}
\dot{\hat{V}}(\hat{x}(t)) &= \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t), \Phi_{si}(\hat{x}(t_k))) \\
&= \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k), \Phi_{si}(\hat{x}(t_k))) \\
&\quad + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t), \Phi_{si}(\hat{x}(t_k))) \\
&\quad - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k), \Phi_{si}(\hat{x}(t_k)))
\end{aligned} \tag{30}$$

Using the inequalities of Eqs. (16a) and (16b),

$$\begin{aligned}
\dot{\hat{V}}(\hat{x}(t)) &\le -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t), \Phi_{si}(\hat{x}(t_k))) \\
&\quad - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k), \Phi_{si}(\hat{x}(t_k)))
\end{aligned} \tag{31}$$

Utilizing the Lipschitz condition stated in Eq. (17) and considering the fact that $\hat{x} \in \Omega_{\hat{\rho}}$ and $u \in U$, we can determine the upper bound of $\dot{\hat{V}}(\hat{x}(t))$ for all $t \in [t_k, t_{k+1})$ as follows:

$$\begin{aligned}
\dot{\hat{V}}(\hat{x}(t)) &\le -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{si}|\hat{x}(t) - \hat{x}(t_k)| \\
&\le -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{si} M_{si} \Delta
\end{aligned} \tag{32}$$

Hence, when Eq. (27a) is fulfilled, the subsequent inequality is valid for any $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$:

$$\dot{\hat{V}}(\hat{x}(t)) \le -\epsilon_s \tag{33}$$

By integrating the aforementioned equation over $t \in [t_k, t_{k+1})$, we can conclude that $V(\hat{x}(t_{k+1})) \le V(\hat{x}(t_k)) - \epsilon_s \Delta$. Hence, we have established that, for all $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the state of the closed-loop SINDy system described by Eq. (10) remains bounded within the closed-loop stability region $\Omega_{\hat{\rho}}$ at all times and converges towards the origin when the controller $u = \Phi_{si}(\hat{x}) \in U$ is implemented in a sample-and-hold manner.

However, it should be noted that Eq. (33) may not hold when $x(t_k) = \hat{x}(t_k) \in \Omega_{\rho_s}$. This implies that the state may exit the region Ω_{ρ_s} within a single sampling period. To address this, we design $\Omega_{\rho_{si}}$ based on Eq. (28a) to ensure that the closed-loop state $\hat{x}(t)$ of the SINDy model remains within $\Omega_{\rho_{si}}$ for all $t \in [t_k, t_{k+1})$, $u \in U$, and $\hat{x}(t_k) \in \Omega_{\rho_s}$ during a sampling period. If the state $\hat{x}(t_{k+1})$ exits Ω_{ρ_s}, the controller $u = \Phi_{si}(x(t_{k+1}))$ will guide the state back towards Ω_{ρ_s} in the subsequent sampling period, as Eq. (33) is satisfied again at $t = t_{k+1}$. Consequently, we have demonstrated the convergence of the state to $\Omega_{\rho_{si}}$ for the closed-loop SINDy system described by Eq. (10) for all initial states $\hat{x}_0 \in \Omega_{\hat{\rho}}$. The next step is to establish that the closed-loop state of the actual nonlinear system governed by Eq. (1) can be bounded within $\Omega_{\hat{\rho}}$ for all times and ultimately bounded within a small neighborhood around the origin when the controller $u = \Phi_{si}(x) \in U$ is implemented using the sample-and-hold technique.

Part 2: Building upon the previous analysis of the SINDy system represented by Eq. (10), we now consider the nonlinear system described by Eq. (1) with $x(t_k) = \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$. We derive the time-derivative of $\hat{V}(x)$ for this nonlinear system, taking into account the presence of bounded disturbances w (with $|w| \le w_m$), as shown below:

$$\begin{aligned}\dot{\hat{V}}(x(t)) &= \frac{\partial \hat{V}(x(t))}{\partial x}F(x(t), \Phi_{si}(x(t_k)), w)\\ &= \frac{\partial \hat{V}(x(t_k))}{\partial x}F(x(t_k), \Phi_{si}(x(t_k)), 0)\\ &\quad + \frac{\partial \hat{V}(x(t))}{\partial x}F(x(t), \Phi_{si}(x(t_k)), w)\\ &\quad - \frac{\partial \hat{V}(x(t_k))}{\partial x}F(x(t_k), \Phi_{si}(x(t_k)), 0)\end{aligned} \tag{34}$$

By referring to Eq. (19), we have $\frac{\partial \hat{V}(x(t_k))}{\partial x}F(x(t_k), \Phi_{si}(x(t_k)), 0) \le -\tilde{c}_3|x(t_k)|^2$ for all $x \in \Omega_{\hat{\rho}}\backslash\Omega_{\rho_s}$. Utilizing Eq. (16a) and the Lipschitz condition stated in Eq. (17), we can derive the following inequality for $\dot{\hat{V}}(x(t))$ within the time interval $t \in [t_k, t_{k+1})$, given that $x(t_k) \in \Omega_{\hat{\rho}}\backslash\Omega_{\rho_s}$:

$$\begin{aligned}\dot{\hat{V}}(x(t)) &\le -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(x(t))}{\partial x}F(x(t), \Phi_{si}(x(t_k)), w)\\ &\quad - \frac{\partial \hat{V}(x(t_k))}{\partial x}F(x(t_k), \Phi_{si}(x(t_k)), 0)\\ &\le -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}|x(t) - x(t_k)| + L_w^{'}|w|\\ &\le -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta + L_w^{'}w_m\end{aligned} \tag{35}$$

If the condition stated in Eq. (27b) is fulfilled, we can establish the following inequality for all $x(t_k) \in \Omega_{\hat{\rho}}\backslash\Omega_{\rho_s}$ and $t \in [t_k, t_{k+1})$:

$$\dot{\hat{V}}(x(t)) \le -\epsilon_w \tag{36}$$

From the inequality in Eq. (36), we can conclude that Eq. (29) holds, ensuring that the state of the closed-loop system described by Eq. (1) remains within the region $\Omega_{\hat{\rho}}$ for all times. Furthermore, this implies that the controller $u = \Phi_{si}(x)$ is capable of driving the state of the actual nonlinear system given by Eq. (1) towards the origin within each sampling period.

In addition to the above, if the initial state $x(t_k)$ belongs to the set Ω_{ρ_s}, we have already demonstrated in Part 1 that the state of the SINDy model described by Eq. (10) remains within the region $\Omega_{\rho_{si}}$ within one sampling period. Taking into account the bounded error between the SINDy model state and the actual nonlinear system state, as indicated by Eq. (19a), we can define a compact set $\Omega_{\rho_{\min}}$ such that $\Omega_{\rho_{si}} \subset \Omega_{\rho_{\min}}$, satisfying the condition stated in Eq. (28b). This guarantees that the state of the actual nonlinear system does not exit $\Omega_{\rho_{\min}}$ during a sampling period if the state of the SINDy model remains bounded within $\Omega_{\rho_{si}}$. If the state $x(t)$ enters the set $\Omega_{\rho_{\min}}\backslash\Omega_{\rho_s}$, we have shown that Eq. (36) holds, and thus, the state will be driven towards the origin again during the next sampling period under the controller $u = \Phi_{si}(x)$.

By establishing the above arguments, we have completed the proof of proposition 3, demonstrating that for any initial state $x_0 = \hat{x}_0 \in \Omega_{\hat{\rho}}$, the closed-loop state trajectories of the nonlinear system described by Eq. (1) remain within the region $\Omega_{\hat{\rho}}$ and ultimately become bounded within $\Omega_{\rho_{\min}}$, provided that the assumptions of proposition 3 are satisfied. □.

The aforementioned stability region, the various Lyapunov level sets and an example of a closed-loop state trajectory under the LMPC are depicted in Fig. 1.

3.2.2. LEMPC using SINDy models

The Lyapunov-based economic model predictive control (LEMPC) approach utilizing a SINDy model is designed to dynamically optimize the economic benefits of a process while ensuring that the closed-loop state remains within a specified stability region at all times (Wu and Christofides, 2019). The LEMPC can be formulated as the following optimization problem:

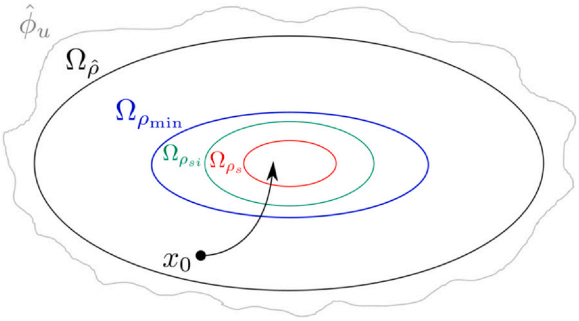

Fig. 1 – A diagram illustrating the sets $\hat{\phi}_u$, $\Omega_{\hat{\rho}}$, $\Omega_{\rho_{\min}}$, $\Omega_{\rho_{si}}$, and Ω_{ρ_s} in concentric ellipses, from outermost to innermost. The LMPC control strategy of Eq. (26) guides the closed-loop state towards the origin and ensures that it eventually remains bounded within $\Omega_{\rho_{\min}}$ for any initial state $x_0 \in \Omega_{\hat{\rho}}$.

$$\mathcal{J} = \max_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_e(\tilde{x}(t), u(t))\, dt \tag{37a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F_{si}(\tilde{x}(t), u(t)) \tag{37b}$$

$$u(t) \in U, \quad \forall\ t \in [t_k, t_{k+N}) \tag{37c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{37d}$$

$$\hat{V}(\tilde{x}(t)) \le \hat{\rho}_e, \quad \forall \quad t \in [t_k, t_{k+N}), \quad \text{if} \quad x(t_k) \in \Omega_{\hat{\rho}_e} \tag{37e}$$

$$\dot{\hat{V}}(x(t_k), u) \le \dot{\hat{V}}(x(t_k), \Phi_{si}(x(t_k))), \quad \text{if} \quad x(t_k) \in \Omega_{\hat{\rho}}\backslash\Omega_{\hat{\rho}_e} \tag{37 f}$$

The notation used in Eq. (37) follows that of Eq. (26). The optimization problem presented in Eq. (37) aims to maximize the time integral of the stage cost function, denoted as $l_e(\tilde{x}(t), u(t))$, over the prediction horizon. The prediction model described in Eq. (37b) and the initial condition given in Eq. (37d) are the same as those used in the LMPC formulation of Eq. (26). The constraint stated in Eq. (37e) ensures that the predicted closed-loop states remain within the region $\Omega_{\hat{\rho}_e}$ over the prediction horizon when the initial state $x(t_k)$ is inside this region. However, if $x(t_k)$ enters the region $\Omega_{\hat{\rho}}\backslash\Omega_{\hat{\rho}_e}$, the contractive constraint expressed in Eq. (37f) drives the state towards the origin during the next sampling period, ultimately causing the state to enter $\Omega_{\hat{\rho}_e}$ within a finite number of sampling periods.

Proposition 4. Consider the system described by Eq. (1) subject to the controller $u = \Phi_{si}(\hat{x}) \in U$. This controller satisfies the conditions specified in Eq. (16) and is implemented using a sample-and-hold approach, meaning that $u(t) = \Phi_{si}(\hat{x}(t_k))\ \forall\, t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$. Further, let ϵ_w and ϵ_s be positive values, and let Δ be a positive time interval. Consider also that $\hat{\rho} > \hat{\rho}_e > \rho_s > 0$, satisfying the conditions given by the following equations:

$$-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_{si}M_{si}\Delta \le -\epsilon_s \tag{38a}$$

$$-\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'}M\Delta + L_w^{'}w_m \le -\epsilon_w \tag{38b}$$

$$\hat{\rho}_e > \max\left\{\hat{V}(\hat{x}(t_k + \Delta)) \ \mid\ \hat{x}(t_k) \in \Omega_{\rho_s}, u \in U, w \in W\right\} \tag{38c}$$

In that case, for any $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the following inequalities hold:

$$\hat{V}(\hat{x}(t)) \le \hat{V}(\hat{x}(t_k)), \quad \forall\, t \in [t_k,\, t_{k+1}) \tag{39a}$$

$$\hat{V}(x(t)) \le \hat{V}(x(t_k)), \quad \forall\, t \in [t_k,\, t_{k+1}) \tag{39b}$$

Proof. In order to demonstrate the decreasing value of $\hat{V}$ along the trajectory $\hat{x}(t)$ of the SINDy model described by Eq. (10) over the interval $t \in [t_k,\, t_{k+1})$, we evaluate the time derivative of $\hat{V}(\hat{x})$ with respect to $\hat{x}(t)$ as follows:

$$\begin{aligned}\dot{\hat{V}}(\hat{x}(t)) &= \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t),\, \Phi_{si}(\hat{x}(t_k)))\\ &= \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k),\, \Phi_{si}(\hat{x}(t_k)))\\ &\quad + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t),\, \Phi_{si}(\hat{x}(t_k)))\\ &\quad - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k),\, \Phi_{si}(\hat{x}(t_k)))\end{aligned} \tag{40}$$

By exploiting the Lyapunov constraints of Eqs. (16a) and (16b) and the Lipschitz condition of Eq. (17), we obtain the following inequalities:

$$\begin{aligned}\dot{\hat{V}}(\hat{x}(t)) &\le -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + \frac{\partial \hat{V}(\hat{x}(t))}{\partial \hat{x}} F_{si}(\hat{x}(t),\, \Phi_{si}(\hat{x}(t_k)))\\ &\quad - \frac{\partial \hat{V}(\hat{x}(t_k))}{\partial \hat{x}} F_{si}(\hat{x}(t_k),\, \Phi_{si}(\hat{x}(t_k)))\\ &\le -\frac{\hat{c}_3}{\hat{c}_2}\rho_s + L_{si} M_{si} \Delta\end{aligned} \tag{41}$$

Hence, if the condition of Eq. (38a) is satisfied, we have $\dot{\hat{V}}(\hat{x}(t)) \le -\epsilon_s$ for all $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k,\, t_{k+1})$. Integrating the above inequality leads to $\hat{V}(\hat{x}(t)) \le \hat{V}(\hat{x}(t_k)) - \Delta\epsilon_s$, for all $\hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k,\, t_{k+1})$ (referred to as Eq. (39a)).

To establish the inequality $\hat{V}(x(t)) \le \hat{V}(x(t_k))$ for all $t \in [t_k,\, t_{k+1})$, we derive the time-derivative of $\hat{V}(x)$ for the nonlinear system described by Eq. (1) (where $\dot{x} = F(x,\, u,\, w)$) in the presence of bounded disturbances (i.e., $|w(t)| \le w_m$) as follows:

$$\begin{aligned}\dot{\hat{V}}(x(t)) &= \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t),\, \Phi_{si}(x(t_k)),\, w)\\ &= \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k),\, \Phi_{si}(x(t_k)),\, 0)\\ &\quad + \frac{\partial \hat{V}(x(t))}{\partial x} F(x(t),\, \Phi_{si}(x(t_k)),\, w)\\ &\quad - \frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k),\, \Phi_{si}(x(t_k)),\, 0)\end{aligned} \tag{42}$$

By utilizing Eq. (19), which states that $\frac{\partial \hat{V}(x(t_k))}{\partial x} F(x(t_k),\, \Phi_{si}(x(t_k)),\, 0) \le -\tilde{c}_3 |x(t_k)|^2$ holds for all $x \in \Omega_{\hat{\rho}}$, we can derive the following inequality for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k,\, t_{k+1})$ using Eq. (16a) and the Lipschitz condition given by Eq. (17):

$$\begin{aligned}\dot{\hat{V}}(x(t)) \le &\; -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'} |x(t) - x(t_k)| + L_w^{'} |w|\\ \le &\; -\frac{\tilde{c}_3}{\hat{c}_2}\rho_s + L_x^{'} M \Delta + L_w^{'} w_m\end{aligned} \tag{43}$$

Therefore, if the condition of Eq. (38b) is satisfied, we can derive the following inequality $\forall\, x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $\forall\, t \in [t_k,\, t_{k+1})$:

$$\dot{\hat{V}}(x(t)) \le -\epsilon_w \tag{44}$$

Likewise, this implies that $\hat{V}(x(t)) \le \hat{V}(x(t_k)) - \Delta\epsilon_w$ holds for all $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$ and $t \in [t_k,\, t_{k+1})$. Hence, if $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_s}$, the state of the nonlinear system described by Eq. (1) will enter Ω_{ρ_s} within a finite number of sampling periods. Moreover, if $x(t_k) \in \Omega_{\rho_s}$, where Eqs. (43) and (44) do not hold, Eq. (38c) ensures that the state will not exit $\Omega_{\hat{\rho}_e}$ within one sampling period for any $u \in U$ and $w \in W$. If the state $x(t_{k+1})$ exits Ω_{ρ_s} but remains within $\Omega_{\hat{\rho}_e}$, then at the subsequent sampling period $t \in [t_{k+1},\, t_{k+2})$, Eq. (44) is satisfied again, causing the state to be driven toward the origin. Consequently, the state of the nonlinear system given by Eq. (1) remains bounded within $\Omega_{\hat{\rho}}$ at all times. □.

4. Error-triggered online update of SINDy models

In this section, we apply the LMPC the LEMPC methods described by Eq. (26) and Eq. (37), respectively, to the nonlinear system of Eq. (1) in the presence of bounded disturbances (i.e., $|w(t)| \le w_m$) that grows due to changes in the dynamics of the nonlinear system of Eq. (1), potentially leading to instability in the closed-loop system. To mitigate the impact of disturbances, we update SINDy models through online learning to capture the nonlinear dynamics of the system described by Eq. (1) while accounting for the influence of disturbances $w(t)$. The subsequent subsections introduce the error-triggering mechanism employed for updating the SINDy models to ensure $|w(t)| \le w_m$ for all time.

4.1. *Error-triggering mechanism*

In this subsection, we develop an event-triggering mechanism based on the errors between predicted states and measured states to update the SINDy model for all $x \in \Omega_{\hat{\rho}}$. This mechanism is referred to as the error-triggered on-line SINDy update throughout the manuscript. Specifically, following the error-triggering mechanism proposed in Alanqar et al. (2017), we introduce a moving horizon error metric denoted as $e_d(t_k)$, which indicates the prediction accuracy of the SINDy model at time $t = t_k$.

$$e_d(t_k) = \sum_{r=0}^{N_w} \sum_{j=1}^{n_x} \frac{|x_{p,j}(t_{k-r}) - x_j(t_{k-r})|}{|x_j(t_{k-r})|} \tag{45}$$

where the quantity N_w represents the number of sampling periods prior to t_k that contribute to the estimation of the prediction error. At each sampling period between t_{k-N_w} and t_k, $x_j(t_{k-r})$ captures the historical measurements of the process states, where r ranges from 0 to N_w. Similarly, $x_{p,j}(t_{k-r})$ represents the predictions of the past states of the system obtained from the SINDy model. The moving horizon error detector triggers a model update if/when the error metric e_d surpasses a predefined threshold $e_{d,T}$, i.e., the following condition:

$$e_d(t_k) > e_{d,T} \tag{46}$$

Determining the parameters for this error-triggered approach involves defining the number of input and output data points N_d that should be retained for model update when it is triggered, the length N_w of the moving horizon used to calculate the error metric e_d, and the threshold $e_{d,T}$ that dictates when a model update should be initiated.

The following strategy is proposed for the selection of the parameter, .

- N_d: The SINDy algorithm is highly dependent on the amount of dynamic information present in a data set rather than simply its size. While the initial SINDy model must be developed with a large data set consisting of many open-loop experiments/simulations, the model update procedure must use a much smaller number of data points to update the coefficients because, firstly, the updates must occur relatively soon after the disturbance takes effect and, secondly, the model is only being updated rather than being identified. Hence, the amount of data required for the initial offline model building procedure has no impact on the amount of data required to efficiently update the model in real-time. The actual amount of data will depend also on the sampling period since a smaller sampling period can yield a large volume of data within a short period of data acquisition. In summary, the value of N_d is highly process-specific, and some insights will be provided in the application section. We note, however, that even if N_d is relatively large for a certain process (or region of a process), one approach to mitigate process deterioration during the time between the error-triggering and the SINDy model update is to use a linear data-driven model as a stopgap solution (Kaiser et al., 2018).

- N_w: The selection of the appropriate value for N_w in the calculation of e_d requires finding a balance. On one hand, N_w should be sufficiently long to ensure that common disturbances during normal operation do not significantly impact e_d, which could lead to unnecessary triggering of errors. On the other hand, N_w should not be longer than necessary to avoid unnecessary data storage and processing. One approach to determine N_w is by evaluating the value of $e_d(t_k)$ at each sampling period for a set of input/output data collected during typical process operation. This evaluation is performed in closed-loop under the SINDy-based MPC, focusing on the region of operation where the initial SINDy model was developed and validated. By repeating this calculation for different values of N_w, it becomes possible to observe the range of minimum and maximum values of e_d. If N_w is small, the minimum and maximum values of e_d may differ significantly since any disturbance or measurement noise within the moving horizon has a considerable impact on e_d. However, as N_w increases, the influence of disturbances and measurement noise becomes less significant. At some point, the minimum and maximum values of e_d are expected to stabilize, indicating that further increases in N_w have minimal effect. In such a case, the smallest value of N_w for which the minimum and maximum values of e_d reach their approximate final value can be selected for use in Eq. (45). It is important to note that the value of N_w for a given process depends on the statistical properties of $w(t)$ and its influence on the system. Therefore, careful consideration of these factors is necessary when determining the appropriate value of N_w.
- $e_{d,T}$: The determination of the threshold value $e_{d,T}$ is performed offline, taking into account the chosen value of N_w. The goal is to set a threshold that avoids triggering model updates in the presence of measurement noise, small constant disturbances, and time-varying disturbances that still result in reasonably accurate predictions using the current model. One approach to achieve this is by analyzing the statistical properties of e_d using a set of closed-loop input/output data corresponding to normal process operation within the region where the initial SINDy model was developed and validated. For example, the maximum value of e_d can be calculated using the selected value of N_w. The threshold can then be set to be a reasonable percentage higher than this maximum value of e_d observed in the normal operating data. This approach ensures that the threshold includes disturbances and measurement noise that regularly affect the system. Alternatively, other statistical measures could be used, such as setting $e_{d,T}$ to be several standard deviations above the mean value of e_d calculated from the normal operating data. The choice of the appropriate measure depends on the specific system being analyzed. It is important to note that, even in the absence of disturbances or measurement noise, e_d may have a non-zero value if the SINDy model captured the dynamics of the system using a different set of basis functions than the actual nonlinearities of the process in consideration. If the exact basis functions and coefficients were correctly identified by the SINDy model, however, e_d can be expected to be very close to zero.

From a practical perspective, due to the continuous, online monitoring of the process performance, even if some parameters are not chosen optimally from the beginning, they can be adjusted based on the incoming data, and the lack of explicit formulae to determine the aforementioned parameters is not a limitation of the adaptive SINDy framework.

Remark 1. While Eq. (45) assumes full-state feedback, if only output measurements are available, the SINDy modeling framework itself can be modified to obtain not the state derivative but the output measurements as functions of the states, i.e., $y = f(x, u)$, where y are the output measurements. Such an approach has been proposed in Wang et al. (2022) under the name of generalized SINDy. Hence, using output measurements as the target variables and nonlinear functions of the states and manipulated inputs as the library functions, once relationships are obtained for the output measurements, the predicted output measurements can be used to compute the error metric in Eq. (45) instead.

4.2. *SINDy model update procedure*

Once the SINDy model update is triggered by the mechanisms of Section 4, the model update is carried out using the last N_d data points. The details of the update procedure are discussed in this subsection.

It is assumed that the structure of the original SINDy model obtained offline does not change due to the presence of disturbances or changes in the process. This is a practical assumption since many changes such as catalyst deactivation or fluctuations in feed flow rates would not add or remove terms from the process model but simply alter the terms, specifically the coefficients associated with the respective terms. Hence, the problem of updating SINDy models in real-time can be reduced to re-identifying a subset of the coefficients in the matrix Ξ. In doing so, however, the remaining coefficients must remain at their original values.

For real-time model updates of q out of p coefficients for each variable, the matrices $\Theta(X, U)$ and Ξ must be split into the part that will remain unchanged and the part that will be re-identified as follows:

$$\Theta = [\Theta_{\text{fixed}} \quad \Theta_{\text{update}}], \qquad \Xi = [\Xi_{\text{fixed}} \quad \Xi_{\text{update}}]^{\top} \tag{47}$$

where $\Theta_{\text{fixed}} \in \mathbb{R}^{m\times(p-q)}$ and $\Theta_{\text{update}} \in \mathbb{R}^{m\times q}$ contain the $m \times 1$ columns of Θ that are fixed and that are updated, respectively. Similarly, $\Xi_{\text{fixed}} \in \mathbb{R}^{(p-q)\times n_x}$ and $\Xi_{\text{update}} \in \mathbb{R}^{q\times n_x}$ contain the corresponding $1 \times n_x$ rows of Ξ. The sparse identification problem then becomes

$$\dot{X} = \Theta\Xi = \Theta_{\text{fixed}}\Xi_{\text{fixed}} + \Theta_{\text{update}}\Xi_{\text{update}} \tag{48}$$

which can be reformulated into the form,

$$\dot{X} - \Theta_{\text{fixed}}\Xi_{\text{fixed}} = \Theta_{\text{update}}\Xi_{\text{update}} \tag{49}$$

In Eq. (49), the left-hand side is a matrix that can be evaluated using the fixed part of the current SINDy model, and Θ_{update} in the right-hand side of Eq. (49) is also known. Therefore, STLSQ can be used to solve for the only unknown in Eq. (49), which is Ξ_{update}, using the same procedure as described in Section 3.1. Specifically, the least-squares formulation now solves for Ξ_{update} as follows:

$$\Xi_{\text{update}} = \arg\min_{\Xi''_{\text{update}}} \|\dot{X} - \Theta_{\text{fixed}}\Xi_{\text{fixed}} - \Theta_{\text{update}}\Xi''_{\text{update}}\|_2 \tag{50}$$

In the above formulation, the notations use the full matrices for the parts of Ξ, which is done for simplicity. However, the same library terms do not need to be fixed between the different variables/ODEs. Since the STLSQ algorithm solves for the coefficients of each state variable sequentially in a loop, the different terms to be updated can be easily incorporated into the solution. Specifically, instead of altering the same q library function coefficients for every variables, let the i^{th} variable update q_i coefficients in its ODE right-hand side ($i = 1, \ldots, n_x$). In this case, the least-squares problem associated with the partial SINDy update for the i^{th} variable is expressed as follows:

$$\dot{x}_i - \Theta_{i,\text{fixed}}\Xi_{i,\text{fixed}} = \Theta_{i,\text{update}}\Xi_{i,\text{update}} \tag{51}$$

For example, consider a system where the first and third library functions are to be updated for the first variable ($q_1 = 2$) but only the second library function for the second variable ($q_2 = 1$). In this case, the least-squares problems to be solved for identifying the updated SINDy coefficients for the ODEs modeling the dynamics of the first two variables would be

$$\dot{x}_1 - [\theta_2 \quad \theta_4 \quad \theta_5 \quad \cdots \quad \theta_p]\begin{bmatrix}\xi_{2,1}\\ \xi_{4,1}\\ \xi_{5,1}\\ \vdots\\ \xi_{p,1}\end{bmatrix} = [\theta_1 \quad \theta_3]\begin{bmatrix}\xi_{1,1}\\ \xi_{3,1}\end{bmatrix} \tag{52}$$

and

$$\dot{x}_2 - [\theta_1 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \cdots \quad \theta_p]\begin{bmatrix}\xi_{1,2}\\ \xi_{3,2}\\ \xi_{4,2}\\ \xi_{5,2}\\ \vdots\\ \xi_{p,2}\end{bmatrix} = [\theta_2]\left[\xi_{2,2}\right] \tag{53}$$

respectively, where $\xi_{i,j}$ denotes the i^{th} row of the j^{th} column of Ξ. Solving Eq. (52) will yield the updated values of $\xi_{1,1}$ and $\xi_{3,1}$, while solving Eq. (53) outputs the updated scalar, $\xi_{2,2}$. In this manner, any number and choice of library functions' coefficients can be updated using the partial re-identification algorithm described in this subsection.

4.3. *Implementation strategy for error-triggered on-line model identification*

After determining the values of N_d, N_w, and $e_{d,T}$ using the methodology described in the previous section, the implementation strategy for the proposed error-triggered on-line model identification is as follows:

1. An initial SINDy model that captures the nonlinear process behavior in the operating region is developed using data from extensive open-loop simulations. This model is used to design the model predictive controller.
2. The system is operated under the MPC that is designed based on the current SINDy model. During operation, N_d values of input/output data are collected and stored for potential future model identification. At t_{N_w}, the moving horizon error detector is activated to compute $e_d(t_k)$.
3. When the current SINDy model becomes inadequate in capturing the dynamics of the process (due to factors such as plant variations or changes in the operating region), the value of $e_d(t_k)$ will rise. Once $e_d(t_k)$ surpasses the threshold $e_{d,T}$, the latest set of N_d input and output data values (collected up until time t_k) are employed for on-line model update. This newly updated SINDy model then replaces the current SINDy model and is utilized as the process model in the MPC.
4. Steps 2 and 3 are repeated as process operation continues.

4.4. *Stability analysis of error-triggered feedback systems*

In this section, we use the propositions developed in Section 3.2.1 and 3.2.2 to develop closed-loop stability results for the nonlinear system of Eq. (1) under the Lyapunov-based controllers.

4.4.1. *Stability analysis for Lyapunov-based tracking MPC*

Based on the LMPC formulation given by Eq. (26), the following theorem establishes that the LMPC optimization problem can be solved with recursive feasibility, ensuring closed-loop stability of the nonlinear system described by Eq. (1) when implementing the optimal control actions computed by LMPC using a sample-and-hold strategy (i.e., $u(t) = \Phi_{si}(\hat{x}(t_k)), \forall\, t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period).

Theorem 1. Consider the closed-loop system of Eq. (1) under the LMPC of Eq. (26) with on-line updates of the SINDy model. Let $\Delta > 0$, $\epsilon_w > 0$ and $\hat{\rho} > \rho_{\min} > \rho_s$ satisfy Eq. (27) and Eq. (28). Then, given any initial state $x_0 \in \Omega_{\hat{\rho}}$, if the conditions of proposition 2 and proposition 3 are satisfied, and if the SINDy model is updated following the implementation strategy in this section with the triggering event of Eq. 46, then it is guaranteed that the LMPC of Eq. (26) has a feasible solution and that under the LMPC of Eq. (26), $x(t) \in \Omega_{\hat{\rho}}, \quad \forall\, t \geq 0$, and $\lim_{t\to\infty} \hat{V}(x(t)) \leq \rho_{\min}$ for the closed-loop system of Eq. (1).

Proof. We will demonstrate that the optimization problem described by Eq. (26) is recursively feasible for all $x \in \Omega_{\hat{\rho}}$. Specifically, if at time $t = t_k$ the state $x(t_k)$ satisfies $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}}$, the control action $u(t) = \Phi_{si}(x(t_k)) \in U$, for $t \in [t_k, t_{k+1})$, obtained based on the state measurement $x(t_k)$, satisfies both the input constraint defined in Eq. (26c) and the Lyapunov-based constraint given by Eq. (26e). Moreover, if $x(t_k) \in \Omega_{\rho_{si}}$, the control actions computed using $\Phi_{si}(x(t_{k+i}))$ for $i = 0, 1, \ldots, N-1$ satisfy the input constraint in Eq. (26c) as well as the Lyapunov-based constraint specified in Eq. (26f). This result is derived from proposition 3, which shows that the predicted states by the SINDy model defined in Eq. (26b) remain within $\Omega_{\rho_{si}}$ when subjected to the controller $\Phi_{si}(x)$.

Consequently, for any initial state $x_0 \in \Omega_{\hat{\rho}}$, the LMPC optimization problem of Eq. (26) can be solved with recursive feasibility if the condition $x(t) \in \Omega_{\hat{\rho}}$ holds for all time instances.

We will now demonstrate that for any initial state $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system described by Eq. (1) remains bounded within $\Omega_{\hat{\rho}}$ for all time and ultimately converges to a small neighborhood around the origin, denoted as $\Omega_{\rho_{\min}}$ and defined by Eq. (28b), under the LMPC controller specified by Eq. (26). Let's consider the case where at time $t = t_k$, the state $x(t_k)$ satisfies $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}}$. In this situation, the constraint defined by Eq. (26e) is activated, ensuring that the control action u is selected to decrease the value of $\hat{V}(\hat{x})$ based on the predicted states provided by the SINDy model described by Eq. (26b) over the next sampling period. Additionally, according to Eq. (36), if the constraint of Eq. (26e) is satisfied, it follows that $\dot{\hat{V}}(x) \leq -\epsilon_w$ for $t \in [t_k, t_{k+1})$ after applying the control action $u^*(t_k)$ to the nonlinear system defined by Eq. (1). Therefore, the value of $\hat{V}(x)$ based on the state of the actual nonlinear system given by Eq. (1) decreases within the next sampling period, implying that the closed-loop state can be driven into $\Omega_{\rho_{si}}$ within a finite number of sampling steps.

Once the state enters $\Omega_{\rho_{si}}$, the constraint specified by Eq. (26f) is activated to ensure that the predicted states of the SINDy model in Eq. (26b) remain within $\Omega_{\rho_{si}}$ throughout the prediction horizon. Due to the existence of a mismatch between the SINDy model described by Eq. (26b) and the nonlinear system represented by Eq. (1), the state of the nonlinear system may exit $\Omega_{\rho_{si}}$ when subject to the constraint defined by Eq. (26f). However, by characterizing a region $\Omega_{\rho_{\min}}$ that satisfies Eq. (28b), proposition 3 guarantees that if the predicted state by the SINDy model remains within $\Omega_{\rho_{si}}$, then the state $x(t)$ of the nonlinear system, for all $t \in [t_k, t_{k+1})$, will remain bounded within $\Omega_{\rho_{\min}}$. Consequently, at the next sampling step $t = t_{k+1}$, if the state $x(t_{k+1})$ is still bounded within $\Omega_{\rho_{si}}$, the constraint defined by Eq. (26f) ensures that the predicted state $\hat{x}$ of the SINDy model in Eq. (26b) remains within $\Omega_{\rho_{si}}$, guaranteeing that the actual state x of the nonlinear system described by Eq. (1) stays within $\Omega_{\rho_{\min}}$.

However, if $x(t_{k+1})$ belongs to $\Omega_{\rho_{\min}} \backslash \Omega_{\rho_{si}}$, similar to the proof provided for the case when $x(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{si}}$, the constraint specified by Eq. (26e) is activated to drive the state towards the origin. This completes the proof that the states of the closed-loop system described by Eq. (1) remain bounded within $\Omega_{\hat{\rho}}$ and converge to $\Omega_{\rho_{\min}}$ for any initial state $x_0 \in \Omega_{\hat{\rho}}$.

We note that the theorem establishes that the closed-loop stability of the nonlinear system described by Eq. (1) is achieved through the implementation of the LMPC controller given by Eq. (26). This controller is designed based on the SINDy model represented by Eq. (10) and incorporates SINDy-based constraints.

It is important to note that the closed-loop state of the nonlinear system, as described by Eq. (1), can be driven to a small neighborhood around the origin. This is possible because the constraints defined by the LMPC controller in Eq. (26) ensure the decrease of $\hat{V}$ during each sampling period, accounting for various factors such as model mismatch (including the modeling error ν between the system in Eq. (1) and the SINDy model in Eq. (10)), the implementation of control actions using a sample-and-hold approach, and the presence of bounded disturbances $w(t)$ in Eq. (1). In other words, closed-loop stability can be maintained under the LMPC controller described by Eq. (26) if the modeling error ν, the sampling period Δ, and the disturbance bound w_m are sufficiently small. This requirement ensures the satisfaction of proposition 2 and proposition 3.

However, the values of ν_m and w_m are generally not known as there is no formula to calculate them. Hence, if ν_m and w_m are not chosen to be sufficiently large and the process is subject to changes such that the modeling error $|\nu|$ exceeds ν_m and large enough disturbances such that the disturbance $|w|$ exceeds w_m, the condition of Eq. (28b) to quantify $\rho_{\min}$ may not hold, and the closed-loop state may exit $\rho_{\min}$. However, the goal of the error-triggering update of SINDy models is to mitigate the model mismatch such that, even if the process changes and $|\nu|$ increases, by choosing the value of $e_{d,T}$ conservatively, the error-triggering and model update procedure is rapidly carried out to reduce $|\nu|$ once again such that it never exceeds ν_m. In this way, the conditions of proposition 3 are met and the proof provided in this theorem hold. Hence, under the sample-and-hold implementation of the LMPC with real-time SINDy model updates, the closed-loop state of the nonlinear system of Eq. (1) remains bounded in $\Omega_{\hat{\rho}}$ and ultimately converges to $\Omega_{\rho_{\min}}$. □.

4.4.2. *Stability analysis for lyapunov-based economic MPC*

Theorem 2. We examine the closed-loop system described by Eq. (1) when using the sample-and-hold implementation of the LEMPC of Eq. (37) with the stabilizing controller $\Phi_{si}(x)$ that fulfills Eq. (16). We consider parameters $\Delta > 0$, $\epsilon_w > 0$, and $\hat{\rho} > \hat{\rho}_e > 0$ that satisfy Eq. (38) along with the inequality,

$$\hat{\rho}_e \leq \hat{\rho} - \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}} f_w(\Delta) - \kappa (f_w(\Delta))^2 \tag{54}$$

Assuming x_0 belongs to $\Omega_{\hat{\rho}}$ and the conditions stated in Proposition 2 and Proposition 4 are fulfilled, it is guaranteed that a feasible solution exists for the optimization problem presented in Eq. (37). Furthermore, the closed-loop state $x(t)$ remains bounded within the closed-loop stability region $\Omega_{\hat{\rho}}$ for all time $t \geq 0$.

Proof. We will first establish the recursive feasibility of the optimization problem presented in Eq. (37) $\forall\, x \in \Omega_{\hat{\rho}}$. Specifically, if $x(t_k)$ is within $\Omega_{\hat{\rho}_e}$, the control actions $\Phi_{si}(x(t_{k+i}))$, where $i = 0, 1, \ldots, N-1$, satisfy both the input constraint of Eq. (37c) and the Lyapunov-based constraint of Eq. (37e). This is due to the fact that Eq. (39a) dictates that the states predicted by the SINDy model described in Eq. (37b) remain inside $\Omega_{\hat{\rho}_e}$ under the controller $\Phi_{si}(x)$.

Furthermore, if $x(t_k)$ falls within $\Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the control action $u(t) = \Phi_{si}(x(t_k)) \in U$, for $t \in [t_k, t_{k+1})$, satisfies both the input constraint defined in Eq. (37c) and the Lyapunov-based constraint outlined in Eq. (37f). This ensures that the state can be driven towards the origin during the subsequent sampling period. As a result, the stabilizing controller $u = \Phi_{si}(x) \in U$ provides a feasible solution that adheres to all the constraints of the LEMPC optimization problem stated in Eq. (37) if $x(t)$ remains within $\Omega_{\hat{\rho}}$ for all time instances.

We will now establish that for $x_0 \in \Omega_{\hat{\rho}}$, the state of the closed-loop system described in Eq. (1) remains bounded within $\Omega_{\hat{\rho}}$ for all time instances. Specifically, if $x(t_k)$ falls within $\Omega_{\hat{\rho}_e}$, the predicted states $\hat{x}(t)$ obtained from the SINDy model presented in Eq. (37b) are guaranteed to stay within $\Omega_{\hat{\rho}_e}$ by adhering to the constraint specified in Eq. (37e). Based on Proposition 2, we can conclude that the actual state $x(t)$, where $t \in [t_k, t_{k+1})$, of the nonlinear system outlined in Eq. (1) is bounded by the following inequality:

$$\begin{aligned}\hat{V}(x) &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}|x - \hat{x}| + \kappa|x - \hat{x}|^2 \\ &\leq \hat{V}(\hat{x}) + \frac{\hat{c}_4\sqrt{\hat{\rho}}}{\sqrt{\hat{c}_1}}f_w(\Delta) + \kappa(f_w(\Delta))^2 \end{aligned} \tag{55}$$

Hence, if we choose $\Omega_{\hat{\rho}_e}$ as a level set of $\hat{V}$ that satisfies the condition in Eq. (54), it guarantees that $V(x)$ based on the actual state $x(t)$ remains bounded within $\Omega_{\hat{\rho}}$ for all $t \in [t_k, t_{k+1})$. However, in the case where $x(t_k)$ lies within $\Omega_{\hat{\rho}} \backslash \Omega_{\hat{\rho}_e}$, the constraint stated in Eq. (37f) becomes active. Consequently, the control action u acts to decrease the value of $\hat{V}(\hat{x})$ based on the states predicted by the RNN model described in Eq. (37b) within the next sampling period.

From Eq. (39b) in proposition 4, it follows that the value of $\hat{V}$ also decreases along the state trajectory of the actual nonlinear system described in Eq. (1) over $t \in [t_k, t_{k+1})$. Thus, we can conclude that for any initial condition within $\Omega_{\hat{\rho}}$, the closed-loop state of the system represented by Eq. (1) remains bounded within $\Omega_{\hat{\rho}}$ $\forall$ t when subjected to the LEMPC of Eq. (37). □.

Remark 2. We note that the condition of Eq. (54), which dictates the size of the level set $\hat{\rho}_e$, includes the class $\mathscr{K}$ function $f_w(\cdot)$, which factors in the uncertainties from the modeling error and the disturbances, ν and w, respectively. Since the bounds on the modeling error and disturbance, ν_m and w_m are not estimable for a general system, if the process disturbance or plant-model mismatch is large and exceed the aforementioned bounds, the size of the level set $\hat{\rho}_e$ may be reduced, which can lead to lower economic benefits of the LEMPC. However, $e_{d,T}$ is chosen conservatively such that the SINDy model of Eq. (37b) is updated as soon as the error-triggering condition of Eq. (46) is violated, such that the modeling error $|\nu|$ is kept low throughout the operation under the LEMPC based on theorem 2.

5. Application of error-triggered on-line model update to plant variations: application to a chemical process example

In this section, we present the application of the proposed error-triggered on-line model identification procedure to control a benchmark chemical reactor. The reactor experiences plant model changes, specifically catalyst deactivation. The system under consideration is a non-isothermal continuous stirred tank reactor (CSTR) involved in the catalytic conversion of reactant species A to product B ($A \rightarrow B$). The reactor operates with an inlet concentration of A denoted by C_{A0}, an inlet temperature of T_0, and a feed volumetric flow rate of F. A heating jacket is employed in the CSTR to supply or remove heat at a rate of Q.

The dynamics of the CSTR are governed by material and energy balance equations given as follows:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{\frac{-E}{RT}} C_A^2 \tag{56a}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_0 - T) + \frac{-\Delta H}{\rho_L C_p} k_0 e^{\frac{-E}{RT}} C_A^2 + \frac{Q}{\rho_L C_p V} \tag{56b}$$

where C_A represents the concentration of reactant A in the reactor, V is the volume of the reacting liquid, T is the temperature of the reactor, and Q is the heat input rate. The concentration of reactant A in the feed is denoted as C_{A0}. The feed temperature and volumetric flow rate are T_0 and F, respectively. The reacting liquid has a constant density of ρ_L and a heat capacity of C_p. Parameters ΔH, k_0, E, and R correspond to the enthalpy of reaction, pre-exponential constant, activation energy, and ideal gas constant, respectively. The values of the process parameters are provided in Table 1.

Table 1 – Parameter values for chemical process example.

$F = 5.0\ \mathrm{m^3/h}$	$V = 1.0\ \mathrm{m^3}$
$k_0 = 8.46e6\ \mathrm{m^3.kmol^{-1}.h^{-1}}$	$E = 5.0e4$ kJ/kmol
$R = 8.314\ \mathrm{kJ.kmol^{-1}.\ K^{-1}}$	$\rho_L = 1000.0\ \mathrm{kg/m^3}$
$\Delta H_r = -1.15e4$ kJ/kmol	$T_0 = 300.0$ K
$Q = 0$ kJ/h	$C_{A0} = 4\ \mathrm{kmol/m^3}$
$C_{As} = 1.95\ \mathrm{kmol/m^3}$	$T_s = 402$ K
$C_p = 0.231\ \mathrm{kJ.kg^{-1}.\ K^{-1}}$	

When the values of Table 1 are substituted into Eq. (56), the CSTR system can be written as

$$\frac{dC_A}{dt} = 5C_{A0} - 5C_A - 8.46{\cdot}10^6 e^{\frac{-6013.95}{T}} C_A^2 \tag{57a}$$

$$\frac{dT}{dt} = 1500 - 5T + 4.21{\cdot}10^8 e^{\frac{-6013.95}{T}} C_A^2 + 0.00433Q \tag{57b}$$

We investigate the operation of the CSTR under LEMPC at an unstable steady-state characterized by $(C_{As},\ T_s) = (1.95\ \mathrm{kmol/m^3},\ 402\ \mathrm{K})$ and $(C_{A0_s}\ \ Q_s) = (4\ \mathrm{kmol/m^3},\ 0\ \mathrm{kJ/h})$. The manipulated inputs in this system are the inlet concentration of species A and the heat input rate, denoted as deviation variables $\Delta C_{A0} = C_{A0} - C_{A0_s}$ and $\Delta Q = Q - Q_s$, respectively. These manipulated inputs have the following bounds: $|\Delta C_{A0}| \leq 3.5\mathrm{kmol/m^3}$ and $|\Delta Q| \leq 5 \times 10^5$ kJ/h. Hence, the states and inputs of the closed-loop system are represented as $x^{\top} = [C_A - C_{As}\ T - T_s]$ and $u^{\top} = [\Delta C_{A0}\ \Delta Q]$, respectively. The equilibrium point of the system is located at the origin of the state-space, denoted as $(x_s{}^*,\ u_s{}^*) = (0,\ 0)$. In this study, we consider the model variations caused by catalyst deactivation during the operation of the CSTR described by Eq. (56). This deactivation leads to a reduction in the reaction pre-exponential factor k_0 within the constraint range of $0 < k_0 < 8.46 \times 10^6\ \mathrm{m^3.\ kmol^{-1}h^{-1}}$.

The control Lyapunov function $V(x) = x^{\top}Px$ is designed, where the positive definite matrix P is given as:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \tag{58}$$

Using this Lyapunov function, the closed-loop stability region Ω_ρ for the CSTR can be defined as a level set of the Lyapunov function, where $\hat{\rho} = 368$ within the region ϕ_u. By employing the controller $u = \Phi(x) \in U$, the origin can be rendered exponentially stable inside this stability region.

To numerically simulate the dynamical model described by Eq. (56), we utilize the explicit Euler method with a time step of $h_c = 10^{-4}$ h. The nonlinear optimization problem of the LEMPC formulation in Eq. (37) is solved using the Python module of the IPOPT software package (Wächter and Biegler, 2006), specifically the PyIpopt module. The sampling period for the optimization problem is set to $\Delta = 10^{-2}$ h.

The main goal of the LEMPC is to maximize the profitability of the CSTR process described by Eq. (56) by manipulating the inlet concentration ΔC_{A0} and the heat input rate ΔQ. It aims to ensure that the closed-loop state trajectories remain within the stability region $\Omega_{\hat{\rho}}$ at all times under LEMPC. The objective function of the LEMPC is designed to optimize the production rate of product B, given by:

$$l_e(\tilde{x}, u) = k_0 e^{-E/RT} C_A^2 \tag{59}$$

In addition, the LEMPC employs a material constraint, specified in Eq. (60), to maintain the average reactant material within an operating period t_p at its steady-state value C_{A0s}. This means that the averaged reactant material deviation, denoted as u_1, should equal zero:

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau)\, d\tau = 0 \quad \text{kmol/m}^3 \tag{60}$$

By incorporating this material constraint, the LEMPC ensures that the average reactant material supplied during the operating period aligns with the steady-state value, facilitating stable and controlled production while maximizing the overall profitability of the CSTR process.

5.1. *Data generation and SINDy model development*

We follow the first type of data generation and model building process described in Abdullah and Christofides (2023). Specifically, we numerically integrate the system of Eq. (56) with an integration time step of $h_c = 10^{-4}$ h and a sampling period of $\Delta = 10^{-2}$ h. 1000 different initial conditions are selected randomly with $C_A \in [0.2, 3.7]$ kmol/m^3 and $T \in [327{,}477]$ K, while the inputs are taken to be step functions with amplitudes $C_{A0} \in [0.5, 7.5]$ kmol/m^3 and $Q \in [-500, 500]$ MJ/h. We note that, although the trajectories settled at the stable steady-states, since the dynamics of the reactor are independent of the specific steady-states, the model development did not suffer. Due to the large variation of the states when settling at other steady-states, however, finite-difference estimates of the time-derivative $\dot{X}$ can be poor when the temperature, for example, goes as low as 1 K or as high as 1000 K, despite the initial conditions and desired steady-state being extremely far from these states. Hence, only trajectories where the temperature never dropped below 300 K nor rose above 500 K were retained, which yielded 53 trajectories.

The function library for SINDy is chosen to be

$$\Theta(C_A, T, C_{A0}, Q) = [1 \quad C_A \quad C_A^2 \quad T \quad C_{A0} \quad Q \quad e^{-\frac{6020}{T}} \quad C_A e^{-\frac{6020}{T}} \quad C_A^2 e^{-\frac{6020}{T}}] \tag{61}$$

The choice for the basis functions is a central problem in SINDy modeling. Due to the presence of nonlinear reaction terms, especially Arrhenius dependence of the temperature and unknown reaction order with respect to reactant A, we consider monomial terms in C_A up to second order and a negative exponential term of the reciprocal of the temperature as well as all possible interactions of these two types of terms. For the input variables, since they often impact the dynamics of the system linearly, we use linear C_{A0} and Q terms in the basis functions to start with. The choice of −6020 as the numerator of the fraction in the exponential term (dependent on the activation energy), denoted by γ, is motivated by first conducting a coarse search of values between −7000 and 0 in steps of 1000. For each value of γ, the maximum absolute error (MAE) of the validation set is calculated for both C_A and T. The results are shown in Fig. 2. As can be seen, the error is very high for larger values of γ and decreases sharply at around −6000, rising again below −6000. For even lower values of $\gamma < -7000$, the models were found to be unstable and, hence, could not be integrated and did not yield an MAE against the validation set. Subsequently, a finer search is conducted in the vicinity of $\gamma = -6000$. Specifically, values of γ between −6500 and −5500 are chosen in steps of 20, yielding 100 values to assess. Similarly, the MAE for both states are recorded for each value of γ and plotted in Fig. 3, which clearly indicates $\gamma = -6020$ is the optimal value for this system. For comparison, the exact value of γ is −6013.95, as seen in Eq. (57). When using steps of 20, the closest value of γ in the trial set is −6020, which is currently identified via the above procedure. An even finer search could be conducted to find $\gamma = -6014$ if using steps of 1 in the vicinity of −6020. However, due to the risk of overfitting and for practical considerations, especially for larger systems as detailed in Abdullah and Christofides (2023), the value of −6020 is considered adequate for this work and is selected as the basis function.

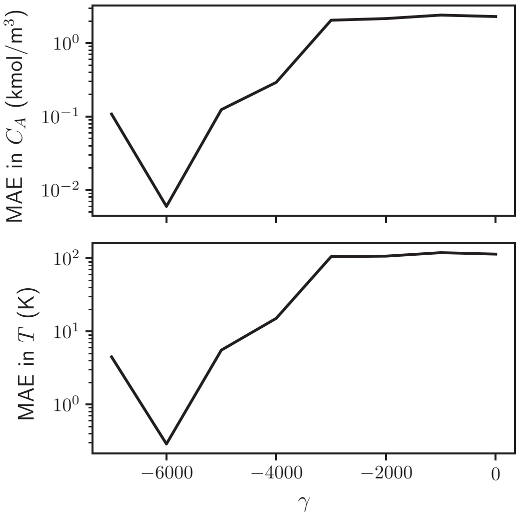

Fig. 2 – Validation error as a function of γ for a coarse search of $\gamma \in [-7000, 0]$.

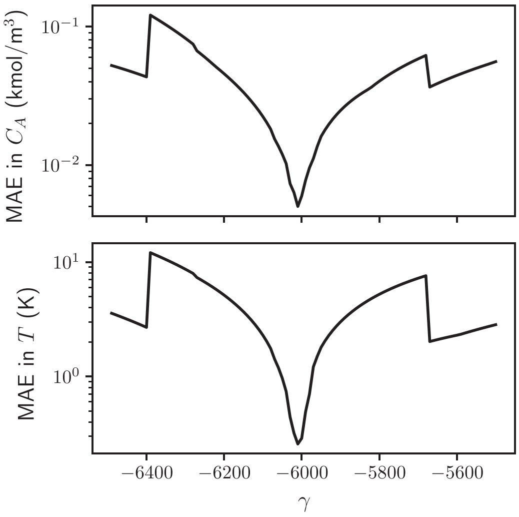

Fig. 3 – Validation error as a function of γ for a fine search of $\gamma \in [-6500, -5500]$.

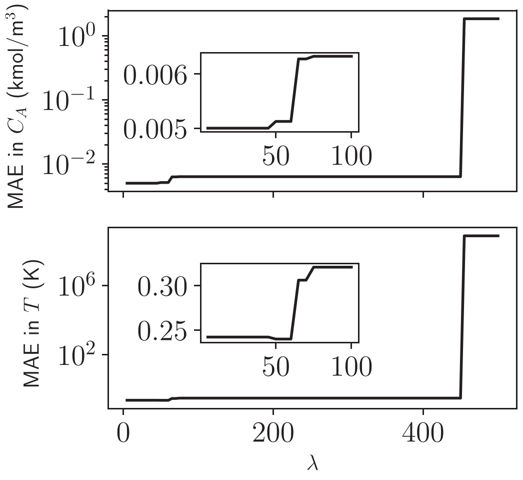

Fig. 4 – Validation error as a function of λ for a coarse search of $\lambda \in [0,500]$ with zoomed-in subplot for $\lambda \in [0,100]$.

For this system, it is imperative to scale the columns of $\Theta(X, U)$ to account for the multiple orders of magnitude of difference between the values of different columns. For instance, the Q column is on the order of magnitude of 10^5 while the final three columns of $\Theta(X, U)$, corresponding to the exponential terms, are on the order of magnitude of 10^{-6}–10^{-7}. We scale every column of Θ by its L_2 norm.

Using the above choice of candidate basis functions and library column normalization, we conduct a similar coarse-to-fine search to tune the sparsification knob λ. The first step is to conduct a coarse search, which is done for $\lambda \in [0,500]$ in steps of 5. For each value of λ, the maximum MAE for both states are plotted in Fig. 4. As can be seen, for $\lambda > 450$, all terms of the C_A ODE are zeroed, leading to an exponential increase in error, which remains constant thereafter. The lowest error is found to be at the lower values of λ, specifically below values of approximately 100. Conducting a finer search for $\lambda \in [0,100]$ in steps of 1 and recording not only the MAE in both states but also the number of terms in the model, we obtain Fig. 5. It can be seen that the MAE for both states is consistently low throughout the entire range. However, the lowest number of terms in the model is at $\lambda = 72$. While a lower value of λ corresponding to a less parsimonious model with lower MAE can be used, in sparse identification, the balance between parsimony and accuracy is important to consider. Hence, since the error does not significantly decrease when λ is reduced further, the model corresponding to $\lambda = 72$ is taken as the optimal SINDy model for the CSTR system of Eq. (56) that balances the model sparsity with accuracy.

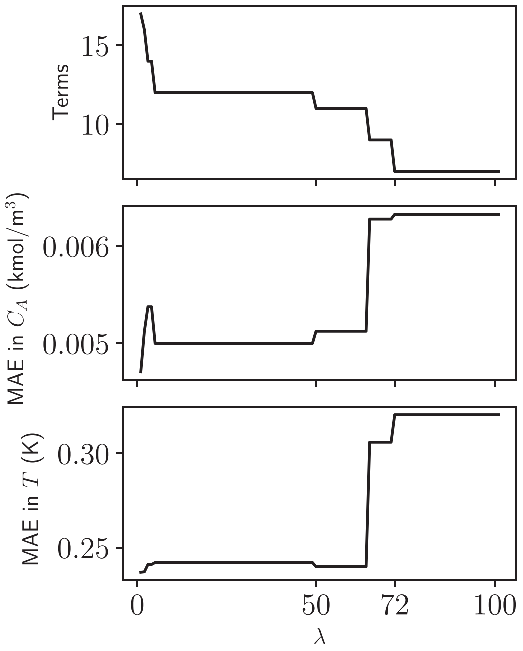

Fig. 5 – Validation error as a function of λ for a fine search of $\lambda \in [0,100]$.

The final SINDy model obtained using the 53 open-loop trajectories using the above procedure is as follows:

$$\frac{dC_A}{dt} = 5.045C_{A0} - 5.049C_A - 8.647{\cdot}10^6 e^{\frac{-6013.95}{T}} C_A^2 \tag{62a}$$

$$\frac{dT}{dt} = 1511.647 - 5.038T + 4.300{\cdot}10^8 e^{\frac{-6020}{T}} C_A^2 + 0.00436Q \tag{62b}$$

where, compared to the first-principles model of Eq. (57), every term has been correctly identified, and all coefficients except the pre-exponential constants have been identified accurately with an error below 1%. As for the pre-exponential constants, since the exponential term in the basis function is $e^{-\frac{6020}{T}}$ rather than $e^{-\frac{6013.95}{T}}$, the pre-exponential constants are slightly larger for the SINDy model, possibly to “compensate for” the more negative exponential term reducing the values of the exponential terms themselves. Therefore, as a result, the final model is extremely accurate as can be seen in the maximum absolute errors in C_A and T, which are 0.00632612 kmol/m^3 and 0.320559 K, respectively.

Remark 3. The accuracy of the SINDy model obtained is most strongly dependent on the data generation and the choice of candidate library functions. Both of these were investigated in-depth in Abdullah and Christofides (2023), Specifically, with respect to data generation, based on three different types of data generation methods studied in Abdullah and Christofides (2023), open-loop step tests in the vicinity of the steady-state of interest were found to be the most appropriate type of data generation for SINDy modeling, which is why the 53 trajectories near the steady-state, $(C_{As}, T_s) = (1.95\text{kmol/m}^3, 402\text{ K})$, were used for model building in this work. Regarding the choice of library functions, two more general frameworks for more complex or larger-scale systems were proposed in Abdullah and Christofides (2023). One effective method was found to be non-dimensionalization, which is a standard practice in engineering modeling. When all variables were scaled to similar orders of magnitudes, it was easier to identify the exact (dimensionless) model since the sensitivity of the library functions was reduced. In contrast, the most general framework proposed was to use polynomial terms in deviation variables since any nonlinearity can be expanded using Taylor expansion to the desired level of generalization using polynomials of a certain order. While the exact ODE model will not be recovered using this approach, from a modeling and control perspective, since only an accurate process model for the operating region is required, this approach is sufficient in terms of both accuracy and computation, possibly faster in terms of computation than the previous approach.

5.2. *Closed-loop simulation results*

5.2.1. *Model assessment before plant disturbances*

Using the chemical process example, we aim to illustrate the error-triggered on-line model update procedure in the presence of plant variations. However, before addressing the effect of catalyst deactivation, we first demonstrate that the initial SINDy model performs just as well as the first-principles model when there is no catalyst deactivation or disturbances. To compare their performance, we designed two LEMPC schemes, both following the structure of Eq. (37). One scheme utilized the exact first-principles model from Eq. (57) as the process model, while the other scheme employed the SINDy model from Eq. (62). Both LEMPC schemes employed the cost function defined in Eq. (59), the additional material constraint of Eq. (60), the upper and lower bounds on u_1 and u_2 described earlier, and the same Lyapunov-based controller and stability region.

For all simulations in this example, the LEMPC designs had a prediction horizon of $N = 5$, a sampling period of $\Delta = 0.01$ h, and an operating period of 20 sampling periods (t_p = 0.2 h). The first-principles LEMPC and the SINDy-based LEMPC were both applied to the CSTR model described by Eq. (56). The reactor was initialized at the unstable steady-state (C_{As}, T_s) = (1.95 kmol/m^3, 402 K), and closed-loop simulations were conducted for 20 operating periods for each case. The resulting closed-loop trajectories for the CSTR under both LEMPC schemes are depicted in Figs. 6 and 7. The average yield of the first-principles LEMPC over twenty operating periods was 62.29, compared to 62.06 for the SINDy-based LEMPC. The agreement between the trajectories for most of the simulation duration and nearly identical yields obtained from the first-principles and SINDy-based LEMPCs further illustrates that the initial SINDy model adequately captures the process behavior in the absence of plant variations. It is worth noting that the periodic nature of the trajectories aligns with prior literature, which has reported that time-varying operation can be economically advantageous for certain processes (e.g., Bailey, 1973; Silveston, 1987).

5.2.2. *LEMPC performance in the presence of plant disturbances*

Next, we investigate the effect of catalyst deactivation on the LEMPC performance using either process model. Specifically, the value of the pre-exponential constant k_0 in Eq. (56) is reduced by 20% of its original value from 8.46×10^6 m^3 kmol^{-1} h^{-1} to 6.77×10^6 m^3 kmol^{-1} h^{-1} after five operating periods (at $t = 1$ h). As a result, the CSTR system of Eq. (57) is altered to

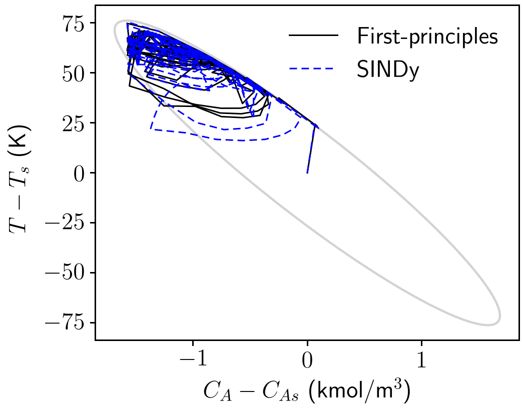

Fig. 6 – State-space trajectories of the CSTR without catalyst deactivation under an LEMPC based on the first-principles model and an LEMPC based on the initial SINDy model.

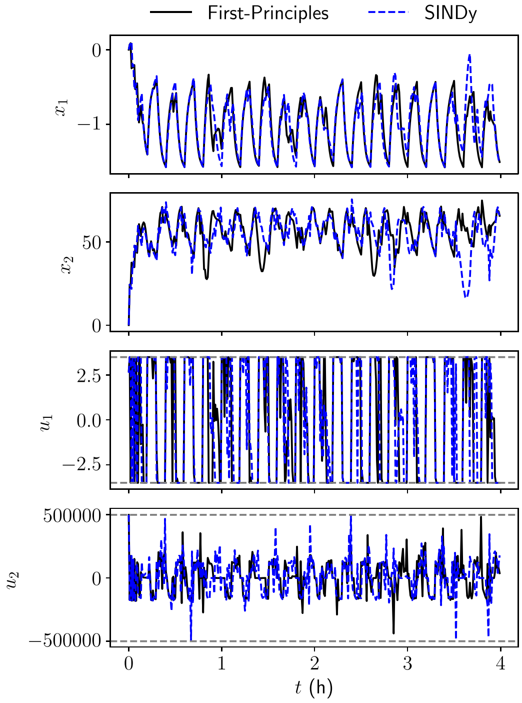

Fig. 7 – State and input trajectories of the CSTR without catalyst deactivation under an LEMPC based on the first-principles model and an LEMPC based on the initial SINDy model. The grey dashed lines represent the upper and lower bound for the inputs.

$$\frac{dC_A}{dt} = 5C_{A0} - 5C_A - \mathbf{6.77}\cdot 10^6 e^{\frac{-6013.95}{T}} C_A^2 \tag{63a}$$

$$\frac{dT}{dt} = 1500 - 5T + \mathbf{3.37}\cdot 10^8 e^{\frac{-6013.95}{T}} C_A^2 + 0.00433Q \tag{63b}$$

where the affected coefficients are boldfaced. In the remainder of the simulations, the material constraint of Eq. (60) is relaxed for the second half of the total simulation duration, from $t = 2$ h to $t = 4$ h, to allow a usage of u_1 that is 10 kmol/m^3 greater than its steady-state value per operating period. This is to allow the LEMPC to find control actions that optimize the average yield further and provide a larger space of control actions for the LEMPC to highlight any possible differences between the LEMPCs with and without SINDy model updates. Hence, the material constraint now takes the form,

$$\frac{1}{t_p}\int_0^{t_p} u_1(\tau)\, d\tau = \begin{cases} 0 \text{ kmol/m}^3, & t < 2 \\ 10 \text{ kmol/m}^3, & t \geq 2 \end{cases} \tag{64}$$

To effectively monitor the prediction error for the SINDy model in the presence of catalyst deactivation, a moving horizon error detector, as described in Section 4.1, was implemented early in the process operation. The detector was activated after collecting a sufficient amount of input/output data points, specifically N_w prior data points. At each sampling time, the detector calculated the value of e_d to assess whether it is necessary to trigger an update of the SINDy model.

The moving horizon error detector calculates the relative prediction error in the concentration of A and the reactor temperature. These errors are evaluated over the past 15

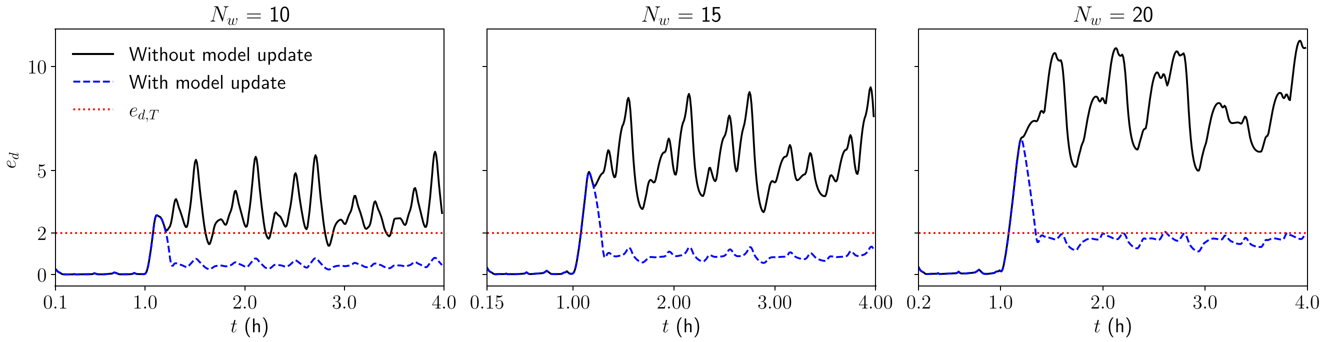

Fig. 8 – Value of error metric e_d using the detector of Eq. (65) over the simulation duration for various values of N_w with and without SINDy model updates. The dashed red line corresponds to $e_{d,T}$ = 2.

sampling periods (i.e., N_w = 15) and the current sampling time using the equation,

$$e_d(t_k) = \sum_{r=0}^{15} \frac{|x_{p,1}(t_{k-r}) - x_1(t_{k-r})|}{|x_1(t_{k-r})|} + \frac{|x_{p,2}(t_{k-r}) - x_2(t_{k-r})|}{|x_2(t_{k-r})|} \tag{65}$$

where the states from 15 sampling periods prior are used to initialize the integration of the current SINDy model over a duration of 15 sampling periods to calculate $x_{p,1}$ and $x_{p,2}$ over the entire window of N_w = 15 (single initialization). For N_w = 15, it was observed that significant discrepancies between the plant and the original SINDy model were marked by the value of e_d exceeding 2 (i.e., $e_{d,T}$ = 2). Therefore, this threshold value was chosen to initiate model updates. The determination of N_w = 15 and $e_{d,T}$ = 2 were as per the guidelines presented in Section 4.1. Specifically, if N_w = 10 (too low), the gap between the error values during normal operation and post-deactivation was not consistently large enough, and the error of the post-deactivated e_d trajectory without the model update periodically dipped very low, as shown in Fig. 8. Hence, if, for example, $e_{d,T}$ was still maintained at a value of 2 while N_w = 10, the post-deactivated e_d trajectory would often dip below 2. While this would still be possible to resolve by slightly reducing $e_{d,T}$ to, for example, 1.25, this would be too finely tuned and not easily generalized. In contrast, at N_w = 15, there is a large and consistent gap between the error trajectories with and without model updates throughout the simulation duration, with $e_d = 2 = e_{d,T}$ being a clear indicator of model performance. The gap can be further increased and the detection made even more robust by further increasing N_w. For example, N_w = 20 with $e_{d,T}$ = 3 is also a valid and possibly even more robust choice of error detection. However, as N_w is increased, the amount of data required to be kept in storage as well as the length of time the SINDy models need to be integrated also increases. Due to the longer data collection, the time elapsed before the error is triggered and the new dynamics are detected, may also increase as the error will *accumulate* at a slower pace. Therefore, for computational considerations, we chose N_w = 15 with $e_{d,T}$ = 2 as a reasonable trade-off that ensures error detection as well as reduces computational burden. Higher values of the window length N_w have been used with higher $e_{d,T}$, however, with success. For example, for the same CSTR system, in Wu et al. (2020), a relatively high error threshold of $e_{d,T}$ = 15 was used, which indicates the window length N_w must also have been quite large, such that normal process operation with an accurate process model could accumulate an error of up to 15 within N_w sampling periods, while, in our work, with N_w = 15, even under the altered process conditions with the old SINDy model used to carry out predictions, the error reached a maximum value of only 9 over the three hours of deactivated process run time.

When an on-line model update is triggered, a few preselected terms in the SINDy model are updated using the most recent input/output data. The coefficients to be updated are selected based on knowledge of the process operation or different subsets of the coefficients can be updated on a trial-and-error basis until the error is reduced to a value below $e_{d,T}$. In this case, since catalyst deactivation is a ubiquitous phenomenon in catalytic reactors, both ODE coefficients corresponding to the nonlinear reaction terms are chosen as candidates to be updated. There may be feed disturbances or other types of process changes as well. Hence, we also consider the coefficient corresponding to the reactor temperature T for both ODEs as coefficients subject to change. Hence, out of 18 coefficients (9 per ODE), 4 coefficients are updated using the model update procedure described in Section 4.2.

As for the input/output data used for the model update, although an immediate update is ideal if possible, this cannot be conducted since sufficient data from the new operating conditions must be present for the model update to succeed. On the other hand, letting the process run for an extended duration with the old model while the error remains high is undesirable as the LEMPC performance will likely deteriorate with time. Hence, the amount of data used to update the model should be the minimum amount required to accurately update the model using post-deactivation data, which will vary from one process to the other (e.g., 5000 data samples were used to update the same CSTR process in Bhadriraju et al. (2019), while 200 data samples were required to re-identify new linear empirical models in Alanqar et al. (2017)). For the reaction studied in this work, the input/output data to update the SINDy model is the state and input data of the current operating period in which the moving horizon error detector was triggered, which would correspond to N_d = 20. However, using the entire operating period of input/output data yielded poor results. Upon further investigation, the cause was found to be the minimum/maximum values of u_1 used at the beginning and end of each operating period, leading to very large changes in the states, which produce accordingly large errors in the estimates of the time-derivative, $\dot{X}$. Hence, the first and last two data points (two due to the use of second-order finite-differences) of the operating period are omitted from the model update algorithm. For this process, it is found that the use of a single operating period without the endpoints is sufficient to update the four coefficients of the SINDy model chosen to be updated. Since the catalyst deactivation, error-triggering,

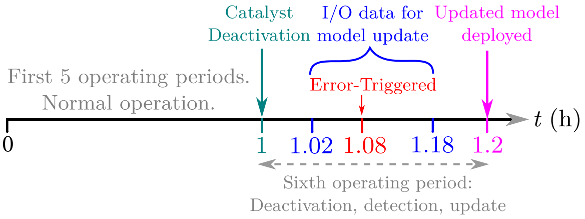

Fig. 9 – Details of the error detection and model update procedure occurring in the sixth operating period from t = 1.0 h to t = 1.2 h.

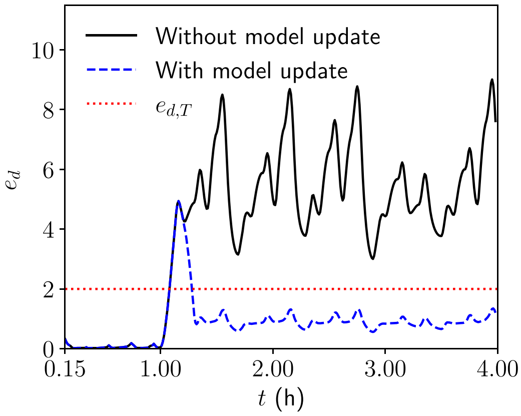

Fig. 10 – Value of error metric e_d using the detector of Eq. (65) and the integrated LEMPC design with error-triggered on-line model updates at each sampling time (the time axis starts from when there is sufficient data to begin calculating e_d, i.e., after N_w = 15 data points are collected).

and model update all occur during the sixth operating period, a detailed illustration of the model update procedure is shown in Fig. 9.

The LEMPC updates the SINDy process model at the end of the sixth operating period to reflect the catalyst deactivation and continues to operate for the remainder of the 4 h of simulation duration. The moving horizon error detector is used to monitor the modeling error throughout the process, and the value of e_d over the simulation duration is shown in Fig. 10. It can be observed that the error e_d exceeds the predetermined threshold of $e_{d,T} = 2$ at t = 1.08 h, during the sixth operating period, and then rapidly decreases below the threshold at t = 1.2 h following the SINDy model update. On the other hand, the value of e_d calculated with the initial SINDy model (without update) continues to remain above $e_{d,t}$ for the remainder of the simulation, indicating that the initial SINDy model cannot very accurately predict the process states once the catalyst is deactivated. Since Fig. 10 spans the entire simulation duration, and the exact details may be difficult to observe, a closer look at the sixth operating period, where all the changes occur, was provided in Fig. 9.

The closed-loop state and input trajectories for the LEMPCs with and without SINDy model updates are shown in Fig. 11 and Fig. 12, which depict the state-space and time-varying trajectories, respectively. From Fig. 11, it is observed that the closed-loop state quickly approaches the boundary of the stability region $\Omega_{\hat{\rho}}$ by using maximal input actions early on in the trajectory and continues to remain as close as possible to the boundary representing the highest possible temperature and lowest possible concentration of A, as this combination maximizes the production of B in the process. The closed-loop state is always maintained within the stability region $\Omega_{\hat{\rho}}$ for 99% of the simulation duration under both LEMPC. The manipulated input profiles indicate, as expected, the cyclic use of u_1, i.e., at the beginning of each operating period, the LEMPC uses the maximum value of ΔC_{A0} allowed to maximize production of B, while reducing the consumption of the reactant at the end of each period to meet the material constraint. For the second half of the simulation, however, due to the relaxation of the material constraint as per Eq. (64), the sum of u_1 per operating period is equal to 10, which is why the initial period of maximum u_1 consumption is observed to be greater than the minimal consumption at the end of the last 10 operating periods.

Finally, the total economic benefits achieved over the 20 operating periods are calculated for three scenarios: the SINDy-based LEMPC without on-line updates, SINDy-based LEMPC with the on-line update, and steady-state operation

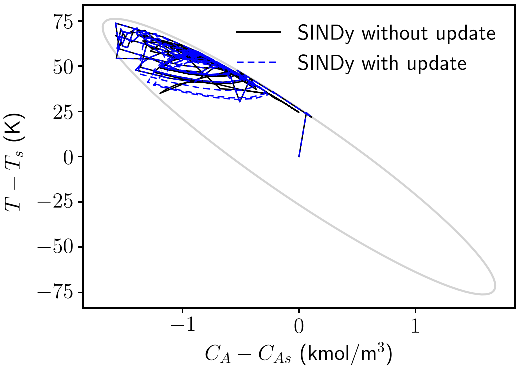

Fig. 11 – State trajectories for the closed-loop CSTR under the LEMPC of Eq. (37) with and without the online update of the SINDy model for the initial condition (0, 0). They grey ellipse represents the stability region $\Omega_{\hat{\rho}}$.

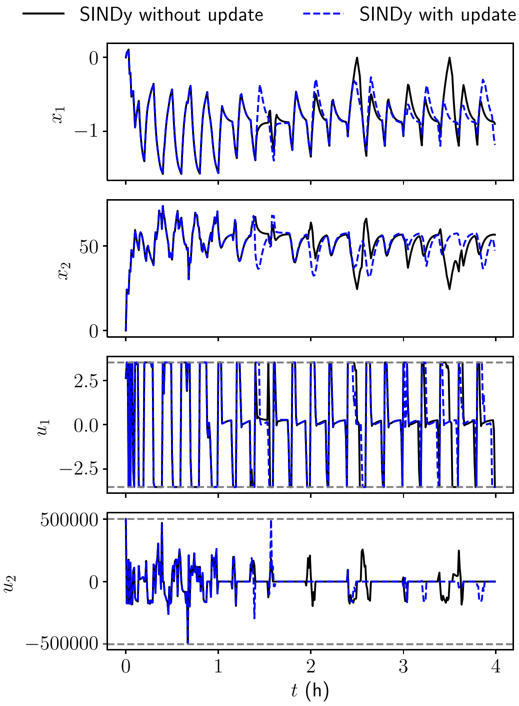

Fig. 12 – State and manipulated input profiles for the initial condition (0, 0) under the LEMPC of Eq. (37) with and without the online update of the SINDy, respectively. The grey dashed lines represent the upper and lower bound for the inputs.

where the system of Eq. (56) is operated at (C_{As}, T_s) for the entire duration. The economic benefits are evaluated using the equation,

$$L_E = \int_0^{5t_f} l_e(x, u)\, dt \tag{66}$$

The closed-loop system under LEMPC with and without SINDy model updates achieves $L_E = 63.45$ and $L_E = 63.72$, respectively, within the four-hour period, while the steady-state operation yields $L_E = 34.79$. This comparison demonstrates that time-varying operation of the system of Eq. (56) under the LEMPC of Eq. (37) with or without on-line updating of SINDy models results in much higher economic benefits of 82% compared to steady-state operation. The economic benefits of the LEMPC with model updates are also 0.42% higher than the benefits of the LEMPC without model updates. Although the difference is small in this specific scenario, this is highly dependent on both the system as well as the model fidelity. For the same CSTR system modeled using RNN models, it was demonstrated in Wu et al. (2020) that a large improvement is possible if the performance of the initial process model deteriorates severely after disturbances are introduced to the system. In our work, however, possibly since the model structure and coefficients were closely identified in the initial SINDy model, the performance deterioration of the LEMPC even without model updates was not sufficiently high to allow the LEMPC with model updates to improve upon. To investigate this, the CSTR was also simulated under an LEMPC using the first-principles model of Eq. (56) but with the relaxed material constraint of Eq. (64) and the process model updated immediately upon deactivation (i.e., at $t = 1$ h, not $t = 1.2$ h). Practically, the first-principles model, the exact moment of catalyst deactivation, and the exact percentage of catalyst deactivation are unknown. However, this is a hypothetical best-case scenario that should yield the highest possible economic benefits possible for this CSTR specification since the LEMPC process model “sees” the exact change immediately upon its occurrence. The economic benefits of this case was found to be $L_E = 63.53$. Since the value is actually lower than L_E for the LEMPC with SINDy, it can be inferred that all three LEMPC perform nearly identically and the minor improvements are more likely due to numerical issues or due to a very small number of sampling periods being significantly different from each other. Hence, the reason for the small, possible improvement of the LEMPC with SINDy model updates can be contributed to the LEMPC performance in general being upper-bounded by other factors due to the specific process, parameters, model structure, and initial SINDy model fidelity of this study.

Remark 4. In this work, a Lyapunov-based tracking MPC using a SINDy model to drive the process to a steady-state was not presented because the development of the SINDy model and its real-time adaptation is similar to the case of economic MPC and would not add any new methodological and/or implementation insights.

6. Conclusions

This study introduces a novel approach for on-line updates of nonlinear ODE models obtained using sparse identification to embed into a model predictive controller (MPC) for nonlinear process systems. The proposed methodology incorporates an error-triggering mechanism through a moving horizon error detector, which evaluates the relative prediction error within a specified horizon. When the prediction error surpasses a predefined threshold, the error-triggering mechanism is activated and the most recent yet sufficient input/output data is used to update specific coefficients of the SINDy model using an efficient algorithm. The results showcase the capability of the proposed approach to improve state predictions crucial for MPC in the presence of plant variations. A chemical process example under the framework of Lyapunov-based empirical model predictive control is employed to illustrate the effectiveness and implementation of real-time updates for the SINDy models. It was demonstrated that a small amount of real-time data could accurately update the SINDy model to adjust to the disturbances, greatly reducing thereby the model prediction error when monitoring the process via the moving horizon error detector. A slight improvement of the economic benefits was also observed when the SINDy model in the LEMPC was updated in real-time, compared to the SINDy-based LEMPC without model updates, although the improvement was limited by the catalyst deactivation and other compounding factors.

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.

Acknowledgement

Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

References

Abdullah, F., Christofides, P.D., 2023. Data-based modeling and control of nonlinear process systems using sparse identification: An overview of recent results. Comput. Chem. Eng. 174, 108247.

Abdullah, F., Wu, Z., Christofides, P.D., 2021a. Data-based reduced-order modeling of nonlinear two-time-scale processes. Chem. Eng. Res. Des. 166, 1–9.

Abdullah, F., Wu, Z., Christofides, P.D., 2021b. Sparse-identification-based model predictive control of nonlinear two-time-scale processes. Comput. Chem. Eng. 153, 107411.

Abdullah, F., Alhajeri, M.S., Christofides, P.D., 2022a. Modeling and Control of Nonlinear Processes Using Sparse Identification: Using Dropout to Handle Noisy Data. Ind. Eng. Chem. Res. 61, 17976–17992.

Abdullah, F., Wu, Z., Christofides, P.D., 2022b. Handling noisy data in sparse model identification using subsampling and co-teaching. Comput. Chem. Eng. 157, 107628.

Alanqar, A., Durand, H., Christofides, P.D., 2017. Error-triggered on-line model identification for model-based feedback control. AIChE J. 63, 949–966.

Amrit, R., Rawlings, J.B., Angeli, D., 2011. Economic optimization using model predictive control with a terminal cost. Annu. Rev. Control 35, 178–186.

Bai, Z., Wimalajeewa, T., Berger, Z., Wang, G., Glauser, M., Varshney, P.K., 2015. Low-Dimensional Approach for Reconstruction of Airfoil Data via Compressive Sensing. AIAA J. 53, 920–933.

Bailey, J.E., 1973. Periodic operation of chemical reactors: a review. Chem. Eng. Commun. 1, 111–124.

Bhadriraju, B., Narasingam, A., 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.

Bhadriraju, B., Bangi, M.S.F., Narasingam, A., Kwon, J.S.I., 2020. Operable adaptive sparse identification of systems: Application to chemical processes. AIChE J. 66, e16980.

Brunton, S.L., Tu, J.H., Bright, I., Kutz, J.N., 2014. Compressive Sensing and Low-Rank Libraries for Classification of Bifurcation Regimes in Nonlinear Dynamical Systems. SIAM J. Appl. Dyn. Syst. 13, 1716–1732.

Brunton, S.L., Noack, B.R., 2015. Closed-loop turbulence control: Progress and challenges. Appl. Mech. Rev. 67.

Christofides, P.D., El-Farra, N.H., 2005. Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty. Constraints and Time-Delays. Springer-Verlag,, Berlin, Germany.

Ellis, M., Durand, H., Christofides, P.D., 2014. A tutorial review of economic model predictive control methods. J. Process Control 24, 1156–1178.

Farsi, M., Liu, J., 2020. Structured online learning-based control of continuous-time nonlinear systems. IFAC-Pap. 53, 8142–8149.

Garcia, C.E., Prett, D.M., Morari, M., 1989. Model predictive control: Theory and practice-A survey. Automatica 25, 335–348.

Ge, S.S., Wang, C., 2004. Adaptive neural control of uncertain MIMO nonlinear systems. IEEE Trans. Neural Netw. 15, 674–692.

Ge, H.W., Liang, Y.C., Marchese, M., 2007. A modified particle swarm optimization-based dynamic recurrent neural network for identifying and controlling nonlinear systems. Comput. Struct. 85, 1611–1622.

Heidarinejad, M., Liu, J., Christofides, P.D., 2012. Economic model predictive control of nonlinear process systems using Lyapunov techniques. AIChE J. 58, 855–870.

Hoffmann, M., Fröhner, C., Noé, F., 2019. Reactive SINDy: Discovering governing reactions from concentration data. J. Chem. Phys. 150, 025101.

Huang, R., Harinath, E., Biegler, L.T., 2011. Lyapunov stability of economically oriented NMPC for cyclic processes. J. Process Control 21, 501–509.

Huang, K., Tao, Z., Liu, Y., Wu, D., Yang, C., Gui, W., 2023, Error-Triggered Adaptive Sparse Identification for Predictive Control and Its Application to Multiple Operating Conditions Processes.IEEE Transactions on Neural Networks and Learning Systems (in press).

Kaiser, E., Kutz, J.N., Brunton, S.L., 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proc. R. Soc. A 474, 20180335.

Khalil, H.K., 2002. Nonlinear Systems, 3rd ed..,. Prentice Hall,, Upper Saddle River, New Jersey.

Lin, Y., Sontag, E.D., 1991. A universal formula for stabilization with bounded controls. Syst. Control Lett. 16, 393–397.

Lin, Y., Sontag, E., Wang, Y., 1996. A smooth converse Lyapunov theorem for robust stability. SIAM J. Control Optim. 34, 124–160.

Mackey, A., Schaeffer, H., Osher, S., 2014. On the compressive spectral method. Multiscale Model. Simul. 12, 1800–1827.

Manzoor, T., Pei, H., Sun, Z., Cheng, Z., 2022. Model Predictive Control Technique for Ducted Fan Aerial Vehicles Using Physics-Informed. Mach. Learn. Drones 7, 4.

Massera, J.L., 1956. Contributions to stability theory. Ann. Math. 64, 182–206.

Mayne, D.Q., 2014. Model predictive control: Recent developments and future promise. Automatica 50, 2967–2986.

Narasingam, A., Kwon, J.S.I., 2018. Data-driven identification of interpretable reduced-order models using sparse regression. Comput. Chem. Eng. 119, 101–111.

Ozolinš, V., Lai, R., Caflisch, R., Osher, S., 2013. Compressed modes for variational problems in mathematics and physics. Proc. Natl. Acad. Sci. 110, 18368–18373.

Proctor, J.L., Brunton, S.L., Brunton, B.W., Kutz, J.N., 2014. Exploiting sparsity and equation-free architectures in complex systems. Eur. Phys. J. Spec. Top. 223, 2665–2684.

Quade, M., Abel, M., NathanKutz, J., Brunton, S.L., 2018. Sparse identification of nonlinear dynamics for rapid model recovery. Chaos: An Interdisciplinary. J. Nonlinear Sci. 28, 063116.

Sarić, A.T., Sarić, A.A., Transtrum, M.K., Stanković, A.M., 2020. Symbolic regression for data-driven dynamic model refinement in power systems. IEEE Trans. Power Syst. 36, 2390–2402.

Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S., 2013. Sparse dynamics for partial differential equations. Proc. Natl. Acad. Sci. 110, 6634–6639.

Silveston, P.L., 1987. Periodic operation of chemical reactors - A review of the experimental literature. Sdhan 10, 217–246.

Stanković, A.M., Sarić, A.A., Sarić, A.T., Transtrum, M.K., 2020, Data-driven symbolic regression for identification of nonlinear dynamics in power systems, in: 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada.1–5.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57.

Wang, W.X., Yang, R., Lai, Y.C., Kovanis, V., Grebogi, C., 2011. Predicting Catastrophes in Nonlinear Dynamical Systems by Compressive Sensing. Phys. Rev. Lett. 106, 154101.

Wang, J., Moreira, J., Cao, Y., Gopaluni, B., 2022, Time-Variant Digital Twin Modeling through the Kalman-Generalized Sparse Identification of Nonlinear Dynamics, 2022 American Control Conference (ACC), IEEE.5217–5222.

Wu, Z., Christofides, P.D., 2019. Economic Machine-Learning-Based Predictive Control of Nonlinear Systems. Mathematics 7, 494.

Wu, Z., Christofides, P.D., 2021. Process Operational Safety and Cybersecurity. Springer,.

Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019a. Machine learning-based predictive control of nonlinear processes. Part I: Theory. AIChE J. 65, e16729.

Wu, Z., Tran, A., Rincon, D., Christofides, P.D., 2019b. Machine learning-based predictive control of nonlinear processes. Part II: Computational implementation. AIChE J. 65, e16734.

Wu, Z., Rincon, D., Christofides, P.D., 2020. Real-time adaptive machine-learning-based predictive control of nonlinear processes. Ind. Eng. Chem. Res. 59, 2275–2290.