



Control Lyapunov barrier function-based predictive control of nonlinear systems using physics-informed recurrent neural networks

Mohammed S. Alhajeri^{a,*}, Fahim Abdullah^b, Panagiotis D. Christofides^{c,d}

^a Department of Chemical Engineering, Kuwait University, P.O.Box 5969, Safat, 13060, Kuwait

^b Computational Sciences and Engineering Division, Oak Ridge National Laboratory, 1 Bethel Valley Road, Oak Ridge, TN, 37830, USA

^c Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA

^d Department of Electrical and Computer Engineering, University of California, Los Angeles, CA, 90095-1592, USA

ARTICLE INFO

Keywords:

Process control
Model predictive control
Nonlinear processes
Machine learning
Physics-informed neural networks

ABSTRACT

Control Lyapunov-barrier functions (CLBF) have been effectively employed in model predictive control (MPC) to ensure both closed-loop stability and operational safety in input-constrained nonlinear systems. In this work, we propose a novel CLBF-MPC framework that leverages physics-informed partially-connected recurrent neural network (PCRNN) models to enhance prediction accuracy by incorporating a priori process structural knowledge. The PCRNN architecture, designed based on known process interconnections, enables improved approximation of nonlinear dynamics which, when incorporated into a CLBF-MPC, allows for improved process operational safety by avoidance of unsafe regions in the state-space that would normally be encountered under regular MPC operation. The effectiveness of the proposed PCRNN-based CLBF-MPC is demonstrated through application to a chemical process example, where it achieves superior predictive performance and successfully maintains system safety by fully avoiding the bounded unsafe region, unlike the fully-connected black-box RNN model when incorporated into the same CLBF-MPC.

1. Introduction

Ensuring safety within the chemical process industry is critical, given the profound impact that unsafe operations can have on both human lives and the environment. During the early years of the twenty-first century, three major chemical incidents in the United States alone led to extensive financial losses, numerous injuries, and tragic fatalities (Sanders, 2015). These severe events underscore the urgent need to develop novel methods for enhancing operational safety (Incidents, 2016). While designing inherently safer processes at the outset is one approach, refining the design of process control systems remains vital for ensuring day-to-day operational safety (Crowl and Louvar, 2011).

Model predictive control (MPC) has become a cornerstone in the control of industrial chemical plants because of its ability to handle complex, multi-input multi-output systems while optimizing process performance (Rawlings and Mayne, 2009). In particular, Lyapunov-based MPC formulations (Mhaskar et al., 2006; Muñoz de la Peña and Christofides, 2008) provide guarantees on system stabilizability and feasibility by explicitly characterizing the region of attraction (often called the stability region) using an appropriately-designed Lyapunov-based control law.

Over the last two decades, a novel approach known as control Lyapunov-barrier function (CLBF)-based control has emerged to simultaneously address both stability and safety concerns (Tee et al., 2009; Niu and Zhao, 2013; Jankovic, 2017). By integrating a control Lyapunov function (CLF) with a control barrier function (CBF)—through methods such as a weighted average (Romdlony and Jayawardhana, 2016) or a quadratic programming strategy (Ames et al., 2014)—CLBFs are capable of characterizing unsafe regions in the state-space. However, because CBFs describe unsafe regions as open and bounded sets rather than as direct state constraints, conventional MPC schemes may struggle to adequately address safety requirements expressed through CBFs. To address merging of CBFs into MPC frameworks to flexibly define unsafe regions while ensuring the system remains in safe operating zones, Wu et al. (2019) proposed a CLBF-MPC formulation that integrated a CLBF with MPC, which was able to provide recursive feasibility, guaranteed process safety, closed-loop stability even in the presence of input limitations, and persistent avoidance of unsafe regions in the state space as the CLBF-MPC drove the state to the setpoint.

Recent advancements in machine learning (ML) have paved the way for significant enhancements in MPC strategies. By incorporating

* Corresponding author.

E-mail addresses: mohammed.alhajeri@ku.edu.kw (M.S. Alhajeri), ABDULLAHF@ORNL.GOV (F. Abdullah), pdc@seas.ucla.edu (P.D. Christofides).

data-driven models, machine learning algorithms can capture complex nonlinear dynamics when trained with large data sets, thereby enhancing prediction accuracy and overall control performance. The literature covering the use of ML in control systems spans a broad range but, for dynamical systems, is primarily comprised of recurrent neural networks (RNNs) and its variants as well as, more recently, transformers for sequence-to-sequence predictions. A recent survey of ML-based MPC can be found in Ren et al. (2022). While the basic case of RNN-based MPC has been studied in-depth, there is a growing interest, especially over the last decade, in more physics-based neural network modeling due to the promising improvements in model performance when ML methods are incorporated with known physical laws. For example, partially-connected RNNs that incorporate knowledge of the process structure in the network structure were found to outperform fully-connected RNNs that neglected domain knowledge in Alhajeri et al. (2022a), with a rigorous mathematical proof based on statistical machine learning conducted in Alhajeri et al. (2023). Furthermore, process structural knowledge-based transfer learning to improve the RNN model training in large systems that can be split into subsystems was shown in Alhajeri et al. (2024). Shah et al. (2022) presented a hybrid deep neural network by integrating a kinetics model for an industrial-scale fermentation process, experimentally validating model predictions and showing that time-varying parameter dependencies can be accurately captured with significantly fewer parameters than a strictly data-driven model. Their hybrid model reduced the root mean squared error (MSE) of product, substrate and biomass concentration predictions by over 25% relative to the kinetic model alone, and yielded a relative error of less than 10% on validation datasets for key states across multiple operating conditions. Silva et al. (2021) developed a hybrid modeling approach combining first-principles exergy analysis with machine learning to predict chemical exergy of biological molecules, demonstrating improved accuracy and interpretability relative to purely empirical or purely mechanistic models. Their method achieved an R^2 of approximately 0.98 and average absolute error below 0.6 MJ/kg when estimating specific chemical exergy across a diverse set of biomolecules. Xiao and Wu (2023) proposed a physics-informed transfer learning framework in which pretrained subsystem RNNs were embedded into a larger RNN model for the full process network, and process-structure and first-principles knowledge were enforced to improve prediction accuracy under limited data. The proposed transfer learning-based RNN models achieved testing errors of 2.014×10^{-5} for the first CSTR and 1.25×10^{-5} for the second CSTR, while also significantly reducing the computation time required for characterizing the stability region using the TL model compared to conventional full search methods. While the previous works have demonstrated improved performance of physics-informed RNNs when incorporated into MPC under normal as well as noisy conditions and to chemical/biological systems, to the best of our knowledge, the case of process safety and incorporation of physics-informed machine learning models in CLBF-MPC has not been studied. In this work, we present a novel CLBF-MPC approach using a physics-informed partially-connected RNN (PCRNN) as the process model in MPC and demonstrate the effectiveness of the CLBF-MPC at preventing the states from entering the unsafe region when using the PCRNN.

The remainder of this paper is structured as follows: Section 2 details the class of systems under consideration, RNN models, the stabilizability assumption, and introduces the concepts of the unsafe region characterization and barrier functions. In Section 3, we described physics-informed RNN models, while the incorporation of the PCRNN into the CLBF-MPC is presented in Section 4. Finally, Section 5 presents a nonlinear chemical process example that demonstrates the practical applicability of the proposed PCRNN-based CLBF-MPC controller, with Section 6 summarizing the conclusions.

2. Preliminaries

2.1. Notations

Throughout this manuscript, the notation $\|\cdot\|$ represents the Euclidean norm of a vector. The notation $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ denotes the standard Lie derivative. For set subtraction, “ $-$ ” is used, i.e., $A - B = \{x \in \mathbb{R}^n | x \in A, x \notin B\}$. A function $f(x)$ is of class C^1 if it is continuously differentiable. For given positive real numbers δ and ϵ , $B_\delta(\epsilon) := \{x \in \mathbb{R}^n | |x - \epsilon| < \delta\}$ is an open ball around ϵ with radius of δ .

2.2. Class of systems

We consider a class of multi-input multi-output (MIMO) nonlinear continuous-time systems represented by the following state-space form:

$$\dot{x} = F(x, u) := f(x) + g(x)u \quad (1)$$

where the state vector of the system is $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, and the manipulated input vector is $u = [u_1, \dots, u_m]^T \in \mathbb{R}^m$. $F(x, u)$ represents a nonlinear vector function of x and u which is assumed to be sufficiently smooth. The constraints on control inputs are given by $u \in U := \{u_i^{\min} \leq u_i \leq u_i^{\max}\}$. The functions $f(\cdot)$ and $g(\cdot)$ are nonlinear vector and matrix functions of dimensions $n \times 1$ and $n \times m$, respectively.

2.3. Stabilizability assumption

For closed-loop stability considerations, we assume that there exists a positive definite control Lyapunov function V for the system of Eq. (1) that satisfies the small control property (i.e., for every $\epsilon > 0$, $\exists \delta > 0$, s.t. $\forall x \in B_\delta(0)$, $\exists u$ that satisfies $|u| < \epsilon$ and $L_f V(x) + L_g V(x)u < 0$, (Sontag, 1989)) and the following condition holds :

$$L_f V(x) < 0, \forall x \in z \in \mathbb{R}^n - \{0\} | L_g V(z) = 0$$

The CLF assumption indicates that there is a stabilizing feedback control law $\Phi(x) \in U$ for the system of Eq. (1) that ensures the origin of the closed-loop system is asymptotically stable for all x within a neighborhood of the origin, meaning that Eq. (1) holds when $u = \Phi(x) \in U$. A common example in practice of such a feedback control law is the Sontag law (Lin and Sontag, 1991). Using the Lyapunov-based control law $\Phi(x)$, one defines the set ϕ_u as all states $x \in \mathbb{R}^n$ for which the time-derivative of $V(x)$ is strictly negative when the input is $u = \Phi(x) \in U$, thereby ensuring a decreasing “energy” under the constrained input law.

This set can be written as $\phi_u = \{x \in \mathbb{R}^n | \dot{V} < 0, u = \Phi(x) \in U\}$ which identifies those states where the Lyapunov condition holds under the admissible control. By then introducing the subset Ω_b as the level set of $V(x)$ contained within ϕ_u , namely $\Omega_b := \{x \in \phi_u | V(x) \leq b, b > 0\}$ so that Ω_b collects all states in ϕ_u whose Lyapunov value does not exceed a positive threshold, using standard results on Lyapunov sublevel sets, any such bounded level set is forward-invariant, i.e., if $x(0)$ lies in Ω_b , then under the dynamics with $u = \Phi(x)$, the trajectory cannot exit Ω_b for all $t \geq 0$. Therefore, for every initial condition $x_0 \in \Omega_b$, the state x of the nominal system under $u = \Phi(x) \in U$ remains in the forward-invariant region Ω_b for all $t \geq 0$, guaranteeing both stability and adherence to the input constraints.

2.4. Recurrent neural network

As mentioned in the introduction, RNN models illustrated in Fig. 1 are suitable for modeling time-series data. The recursive action in the hidden layer neurons allows the RNN to hold the memory of the previous states, such that it can adequately approximate a time-series dataset. In this work, an RNN model is used to approximate the nonlinear system

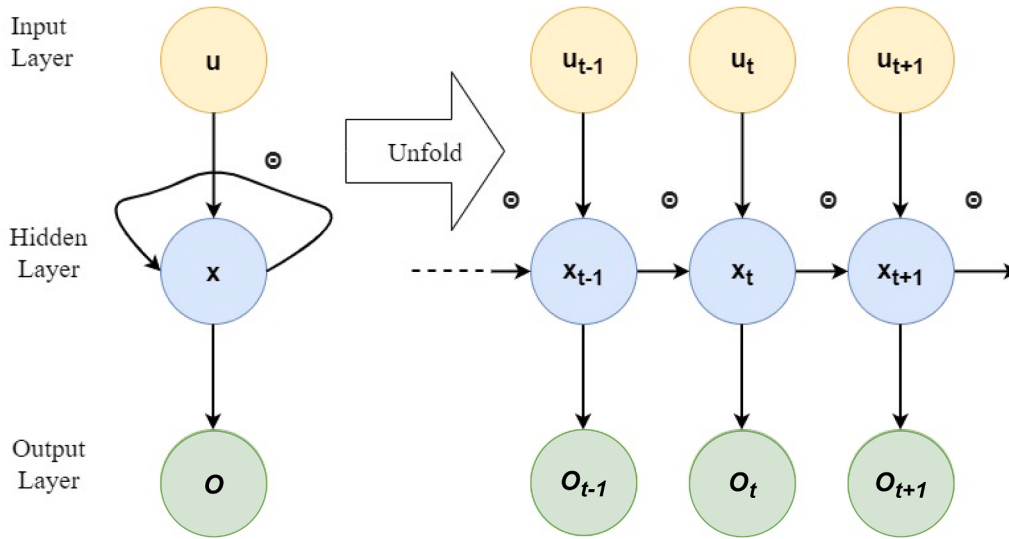


Fig. 1. A schematic of a recurrent neural network.

of Eq. (1) using process operational data and can be represented as:

$$\dot{\bar{x}} = F_{rnn}(\bar{x}, u) := A\bar{x} + \Theta^T y \quad (2)$$

where $\bar{x} = [\bar{x}_1, \dots, \bar{x}_n]$ is the state vector of the RNN, and the manipulated input vector is $u = [u_1, \dots, u_m]$. As a vector of both $\bar{x}$ and u , the vector y is defined as $[y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}] = [\sigma(\bar{x}_1), \dots, \sigma(\bar{x}_n), u_1, \dots, u_m] \in \mathbb{R}^{n+m}$. The notation $\sigma(\cdot)$ represents a nonlinear activation function, such as the hyperbolic tangent function, used in the hidden layer(s). $A = \text{diag}[-a_1, \dots, -a_n]$ is a diagonal matrix with negative coefficients ($a_i > 0$) such that each state $\bar{x}$ is stable in terms of bounded-input bounded-state stability. The notation $\Theta = [\theta_1, \dots, \theta_n] \in \mathbb{R}^{(n+m) \times n}$ denotes a matrix containing associated weights to be optimized during the neural network training process. Therefore, the vector $\theta_i = b_i[w_{i1}, \dots, w_{i(n+m)}]$ is an element of Θ where b_i is a constant, and w_{ij} stands for the weight on the connection from the j th input to the i th neuron where $j = 1, \dots, (n+m)$ and $i = 1, \dots, n$.

The RNN is subsequently trained using a conventional learning approach previously outlined in Alhajeri et al. (2021). The datasets for training, validation, and testing are created through extensive open-loop simulations of the process model, ensuring sufficient variation in initial conditions and control actions. Specifically, the continuous-time system described by Eq. (1) is numerically integrated via the explicit Euler method using a suitably small integration step size h_c , while the control inputs u are applied in a sample-and-hold manner, meaning $u(t) = u(t_k)$ for all $t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ and Δ is the sampling period. Given that RNNs are known for their strong ability to approximate nonlinear dynamical behavior from time-series data (Park and Sandberg, 1991; Chen and Chen, 1995), the RNN model can be trained using either all or a subset of the data points generated at each integration step h_c within a sampling interval to accurately learn the system's state evolution. Moreover, it is essential that the RNN model achieves a sufficiently low modeling error v during training, where $|v| = |F(x, u) - F_{rnn}(\bar{x}, u)| \leq \gamma|x| \leq v_{\min}$ with positive constants γ and $v_{\min}$, to ensure that it reliably represents the process dynamics within the defined operating region.

Remark 1. The modeling error v is not constant across all possible states and inputs. However, by restricting operation to the stability region Ω_p , which is compact, both states and inputs remain bounded. Therefore, training the RNN on data sampled from Ω_p ensures that the modeling error v can be uniformly upper bounded by a sufficiently small positive constant $v_{\min}$ across this region.

2.5. Characterization of unsafe regions

We assume the existence of a set $D_{us} \subset \mathbb{R}^n$ representing states where it is unsafe for the system of Eq. (1) to operate, and a safe region $\mathcal{X}_0 := \{x \in \mathbb{R}^n - D_{us}\}$ such that $\mathcal{X}_0 \cap D_{us} = \emptyset$ and $0 \in \mathcal{X}_0$, within which both closed-loop stability and operational safety can be simultaneously ensured. These are formalized as follows:

Definition 1 (Wu et al. (2019), Wu and Christofides (2019)). Consider the system described in Eq. (1) with input constraints $u \in U$. If there exists a control law $u = \Phi(x) \in U$ such that for any initial condition $x(t_0) = x_0 \in \mathcal{X}_0$, the trajectory $x(t)$ remains within $\mathcal{X}_0$ for all $t \geq 0$, and the origin of the closed-loop system of Eq. (1) can be rendered asymptotically stable, then the control law $\Phi(x)$ is said to maintain the system state within a safe and stable region $\mathcal{X}_0$ at all times.

The unsafe region D_{us} is identified through safety analysis based either on first-principles process models or on operational data. According to Wu and Christofides (2019), there are two general classes of unsafe regions: (1) bounded unsafe sets, which are commonly encountered in fields such as motion planning for robotics and autonomous vehicles; and (2) unbounded unsafe sets, which are prevalent in chemical process systems, e.g., operating conditions where a reactor's temperature exceeds a critical threshold, signaling unsafe operation.

This work focuses on the first case, i.e. bounded unsafe regions. A predictive controller based on control Lyapunov-barrier functions CLBF and machine learning models is developed to guarantee that the closed-loop state can be steered toward the steady state while avoiding entry into the unsafe region.

2.6. Barrier function

The concept of control Lyapunov-barrier functions (CLBFs) has been employed in the works of Romdlony and Jayawardhana (2016) and Wu et al. (2019) to develop stabilizing controllers that enforce the state of nonlinear systems, such as the one described in Eq. (1), to move toward the origin while ensuring avoidance of unsafe regions within the state-space. These CLBFs integrate the stabilizing properties of CLFs with the safety-enforcing characteristics of CBFs, resulting in a unified framework that facilitates both objectives. In the context of systems modeled by recurrent neural networks (RNNs), as represented by Eq. (2), and subject to input constraints, constrained CLBFs are defined as follows to ensure that the control inputs remain within permissible bounds while

achieving the desired stability and safety objectives:

$$W_c(x) \geq \rho, \forall x \in D_{us} \subset \phi_{uc} \quad (3a)$$

$$L_{\hat{f}} W_c(x) < 0, \forall x \in \{z \in \phi_{uc} - (D_{us} \cup \{0\} \cup \mathcal{X}_e) \mid L_{\hat{g}} W_c(z) = 0\} \quad (3b)$$

$$\mathcal{U}_\rho := \{x \in \phi_{uc} \mid W_c(x) \leq \rho\} \neq \emptyset \quad (3c)$$

$$\overline{\phi_{uc}} - (D_{us} \cup \mathcal{U}_\rho) \cap \overline{D_{us}} = \emptyset \quad (3d)$$

where $\rho \in \mathbb{R}$, and $\mathcal{X}_e := \{x \in \phi_{uc} - (D \cup \{0\}) \mid \partial W_c(x)/\partial x = 0\}$ is a set of states for the RNN model of Eq. (2) where $L_{\hat{f}} W_c(x) = 0$ (for $x \neq 0$) due to $\partial W_c(x)/\partial x = 0$. $\hat{f}$ and $\hat{g}$ are from the RNN model in the form of Eq. (2). The methodology for constructing the constrained CLBF as presented in Eq. (3) is detailed in the works of Romdlony and Jayawardhana (2016) and Wu et al. (2019). In this approach, a CLF and a CBF are independently designed and subsequently combined through a weighted average to form the CLBF. It is assumed that there exists a feedback control law $u = \Phi_{nn}(x) \in U$ that ensures exponential stability of the origin within an open neighborhood ϕ_{uc} , which contains the origin. This assumption pertains to the RNN system described in Eq. (2), which also conforms to the structure outlined in Eq. (2). Under this assumption, there exists a continuously differentiable (C^1) constrained CLBF $W_c(x)$ that attains its minimum at the origin and satisfies the following inequalities for all $x \in \phi_{uc}$:

$$\hat{c}_1 |x|^2 \leq W_c(x) - \rho_o \leq \hat{c}_2 |x|^2, \quad (4a)$$

$$\frac{\partial W_c(x)}{\partial x} F_{nn}(x, \Phi_{nn}(x)) \leq -\hat{c}_3 |x|^2, \forall x \in \phi_{uc} - B_\delta(x_e) \quad (4b)$$

$$\left| \frac{\partial W_c(x)}{\partial x} \right| \leq \hat{c}_4 |x| \quad (4c)$$

where $\hat{c}_j(\cdot)$ for $j = 1, 2, 3, 4$ are positive real constants. The function $W_c(x)$ attains its global minimum at the origin, i.e., $W_c(0) = \rho_o$, within the domain ϕ_{uc} . The notation $B_\delta(x_e)$ denotes a small neighborhood around the point $x_e \in \mathcal{X}_e$. The function $F_{nn}(x, u)$ represents the RNN model of Eq. (2). Furthermore, assuming the continuity and smoothness of the functions f , g , and h in the nonlinear system described by Eq. (1), there exist positive constants M , L_x , L_w , $L_{\bar{x}}$, and $L_{\bar{w}}$ such that the following inequalities hold for all $x, \bar{x} \in \Omega_\rho$, and $u \in U$:

$$|F(x, u)| \leq M \quad (5a)$$

$$|F(x, u) - F(x', u)| \leq L_x |x - x'| \quad (5b)$$

$$\left| \frac{\partial W_c(x)}{\partial x} F(x, u) - \frac{\partial W_c(x')}{\partial x} F(x', u) \right| \leq L'_x |x - x'| \quad (5c)$$

The following theorem is established to demonstrate that, in the presence of an unsafe region D_{us} , closed-loop stability and operational safety are achieved simultaneously for the RNN system of Eq. (2) under the CLBF-based controller.

Theorem 1. Assume the existence of a constrained control Lyapunov-barrier function (CLBF), $W_c(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, that attains its minimum at the origin and satisfies the conditions outlined in Eq. (3) for the RNN model described by Eq. (2). The control law $u = \Phi_{nn}(x) \in U$, which meets the criteria specified in Eq. (5), ensures that the closed-loop state remains within the safe set $\mathcal{X}_0$ and avoids the unsafe region D_{us} for all time instances, given any initial condition $x_0 \in \mathcal{X}_0 - D_{us}$. Furthermore, when dealing with an unbounded unsafe region, the origin can be rendered exponentially stable under the control law $u = \Phi_{nn}(x) \in U$ for all $x_0 \in \mathcal{X}_0$.

Proof. As affirmed in Wu and Christofides (2019), in scenarios involving unbounded unsafe regions, the origin, representing the steady-state of the nonlinear system described by Eq. (1), serves as the unique stationary point within the state-space. Consequently, under the control law $u = \Phi_{nn}(x) \in U$, both closed-loop stability and process operational safety can be readily ensured. The detailed proof pertaining to unbounded unsafe regions aligns closely with the methodologies presented in Theorems 1 and 2 of Wu and Christofides (2019) and is therefore omitted here. $\square$

3. Partially-connected recurrent neural network (PCRNN) models

From a modeling standpoint, even state-of-the-art black-box ML models (such as dense fully-connected RNNs) have shown only limited success in scientific fields (Karpatne et al., 2017), largely because they require large datasets, often produce outputs inconsistent with physical laws, and struggle to generalize to ranges outside of the training data. To address this, researchers are increasingly exploring the integration of scientific knowledge with ML models, recognizing that neither pure machine learning nor purely theoretical models alone may suffice for complex applications (e.g. Alber et al., 2019; Baker et al., 2019; Rai and Sahu, 2020). In physics-based ML approaches, domain expertise informs areas like feature engineering and model refinement, distinguishing them from conventional ML practices. Although the idea of merging science and ML is gaining popularity only recently (Karpatne et al., 2017), significant prior work exists across various disciplines, including earth sciences (Reichstein et al., 2019), climatology (Krasnopolsky and Fox-Rabinovitz, 2006; O'Gorman and Dwyer, 2018), material discovery (Cang et al., 2018; Schleider et al., 2019), quantum chemistry (Chakraborty et al., 2014; Schütt et al., 2017), biological sciences (Yazdani et al., 2020), and hydrology (Xu and Valocchi, 2015). Early successes in simple case studies have been promising, raising hopes that this paradigm can accelerate scientific progress and contribute to tackling global challenges in areas such as the environment (Faghmous and Kumar, 2014), healthcare (Wang et al., 2020a), and food and nutrition security (Jia et al., 2019).

In the field of process systems engineering, traditional methodologies for approximating solutions have predominantly relied on physics-based numerical techniques. These methods employ numerical differentiation and integration to solve systems of differential equations that encapsulate established physical laws across spatial and temporal domains (Butcher, 1996; Sagaut et al., 2013; Houska et al., 2012). An alternative approach involves the development of simplified models that approximate system dynamics, such as the Euler equations for gas dynamics and the Reynolds-averaged Navier-Stokes equations for turbulent flows (Chaouat, 2017; Tompson et al., 2017). However, constructing such simplified models that accurately represent complex phenomena poses significant challenges. Moreover, these models often capture only a subset of the dynamics inherent in real-world processes, potentially leading to discrepancies between the model predictions and the actual system behavior. Recent advancements have demonstrated that machine learning (ML) models can generate realistic predictions and significantly expedite the simulation of complex dynamics, outperforming traditional numerical solvers in various applications ranging from turbulence modeling to chemical reaction processes (Wang et al., 2020b; Kochkov et al., 2021; Luo et al., 2022). Despite these promising developments, ML models are inherently data-driven and often lack embedded physical constraints, which can result in predictions that violate fundamental physical principles. Consequently, accurately modeling complex dynamical systems in scientific applications cannot be effectively achieved through ML models or physics-based theories in isolation. Integrating machine learning techniques with conventional physics-based methodologies offers a synergistic approach, leveraging the strengths of both to enhance model accuracy and reliability.

The exploration of structured system modeling is motivated by several considerations. Unlike systems characterized by structured local connectivity, fully-connected systems require long-range connections and exhibit slower communication times between neurons. Real-world problems often exhibit local correlations, and constructing networks with organized neighborhoods is more efficient and memory-conserving than developing fully-connected networks (Canning and Gardner, 1988). Typically, when developing a dynamic model for a general nonlinear process, a neural network that utilizes all available process inputs to predict the desired output is preferred. Constructing a fully connected, black-box dynamic model for these processes is relatively straightforward using open-source machine learning tools; such

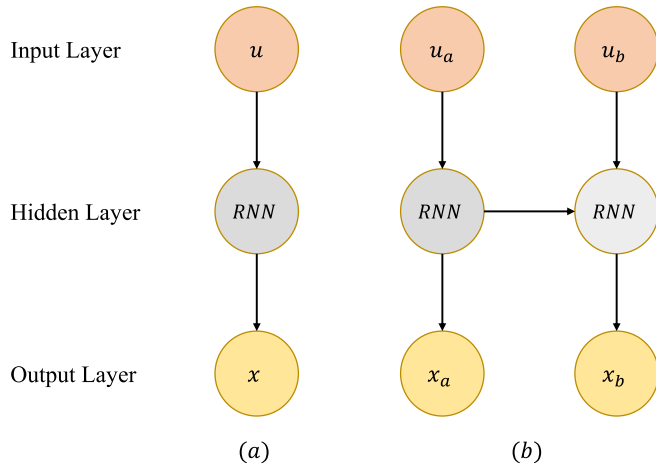


Fig. 2. Structure of (a) standard fully-connected and (b) partially-connected RNN.

a model endeavors to capture any potential relationships between each input and each output of the underlying process. As illustrated in Fig. 2, the general structure of a fully-connected RNN comprises at least three layers: an input layer, hidden layers, and an output layer. Consequently, fully-connected RNN models are often the optimal choice for modeling processes where no prior knowledge is available.

Standard RNNs typically do not incorporate domain-specific knowledge during model development and generally employ fully-connected layers to capture input-output relationships using the provided training dataset. However, it has been demonstrated in Wu et al. (2020) that a priori process structural knowledge can be utilized to enhance an RNN's performance by employing a partially-connected architecture. Fig. 2 illustrates the distinction between fully-connected and partially-connected RNNs, from which it can be observed that certain connections between neurons are removed in a partially-connected structure to better align with the underlying input-output relationships based on a priori process structural knowledge. Partially-connected RNNs are suitable for modeling multi-unit processes where upstream units influence downstream units, but not vice versa. For instance, consider the nonlinear system described by Eq. (1), where the input vector u_1 affects only the state x_1 , while both u_1 and u_2 influence the state x_2 . In this system, $x = [x_1 \in \mathbb{R}^{n_{x1}}, x_2 \in \mathbb{R}^{n_{x2}}]$ and $u = [u_1 \in \mathbb{R}^{n_{u1}}, u_2 \in \mathbb{R}^{n_{u2}}] \in \mathbb{R}^{n_u}$ with $n_{u1} + n_{u2} = n_u$ and $n_{x1} + n_{x2} = n_x$. Wu et al. (2020) demonstrated that by employing a partially-connected architecture, the number of weight parameters can be significantly reduced to achieve the desired model accuracy compared to a fully-connected RNN model. Additionally, in Alhajeri et al. (2022b), an Aspen simulation study of two CSTRs in-series was conducted to show that the MPC using partially-connected RNN models achieved improved closed-loop performances with reduced computational time. Nevertheless, to the best of our knowledge, investigation into the impact of physics-informed recurrent neural networks in the design of safety-critical control systems, where an inaccurate model may result in control actions that drive the state to unsafe conditions, remains an open area of research and motivate our consideration of incorporating such models into MPC.

Remark 2. Attention-based sequence models are promising for complex, long-horizon time dependencies. Recent process systems engineering (PSE) studies have demonstrated the viability of time-series transformers (TSTs) for process modeling and hybrid (physics-based modeling coupled with ML models) modeling in batch crystallization and related tasks, reporting improved accuracy and interpretability compared to conventional recurrent/convolutional NNs. Examples include Sitapure and Sang-Il Kwon (2023) on TSTs for process modeling and control, and a first-of-a-kind hybrid TST framework for batch crystallization

that achieved strong normalized MSE and R^2 performance (Sitapure and Kwon, 2023). These results suggest that attention mechanisms could be either considered as an alternative to PCRNN models when FCRNN is insufficient or could even be incorporated into our framework for further improvement if/where necessary for more complex systems where PCRNN does not perform adequately.

Remark 3. When data is limited, deep learning models risk overfitting and poor extrapolation. In such settings, depending on the range of operation covered in the data set, a more viable approach can be to use simpler model formulations with fewer parameters to fit, such as, for dynamical systems, sparse identification (SINDy) or autoregressive (e.g., NARMAX) type models. These methods can leverage low-order structure and yield identifiable models with a user-defined level of non-linearity and time-lag. Secondly, if a similar process with a large data set can be obtained, a transfer learning approach can adapt a neural network model from the large data set to the new process with limited data and possibly generalize accurately. Finally, simpler hybrid models where some first-principles equations are used and the ML model is used only on the residuals could also be considered.

4. RNN-based model predictive control

In this section, we integrate an RNN model into a CLBF-based model predictive control (CLBF-MPC) framework. Specifically, the PCRNN serves as the predictive model for state predictions required for solving the following optimization problem:

$$\mathcal{J} = \min_{u \in S(\Delta)} \int_{t_k}^{t_{k+P}} L(\tilde{x}(t), u(t)) dt \quad (6a)$$

$$\text{s.t. } \dot{\tilde{x}}(t) = F_{rm}(\tilde{x}(t), u(t)) \quad (6b)$$

$$\tilde{x}(t_k) = x(t_k) \quad (6c)$$

$$u(t) \in U, \forall t \in [t_k, t_{k+P}) \quad (6d)$$

$$\begin{aligned} \dot{\tilde{W}}(x(t_k), u(t_k)) &\leq \dot{\tilde{W}}(x(t_k), \Phi_{nn}(x(t_k))) \\ &\text{if } x(t_k) \notin \mathcal{B}_\delta(x_e) \text{ and } \dot{\tilde{W}}(x(t_k)) > \rho_{\min} \end{aligned} \quad (6e)$$

$$\dot{\tilde{W}}(\tilde{x}(t)) \leq \rho_{\min}, \forall t \in [t_k, t_{k+P}), \text{ if } \dot{\tilde{W}}(x(t_k)) \leq \rho_{\min} \quad (6f)$$

$$\dot{\tilde{W}}(\tilde{x}(t)) < \dot{\tilde{W}}(x(t_k)), \forall t \in (t_k, t_{k+P}) \text{ if } x(t_k) \in \mathcal{B}_\delta(x_e) \quad (6g)$$

where the predicted state trajectory is denoted as $\tilde{x}$. $S(\Delta)$ represents the set of constant piecewise functions with period Δ , and the prediction horizon is P . The objective function in Eq. (6a) is expressed as the integral of $L(\tilde{x}(t), u(t)) = \tilde{x}^T Q \tilde{x} + u^T R u$ over the prediction horizon, where Q and R are positive definite weighting matrices. This formulation ensures that the objective function attains its minimum at the origin. The predicted state trajectory $\tilde{x}(t)$ used to compute $L(\tilde{x}(t), u(t))$ is obtained from the RNN model of Eq. (2) and the initial condition provided by the current state measurement, represented by F_{rm} in Eq. (6b) and $x(t_k)$ in Eq. (6c), respectively. Eq. (6d) imposes constraints on the input vector along the predicted trajectory. The constraints in Eqs. (6e)–(6g) guarantee closed-loop stability and safety, where $\dot{\tilde{W}}_c = \frac{\partial W_c(x)}{\partial x} (f(x) + g(x)u)$. Specifically, when $x(t_k) \notin \mathcal{B}_\delta(x_e)$ and $\dot{\tilde{W}}(x(t_k)) > \rho_{\min}$, the constraint in Eq. (6e) enforces a decrease in $\dot{\tilde{W}}(\tilde{x})$ at a rate at least equal to that achieved by the CLBF-based controller $u = \Phi_{nn}(x) \in U$. If $\dot{\tilde{W}}(x(t_k)) \leq \rho_{\min}$, Eq. (6f) ensures that the closed-loop state trajectory remains inside the level set $\mathcal{U}_{\rho_{\min}}$ throughout the prediction horizon. When $x(t_k) \in \mathcal{B}_\delta(x_e)$, Eq. (6g) is activated to reduce $\dot{\tilde{W}}(x)$ over the following sampling period, ensuring that the state escapes the saddle point within a finite number of steps. The first control action $u^*(t_k)$ from the optimized input trajectory $u^*(t)$ is applied in a sample-and-hold fashion during the next sampling period. Subsequently, the horizon shifts forward by one sampling period, and the optimization problem of Eq. (6) is resolved again by the MPC at every new measurement update (i.e., every Δ).

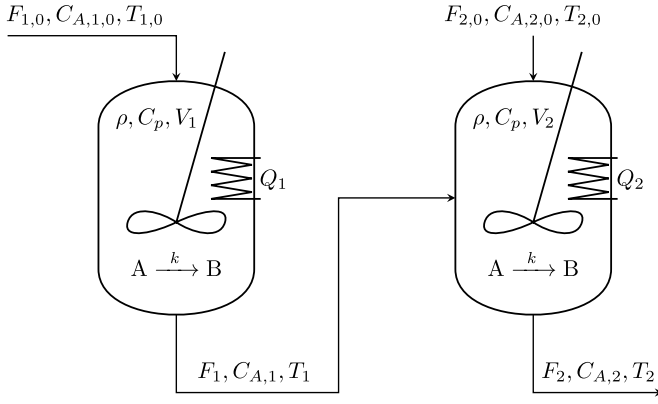


Fig. 3. Two continuous-stirred tank reactors in series.

5. Application to a chemical process

In this section, we present a chemical process example to assess the effectiveness of the proposed physics-informed PCRNN-based CLBF-MPC framework. We consider a process with two CSTRs in series, where information flows from CSTR 1 to CSTR 2 but not in the reverse direction. This motivates the use of a partially connected RNN (PCRNN) because the upstream-to-downstream structure of the two-CSTRs-in-series process can cause model deterioration when a conventional RNN is used [Alhajeri et al. \(2022b\)](#). Enforcing the known structured sparsity in the RNN as in a PCRNN reduces erroneous cross-couplings and improves model fidelity, especially at the same dataset size ([Alhajeri et al., 2023](#)). The study begins with the development of a dynamic model for the two-CSTRs-in-series chemical process based on first principles. A time-series dataset is then generated through extensive open-loop simulations of the first-principles model to train and test the RNN models. Finally, closed-loop simulations are conducted using the FP- and RNN-based MPC and CLBF-MPC, and the results are analyzed and discussed.

The two-CSTRs-in-series system consists of two sequential non-isothermal continuous stirred tank reactors (CSTRs) with ideal mixing, as depicted in [Fig. 3](#). Each reactor hosts an irreversible second-order exothermic reaction where raw material A is converted to product B (i.e., $A \rightarrow B$). The feed flow rate to each reactor, denoted $F_{i,0}$, contains only chemical A with initial concentration and temperature $C_{A,i,0}$ and $T_{i,0}$, respectively, where $i = 1, 2$ corresponds to the reactor index. Each reactor is equipped with a heating jacket that supplies or removes heat at a rate Q_i . The dynamic model governing the two CSTRs is derived from material and energy balance equations and expressed as a set of ordinary differential equations (ODEs):

$$\frac{dC_{A,1}}{dt} = \frac{F_{1,0}}{V_1} (C_{A,1,0} - C_{A,1}) - k_0 e^{\frac{-E}{RT_1}} C_{A,1}^2 \quad (7a)$$

$$\frac{dT_1}{dt} = \frac{F_{1,0}}{V_1} (T_{1,0} - T_1) + \frac{-\Delta H}{\rho C_p} k_0 e^{\frac{-E}{RT_1}} C_{A,1}^2 + \frac{Q_1}{\rho C_p V_1} \quad (7b)$$

$$\frac{dC_{A,2}}{dt} = \frac{F_{1,0}}{V_2} C_{A,1} + \frac{F_{2,0}}{V_2} C_{A,2,0} - \frac{F_{1,0} + F_{2,0}}{V_2} C_{A,2} - k_0 e^{\frac{-E}{RT_2}} C_{A,2}^2 \quad (7c)$$

$$\frac{dT_2}{dt} = \frac{F_{2,0}}{V_2} T_{2,0} + \frac{F_{1,0}}{V_2} T_1 - \frac{F_{1,0} + F_{2,0}}{V_2} T_2 + \frac{-\Delta H}{\rho C_p} k_0 e^{\frac{-E}{RT_2}} C_{A,2}^2 + \frac{Q_2}{\rho C_p V_2} \quad (7d)$$

The variables $C_{A,i}$, T_i , and Q_i denote the concentration of reactant A, the reactor temperature, and the heating (supply) rate, respectively, within the i th reactor. The inlet stream's concentration of A, feed flow rate, and temperature are denoted by $C_{A,i,0}$, $F_{i,0}$, and $T_{i,0}$, respectively. The volume of reacting liquid in each reactor, V_i , has a constant density ρ and heat capacity C_p for both reactors. The parameters ΔH , k_0 , R , and E represent the reaction enthalpy, pre-exponential factor, ideal gas

Table 1

Parameter and steady-state values for the CSTRs.

$C_{A,1s} = 1.95 \text{ kmol/m}^3$	$T_{1s} = 402 \text{ K}$
$C_{A,1,0s} = 4 \text{ kmol/m}^3$	$T_{2s} = 402 \text{ K}$
$C_{A,2s} = 1.95 \text{ kmol/m}^3$	$Q_{1s} = 0.0 \text{ kJ/h}$
$C_{A,2,0s} = 4 \text{ kmol/m}^3$	$Q_{2s} = 0.0 \text{ kJ/h}$
$T_{1,0} = 300 \text{ K}$	$T_{2,0} = 300 \text{ K}$
$F_{1,0} = 5 \text{ m}^3/\text{h}$	$F_{2,0} = 5 \text{ m}^3/\text{h}$
$V_1 = 1 \text{ m}^3$	$V_2 = 1 \text{ m}^3$
$k_0 = 8.46 \times 10^6 \text{ m}^3/(\text{kmol h})$	$E = 5 \times 10^4 \text{ kJ/kmol}$
$R = 8.314 \text{ kJ}/(\text{kmol K})$	$\Delta H = -1.15 \times 10^4 \text{ kJ/kmol}$
$\rho = 1000 \text{ kg/m}^3$	$C_p = 0.231 \text{ kJ}/(\text{kg K})$

constant, and activation energy, respectively, and are the same for both reactors. The process parameter values are summarized in [Table 1](#).

The manipulated inputs in this process are, for both reactors, the inlet concentration of A and the heat supply rate, expressed as deviations from their steady-state values, i.e., $u_1 = C_{A,1,0} - C_{A,1,0s}$, $u_2 = Q_1 - Q_{1s}$, $u_3 = C_{A,2,0} - C_{A,2,0s}$, and $u_4 = Q_2 - Q_{2s}$. The physical input bounds are given by $\pm 3.5 \text{ kmol/m}^3$

and $\pm 5 \times 10^5 \text{ kJ/h}$ for the concentration and heat rate, respectively. The states are also expressed as deviations from their steady-state values, such that $[x_1, x_2, x_3, x_4] = [C_{A,1} - C_{A,1s}, T_1 - T_{1s}, C_{A,2} - C_{A,2s}, T_2 - T_{2s}]$, making the origin the equilibrium point of the system's state-space representation.

5.1. Data generation and RNN models development

Large data sets are essential for developing machine-learning-based models, and generally, larger data sets lead to more accurate models ([Wu et al., 2022](#)), assuming the data is independent and identically distributed. Such large data sets can be sourced from industries, pilot plants, laboratories, and computer simulations. However, industrial data is typically not publicly accessible, while data collection from pilot plants and laboratory experiments is both costly and time-intensive. Therefore, in this work, extensive open-loop simulations of the first-principles model of [Eq. \(7\)](#) are employed to generate the required data set.

For numerically integrating the ODEs, the explicit Euler method with an integration time step of $h_c = 5 \times 10^{-4} \text{ h}$ is used. The integration is carried out over one sampling period Δ under various initial conditions, totaling 3000 different combinations. MATLAB is utilized to generate a data set of size m_{data} . This data set is divided into two matrices: an output matrix containing x_1 through x_4 at time $t = t_k + \Delta$, and an input matrix containing u_1, u_2 and x_1 through x_4 at time $t = t_k$.

Subsequently, two RNN models are built using the generated data and the Keras library. Each model consists of two hidden layers, each containing 30 neurons, with the hyperbolic tangent function (i.e., $\tanh(x)$) serving as the activation function for all layers except the input and output layers. The output layer uses a linear activation function to ensure all real values are possible to be mapped to. In the fully-connected RNN, connections between layers remain unchanged, whereas, in the partially-connected RNN, inputs are directed to different layers in a way that mirrors the physical structure of the underlying process. The partially-connected RNN model follows the algorithm detailed in [Alhajeri et al. \(2022b\)](#).

Using the input data from the previous sampling interval, the state evolution over the next 0.01 h (equivalent to one sampling time Δ) is forecasted. The Adam optimizer, which combines RMSprop with gradient descent incorporating momentum, is employed instead of standard gradient descent. Furthermore, five-fold cross-validation is performed on the RNN models to ensure robustness, with models selected based on the lowest validation MSE.

To assess the model accuracy, open-loop simulations of the two models under varying input conditions from known initial conditions are conducted, and the MSE is then calculated and tabulated in [Table 2](#).

Table 2
Open-loop prediction results (MSE).

State	Modeling architecture	
	FCRNN	PCRNN
x_1	0.0065	0.0011
x_2	125.4551	52.2929
x_3	0.0134	0.0031
x_4	156.3076	15.1736

It can be observed that the PCRNN modeling error is low enough for control purposes and, in particular, the ratio of the FCRNN MSE to the PCRNN MSE for x_1 , x_2 , x_3 , and x_4 are 6.0, 2.4, 4.3, and 10.3, respectively. Since all ratios exceed 1, these results demonstrate that the PCRNN consistently provides superior prediction accuracy compared to the FCRNN.

5.2. Closed-loop simulations

In this section, we conduct closed-loop simulations of an FCRNN- and a PCRNN-based LMPC with and without the inclusion of a barrier function. These simulations are designed to highlight the role of the barrier function in ensuring safe operating dynamics as well as the importance of model accuracy within the CLB-MPC. To evaluate the controller's performance, we simulate trajectories starting from two different initial conditions, demonstrating how the system behaves under both configurations. The barrier function is introduced as a safeguard to prevent the system from entering unsafe operating regions, ensuring that the dynamics remain within acceptable boundaries throughout the process.

The control objective is to maintain the two-CSTR system at the unstable equilibrium point $(C_{A,1s}, T_{1s}) = (1.95 \text{ kmol/m}^3, 402 \text{ K})$ for $i = 1, 2$ while ensuring avoidance of the bounded unsafe operating region in the state-space by manipulating the inlet concentrations $C_{A,i,0}$ and heat input rates Q_i . Specifically, we aim to show that, under the CLB-MPC of Eq. (6), starting from an initial state away from the origin, the closed-loop system of Eq. (7) can successfully avoid a bounded unsafe region D_b within the state-space as it approaches the origin, ultimately converging to a small neighborhood surrounding the origin.

The unsafe region D_b is characterized as an ellipse in the state-space,

$$D_b := \left\{ x \in \mathbb{R}^2 \mid F(x) = \frac{(x_1 + 0.92)^2}{0.5} + \frac{(x_2 - 42)^2}{500} < 0.06 \right\} \quad (8)$$

for the first CSTR and similarly for the second CSTR but with x_3 and x_4 in place of x_1 and x_2 , respectively. Following the CLB construction approach outlined in Wu et al. (2019), Wu and Christofides (2019), we begin by designing a control Lyapunov function in the standard quadratic form, defined as $V(x) = x^T P x$, where P is a positive definite matrix specified as follows:

$$P = \begin{bmatrix} 1060 & 22 \\ 22 & 0.52 \end{bmatrix} \quad (9)$$

Next, we define a set H that encloses the bounded unsafe region D_b as $H := \{x \in \mathbb{R}^2 \mid F(x) < 0.07\}$, and proceed to design the Control Barrier function $B(x)$ as follows:

$$B(x) = \begin{cases} e^{\frac{F(x)}{F(x)-0.07} - e^{-6}}, & \text{if } x \in H \\ -e^{-6}, & \text{if } x \notin H \end{cases} \quad (10)$$

The control Lyapunov-barrier function $W_c(x) = V(x) + \mu B(x) + v$ is constructed with the following parameters: $\rho = 0, c_1 = 0.1, c_2 = 1061, c_3 = \max_{x \in \partial H} |x|^2 = 2295, c_4 = \min_{x \in \partial D} |x|^2 = 1370, v = \rho_c - c_1 c_4 = -160, \mu = 10^9$. The definitions of the above parameters can be found in Wu et al. (2019), Wu and Christofides (2019). Finally, the stationary point of $W_c(x)$, calculated by setting $\frac{\partial W_c(x)}{\partial x}$ to 0, is found to be $x_e = (-1.004, 47.48)$ and is ensured to be contained within $U_{\hat{\rho}}$, where $\hat{\rho} = 500$, to not affect closed-loop stability.

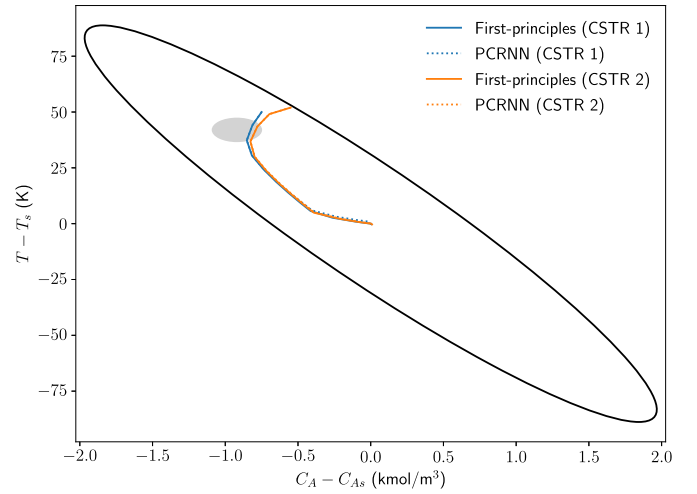


Fig. 4. Closed-loop state trajectories for the system of Eq. (7) under MPC without the CLB constraints using the FP model (solid lines) and PCRNN model (dotted lines), starting from the first initial condition, $[-0.75, 50, -0.55, 52]$. The gray ellipse is the set of bounded unsafe states D_b in each reactor's state-space.

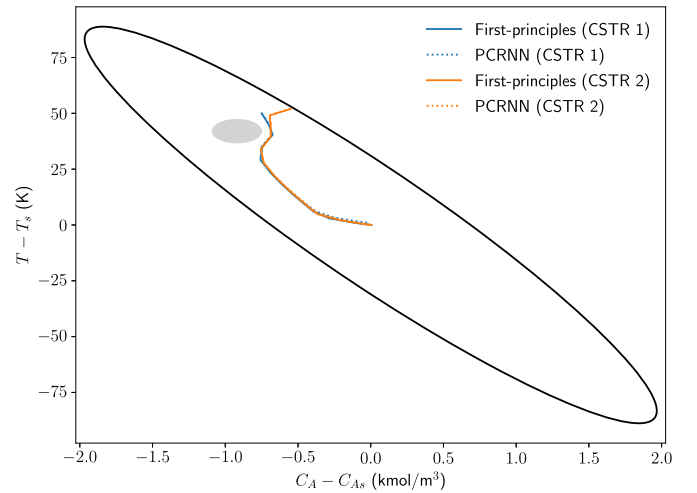


Fig. 5. Closed-loop state trajectories for the system of Eq. (7) under the CLB-MPC using the FP model (solid lines) and PCRNN model (dotted lines), starting from the first initial condition, $[-0.75, 50, -0.55, 52]$. The gray ellipse is the set of bounded unsafe states D_b in each reactor's state-space.

The state-space trajectories of the two CSTRs under MPC, starting from the first initial condition of $x = [-0.75, 50, -0.55, 52]^T$, without and with the barrier function are shown in Figs. 4 and 5, respectively, with their corresponding time-varying state and input trajectories (for the PCRNN) depicted in Figs. 6 and 7. It can be observed in Fig. 4 that, in absence of the barrier function, the state enters the unsafe region on its way to the origin. Since this occurs even with the first-principles (FP) model of Eq. (7) used as the predictive model for the MPC, this is not due to the modeling error of the PCRNN but the expected trajectory of the MPC for this initial condition without any safety considerations. In contrast, under the CLB-MPC, the states of both CSTRs avoid the unsafe region using both the FP and PCRNN models, as seen in Fig. 5.

Furthermore, to highlight the advantages of the PCRNN-based CLB-MPC in achieving the required prediction accuracy and ensuring process operational safety, an FCRNN model is developed using the same dataset as the PCRNN model. Specifically, the FCRNN utilizes all states x_1 through x_4 and inputs u_1 through u_4 at time t_{k-1} to forecast the states at time t_k . Simulations identical to the previous case are con-

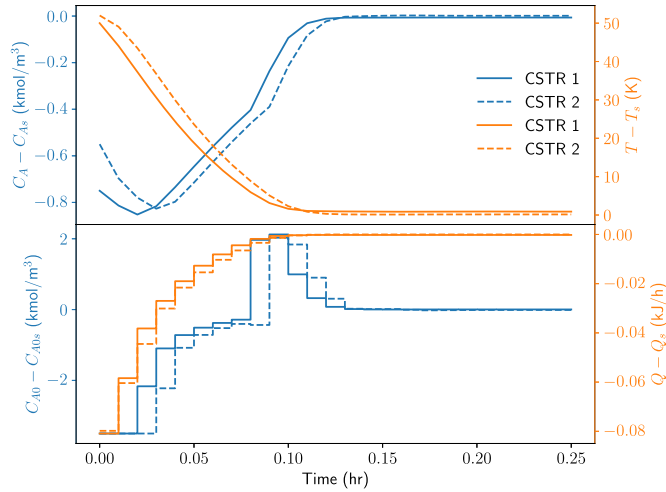


Fig. 6. Time-varying state and input trajectories for the system of Eq. (7) under MPC without the CLBF constraints using the PCRNN model, starting from the first initial condition, $[-0.75, 50, -0.55, 52]$.

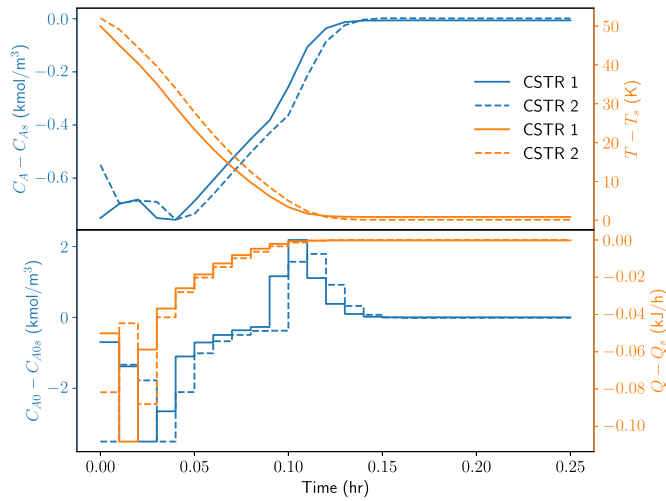


Fig. 7. Time-varying state and input trajectories for the system of Eq. (7) under the CLBF-MPC using the PCRNN model, starting from the first initial condition, $[-0.75, 50, -0.55, 52]$.

ducted but from another initial condition, $x = [-1, 55, -1, 55]^T$, to assess the generalizability of the CLBF-MPC, showcase the drawback of the FCRNN model, and stress-test the controller near the unsafe set boundary. The state-space trajectories for both CSTRs under the MPC using the FP model, the FCRNN model, and the PCRNN model, without and with the barrier function, are shown in Figs. 8 and 9, respectively, with their corresponding time-varying state and input trajectories depicted (for the PCRNN) in Figs. 10 and 11. Once again, it is observed from Fig. 8 that the trajectory to the origin without safety constraints would pass through the unsafe region even under the FP-based MPC. However, under the CLBF-MPC using either the FP or PCRNN model, the states successfully avoid the unsafe region as seen in Fig. 9. In contrast, the states for CSTR 1 enters the unsafe region even under the CLBF-MPC using the FCRNN model, unlike the FP- and PCRNN-based CLBF-MPC. It is noted that the FCRNN-based CLBF-MPC can typically stabilize the nonlinear system within a neighborhood of the steady-state, assuming the modeling error is small in that region. However, in this particular example, the CLBF-MPC utilizing an FCRNN model fails to ensure satisfactory performance because, beyond achieving closed-loop stability, the control strategy must also guarantee process operational safety, which necessitates an even more accurate model, especially for this initial con-

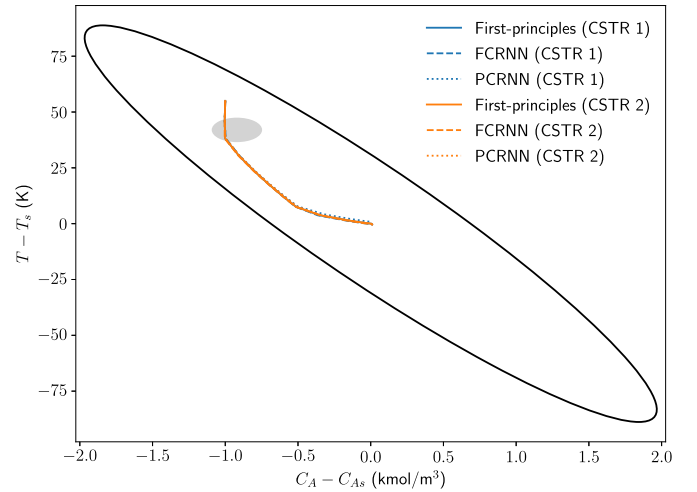


Fig. 8. Closed-loop state trajectories for the system of Eq. (7) under MPC without the CLBF constraints using the FP model (solid lines), FCRNN model (dashed lines), and PCRNN model (dotted lines), starting from the second initial condition, $[-1, 55, -1, 55]$. The gray ellipse is the set of bounded unsafe states D_b in each reactor's state-space.

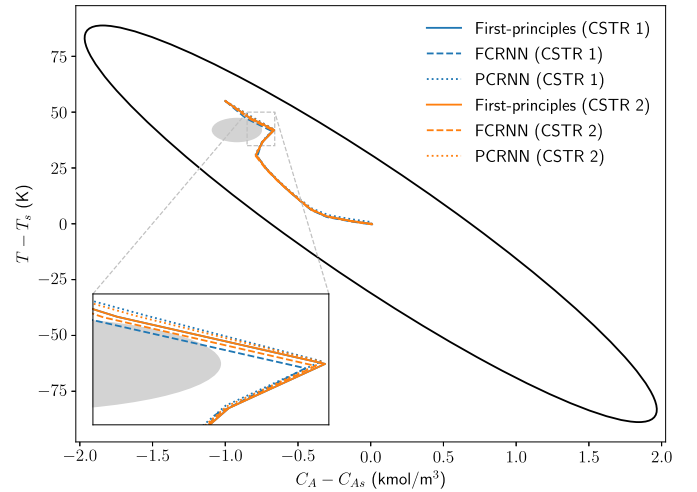


Fig. 9. Closed-loop state trajectories for the system of Eq. (7) under the CLBF-MPC using the FP model (solid lines), FCRNN model (dashed lines), and PCRNN model (dotted lines), starting from the second initial condition, $[-1, 55, -1, 55]$. The gray ellipse is the set of bounded unsafe states D_b in each reactor's state-space.

dition as the controller has limited actions to take while avoiding the unsafe set due to initialization in close proximity to its boundary. Feedback control relying on an inaccurate FCRNN model cannot assure that the system state will avoid entering unsafe regions at all times. Specifically, when there exists a significant model mismatch, the controller may predict the state to be outside unsafe regions, while in reality, the true state violates these safety boundaries within one sampling period. This limitation motivates the use of the PCRNN model that offers a closer approximation of the nonlinear system dynamics across the operating region, thereby providing more accurate predictions necessary for effective CLBF-MPC. Finally, to numerically confirm the above observations, Fig. 12 shows the magnitude of violation of the safety constraint, i.e., how far inside the unsafe region the state is, for both CSTRs, under the FCRNN- and PCRNN-based CLBF-MPC. We quantify the barrier function constraint violation by evaluating the ellipse's implicit level-set function, $\frac{(x_1+0.92)^2}{0.5} + \frac{(x_2-42)^2}{500} - 0.06$, which is negative inside the unsafe

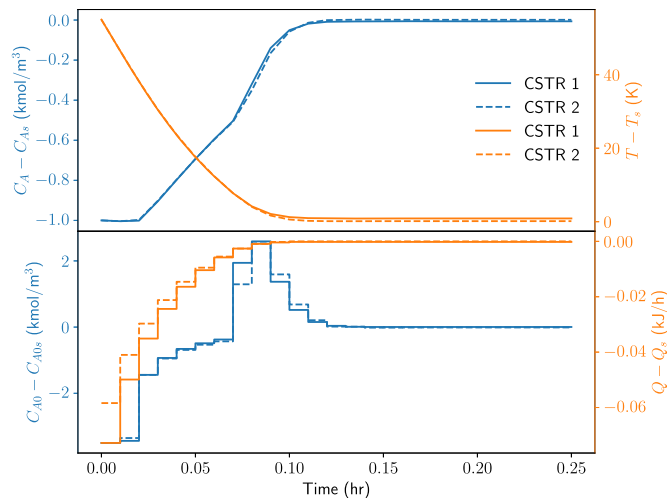


Fig. 10. Time-varying state and input trajectories for the system of Eq. (7) under MPC without the CLBF constraints using the PCRNN model, starting from the second initial condition, $[-1, 55, -1, 55]$.

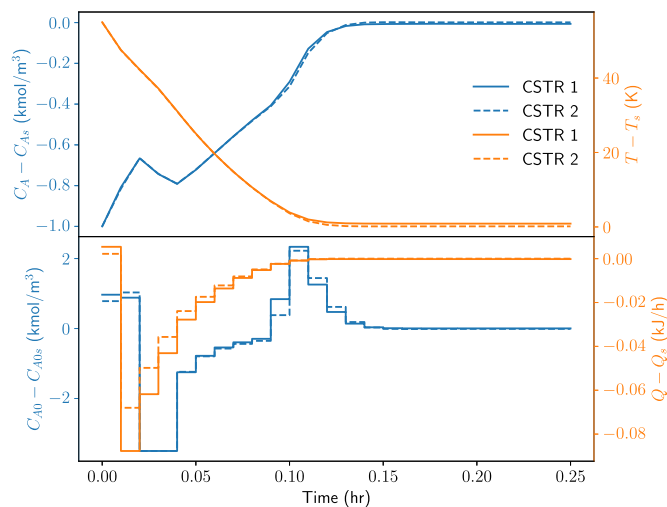


Fig. 11. Time-varying state and input trajectories for the system of Eq. (7) under the CLBF-MPC using the PCRNN model, starting from the second initial condition, $[-1, 55, -1, 55]$.

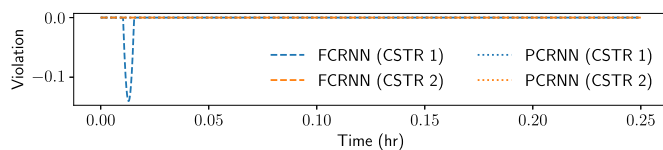


Fig. 12. Violation of the safety constraint measured in terms of depth of intrusion into the unsafe set D_b for the FCRNN and PCRNN trajectories shown in Fig. 9.

region, and enforced to be zero otherwise (outside), i.e.,

$$\text{Violation} = \min \left\{ \frac{(x_1 + 0.92)^2}{0.5} + \frac{(x_2 - 42)^2}{500} - 0.06, 0 \right\} \quad (11)$$

At each time instance along the CSTR trajectories, we compute the violation using Eq. (11), zero any positive values, and plot the resulting depth of intrusion into the ellipse, such that larger negative values correspond to points farther inside the unsafe region. It can be confirmed from Fig. 12 that the trajectory of CSTR 1 under the FCRNN-based CLBF-MPC does indeed enter the unsafe region, while that under the PCRNN-based MPC always stays outside D_b .

The two case studies above demonstrate that a PCRNN model developed from extensive open-loop simulations to replace the CSTR process described in Eq. (7) within the CLBF-MPC framework achieves the desired approximation accuracy and can guarantee that for any initial condition in the safe operating region, the closed-loop state of the system of Eq. (7) remains within the safe operating region at all times and ultimately converges to a small neighborhood $U_{\rho_{\min}}$ around the origin, while avoiding bounded unsafe regions in the state-space.

6. Conclusion

In this study, a control Lyapunov-barrier function-based model predictive controller (CLBF-MPC) using physics-informed partially-connected recurrent neural network (PCRNN) models was developed for nonlinear process systems. Assuming the modeling error between the PCRNN model and the actual nonlinear process is sufficiently small, it was shown that, in the presence of a bounded unsafe region, discontinuous control actions can be employed in the vicinity of the saddle point to facilitate the system state's escape from such points and progression toward the origin. The effectiveness of the proposed method was illustrated through a chemical process example involving bounded unsafe regions. Simulation results confirmed that the PCRNN-based CLBF-MPC achieved the desired predictive performance and successfully drove the system state to the origin while avoiding unsafe regions in the state-space, while the fully-connected RNN model (without physics-knowledge) caused the state to enter the unsafe operating region, demonstrating the need for accurate process models, incorporating information of physics if required, for safety-critical systems.

CRedit authorship contribution statement

Mohammed S. Alhajeri: Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Funding acquisition, Data curation, Conceptualization; **Fahim Abdullah:** Writing – review & editing, Writing – original draft, Visualization; **Panagiotis D. Christofides:** Writing – review & editing, Supervision, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mohammed S. Alhajeri reports financial support was provided by Kuwait University. Mohammed S. Alhajeri reports a relationship with Kuwait University that includes: employment. If there are other authors, they 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

This work was supported and funded by Kuwait University Research Grant No. EC01/25.

Literature Cited

- Alber, M., Buganza Tepole, A., Cannon, W.R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W.W., Perdikaris, P., Petzold, L., et al., 2019. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digit. Med.* 2, 115.
- Alhajeri, M.S., Abdullah, F., Wu, Z., Christofides, P.D., 2022a. Physics-informed machine learning modeling for predictive control using noisy data. *Chem. Eng. Res. Des.* 186, 34–49.

- Alhajeri, M.S., Alnajdi, A., Abdullah, F., Christofides, P.D., 2023. On generalization error of neural network models and its application to predictive control of nonlinear processes. *Chem. Eng. Res. Des.* 189, 664–679.
- Alhajeri, M.S., Luo, J., Wu, Z., Albalawi, F., Christofides, P.D., 2022b. Process structure-based recurrent neural network modeling for predictive control: a comparative study. *Chem. Eng. Res. Des.* 179, 77–89.
- Alhajeri, M.S., Ren, Y.M., Ou, F., Abdullah, F., Christofides, P.D., 2024. Model predictive control of nonlinear processes using transfer learning-based recurrent neural networks. *Chem. Eng. Res. Des.* 205, 1–12.
- Alhajeri, M.S., Wu, Z., Rincon, D., Albalawi, F., Christofides, P.D., 2021. Machine-learning-based state estimation and predictive control of nonlinear processes. *Chem. Eng. Res. Des.* 167, 268–280.
- Ames, A.D., Grizzle, J.W., Tabuada, P., 2014. Control barrier function based quadratic programs with application to adaptive cruise control. In: *Proceedings of the 53rd IEEE Conference on Decision and Control*. Los Angeles, California, pp. 6271–6278.
- Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., Parashar, M., Patra, A., Sethian, J., Wild, S., et al., 2019. Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Technical Report. USDOE Office of Science (SC), Washington, DC (United States).
- Butcher, J.C., 1996. A history of Runge-Kutta methods. *Appl. Numer. Math.* 20, 247–260.
- Cang, R., Li, H., Yao, H., Jiao, Y., Ren, Y., 2018. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *Comput. Mater. Sci.* 150, 212–221.
- Canning, A., Gardner, E., 1988. Partially connected models of neural networks. *J. Phys. A Math. Gen.* 21, 3275–3283.
- Chakraborty, I., Bodurtha, K.J., Heeder, N.J., Godfrin, M.P., Tripathi, A., Hurt, R.H., Shukla, A., Bose, A., 2014. Massive electrical conductivity enhancement of multilayer graphene/polystyrene composites using a nonconductive filler. *ACS Appl. Mater. Interfaces* 6, 16472–16480.
- Chaouat, B., 2017. The state of the art of hybrid RANS/LES modeling for the simulation of turbulent flows. *Flow Turbul. Combust.* 99, 279–327.
- Chen, T., Chen, H., 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* 6, 911–917.
- Crowl, D.A., Louvar, J.F., 2011. *Chemical Process Safety: Fundamentals with Applications*. Vol. 30. Pearson Education, Upper Saddle River, NJ. Third Edition.
- Faghmous, J.H., Kumar, V., 2014. A big data guide to understanding climate change: the case for theory-guided data science. *Big Data* 2, 155–163.
- Houska, B., Logist, F., Diehl, M., Van Impe, J., 2012. A Tutorial on Numerical Methods for State and Parameter Estimation in Nonlinear Dynamic Systems. *Identification and Automation Systems* 67–88.
- Incidents, C.I.O.C., 2016. Final Report of the Investigations of Chemical Incidents. Technical Report. U.S. Chemical Safety and Hazard Investigation Board.
- Jankovic, M., 2017. Combining control Lyapunov and barrier functions for constrained stabilization of nonlinear systems. In: *Proceedings of the American Control Conference*. Seattle, Washington, pp. 1916–1922.
- Jia, X., Khandelwal, A., Mulla, D.J., Pardey, P.G., Kumar, V., 2019. Bringing automated, remote-sensed, machine learning methods to monitoring crop landscapes at scale. *Agric. Econ.* 50, 41–50.
- Karpate, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2017. Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* 29, 2318–2331.
- Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S., 2021. Machine learning-accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci.* 118, e2101784118.
- Krasnopolsky, V.M., Fox-Rabinovitz, M.S., 2006. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Netw.* 19, 122–134.
- Lin, Y., Sontag, E.D., 1991. A universal formula for stabilization with bounded controls. *Syst. Contr. Lett.* 16, 393–397.
- Luo, J., Canuso, V., Jang, J.B., Wu, Z., Morales-Guio, C.G., Christofides, P.D., 2022. Machine learning-based operational modeling of an electrochemical reactor: handling data variability and improving empirical models. *Ind. Eng. Chem. Res.* 61, 8399–8410.
- Mhaskar, P., El-Farra, N.H., Christofides, P.D., 2006. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Syst. Contr. Lett.* 55, 650–659.
- Muñoz de la Peña, D., Christofides, P.D., 2008. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Trans. Autom. Contr.* 53, 2076–2089.
- Niu, B., Zhao, J., 2013. Barrier Lyapunov functions for the output tracking control of constrained nonlinear switched systems. *Syst. Contr. Lett.* 62, 963–971.
- O’Gorman, P.A., Dwyer, J.G., 2018. Using machine learning to parameterize moist convection: potential for modeling of climate, climate change, and extreme events. *J. Adv. Model. Earth Syst.* 10, 2548–2563.
- Park, J., Sandberg, I.W., 1991. Universal approximation using radial-basis-function networks. *Neural Comput.* 3, 246–257.
- Rai, R., Sahu, C.K., 2020. Driven by data or derived through physics? A review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access* 8, 71050–71073.
- Rawlings, J.B., Mayne, D.Q., 2009. *Model Predictive Control: Theory and Design*. Nob Hill Pub.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al., 2019. Deep learning and process understanding for data-driven earth system science. *Nature* 566, 195–204.
- Ren, Y.M., Alhajeri, M.S., Luo, J., Chen, S., Abdullah, F., Wu, Z., Christofides, P.D., 2022. A tutorial review of neural network modeling approaches for model predictive control. *Comput. Chem. Eng.* 165, 107956.
- Romdlony, M.Z., Jayawardhana, B., 2016. Stabilization with guaranteed safety using control Lyapunov-barrier function. *Automatica* 66, 39–47.
- Sagaut, P., Terracol, M., Deck, S., 2013. Multiscale and Multiresolution Approaches in Turbulence-LES, DES and Hybrid RANS/LES Methods: Applications and Guidelines. World Scientific.
- Sanders, R.E., 2015. *Chemical Process Safety: Learning From Case Histories*. Butterworth-Heinemann.
- Schleder, G.R., Padilha, A. C.M., Acosta, C.M., Costa, M., Fazzio, A., 2019. From DFT to machine learning: recent approaches to materials science—a review. *J. Phys. Mater.* 2, 032001.
- Schütt, K., Kindermans, P.-J., Saucedo Felix, H.E., Chmiela, S., Tkatchenko, A., Müller, K.-R., 2017. SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. *Advances in Neural Information Processing Systems* 30.
- Shah, P., Sherif, M.Z., Bangi, M. S.F., Kravaris, C., Kwon, J. S.-I., Botre, C., Hirota, J., 2022. Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: identification of time-varying dependencies among parameters. *Chem. Eng. J.* 441, 135643.
- Silva, S.R., Bonanato, G., da Costa, J. E.F., Sarrouh, B., da Costa, A. O.S., 2021. Specific chemical exergy prediction for biological molecules using hybrid models. *Chem. Eng. Sci.* 235, 116462.
- Sitapure, N., Kwon, J. S.-I., 2023. CrystalGPT: enhancing system-to-system transferability in crystallization prediction and control using time-series-transformers. *Comput. Chem. Eng.* 177, 108339.
- Sitapure, N., Sang-I Kwon, J., 2023. Introducing hybrid modeling with time-series-transformers: a comparative study of series and parallel approach in batch crystallization. *Ind. Eng. Chem. Res.* 62 (49), 21278–21291.
- Sontag, E.D., 1989. A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization. *Syst. Contr. Lett.* 13, 117–123.
- Tee, K.P., Ge, S.S., Tay, E.H., 2009. Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica* 45, 918–927.
- Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K., 2017. Accelerating Eulerian fluid simulation with convolutional networks. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3424–3433.
- Wang, L., Chen, J., Marathe, M., 2020a. TDEFSI: theory-guided deep learning-based epidemic forecasting with synthetic information. *ACM Trans. Spatial Algorithms Syst.* 6, 1–39.
- Wang, R., Kashinath, K., Mustafa, M., Albert, A., Yu, R., 2020b. Towards physics-informed deep learning for turbulent flow prediction. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466.
- Wu, Z., Albalawi, F., Zhang, Z., Zhang, J., Durand, H., Christofides, P.D., 2019. Control lyapunov-barrier function-based model predictive control of nonlinear systems. *Automatica* 109, 108508.
- Wu, Z., Alnajdi, A., Gu, Q., Christofides, P.D., 2022. Statistical machine-learning-based predictive control of uncertain nonlinear processes. *AIChE J.* 68, e17642.
- Wu, Z., Christofides, P.D., 2019. Handling bounded and unbounded unsafe sets in control lyapunov-barrier function-based model predictive control of nonlinear processes. *Chem. Eng. Res. Des.* 143, 140–149.
- Wu, Z., Rincon, D., Christofides, P.D., 2020. Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *J. Process Contr.* 89, 74–84.
- Xiao, M., Wu, Z., 2023. Modeling and control of a chemical process network using physics-informed transfer learning. *Ind. Eng. Chem. Res.* 62 (42), 17216–17227.
- Xu, T., Valocchi, A.J., 2015. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Comput. Geosci.* 85, 124–136.
- Yazdani, A., Lu, L., Raissi, M., Karniadakis, G.E., 2020. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS Comput. Biol.* 16, e1007575.