Automatica 189 (2026) 113005

Contents lists available at ScienceDirect

Automatica

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

Stability and learning of networked systems with application to physics-informed neural networks☆

Laura Boca de Giuli *, Alessio La Bella, Marcello Farina, Riccardo Scattolini

Department of Electronics, Information, and Bioengineering, Politecnico di Milano, 20133 Milan, Italy

ARTICLE INFO

Article history:
Received 14 October 2024
Received in revised form 24 November 2025
Accepted 3 March 2026

Keywords:
Networked systems
Incremental input-to-state stability
Physics-informed neural networks
Plug-and-play

ABSTRACT

This article addresses learning and stability of networked systems. A general methodology is proposed to devise physics-informed data-based networked models by interconnecting multiple submodels according to the networked system topology and jointly training them exploiting input–output data. Since stability properties are crucial in data-based modeling and cannot always be ensured when interconnecting even stable submodels, a novel sufficient condition is proposed guaranteeing incremental input-to-state stability (δISS) of discrete-time networked models. It is shown that this condition can be easily enforced during the training of physics-informed recurrent neural networks, achieving guaranteed stability properties and improved modeling performance compared to standard black-box approaches. Moreover, the enforced δISS property enables *(i)* stable plug-and-play operations on the networked system model, *(ii)* the development of a convergent decentralized state observer, and *(iii)* the design of a convergent nonlinear model predictive control regulator. The presented strategies are tested in simulation on a realistic large-scale networked system, i.e., a benchmark chemical plant, showing promising results in both modeling and control design.

1. Introduction

A growing interest is arising in the scientific community in modeling and controlling large-scale networked systems (Tang & Daoutidis, 2018), which exhibit a complex structure, typically represented as a graph containing nodes (subsystems) and edges (interconnections among subsystems). Examples include electricity networks (La Bella et al., 2022), district heating networks (Nigro et al., 2025), chemical plants (Luyben & Luyben, 1995), hydropower valleys (Maestre et al., 2015), and cyber–physical systems (Kim & Kumar, 2012). These systems are often complex to model accurately using physical equations due to their large-scale dimensions, the numerous parameters, and the associated uncertainties. On the other hand, they are usually characterized by a large availability of monitoring data, which can be exploited to recover an estimated model. Purely data-based models, though, may be unreliable, as they lack any available physical insight. This has sparked increasing interest within the control community in physics-informed machine learning techniques, which aim to embed physical knowledge in the training procedure of data-based models to enhance their physical coherence and modeling accuracy (Nghiem et al., 2023). In this context, neural networks (NNs), and in particular recurrent neural networks (RNNs), have shown to be well-suited for embedding physical knowledge, in addition to their enhanced learning capability (Karpatne et al., 2022).

Besides, a key feature of data-based modeling is the ability to inherit stability properties from the system to be modeled, provided that the system itself possesses such properties. For instance, considering RNN models, incremental input-to-state stability (δISS) is crucial, as it ensures that the state trajectories of the model are independent of their initialization and that the effect of input perturbations is bounded (Bonassi et al., 2022). Having a δISS model is also beneficial for designing model predictive control (MPC) schemes and state observers with closed-loop guarantees, as discussed in Bonassi et al. (2024), Masero et al. (2024), Terzi et al. (2021). In this context, (Xie et al., 2025) experimentally verified on a real system that a δISS model leads to improved closed-loop performance also with an internal model controller (IMC).

In view of this, the objective of this work is to develop a physics-informed methodology for networked systems to learn accurate RNN-based models that enjoy δISS properties and enable the design of MPC schemes with closed-loop convergence guarantees.

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Vincent Andrieu under the direction of Editor Luca Zaccarian (Nonlinear Systems and Control).

* Corresponding author.
E-mail addresses: laura.bocadegiuli@polimi.it (L.B.d. Giuli), alessio.labella@polimi.it (A.L. Bella), marcello.farina@polimi.it (M. Farina), riccardo.scattolini@polimi.it (R. Scattolini).

https://doi.org/10.1016/j.automatica.2026.113005

1.1. Related work

Data-based methods have been proposed in the literature to recover control-oriented models of networked systems for MPC design, as discussed in La Bella and Del Corno (2023), Terzi et al. (2020), neglecting however stability properties and available physical knowledge. The latter can be effectively incorporated in physics-informed neural networks by defining suitable training functions, e.g., including regularization terms expressing known physical relations among variables (Daw et al., 2022; Drgoňa et al., 2021). Another method incorporates available physical insights, like partially known dynamical relationships, directly into the NN architecture (Bolderman et al., 2021; Daw et al., 2020). However, these physics-informed methods do not specifically address the modeling of networked systems. In these systems, the primary available knowledge is their topology, which can be leveraged to obtain a physics-informed model by accordingly interconnecting different submodels. This idea has been applied to RNN-based models in Alhajeri et al. (2022) for chemical plants and in Boca de Giuli, La Bella, and Scattolini (2024) for district heating systems. Nonetheless, these topology-based approaches are tailored to specific applications and do not account for stability properties, which are crucial for the practical deployment of data-based models (Bolderman et al., 2024).

Sufficient conditions for guaranteeing δISS in different RNN-based architectures are discussed in Bonassi et al. (2022), D'Amico et al. (2023), but these works do not incorporate physical knowledge or address the stability of multiple interconnected submodels. This latter aspect is a well-known challenge in the literature because, even if each submodel is stable, the overall stability of a networked model is not guaranteed due to the possible presence of inner loops. Note that the cascade of stable submodels is stable. In this context, (Marino et al., 2024) provide stability conditions specifically for gated graph neural networks. Sufficient conditions for ensuring input-to-state stability (ISS) of generically interconnected subsystems are established for continuous-time systems in Dashkovskiy et al. (2007, 2010) and for discrete-time systems in Boca de Giuli, La Bella, Farina and and Scattolini (2024), Jiang and Wang (2001). However, ISS is a weaker property than δISS, e.g., it does not guarantee that the closer two input sequences are, the smaller the difference between the resulting state trajectories is, regardless of their initialization (Angeli, 2009). Concerning δISS, a sufficient condition for interconnected δISS continuous-time subsystems is discussed in Angeli (2002), considering only single-loop topologies. Ultimately, the aforementioned references do not address the δISS property of interconnected δISS discrete-time subsystems, which is a key challenge in modeling frameworks that rely on RNN-based submodels interconnected, for instance, according to physics-informed strategies.

1.2. Main contribution

In light of the above discussion, this work proposes a methodology for learning generic networked system models by leveraging their physical topology and ensuring that the resulting networked model enjoys δISS. The contributions of the work are hereafter synthesized.

(1) *δISS condition for networked models:* We derive a new sufficient condition to ensure the δISS property of a model composed of arbitrarily interconnected discrete-time δISS submodels. We also show that this property can be exploited in a plug-and-play framework, establishing novel conditions under which submodels can be added or removed without affecting the δISS of the overall networked model.

(2) *Physics-informed RNN-based modeling of networked systems with δISS:* We propose a physics-informed RNN (PI-RNN) methodology for modeling generic networked systems by leveraging their topology as physical information. Specifically, by associating each subsystem with an RNN-based submodel, the overall PI-RNN networked model is recovered through an interconnection of these submodels resembling the system topology. It is then shown that the PI-RNN model can be identified from data through a unique training procedure which enforces the proposed δISS conditions for networked models by using suitably defined loss functions.

The proposed methodology yields several advantages. First, embedding the networked system topology in the physics-informed model enhances modeling accuracy compared to standard data-based approaches, while also ensuring the δISS property. Moreover, the approach enables stable plug-and-play operations on the developed physics-informed model, making it highly effective for practical applications. This implies that, if some subsystems are modified or replaced over time, it is not necessary to reidentify the entire networked model and reestablish its stability properties. Instead, only the submodels corresponding to the updated subsystems need to be identified, with suitable conditions enforced locally during their training procedures to preserve the δISS of the networked model. Finally, the proposed δISS physics-informed model enables the design of a decentralized observer and a nonlinear MPC (NMPC) regulator, both with proven convergence guarantees.

The proposed methodology is tested in simulation on a large-scale chemical plant (Luyben & Luyben, 1995), showing promising results from both modeling and control perspectives.

1.3. Paper outline

This article is organized as follows. Section 2 introduces the notation and basic definitions used throughout the paper. Section 3 presents the physics-informed methodology to model networked systems leveraging their topology. The established δISS condition for the stability analysis of the interconnection of δISS discrete-time submodels is provided in Section 4, where also different plug-and-play scenarios are considered. In Section 5, it is shown how δISS conditions can be enforced during the training procedure of a physics-informed recurrent neural network model. The proposed framework is tested in simulation on a reference chemical plant in Section 6, with Section 6.1 focusing on the design of convergent decentralized state observers and NMPC architectures, and Section 6.2 focusing on the experimental design. Final considerations are given in Section 7.

2. Notation and preliminaries

Let $\mathbb{N}$ denote the set of natural numbers, $\mathbb{R}$ the set of real numbers, $\mathbb{R}_{\geq 0}$ the set of positive or null real numbers, and $\mathbb{R}_{>0}$ the set of positive real numbers. Given a vector v, $v^\top$ denotes its transpose, whereas $\|v\|_p$ the p-norm of v, with $p > 0$. Given a matrix A, A_{ij} denotes its component in row i and column j, $\rho(A)$ indicates its spectral radius, $\|v\|_A^2 = v^\top A v$, and its induced norm is defined as $\|A\|_p = \max_v \frac{\|Av\|_p}{\|v\|_p}$. I_n denotes the identity matrix of dimension n. Sets are generically indicated by calligraphic letters, e.g., $\mathcal{N}$, their cardinality is denoted as $|\mathcal{N}|$ and their interior is denoted as $\text{Int}(\mathcal{N})$. Given n variables $a^{[1]}, \dots, a^{[n]}$ and the set of their indices $\mathcal{N} = \{1, \dots, n\}$, $a = [a^{[1]\top} \ \dots \ a^{[n]\top}]^\top$ is compactly written as $a = \{a^{[i]}\}_{\forall i \in \mathcal{N}}$. Given a vector variable v varying according to a discrete-time index $k \in \mathbb{N}$, the value of v at time k is indicated as v_k. Given v_k extracted from a set $\mathcal{V}$, the sequence

spanning from time index k_1 to $k_2 > k_1$, i.e., $\{v_{k_1}, \dots, v_{k_2}\}$, is indicated by $v_{k_1:k_2}$, whereas the set of possible sequences is indicated by $\mathcal{V}_{k_1:k_2} = \{v_{k_1:k_2} \mid v_\tau \in \mathcal{V}\ \forall \tau \in \{k_1, \dots, k_2\}\}$. The $l_{p,\infty}$ norm of a sequence is defined as $\|v_{k_1:k_2}\|_{p,\infty} = \|[\|v_{k_1}\|_p, \dots, \|v_{k_2}\|_p]^\top\|_\infty$.

Finally, some notions used in the paper are introduced below, with more details available in Bayer et al. (2013). Consider the standard definitions of functions of classes $\mathcal{K}$, $\mathcal{K}_\infty$, and $\mathcal{KL}$ as presented in Rawlings et al. (2017). Consider a discrete-time system in the following form:

$$\mathcal{S}: \begin{cases} \chi_{k+1} = f_s(\chi_k, u_k) \\ \zeta_k = g_s(\chi_k, u_k) \end{cases}, \tag{1}$$

where f_s and g_s are generic functions, whereas $\chi \in \mathbb{R}^{n_\chi}$, $u \in \mathbb{R}^{n_u}$, and $\zeta \in \mathbb{R}^{n_\zeta}$ are the system state, input, and output variables, respectively. Hereafter, stability properties are stated with respect to the sets $\mathcal{X} \subseteq \mathbb{R}^{n_\chi}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, where $\mathcal{X}$ is assumed to be positively invariant, that is, for any $u \in \mathcal{U}$, $\chi \in \mathcal{X} \Longrightarrow f_s(\chi, u) \in \mathcal{X}$.

Definition 2.1 (δISS)**.** System (1) is said to be incrementally input-to-state stable (δISS) with respect to $\mathcal{X}$ and $\mathcal{U}$ if there exist functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that, for any $k \in \mathbb{N}$, any pair of initial states $\chi_{a,0}$, $\chi_{b,0} \in \mathcal{X}$, and any pair of input sequences $u_{a,0:k}$, $u_{b,0:k} \in \mathcal{U}_{0:k}$, it holds that

$$\|\chi_{a,k} - \chi_{b,k}\|_p \leq \beta(\|\chi_{a,0} - \chi_{b,0}\|_p, k) + \gamma(\|u_{a,0:k} - u_{b,0:k}\|_{p,\infty}), \tag{2}$$

where $\chi_{l,k}$ denotes the state trajectory of (1) initialized in $\chi_{l,0}$ and fed with the input $u_{l,0:k}$ for $l \in \{a, b\}$. If there exist $\mu \in \mathbb{R}_{>0}$ and $\lambda \in (0, 1)$ such that system (1) enjoys (2) with $\beta(\|\chi_{a,0} - \chi_{b,0}\|_p, k) = \mu\lambda^k\|\chi_{a,0} - \chi_{b,0}\|_p$, system (1) is said to be exponentially δISS.

Definition 2.2 (δISS *Lyapunov Function*)**.** A function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called a δISS Lyapunov function for system (1) if, for any $k \in \mathbb{N}$, any $\chi_{a,k}$, $\chi_{b,k} \in \mathcal{X}$, and any $u_{a,k}$, $u_{b,k} \in \mathcal{U}$, it holds that

(1) there exist $\underline{\phi}, \overline{\phi} \in \mathcal{K}_\infty$ such that

$$\underline{\phi}(\|\chi_{a,k} - \chi_{b,k}\|_p) \leq V(\chi_{a,k}, \chi_{b,k}) \leq \overline{\phi}(\|\chi_{a,k} - \chi_{b,k}\|_p); \tag{3}$$

(2) there exist $\xi \in \mathcal{K}_\infty$ and $\gamma \in \mathcal{K}$ such that

$$\begin{aligned} &V(\chi_{a,k+1}, \chi_{b,k+1}) - V(\chi_{a,k}, \chi_{b,k}) \leq \\ &-\xi(\|\chi_{a,k} - \chi_{b,k}\|_p) + \gamma(\|u_{a,k} - u_{b,k}\|_p). \end{aligned} \tag{4}$$

Condition (4) corresponds to claim that there exist $\kappa, \alpha \in \mathcal{K}_\infty$ such that

$$\begin{aligned} &\kappa(\|\chi_{a,k} - \chi_{b,k}\|_p) \geq (\|u_{a,k} - u_{b,k}\|_p) \Longrightarrow \\ &V(\chi_{a,k+1}, \chi_{b,k+1}) - V(\chi_{a,k}, \chi_{b,k}) \leq -\alpha(\|\chi_{a,k} - \chi_{b,k}\|_p). \end{aligned} \tag{5}$$

Theorem 2.1 (*Bayer et al., 2013*)**.** *If system (1) admits a δISS Lyapunov function, then it is δISS.*

3. Physics-informed data-based modeling via networked system topology

Consider a physical plant $\mathcal{S}$ as in (1) and suppose it is a networked system composed of n subsystems $\mathcal{S}^{[i]}$, for $i = 1, \dots, n$. These subsystems are interconnected according to a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \dots, n\}$ represents the set of nodes, i.e., the set of subsystems $\mathcal{S}^{[i]}$, with $i \in \mathcal{N}$, whereas $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ identifies the set of edges, expressing the interconnections among different subsystems. In particular, an edge $(i, j) \in \mathcal{E}$ indicates that $\mathcal{S}^{[j]}$ has a physical influence on $\mathcal{S}^{[i]}$, and $(i, i) \notin \mathcal{E}$ by convention. For instance, in electricity networks, nodes represent generators and loads, while edges correspond to the network lines (La Bella et al., 2022). Similarly, district heating networks are composed of thermal sources and users (nodes), and of water pipelines (edges) (Nigro et al., 2025).

For a given subsystem $\mathcal{S}^{[i]}$, the set of inlet neighbors, i.e., the ones having a direct physical influence on $\mathcal{S}^{[i]}$, is denoted as $\mathcal{I}_i := \{j \in \mathcal{N} \mid (i, j) \in \mathcal{E}\}$, whereas the set of outlet neighbors, i.e., the ones directly physically influenced by $\mathcal{S}^{[i]}$, is denoted as $\mathcal{O}_i := \{j \in \mathcal{N} \mid (j, i) \in \mathcal{E}\}$. Thus, similarly to Lunze (1992), each $\mathcal{S}^{[i]}$ is assumed to be characterized by an output $\zeta^{[i]}$ and an augmented input $v^{[i]} := [u^{[i]\top} \{\zeta^{[j]}\}^\top_{\forall j \in \mathcal{I}_i}]^\top$ comprising the local inputs of $\mathcal{S}^{[i]}$, i.e., $u^{[i]}$, and the outputs of those subsystems that have a physical influence on $\mathcal{S}^{[i]}$, i.e., $\{\zeta^{[j]}\}_{\forall j \in \mathcal{I}_i}$. The local input of each $\mathcal{S}^{[i]}$ is assumed to belong to a compact set, i.e., $u^{[i]} \in \mathcal{U}^{[i]} := \{u^{[i]} \in \mathbb{R}^{n_u^{[i]}} : \|u^{[i]}\|_\infty \leq \breve{u}^{[i]}\}$, where $\breve{u}^{[i]} \in \mathbb{R}^{n_u^{[i]}}$. On the whole, considering the formulation of the networked system $\mathcal{S}$ in (1), its overall input and output variables include the ones of each $\mathcal{S}^{[i]}$, i.e., $u = \{u^{[i]}\}_{i=1}^n$ and $\zeta = \{\zeta^{[i]}\}_{i=1}^n$. Note that since networked systems may feature a topology with a large number of nodes and edges, this can be simplified using clustering algorithms (Tang & Daoutidis, 2018; Xu et al., 2021), yielding graphs with reduced complexity while preserving the main underlying structure and interactions of the system.

The first objective of this work is to develop a data-based model $\mathcal{M}$ of the networked system $\mathcal{S}$, exploiting its topology as prior physical knowledge to achieve improved performance with respect to traditional data-based approaches. To accomplish this, it is assumed that local input–output measurements of each $\mathcal{S}^{[i]}$, i.e., $u^{[i]}$ and $\zeta^{[i]}$, are available, and that the topology of $\mathcal{S}$, i.e., $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, is known.

The proposed physics-informed modeling approach first involves defining for each subsystem $\mathcal{S}^{[i]}$, with $i \in \mathcal{N}$, a corresponding submodel $\mathcal{M}^{[i]}$ of type

$$\mathcal{M}^{[i]}: \begin{cases} x^{[i]}_{k+1} = f^{[i]}(x^{[i]}_k, v^{[i]}_k; \tilde{\Theta}^{[i]}) \\ y^{[i]}_k = W^{[i]}x^{[i]}_k + b^{[i]} \end{cases}, \tag{6}$$

where $f^{[i]}$ is a generic bounded function, $x^{[i]} \in \mathbb{R}^{n_x^{[i]}}$, $v^{[i]} \in \mathbb{R}^{n_v^{[i]}}$, and $y^{[i]} \in \mathbb{R}^{n_y^{[i]}}$ are the ith submodel state, input, and output variables, respectively. Additionally, $\Theta^{[i]} := \tilde{\Theta}^{[i]} \cup \{W^{[i]}, b^{[i]}\}$ contains the submodel parameters that must be tuned during the training procedure, as discussed in Section 5.

Then, the n submodels $\mathcal{M}^{[i]}$ are interconnected according to the networked system topology described by the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Consistently with the augmented input of each subsystem $v^{[i]}$, each $\mathcal{M}^{[i]}$ is designed to have as input

$$v^{[i]} := [u^{[i]\top} \{y^{[j]}\}^\top_{\forall j \in \mathcal{I}_i}]^\top, \tag{7}$$

which includes the local inputs $u^{[i]}$ and the outputs of the inlet neighbor submodels $\{y^{[j]}\}_{\forall j \in \mathcal{I}_i}$, so as to embed the topology of the networked system. By defining the variables $x := \{x^{[i]}\}_{i=1}^n \in \mathbb{R}^{n_x}$, $u := \{u^{[i]}\}_{i=1}^n \in \mathbb{R}^{n_u}$, $y := \{y^{[i]}\}_{i=1}^n \in \mathbb{R}^{n_y}$, and the parameters $\tilde{\Theta} := \{\tilde{\Theta}^{[i]}\}_{i=1}^n$, $W := \{W^{[i]}\}_{i=1}^n$, $b := \{b^{[i]}\}_{i=1}^n$, the overall physics-informed networked model $\mathcal{M}$ is expressed as follows:

$$\mathcal{M}: \begin{cases} x_{k+1} = f(x_k, u_k; \tilde{\Theta}) \\ y_k = Wx_k + b \end{cases}, \tag{8}$$

where $\Theta := \tilde{\Theta} \cup \{W, b\}$ contains all model parameters.

Note that the formulation of each $\mathcal{M}^{[i]}$ in (6), featuring a nonlinear (or linear) state dynamics and a linear output relationship, is representative of various RNN architectures, such as gated recurrent unit (GRU), long short-term memory (LSTM), neural nonlinear autoregressive exogenous (NNARX), and echo state network (ESN) (Bonassi et al., 2022). Given their strong

modeling capabilities, these RNN architectures will be considered in the following of the work, and thus (8) will be referred to as a physics-informed RNN (PI-RNN), reflecting its topology-based structure. Nonetheless, the proposed approach can be applied to any data-based model class formulated as in (6), where different submodels are interconnected according to the networked system topology.

Assuming that systems $\mathcal{S}^{[i]}$, $\forall i \in \mathcal{N}$, and $\mathcal{S}$ enjoy the δISS property, the stability of each submodel $\mathcal{M}^{[i]}$ and of the resulting model $\mathcal{M}$ is not guaranteed a priori. On the one hand, enforcing δISS for each submodel $\mathcal{M}^{[i]}$ in (6) can be accomplished for the aforementioned RNN architectures by exploiting suitable conditions on $\Theta^{[i]}$, as outlined in Bonassi et al. (2022). However, an arbitrary interconnection of even δISS submodels does not necessarily imply the δISS of the networked model $\mathcal{M}$ in (8), due to the possible presence of inner loops. In view of this, a novel condition ensuring the δISS property for arbitrarily interconnected δISS discrete-time submodels is presented in the next section.

4. A novel δiss condition for discrete-time networked models

In this section, we consider the networked model $\mathcal{M}$ in (8) and we derive a sufficient condition to ensure its δISS property. The following assumption is first introduced.

Assumption 4.1. For each submodel $\mathcal{M}^{[i]}$ in (6), for any input sequence $v^{[i]} \in \mathbb{R}^{n_v^{[i]}}$ there exists $\check{x}^{[i]} \in \mathbb{R}_{>0}$ such that $x^{[i]} \in \check{\mathcal{X}}^{[i]} := \{x^{[i]} \in \mathbb{R}^{n_x^{[i]}} : \|x^{[i]}\|_\infty \leq \check{x}^{[i]}\}$. Moreover, considering a given compact set $\mathcal{V}^{[i]} \subset \mathbb{R}^{n_v^{[i]}}$, if $v^{[i]} \in \mathcal{V}^{[i]}$, then $\mathcal{M}^{[i]}$ admits a compact invariant set $\mathcal{X}^{[i]} \subset \check{\mathcal{X}}^{[i]}$, that is, $x^{[i]} \in \mathcal{X}^{[i]} \implies f^{[i]}(x^{[i]}, v^{[i]}; \widetilde{\Theta}^{[i]}) \in \mathcal{X}^{[i]}$ for any $v^{[i]} \in \mathcal{V}^{[i]}$.

Note that Assumption 4.1 is not restrictive, but rather it is applicable to many classes of data-based models, such as the aforementioned RNNs, whose state dynamics are characterized by bounded activation functions (e.g., sigmoidal and hyperbolic tangent functions). In a few words, Assumption 4.1 first implies the existence of a compact set $\check{\mathcal{X}}^{[i]}$ that bounds all possible state trajectories. Regarding the invariant set $\mathcal{X}^{[i]} \subset \check{\mathcal{X}}^{[i]}$, it depends on the specific RNN class: for further details, see Armenio et al. (2019a), Bonassi et al. (2021b), Terzi et al. (2021), where invariant sets are derived for LSTMs, GRUs, and ESNs, respectively. Additionally, the following assumption is introduced.

Assumption 4.2. Each submodel $\mathcal{M}^{[i]}$ in (6) is characterized by a δISS Lyapunov function $V^{[i]}$ such that, for any $k \in \mathbb{N}$, any $x_{a,k}^{[i]}, x_{b,k}^{[i]} \in \mathcal{X}^{[i]}$, and any $v_{a,k}^{[i]}, v_{b,k}^{[i]} \in \mathcal{V}^{[i]}$,

(1) there exist $\underline{\phi}^{[i]}, \overline{\phi}^{[i]} \in \mathbb{R}_{>0}$ such that

$$\underline{\phi}^{[i]} \|x_{a,k}^{[i]} - x_{b,k}^{[i]}\|_p^q \leq V^{[i]}(x_{a,k}^{[i]}, x_{b,k}^{[i]}) \leq \overline{\phi}^{[i]} \|x_{a,k}^{[i]} - x_{b,k}^{[i]}\|_p^q, \tag{9}$$

(2) there exist $\xi^{[i]}, \gamma^{[i]} \in \mathbb{R}_{>0}$ such that

$$\begin{aligned} &V^{[i]}(x_{a,k+1}^{[i]}, x_{b,k+1}^{[i]}) - V^{[i]}(x_{a,k}^{[i]}, x_{b,k}^{[i]}) \leq \\ &- \xi^{[i]} \|x_{a,k}^{[i]} - x_{b,k}^{[i]}\|_p^q + \gamma^{[i]} \|v_{a,k}^{[i]} - v_{b,k}^{[i]}\|_p^q, \end{aligned} \tag{10}$$

where $p = q$ or $p = \infty \wedge q = 1$.

Assumption 4.2 implies that each $\mathcal{M}^{[i]}$ is exponentially δISS, since the functions $\xi(\cdot)$ and $\gamma(\cdot)$ in (4) are linear, with coefficients $\xi^{[i]}$ and $\gamma^{[i]}$ (Bof et al., 2018). Note that δISS Lyapunov functions are available in the literature for many RNN architectures — for instance, (Bonassi et al., 2021a) for GRUs, (Bonassi et al., 2023) for LSTMs, (Bonassi et al., 2021b) for NNARXs, (Armenio et al., 2019b) for ESNs, and D'Amico et al. (2023) for a generic class of RNNs. As will be evident from Section 5, this assumption can be enforced for the mentioned RNN-based classes through suitable conditions on the model parameters during training. Note that the results provided in the paper could be extended for any $p, q \in \mathbb{R}_{>0}$ at the price of additional mathematical effort, whose details are not reported here for the sake of clarity. Before presenting the main result on the δISS of networked models, the following lemma must be introduced.

Lemma 4.1. *Consider the input u of the networked model $\mathcal{M}$ in (8) constrained in a compact set $\mathcal{U} := \bigtimes_{i=1}^{n} \mathcal{U}^{[i]} \subset \mathbb{R}^{n_u}$. Then, $\mathcal{M}$ admits a compact invariant set defined as $\mathcal{X} := \bigtimes_{i=1}^{n} \mathcal{X}^{[i]} \subset \mathbb{R}^{n_x}$.*

The proof of Lemma 4.1 is reported in Appendix A.1.

The previous assumptions and lemma can be used to establish a sufficient condition to ensure the δISS of interconnected δISS discrete-time submodels, as outlined below.

Theorem 4.1. *Consider the networked model $\mathcal{M}$ in (8) composed of n submodels $\mathcal{M}^{[i]}$ interconnected according to a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with each $\mathcal{M}^{[i]}$ respecting Assumptions 4.1 and 4.2. Consider a matrix $\Gamma \in \mathbb{R}^{n \times n}$, where each element is defined as*

$$\Gamma_{ij} := \begin{cases} \dfrac{\gamma^{[i]} \|W^{[j]}\|_p^q}{\beta^{[i]} \underline{\phi}^{[j]}} & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $\beta^{[i]} \in (0, \xi^{[i]}/\overline{\phi}^{[i]})$. If

$$\rho(\Gamma) < 1, \tag{12}$$

then $\mathcal{M}$ is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$. In particular, $\mathcal{M}$ admits the δISS Lyapunov function

$$V(x_{a,k}, x_{b,k}) := \sum_{i=1}^{n} \sigma^{[i]} V^{[i]}(x_{a,k}^{[i]}, x_{b,k}^{[i]}), \tag{13}$$

where $\sigma^{[i]} \in \mathbb{R}_{>0}$ is defined such that $\sigma^{[i]} \beta^{[i]} - \sum_{j \in \mathcal{O}_i} \Gamma_{ji} \sigma^{[j]} \beta^{[j]} > 0$, $\forall i \in \mathcal{N}$.

The proof of Theorem 4.1 is reported in Appendix A.2.

Condition (12) can be used to verify the δISS property of a networked model $\mathcal{M}$. Interestingly, two alternative conditions derived from (12) are hereafter established, which, despite being more conservative, show to be simpler to enforce during the training procedure of $\mathcal{M}$, see Section 5.

Corollary 4.1. *Consider the networked model $\mathcal{M}$ in (8) composed of n submodels $\mathcal{M}^{[i]}$ interconnected according to $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with each $\mathcal{M}^{[i]}$ respecting Assumptions 4.1 and 4.2. Let the matrix Γ be defined as in (11). If*

$$\sum_{j=1, j \neq i}^{n} \Gamma_{ij} < 1, \quad \forall i \in \mathcal{N}, \tag{14}$$

then $\mathcal{M}$ is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$.

Corollary 4.2. *Consider the networked model $\mathcal{M}$ in (8) composed of n submodels $\mathcal{M}^{[i]}$ interconnected according to $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with each $\mathcal{M}^{[i]}$ respecting Assumptions 4.1 and 4.2. Let Γ be defined as in (11), and let its characteristic polynomial be $\Phi(\lambda) = \lambda^n + c_1 \lambda^{n-1} + \ldots + c_n$, where each c_i explicitly depends on the elements of Γ. If*

$$\sum_{i=1}^{n} |c_i| < 1, \tag{15}$$

then $\mathcal{M}$ is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$.

The proofs of Corollaries 4.1 and 4.2 are reported in Appendices A.3 and A.4, respectively.

4.1. Plug-and-play operations

The derived δISS conditions enable plug-and-play (P&P) operations on the networked model $\mathcal{M}$ due to its modularity. P&P schemes are commonly employed in decentralized and distributed control frameworks to manage the integration or removal of subsystems from a networked system, allowing for the redesign of only the corresponding local regulators to maintain the overall stability (Riverso et al., 2013). Inspired by this framework, we consider here the scenario where some subsystems undergo changes, implying that the corresponding submodels must be redefined. For example, in a chemical plant, certain components may deteriorate due to wear and require replacement; in an electricity network, a power generator might be isolated or connected; in a district heating network, thermal users may be added or disconnected.

Therefore, novel local conditions for the modified submodels are introduced in the following, guaranteeing that the δISS of the overall networked model is preserved despite these changes.

First, consider the "unplug" case, where a subsystem $\mathcal{S}^{[a]}$ is removed from $\mathcal{S}$, yielding a new directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, with $\tilde{\mathcal{N}} = \mathcal{N}\backslash\{a\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{N}} \times \tilde{\mathcal{N}}$, being a the removed node. At the same time, assume that the corresponding submodel $\mathcal{M}^{[a]}$ is removed from $\mathcal{M}$, leading to a new networked model $\tilde{\mathcal{M}}$, still consistent with the networked system. The following result holds.

Proposition 4.1. *Consider to remove a submodel $\mathcal{M}^{[a]}$ from the networked model $\mathcal{M}$, such that a new model $\tilde{\mathcal{M}}$ is obtained, represented by a directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, with $\tilde{\mathcal{N}} = \mathcal{N}\backslash\{a\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{N}} \times \tilde{\mathcal{N}}$. If the original networked model $\mathcal{M}$ respects (12), i.e., it is δISS, then the new model $\tilde{\mathcal{M}}$ also respects (12), i.e., it is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$.*

The proof of Proposition 4.1 is reported in Appendix A.5.

Consider now the "plug-in" case, where a subsystem $\mathcal{S}^{[a]}$ is added to the original networked system $\mathcal{S}$, yielding a new directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, with $\tilde{\mathcal{N}} = \mathcal{N} \cup \{a\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{N}} \times \tilde{\mathcal{N}}$, being $a = n + 1$ the added node. Consequently, a new corresponding submodel $\mathcal{M}^{[a]}$ must be introduced and plugged into $\mathcal{M}$, forming a new networked model $\tilde{\mathcal{M}}$. The new matrix $\tilde{\Gamma}$, defined as in (11) and associated with $\tilde{\mathcal{M}}$, can be written as

$$\tilde{\Gamma} = \begin{bmatrix} \Gamma & \Gamma_{1a} \\ \Gamma_{a1} & 0 \end{bmatrix}, \tag{16}$$

where $\Gamma_{1a} \in \mathbb{R}^{n\times 1}_{\geq 0}$ and $\Gamma_{a1} \in \mathbb{R}^{1\times n}_{\geq 0}$ include the elements associated with the interconnections between the new submodel $\mathcal{M}^{[a]}$ and the original networked model. As discussed, adding a new submodel might cause instability in a networked model, especially if loops are introduced in the graph. Nevertheless, a sufficient condition on Γ_{1a} and Γ_{a1} can be established such that the δISS property of the networked model is preserved even after the plug-in operation.

Proposition 4.2. *Consider to add a submodel $\mathcal{M}^{[a]}$ to the networked model $\mathcal{M}$, such that the resulting model $\tilde{\mathcal{M}}$ is represented by a directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, with $\tilde{\mathcal{N}} = \mathcal{N} \cup \{a\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{N}} \times \tilde{\mathcal{N}}$. Assume that $\mathcal{M}^{[a]}$ respects Assumptions 4.1 and 4.2, and that the matrix Γ associated with $\mathcal{M}$ respects (12), implying the existence of $v \in \mathbb{R}^n_{>0}$ such that $\Gamma v < v$. Considering the matrix $\tilde{\Gamma}$ associated with $\tilde{\mathcal{M}}$ in (16), if*

$$\Gamma_{1a}\Gamma_{a1}v < (I_n - \Gamma)v, \tag{17}$$

then the new model $\tilde{\mathcal{M}}$ respects (12), i.e., it is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$.

The proof of Proposition 4.2 is reported in Appendix A.6.

Condition (17) is advantageous for stability verification because it involves only the parameters associated with the newly added submodel $\mathcal{M}^{[a]}$, i.e., the ones included in Γ_{1a} and Γ_{a1}. Note also that the vector $v \in \mathbb{R}^n_{>0}$ in (17) verifying $\Gamma v < v$ can be easily computed from Γ. Specifically, if Γ is an irreducible matrix, being Schur stable and non-negative, then v can be set to the Perron–Frobenius eigenvector which solves $\Gamma v = \rho(\Gamma)v$ (Berman & Plemmons, 1994). Conversely, if Γ is reducible, heuristic methods to find $v \in \mathbb{R}^n_{>0}$ such that $\Gamma v < v$ are available in the literature, as the one described in Farina and Scattolini (2012, Algorithm 2).

Finally, if the original model $\mathcal{M}$ respects Corollary 4.1, a simpler sufficient condition for ensuring the δISS of the updated model $\tilde{\mathcal{M}}$ in the plug-in scenario can be derived, as reported below.

Corollary 4.3. *Consider to add a submodel $\mathcal{M}^{[a]}$ to the networked model $\mathcal{M}$, such that the resulting model $\tilde{\mathcal{M}}$ is represented by a directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, with $\tilde{\mathcal{N}} = \mathcal{N} \cup \{a\}$ and $\tilde{\mathcal{E}} \subseteq \tilde{\mathcal{N}} \times \tilde{\mathcal{N}}$. Assume that $\mathcal{M}^{[a]}$ respects Assumptions 4.1 and 4.2, and that the matrix Γ associated with $\mathcal{M}$ respects (14). Considering the matrix $\tilde{\Gamma}$ associated with $\tilde{\mathcal{M}}$ in (16), and defining $\mathcal{O}_a = \{i \in \mathcal{N} \mid (i, a) \in \tilde{\mathcal{E}}\}$, if*

$$\sum_{j=1, j\neq i}^{n+1} \tilde{\Gamma}_{ij} < 1, \quad \forall i \in \mathcal{O}_a \cup \{a\}, \tag{18}$$

then the new model $\tilde{\mathcal{M}}$ is δISS with respect to $\mathcal{X}$ and $\mathcal{U}$.

The proof of Corollary 4.3 is reported in Appendix A.7.

5. Training procedure of δiss physics-informed recurrent neural networks

In this section, we show how a δISS networked model $\mathcal{M}$ can be identified from data, enforcing the sufficient conditions provided in Section 4.

We consider submodels belonging to the aforementioned RNN architectures, i.e., those presented in Bonassi et al. (2022). In this case, each $\mathcal{M}^{[i]}$ can be ensured to satisfy Assumption 4.2 under suitable conditions on its parameters $\Theta^{[i]}$, thereby implying its δISS. These conditions can generally be expressed as nonlinear inequalities on the parameters of the RNN submodel $\mathcal{M}^{[i]}$:

$$\tilde{\eta}^{[i]}(\Theta^{[i]}) < 0, \quad \forall i \in \mathcal{N}. \tag{19}$$

Further details regarding the explicit formulation of (19) can be found in Bonassi et al. (2021a) for GRUs, in Bonassi et al. (2023) for LSTMs, in Bonassi et al. (2021b) for NNARXs, and in Armenio et al. (2019b) for ESNs. In these references, it is also shown that δISS Lyapunov functions respecting Assumption 4.2 exist for the aforementioned RNN classes, and their expressions are known. Consequently, the parameters $\underline{\phi}^{[i]}$ and $\overline{\phi}^{[i]}$ in condition (9) can be easily derived. Moreover, as shown in Bonassi (2023a), $\xi^{[i]}$ and $\gamma^{[i]}$ in condition (10) can be expressed as functions of the model parameters $\Theta^{[i]}$ for all the mentioned RNN architectures. This implies that each element of Γ in (11) depends on the model parameters, i.e., $\Gamma_{ij} = \Gamma_{ij}(\Theta^{[i]}, \Theta^{[j]})$, and, consequently, $\Gamma = \Gamma(\Theta)$. It is worth noting that the parameters composing each element of Γ are derived through conservative relationships based on Lyapunov functions. This may affect the modeling performance of the trained PI-RNN, although in the case study of Section 6.2 the reduction in accuracy is minor. On the other hand, imposing the δISS constraint can also be beneficial, as it reduces the model's degrees of freedom and confers robustness properties that may be helpful for control design (Xie et al., 2025). Note also that $\beta^{[i]}$

in (11) is a degree of freedom, initially set to a fixed value and, if the condition $\beta^{[i]} \in (0, \xi^{[i]}/\overline{\phi}^{[i]})$ is not met after training, its value is reduced and the training is repeated.

At this point, to impose the δISS property of the overall networked PI-RNN model (8), it is possible to express condition (12) in Theorem 4.1 as a generic function of Θ, given that $\Gamma = \Gamma(\Theta)$, i.e.,

$$\eta(\Theta) < 0. \tag{20}$$

It is worth noting that the conditions outlined in Corollaries 4.1 and 4.2, although more conservative than (12), can be expressed more easily as (20) due to their simpler explicit formulation.

We here propose to identify the networked model $\mathcal{M}$ in (8) through a unique training procedure that leverages available input–output measurements of each subsystem $\mathcal{S}^{[i]}$ and enforces conditions (19) and (20), implying the δISS of $\mathcal{M}$. These conditions can be included in the loss function used to train $\mathcal{M}$ through regularization terms designed to be monotonically increasing when their argument is positive and zero otherwise, e.g., by exploiting piecewise-linear functions. In particular, $\tilde{\omega}^{[i]}(\tilde{\eta}^{[i]}(\Theta^{[i]}))$ is introduced as the regularization term that enforces (19) for each $\mathcal{M}^{[i]}$, verifying Assumption 4.2. On the other hand, we introduce $\omega(\eta(\Theta))$ as the regularization term enforcing (20), thereby ensuring the δISS property of the networked PI-RNN model $\mathcal{M}$. Altogether, the loss function used to train $\mathcal{M}$ is formulated as follows:

$$\mathcal{L}(\mathcal{D}_{\text{tr}}, \Theta) = \text{MSE}(\mathcal{D}_{\text{tr}}; \Theta) + \sum_{i=1}^{n} \tilde{\omega}^{[i]}(\tilde{\eta}^{[i]}(\Theta^{[i]})) + \omega(\eta(\Theta)), \tag{21}$$

where $\text{MSE}(\mathcal{D}_{\text{tr}}; \Theta)$ denotes the mean squared error, which measures the fitting quality of the PI-RNN model onto the training dataset $\mathcal{D}_{\text{tr}}$ (Bonassi, 2023a). As evident, the loss function (21) aims to seek the model parameters Θ that maximize modeling accuracy by minimizing the prediction error with respect to data, while enforcing the δISS conditions (19) and (20). If the latter are respected, the proposed modeling approach preserves the δISS property of the individual submodels $\mathcal{M}^{[i]}$ and transfers it to the networked model $\mathcal{M}$ in view of the results presented in Section 4. As witnessed by the numerical results in Section 6.2, the PI-RNN model is expected to achieve superior fitting performance compared to standard RNNs, thanks to the embedded networked system topology, while also ensuring the δISS property.

Remark 5.1. Since the stability conditions are enforced through regularization terms in the loss function (21), it is possible that the trained model does not satisfy these conditions. In such cases, the training process is repeated with an increased weight on the corresponding regularization terms until stability properties are met.

Final remarks regarding plug-and-play features are in order, considering the discussion in Section 4.1. Thanks to the modularity of the PI-RNN and of its δISS property, plug/unplug-and-play operations on $\mathcal{M}$ can be easily performed without retraining the entire model, while still ensuring physical consistency with the networked system and preserving the overall δISS property of the networked model.

In particular, the first time a system is identified, (21) is leveraged to train the networked model $\mathcal{M}$. Then, if a subsystem $\mathcal{S}^{[a]}$ is unplugged from $\mathcal{S}$, resulting in a new directed graph, a new training is not necessary. The corresponding submodel $\mathcal{M}^{[a]}$ must be removed from $\mathcal{M}$, leading to a reduced model $\tilde{\mathcal{M}}$ that still accurately resembles the updated system, thanks to the modularity of the PI-RNN. Moreover, given Proposition 4.1, it follows that $\tilde{\mathcal{M}}$ automatically inherits the δISS property from $\mathcal{M}$.

If instead a subsystem $\mathcal{S}^{[a]}$ is plugged into $\mathcal{S}$, it is sufficient to train a new submodel $\mathcal{M}^{[a]}$, enforcing its individual δISS condition by means of Proposition 4.2, and then integrate it into the original networked model $\mathcal{M}$ consistently with the new topology. In this way, the resulting model $\tilde{\mathcal{M}}$ accurately represents the updated system while preserving its δISS property.

6. Case study and control application

Before presenting the case study, we provide some guidelines for devising a control system, as an NMPC scheme will be tested in the numerical experiments to further motivate the benefits of enforcing δISS on the PI-RNN. Having a model with this property allows in fact to design *(i)* decentralized observers with guaranteed convergence of the overall state estimate to the true one, and *(ii)* NMPC schemes with proven convergence guarantee to the equilibrium. The design of these elements, which constitute the control system depicted in Fig. 1, is described in Section 6.1, while Section 6.2 is devoted to present the numerical results.

6.1. Control system

Consider that the physical plant $\mathcal{S}$ has been modeled as described in Section 3, where the networked model $\mathcal{M}$ and the submodels $\mathcal{M}^{[i]}$ have been trained to enjoy the δISS property as described in Section 5. Since the states have no physical meaning in most RNN models, they cannot be measured, and an observer is therefore required to estimate them for MPC initialization. Thanks to the modular structure of the proposed PI-RNN model, a state observer can be designed for each $\mathcal{M}^{[i]}$ using input–output measurements of the corresponding $\mathcal{S}^{[i]}$. The associated state observer $\mathcal{SO}^{[i]}$ reads as

$$\mathcal{SO}^{[i]}: \begin{cases} \hat{x}^{[i]}_{k+1} = f_o^{[i]}(\hat{x}^{[i]}_k, u^{[i]}_k, \zeta^{[i]}_k; \tilde{\Theta}^{[i]}_o) \\ \hat{y}^{[i]}_k = W_o^{[i]} \hat{x}^{[i]}_k + b_o^{[i]} \end{cases}, \tag{22}$$

where $\hat{x}^{[i]} \in \mathbb{R}^{n_x^{[i]}}$ and $\hat{y}^{[i]} \in \mathbb{R}^{n_y^{[i]}}$ are the estimated states and outputs, respectively, whereas the overall set of parameters of the ith observer is $\Theta_o^{[i]} := \tilde{\Theta}_o^{[i]} \cup \{W_o^{[i]}, b_o^{[i]}\}$. Considering the aforementioned class of RNNs, each observer $\mathcal{SO}^{[i]}$ can be designed to be weakly exponential, as defined in Vidyasagar (2002). Observers with exponential convergence guarantees have been proposed in the literature for δISS RNN models, e.g., in Bonassi et al. (2024) for GRU networks, in Bonassi et al. (2023) for LSTM, and in Armenio et al. (2019a) for ESN, but their design procedures do not apply when dealing with multiple interconnected RNNs. Nevertheless, the modular PI-RNN structure enables to design a decentralized exponentially convergent state observer $\mathcal{SO}$, composed of independent exponentially convergent $\mathcal{SO}^{[i]}$, to retrieve the overall state estimate $\hat{x}$ of $\mathcal{M}$ from $u^{[i]}$ and $\zeta^{[i]}$ (Vidyasagar, 2002).

To optimally operate the considered networked system, an NMPC regulator can be designed under the well-known certainty equivalence principle (James, 1994). For instance, the control formulation proposed in Bonassi et al. (2024) can be employed, with the objective to track a specified equilibrium $\bar{y}$ corresponding to the pair $(\bar{x}, \bar{u})$, assumed to exist. Thanks to the δISS property of the PI-RNN model $\mathcal{M}$, enforced as discussed in Sections 4 and 5, the convergence result established in Bonassi et al. (2024) can be leveraged to derive an NMPC law that ensures recursive feasibility and convergence to the equilibrium.

Remark 6.1. The plug-and-play feature of the proposed approach is also effective from a control perspective. In particular, if a subsystem $\mathcal{S}^{[a]}$ is added to (or removed from) the networked system $\mathcal{S}$, the predictive model embedded in an NMPC regulator can be partially updated as described in Section 5. Similarly, the decentralized observer $\mathcal{SO}$ can be modified by adding (or removing) a sub-observer $\mathcal{SO}^{[a]}$ corresponding to the subsystem $\mathcal{S}^{[a]}$.

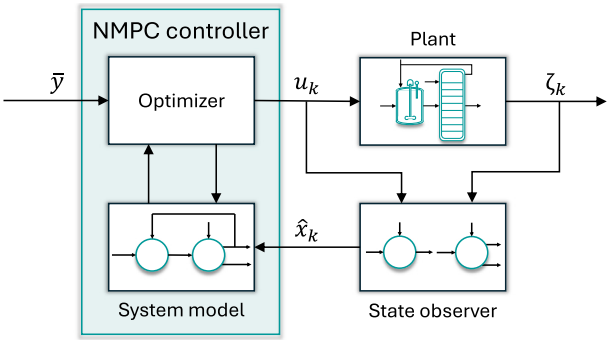

Fig. 1. Scheme of the considered control architecture.

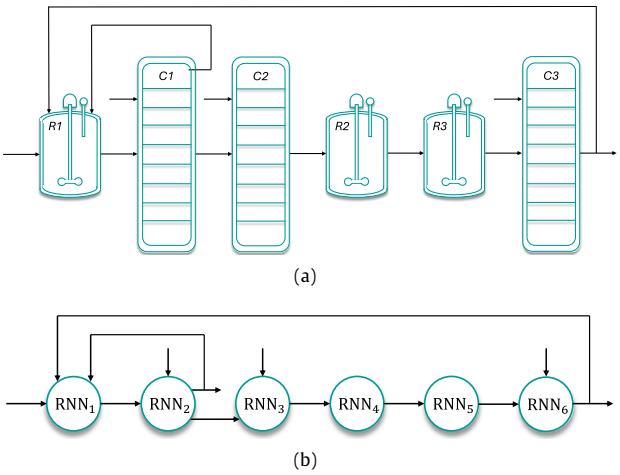

Fig. 2. (a) Schematics of the chemical plant: $R1$, $R2$, $R3$ are the reactors, $C1$, $C2$, $C3$ are the distillation columns. (b) PI-RNN architecture for the considered chemical plant.

6.2. Case study

The proposed framework is tested on a large-scale chemical plant. This system was initially studied in Luyben and Luyben (1995) and further explored in Scattolini et al. (2009), making it a significant benchmark for plantwide control of complex processes. The considered plant is schematically depicted in Fig. 2(a). It is a large-scale networked system featuring a complex flow sheet, significant interactions among units with recycle streams, and numerous byproduct/intermediate components. The physical model of the plant, characterized by 120 state variables, is not reported here for space reasons, but more details are available in the mentioned references. This model has been leveraged to develop a dynamic simulator within the Simulink environment in MATLAB, exploiting the parameters reported in Scattolini et al. (2009). The simulator based on the physical model is used both for data collection and for control testing.

The considered plant comprises three binary distillation columns ($C1$, $C2$, $C3$) and three chemical reactors ($R1$, $R2$, $R3$). Moreover, two recycle streams are present (from $C1$ to $R1$ and from $C3$ to $R1$), and six chemical components (A, B, C, D, E, F) are processed. In particular, the plant contains two reaction steps, i.e., $A + B \rightarrow C + D$ producing desired product C and $D + E \rightarrow F + B$ producing desired product F and regenerating B for the first reaction step. Following the notation used in this paper, the considered chemical plant is a networked system $\mathcal{S}$ composed of $n = 6$ subsystems $\mathcal{S}^{[i]}$, interconnected according to a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where input–output measurements are available. The subsystems $\mathcal{S}^{[i]}$, with $i \in \{1, \dots, n\}$, correspond to the first chemical reactor $R1$, the first and second distillation columns $C1$ and $C2$, the second and third reactors $R2$ and $R3$, and the third column $C3$, respectively.

The control variables selected for the system $\mathcal{S}$ consist of the inlet feed flow rate of the first reactor and the reflux flow rates of the three distillation columns, i.e., $u = [u_{R1}^\top \, u_{C1}^\top \, u_{C2}^\top \, u_{C3}^\top]^\top$. The output vector is $\zeta = [\zeta_{R1\to C1}^\top \zeta_{C1\to R1}^\top \zeta_{C1\to C2}^\top \zeta_{C2\to R2}^\top \zeta_{R2\to R3}^\top \zeta_{R3\to C3}^\top \zeta_{C3\to R1}^\top]^\top$, where each $\zeta_{\alpha\to\beta}$ includes the molar fractions of four chemical components multiplied by the corresponding output flow rates. Overall, the system comprises four input and twenty-eight output variables.

The presence of recirculating flows among various elements within the chemical plant makes the considered system strongly interacting. Furthermore, despite its modeling complexity, the topology of the plant is known and a large amount of data is assumed to be available through measurements of the input and output variables of each subsystem. This motivates the application of the proposed physics-informed data-based modeling approach to the aforementioned chemical plant. In particular, the implementation of the PI-RNN networked model is carried out in Python (version 3.10), customizing the library developed in Bonassi (2023b) to enforce the physics-informed structure and the δISS property for interconnected submodels. An NMPC problem is also tested and implemented in MATLAB R2023a using the CasADi environment and the Ipopt solver. All computations are performed on a laptop equipped with an Intel Core i7-1195G7 processor.

To accurately identify a prediction model $\mathcal{M}$ of the considered plant $\mathcal{S}$, a dataset of input–output samples is collected using multilevel pseudorandom binary sequences to vary the inputs u. Overall, 10000 samples are collected with a sampling time of $\tau_s = 1$s, properly divided into training (80%), validation (10%), and test (10%) sets.

For a fair comparison, this large-scale system is initially identified using a standard black-box RNN model with 3 neurons per 6 hidden layers, having four inputs and twenty-eight outputs. Then, a networked PI-RNN is identified, employing $n = 6$ single-layer RNNs interconnected to resemble the chemical plant topology, as shown in Fig. 2(b). According to the modeling approach proposed in Section 3, the resulting model is thus a PI-RNN with 3 neurons per 6 interconnected RNNs. Ultimately, both models employ GRU networks and are characterized by a total of 18 states, one for each neuron. Moreover, they are trained using the truncated backpropagation through time (TBPTT) method, which is thoroughly described in Bonassi (2023a). For the standard RNN, the δISS property is enforced on the overall model during its training process as discussed in Bonassi et al. (2021a). Conversely, for training the PI-RNN model, the loss function (21) is exploited, where the δISS of the single submodels is enforced using the condition in Bonassi et al. (2021a) to define (19), whereas the δISS of the networked model is enforced using condition (15) to define (20). The regularization terms in (21) are both designed as piecewise-linear functions, e.g., $\omega(\eta(\Theta)) = \underline{\pi} \min(0, \eta(\Theta)) + \overline{\pi} \max(0, \eta(\Theta))$, with $0 < \underline{\pi} \ll \overline{\pi}$ (Bonassi et al., 2022). Additionally, both models are trained with the 10000-sample dataset over 2000 epochs, employing the adaptive moment estimation (ADAM) optimizer and a learning rate of 0.003.

To quantitatively evaluate the modeling performance of the developed data-based models, the set $\mathcal{D}_{\text{te}}$ is introduced, including the time instants of the samples contained in the test set. Hence, the FIT index is employed:

$$\text{FIT} = \left(1 - \frac{\sum_{k\in\mathcal{D}_{\text{te}}} \|\zeta_k - y_k\|_2}{\sum_{k\in\mathcal{D}_{\text{te}}} \| \zeta_k - \zeta_{\text{avg}}\|_2}\right) \cdot 100, \tag{23}$$

Table 1
Comparison between the average FIT and its standard deviation for an RNN and a PI-RNN, both having 18 states and trained five times with 10000 samples, together with the average epoch time.

	FIT [%]	Average epoch time [s]
Standard RNN	67.5 ± 1.4	7.9 ± 1.4
PI-RNN	82.4 ± 2.9	9.0 ± 1.1

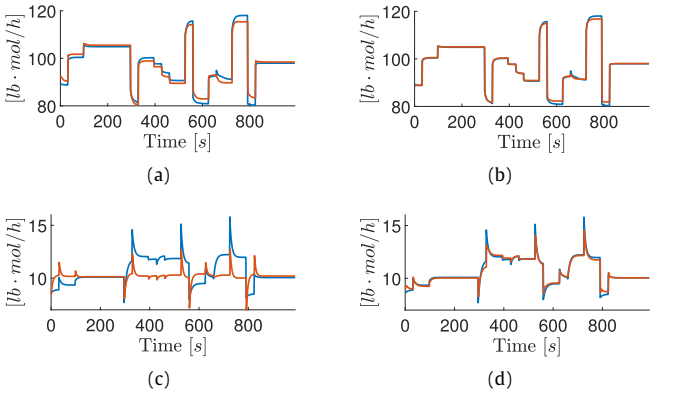

Fig. 3. RNN and PI-RNN identification results. The identified variable is depicted in orange, the measured one in blue. (a) $y_{R1\to C1,A}$ identified by the standard RNN; (b) $y_{R1\to C1,A}$ identified by the PI-RNN; (c) $y_{C2\to R2,D}$ identified by the standard RNN; (d) $y_{C2\to R2,D}$ identified by the PI-RNN.

where y and ζ are the identified and measured outputs over the test set, respectively, whereas $\zeta_{\text{avg}} = \frac{1}{|\mathcal{D}_{\text{te}}|}\sum_{k\in\mathcal{D}_{\text{te}}}\zeta_k$ corresponds to the average of the measured outputs.

To ensure robustness, the identification procedure is repeated five times due to the random initialization of RNNs weights and biases, as well as the random subsequences extracted by the TBPTT method. Note that the final model selected for trend prediction and NMPC is the one achieving the highest FIT. The obtained results are reported in Table 1, which includes the average FIT value and its standard deviation for the standard RNN and the PI-RNN, as well as the average computation time per epoch. These results, together with the predicted trends shown in Fig. 3, indicate that the PI-RNN consistently outperforms the standard RNN, even though both models use the same number of states. This improvement is achieved by leveraging the networked system topology. It is worth noting that the implemented models verify the derived δISS conditions, being the latter enforced in their loss functions. Fig. 4(a) reports the evolution of $\|x_{a,k} - x_{b,k}\|_\infty$ under different initial conditions and constant inputs, for the δISS PI-RNN model. As visible, all trajectories converge to 0, confirming the δISS condition (2) for the overall networked model. For the sake of completeness, the standard RNN and the PI-RNN models have also been trained without enforcing the δISS regularization terms. As expected, the resulting models do not verify the δISS conditions when these are checked a posteriori, highlighting the importance of enforcing the δISS property during training. Nonetheless, the FIT performance is only slightly affected compared to the δISS-constrained models (see Table 1), reaching 68.7% for the standard RNN and 86.4% for the PI-RNN.

A final test is carried out to evaluate the plug-and-play feature of the δISS PI-RNN model. Specifically, it is assumed that column $C2$ of the chemical plant undergoes a parameter change, with its vapor flow rate increasing from $385\frac{lb\cdot mol}{h}$ to $435\frac{lb\cdot mol}{h}$. This change causes the fitting performance of the PI-RNN model to drop to 78.4%. Nevertheless, thanks to the modularity of the PI-RNN approach, it is not necessary to reidentify the entire networked system model, as would be required with standard RNNs. Instead,

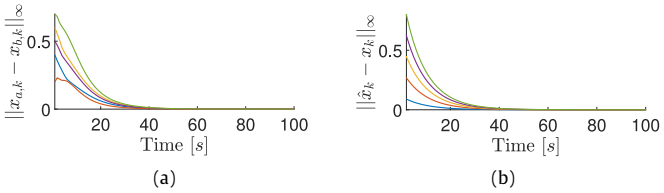

Fig. 4. (a) Convergence of the difference between two model states for different initial conditions and constant inputs. (b) Convergence of the difference between observer and model states for different initial conditions and constant inputs.

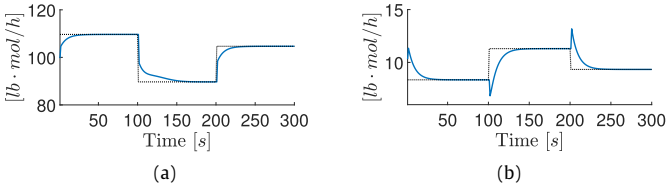

Fig. 5. NMPC results. Set-points are depicted in black dotted lines. (a) $y_{R1\to C1,A}$ trend with respect to its set-point. (b) $y_{C2\to R2,D}$ trend with respect to its set-point.

only the submodel related to $C2$ needs to be updated. This new submodel is trained using new input-ouput data and a loss function that enforces condition (17) as a penalty term. This condition ensures that the δISS property of the overall networked model remains intact, as only the parameters related to the new submodel of $C2$ are learned, while the parameters of the other submodels are unchanged, as discussed in Section 5. This submodel is then plugged into the original PI-RNN model by replacing the old submodel of $C2$. As a result, the updated networked model still verifies the condition in Theorem 4.1, i.e., $\rho(\tilde{\Gamma}) = 0.8 < 1$, without the necessity of retraining the entire model due to a local change. Moreover, the fitting performance of the updated networked model remains satisfactory, i.e., FIT = 80.3%, which is comparable to the results reported in Table 1.

Finally, an NMPC regulator is designed following the approach proposed in Bonassi et al. (2024), highlighting the benefits of leveraging the derived δISS PI-RNN model. First, a set of $n = 6$ weak exponential observers $\mathcal{SO}^{[i]}$ are employed, each estimating the state related to the corresponding submodel $\mathcal{M}^{[i]}$. All observers $\mathcal{SO}^{[i]}$ are designed using the method described in Bonassi et al. (2024), being GRU models employed in the PI-RNN. As discussed in Section 6.1 and illustrated in Fig. 4(b), the overall decentralized observer $\mathcal{SO}$ is also weakly exponentially convergent. The NMPC is executed with a sampling time $\tau_s = 1$s and features a prediction horizon of $N = 12$ and a simulation horizon related to the terminal cost of $M = 12$. The goal of the NMPC problem is to track specific output references. As an example, the trends over 300 s of simulation of $y_{R1\to C1,A}$ with respect to its set-point is depicted in Fig. 5(a) and the one of $y_{C2\to R2,D}$ in Fig. 5(b). The average computation time per iteration is 0.5s. In conclusion, the NMPC regulator exploiting the proposed δISS PI-RNN model is closed-loop convergent, and it is able to steer the desired output variables to their target values exhibiting negligible steady-state error.

7. Conclusions

This article proposes a general methodology for learning large-scale networked systems using physics-informed data-based models with stability properties. The developed algorithm incorporates the physical system topology into the model architecture,

achieving superior fitting performance compared to standard data-based models. Additionally, a novel condition is derived to ensure the δISS property for arbitrarily interconnected δISS submodels by leveraging their δISS Lyapunov functions. This stability condition is enforced in the loss function during the training procedure of a physics-informed recurrent neural network. The modularity of the proposed modeling approach, combined with its derived δISS property, facilitates smooth plug-and-play operations on the model while preserving high fitting performance and stability. The δISS property of the model also allows for the design of a suitable decentralized observer and an NMPC regulator that ensures convergence to equilibrium. The proposed methods are tested in simulation on a large-scale chemical plant referenced in the literature, achieving promising results from both modeling and control perspectives.

Future work will be dedicated to the development of a detection scheme for the identification of potential modifications in a physical system and their specific sources. This will allow to adapt the proposed physics-informed data-based model to such system changes in an online fashion, leveraging active learning.

Acknowledgments

The work of Laura Boca de Giuli, Alessio La Bella, and Riccardo Scattolini was carried out within the MICS (Made in Italy - Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR - M4 C2, Invest 1.3 - D.D. 1551.11-10-2022, PE00000004). CUP MICS D43C22003120001. The work of Marcello Farina has been financially supported by the PRIN 2022 project Control of Assistive Robots in crowded Environments (CARE, Id. 20225LX9M3), and by the Italian Ministry of Enterprises through the project 4D Drone Swarms (4DDS) under grant no. F/310097/01-04/X56.

Appendix

A.1. Proof of Lemma 4.1

Consider the augmented input of $\mathcal{M}^{[i]}$, i.e., $v^{[i]}$, defined as in (7), where $u^{[i]} \in \mathcal{U}^{[i]}$ and $y^{[j]} \in \mathcal{Y}^{[j]} := \{y^{[j]} \in \mathbb{R}^{n_y^{[j]}} : y^{[j]} = W^{[j]}x^{[j]} + b^{[j]} \wedge x^{[j]} \in \check{\mathcal{X}}^{[j]}\}$, $\forall j \in \mathcal{I}_i$.$\mathcal{Y}^{[j]}$ is a compact set since $\check{\mathcal{X}}^{[j]}$ is compact as evident from Assumption 4.1. Therefore, $v^{[i]} \in \mathcal{V}^{[i]} := \mathcal{U}^{[i]} \times \left(\bigtimes_{j\in\mathcal{I}_i} \mathcal{Y}^{[j]}\right)$, where $\mathcal{V}^{[i]}$ is a compact set as well. Thus, given Assumption 4.1, $\mathcal{M}^{[i]}$ admits a compact invariant set $\mathcal{X}^{[i]}$ for any $v^{[i]} \in \mathcal{V}^{[i]}$ also when interconnected to other submodels. In conclusion, defining $\mathcal{X} := \bigtimes_{i=1}^{n} \mathcal{X}^{[i]}$, it holds that $\mathcal{X}$ is a compact invariant set for the networked model $\mathcal{M}$. □

A.2. Proof of Theorem 4.1

Considering (9) and (13), it is possible to bound the candidate δISS Lyapunov function of $\mathcal{M}$ as follows:

$$\underline{\phi}\|x_{a,k} - x_{b,k}\|_p^q \le V(x_{a,k}, x_{b,k}) \le \overline{\phi}\|x_{a,k} - x_{b,k}\|_p^q, \tag{A.1}$$

with $\underline{\phi} = \min_{i\in\mathcal{N}}(\sigma^{[i]}\underline{\phi}^{[i]})$ and $\overline{\phi} = \max_{i\in\mathcal{N}}(\sigma^{[i]}\overline{\phi}^{[i]})$ if $p = q$, whereas $\underline{\phi} = \min_{i\in\mathcal{N}}(\sigma^{[i]}\underline{\phi}^{[i]})$ and $\overline{\phi} = n \cdot \max_{i\in\mathcal{N}}(\sigma^{[i]}\overline{\phi}^{[i]})$ if $p = \infty \wedge q = 1$ (see Assumption 4.2).

The δISS of the overall model is proved exploiting condition (5) in Definition 2.2. In particular, we show that if

$$\psi\|x_{a,k} - x_{b,k}\|_p^q \ge \|u_{a,k} - u_{b,k}\|_p^q, \tag{A.2}$$

with $\psi = \dfrac{\underline{\phi}\cdot\min_{i\in\mathcal{N}}\left(\beta^{[i]} - \sum_{j\in\mathcal{O}_i}\frac{\Gamma_{ji}\beta^{[j]}\sigma^{[j]}}{\sigma^{[i]}}\right)}{n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})}$, then there exists $\epsilon \in \mathbb{R}_{>0}$ such that

$$V(x_{a,k+1}, x_{b,k+1}) - V(x_{a,k}, x_{b,k}) \le -\epsilon\underline{\phi}\|x_{a,k} - x_{b,k}\|_p^q, \tag{A.3}$$

implying $V(x_{a,k}, x_{b,k})$ to be a δISS Lyapunov function.

Note that $\psi > 0$ since $\left(\beta^{[i]} - \sum_{j\in\mathcal{O}_i}\frac{\Gamma_{ji}\beta^{[j]}\sigma^{[j]}}{\sigma^{[i]}}\right) > 0$, $\forall i \in \mathcal{N}$. The latter holds since $\rho(\Gamma) < 1$, as assumed in Theorem 4.1. In fact, considering that $\sigma^{[i]} > 0$ and naming $s^{[i]} = \beta^{[i]}\sigma^{[i]} > 0$, it must hold that $s^{[i]} - \sum_{j\in\mathcal{O}_i}\Gamma_{ji}s^{[j]} > 0$, $\forall i \in \mathcal{N}$, which can be rewritten as $s - \Gamma^\top s > 0$. As discussed in Rüffer (2010), there exists $s \in \mathbb{R}^n_{>0}$ such that $s - \Gamma^\top s > 0$ if, and only if, $\rho(\Gamma^\top) < 1$, i.e., $\rho(\Gamma) < 1$.

To prove the validity of (A.2)–(A.3), we first note that from (9)–(10) it follows that

$$\begin{aligned} &V^{[i]}(x^{[i]}_{a,k+1}, x^{[i]}_{b,k+1}) - V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) \le \\ &- \xi^{[i]}/\overline{\phi}^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) + \gamma^{[i]}\|v^{[i]}_{a,k} - v^{[i]}_{b,k}\|_p^q. \end{aligned} \tag{A.4}$$

Considering (13) and (A.4), it is possible to write:

$$\begin{aligned} &V(x_{a,k+1}, x_{b,k+1}) - V(x_{a,k}, x_{b,k}) = \\ &\sum_{i=1}^{n}\sigma^{[i]}\left(V^{[i]}(x^{[i]}_{a,k+1}, x^{[i]}_{b,k+1}) - V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k})\right) \le \\ &\sum_{i=1}^{n}\sigma^{[i]}\left(-\epsilon^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) - \beta^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) + \gamma^{[i]}\|v^{[i]}_{a,k} - v^{[i]}_{b,k}\|_p^q\right), \end{aligned} \tag{A.5}$$

where $\epsilon^{[i]} + \beta^{[i]} = \xi^{[i]}/\overline{\phi}^{[i]}$, with $\epsilon^{[i]}, \beta^{[i]} \in \mathbb{R}_{>0}$. In view of (7) and of Assumption 4.2, one can write

$$\begin{aligned} &\|v^{[i]}_{a,k} - v^{[i]}_{b,k}\|_p^q \le \|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q + \sum_{j\in\mathcal{I}_i}\|y^{[j]}_{a,k} - y^{[j]}_{b,k}\|_p^q \le \\ &\|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q + \sum_{j\in\mathcal{I}_i}\|W^{[j]}\|_p^q\ \|x^{[j]}_{a,k} - x^{[j]}_{b,k}\|_p^q \le \\ &\|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q + \sum_{j\in\mathcal{I}_i}\frac{\|W^{[j]}\|_p^q}{\underline{\phi}^{[j]}}V^{[j]}(x^{[j]}_{a,k}, x^{[j]}_{b,k}). \end{aligned} \tag{A.6}$$

Therefore, (A.5) can be written as

$$\begin{aligned} &V(x_{a,k+1}, x_{b,k+1}) - V(x_{a,k}, x_{b,k}) \le \\ &\sum_{i=1}^{n}\sigma^{[i]}\Bigg(-\epsilon^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) - \beta^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) + \\ &\gamma^{[i]}\|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q + \sum_{j\in\mathcal{I}_i}\frac{\gamma^{[i]}\|W^{[j]}\|_p^q}{\underline{\phi}^{[j]}}V^{[j]}(x^{[j]}_{a,k}, x^{[j]}_{b,k})\Bigg). \end{aligned} \tag{A.7}$$

In view of (11) and setting $\epsilon = \min_{i\in\mathcal{N}}\epsilon^{[i]}$, it follows that

$$\begin{aligned} &V(x_{a,k+1}, x_{b,k+1}) - V(x_{a,k}, x_{b,k}) \le -\epsilon V(x_{a,k}, x_{b,k}) - \\ &\sum_{i=1}^{n}\sigma^{[i]}\beta^{[i]}V^{[i]}(x^{[i]}_{a,k}, x^{[i]}_{b,k}) + \sum_{i=1}^{n}\gamma^{[i]}\sigma^{[i]}\|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q + \\ &\sum_{i=1}^{n}\sum_{j\in\mathcal{I}_i}\Gamma_{ij}\beta^{[i]}\sigma^{[i]}V^{[j]}(x^{[j]}_{a,k}, x^{[j]}_{b,k}). \end{aligned} \tag{A.8}$$

At this point, note that $\sum_{i=1}^{n}\sum_{j\in\mathcal{O}_i}\Gamma_{ji} = \sum_{i=1}^{n}\sum_{j\in\mathcal{I}_i}\Gamma_{ij}$, and that $\sum_{i=1}^{n}\gamma^{[i]}\sigma^{[i]}\|u^{[i]}_{a,k} - u^{[i]}_{b,k}\|_p^q \le n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})\|u_{a,k} - u_{b,k}\|_p^q$.

Thus, (A.8) implies that

$$V(x_{a,k+1}, x_{b,k+1}) - V(x_{a,k}, x_{b,k}) \le -\epsilon V(x_{a,k}, x_{b,k}) -$$

$$\sum_{i=1}^{n}\sigma^{[i]}\beta^{[i]}V^{[i]}(x_{a,k}^{[i]},x_{b,k}^{[i]})+\sum_{i=1}^{n}\gamma^{[i]}\sigma^{[i]}\|u_{a,k}^{[i]}-u_{b,k}^{[i]}\|_p^q+$$
$$\sum_{i=1}^{n}\sum_{j\in\mathcal{O}_i}\Gamma_{ji}\beta^{[j]}\sigma^{[j]}V^{[i]}(x_{a,k}^{[i]},x_{b,k}^{[i]})\le-\epsilon V(x_{a,k},x_{b,k})-$$
$$\sum_{i=1}^{n}\left(\beta^{[i]}-\sum_{j\in\mathcal{O}_i}\frac{\Gamma_{ji}\beta^{[j]}\sigma^{[j]}}{\sigma^{[i]}}\right)\sigma^{[i]}V^{[i]}(x_{a,k}^{[i]},x_{b,k}^{[i]})+$$
$$n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})\|u_{a,k}-u_{b,k}\|_p^q\qquad\le-\epsilon V(x_{a,k},x_{b,k})-$$
$$\min_{i\in\mathcal{N}}\left(\beta^{[i]}-\sum_{j\in\mathcal{O}_i}\frac{\Gamma_{ji}\beta^{[j]}\sigma^{[j]}}{\sigma^{[i]}}\right)\sum_{i=1}^{n}\sigma^{[i]}V^{[i]}(x_{a,k}^{[i]},x_{b,k}^{[i]})+$$
$$n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})\|u_{a,k}-u_{b,k}\|_p^q,\tag{A.9}$$

which, given the expression of ψ, (13), (A.1), and (A.2), reads as

$$V(x_{a,k+1},x_{b,k+1})-V(x_{a,k},x_{b,k})\le-\epsilon V(x_{a,k},x_{b,k})-$$
$$n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})\left(\frac{\psi}{\phi}V(x_{a,k},x_{b,k})-\|u_{a,k}-u_{b,k}\|_p^q\right)\le$$
$$-\epsilon\underline{\phi}\|x_{a,k}-x_{b,k}\|_p^q-n\cdot\max_{i\in\mathcal{N}}(\gamma^{[i]}\sigma^{[i]})\Big(\psi\|x_{a,k}-x_{b,k}\|_p^q-$$
$$\|u_{a,k}-u_{b,k}\|_p^q\Big)\le-\epsilon\underline{\phi}\|x_{a,k}-x_{b,k}\|_p^q,\tag{A.10}$$

implying that (A.2) entails (A.3). Thus, $V(x_{a,k},x_{b,k})$ is a δISS Lyapunov function respecting (5) and, in view of Theorem 2.1, the networked model (8) is δISS. This concludes the proof. □

A.3. Proof of Corollary 4.1

According to Geršgorin's Theorem (Varga, 2011), the eigenvalues of Γ lie within the union of the discs centered at Γ_{ii} with radius $\sum_{j=1,j\neq i}^{n}|\Gamma_{ij}|$, $\forall i\in\mathcal{N}$. Since $\Gamma_{ii}=0$ for all $i\in\mathcal{N}$ and Γ is a non-negative matrix, see (11), if (14) holds, then all Geršgorin's discs have radius smaller than 1, implying that $\rho(\Gamma)<1$. Thus, by Theorem 4.1, $\mathcal{M}$ is δISS. □

A.4. Proof of Corollary 4.2

According to Enders (2008), if (15) holds, then the eigenvalues of the associated characteristic polynomial have magnitude smaller than 1. This implies that $\rho(\Gamma)<1$, and thus $\mathcal{M}$ is δISS. □

A.5. Proof of Proposition 4.1

According to Farina and Rinaldi (2011, Theorem 13), the matrix Γ in (11), being a non-negative matrix, is Schur stable, i.e., $\rho(\Gamma)<1$, if, and only if, all leading minors of $(I_n-\Gamma)$ are positive. Now, consider unplugging a submodel $\mathcal{M}^{[a]}$. By removing the ath row and column from Γ, the resulting matrix is $\tilde{\Gamma}\in\mathbb{R}_{\geq 0}^{n-1\times n-1}$, corresponding to the new model $\tilde{\mathcal{M}}$. Since all leading minors of $(I_n-\Gamma)$ are positive, all leading minors of $(I_{n-1}-\tilde{\Gamma})$ are also positive. This implies that $\tilde{\Gamma}$ is Schur stable and, consequently, the δISS condition in Theorem 4.1 is satisfied for $\tilde{\mathcal{M}}$ as well. □

A.6. Proof of Proposition 4.2

Being Γ a non-negative matrix, there exists $v\in\mathbb{R}_{>0}^{n}$ that verifies $\Gamma v<v$ if, and only if, (12) holds (Rüffer, 2010). Considering that $\tilde{\Gamma}$ in (16) is non-negative as well, we need to show that, under (17), there exists $\tilde{v}$ such that $\tilde{\Gamma}\tilde{v}<\tilde{v}$. In particular, under (17), there exists a small enough $\epsilon\in\mathbb{R}_{>0}$ such that

$$\Gamma_{1a}(\Gamma_{a1}v+\epsilon)<(I_n-\Gamma)v.\tag{A.11}$$

Naming $v_a=\Gamma_{a1}v+\epsilon>0$, we have that

$$\begin{cases}\Gamma_{1a}v_a+\Gamma v<v\\ \Gamma_{a1}v<v_a\end{cases},\tag{A.12}$$

which can be written as

$$\begin{bmatrix}\Gamma & \Gamma_{1a}\\ \Gamma_{a1} & 0\end{bmatrix}\begin{bmatrix}v\\ v_a\end{bmatrix}<\begin{bmatrix}v\\ v_a\end{bmatrix}.\tag{A.13}$$

In conclusion, there exists $\tilde{v}=[v^\top\ v_a^\top]^\top$ such that $\tilde{\Gamma}\tilde{v}<\tilde{v}$ holds, implying that $\rho(\tilde{\Gamma})<1$. According to Theorem 4.1, the new model $\tilde{\mathcal{M}}$ is δISS. □

A.7. Proof of Corollary 4.3

The proof straightforwardly derives from Corollary 4.1. After a plug-in operation, condition (14) needs to be verified only for the rows of $\tilde{\Gamma}$ that include parameters related to the newly added submodel $\mathcal{M}^{[a]}$, since the remaining rows already satisfy (14). In other words, this condition must be verified for all the ith rows, with $i\in\mathcal{O}_a$, which include the elements of $\tilde{\Gamma}$ corresponding to the outlet neighbors of the added submodel $\mathcal{M}^{[a]}$, and for the ath row, which contains the elements related to the inlet neighbors of $\mathcal{M}^{[a]}$. □

References

Alhajeri, M. S., Luo, J., Wu, Z., Albalawi, F., & Christofides, P. D. (2022). Process structure-based recurrent neural network modeling for predictive control: A comparative study. *Chemical Engineering Research and Design*, *179*, 77–89.

Angeli, D. (2002). A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, *47*(3), 410–421.

Angeli, D. (2009). Further results on incremental input-to-state stability. *IEEE Transactions on Automatic Control*, *54*(6), 1386–1391.

Armenio, L. B., Terzi, E., Farina, M., & Scattolini, R. (2019a). Echo state networks: analysis, training and predictive control. In *2019 European control conference* (pp. 799–804). IEEE.

Armenio, L. B., Terzi, E., Farina, M., & Scattolini, R. (2019b). Model predictive control design for dynamical systems learned by echo state networks. *IEEE Control Systems Letters*, *3*(4), 1044–1049.

Bayer, F., Bürger, M., & Allgöwer, F. (2013). Discrete-time incremental ISS: A framework for robust NMPC. In *2013 European control conference* (pp. 2068–2073). IEEE.

Berman, A., & Plemmons, R. J. (1994). *Nonnegative matrices in the mathematical sciences*. SIAM.

Boca de Giuli, L., La Bella, A., Farina, M., & Scattolini, R. (2024). Modeling and predictive control of networked systems via physics-informed neural networks. In *2024 IEEE conference on decision and control* (pp. 3005–3010). IEEE.

Boca de Giuli, L., La Bella, A., & Scattolini, R. (2024). Physics-informed neural network modeling and predictive control of district heating systems. *IEEE Transactions on Control Systems Technology*, *32*(4), 1182–1195.

Bof, N., Carli, R., & Schenato, L. (2018). Lyapunov theory for discrete time systems. arXiv preprint arXiv:1809.05289.

Bolderman, M., Butler, H., Koekebakker, S., van Horssen, E., Kamidi, R., Spaan-Burke, T., Strijbosch, N., & Lazar, M. (2024). Physics-guided neural networks for feedforward control with input-to-state-stability guarantees. *Control Engineering Practice*, *145*, Article 105851.

Bolderman, M., Lazar, M., & Butler, H. (2021). Physics–guided neural networks for inversion–based feedforward control applied to linear motors. In *2021 IEEE conference on control technology and applications* (pp. 1115–1120). IEEE.

Bonassi, F. (2023a). *Reconciling deep learning and control theory: recurrent neural networks for model-based control design*. (P.hD. thesis), URL https://www.politesi.polimi.it/handle/10589/196384.

Bonassi, F. (2023b). Ssnet: a python module for training state space neural NETworks. URL https://github.com/bonassifabio/ssnet.

Bonassi, F., Farina, M., & Scattolini, R. (2021a). On the stability properties of gated recurrent units neural networks. *Systems & Control Letters*, *157*, Article 105049.

Bonassi, F., Farina, M., & Scattolini, R. (2021b). Stability of discrete-time feed-forward neural networks in NARX configuration. *IFAC-PapersOnLine*, *54*(7), 547–552.

Bonassi, F., Farina, M., Xie, J., & Scattolini, R. (2022). On recurrent neural networks for learning-based control: recent results and ideas for future developments. *Journal of Process Control*, *114*, 92–104.
Bonassi, F., La Bella, A., Farina, M., & Scattolini, R. (2024). Nonlinear MPC design for incrementally ISS systems with application to GRU networks. *Automatica*, *159*, Article 111381.
Bonassi, F., La Bella, A., Panzani, G., Farina, M., & Scattolini, R. (2023). Deep long-short term memory networks: Stability properties and experimental validation. In *2023 European control conference* (pp. 1–6). IEEE.
D'Amico, W., La Bella, A., & Farina, M. (2023). An incremental input-to-state stability condition for a class of recurrent neural networks. *IEEE Transactions on Automatic Control*, *69*(4), 2221–2236.
Dashkovskiy, S., Rüffer, B. S., & Wirth, F. R. (2007). An ISS small gain theorem for general networks. *Mathematics of Control, Signals, and Systems*, *19*, 93–122.
Dashkovskiy, S. N., Rüffer, B. S., & Wirth, F. R. (2010). Small gain theorems for large scale systems and construction of ISS Lyapunov functions. *SIAM Journal on Control and Optimization*, *48*(6), 4089–4118.
Daw, A., Karpatne, A., Watkins, W. D., Read, J. S., & Kumar, V. (2022). Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge guided machine learning* (pp. 353–372). Chapman and Hall/CRC.
Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., & Karpatne, A. (2020). Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 SIAM international conference on data mining* (pp. 532–540). SIAM.
Drgoňa, J., Tuor, A. R., Chandan, V., & Vrabie, D. L. (2021). Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, *243*, Article 110992.
Enders, W. (2008). *Applied econometric time series*. John Wiley & Sons.
Farina, L., & Rinaldi, S. (2011). *Positive linear systems: theory and applications*. John Wiley & Sons.
Farina, M., & Scattolini, R. (2012). Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, *48*(6), 1088–1096.
James, M. R. (1994). On the certainty equivalence principle and the optimal control of partially observed dynamic games. *IEEE Transactions on Automatic Control*, *39*(11), 2321–2324.
Jiang, Z.-P., & Wang, Y. (2001). Input-to-state stability for discrete-time nonlinear systems. *Automatica*, *37*(6), 857–869.
Karpatne, A., Kannan, R., & Kumar, V. (2022). *Knowledge guided machine learning: Accelerating discovery using scientific knowledge and data*. CRC Press.
Kim, K.-D., & Kumar, P. R. (2012). Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1287–1308.
La Bella, A., & Del Corno, A. (2023). Optimal management and data-based predictive control of district heating systems: The Novate Milanese experimental case-study. *Control Engineering Practice*, *132*, Article 105429.
La Bella, A., Klaus, P., Ferrari-Trecate, G., & Scattolini, R. (2022). Supervised model predictive control of large-scale electricity networks via clustering methods. *Optimal Control Applications and Methods*, *43*(1), 44–64.
Lunze, J. (1992). *Feedback control of large-scale systems*. Prentice Hall New York.
Luyben, M. L., & Luyben, W. L. (1995). Design and control of a complex process involving two reaction steps, three distillation columns, and two recycle streams. *Industrial & Engineering Chemistry Research*, *34*(11), 3885–3898.
Maestre, J., Ridao, M., Kozma, A., Savorgnan, C., Diehl, M., Doan, M., Sadowska, A., Keviczky, T., De Schutter, B., Scheu, H., et al. (2015). A comparison of distributed MPC schemes on a hydro-power plant benchmark. *Optimal Control Applications and Methods*, *36*(3), 306–332.
Marino, A., Pacchierotti, C., & Giordano, P. R. (2024). Input state stability of gated graph neural networks. *IEEE Transactions on Control of Network Systems*, *11*(4), 2052–2063.
Masero, E., Bonassi, F., La Bella, A., & Scattolini, R. (2024). Estimation and MPC control based on gated recurrent unit neural networks with unknown disturbances. In *2024 IEEE conference on decision and control* (pp. 120–125). IEEE.
Nghiem, T. X., Drgoňa, J., Jones, C., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Di Cairano, S., Paulson, J. A., Carron, A., et al. (2023). Physics-informed machine learning for modeling and control of dynamical systems. In *2023 American control conference* (pp. 3735–3750). IEEE.
Nigro, L., La Bella, A., Casella, F., & Scattolini, R. (2025). Control-oriented modeling, simulation, and predictive control of district heating networks. *IEEE Transactions on Automation Science and Engineering*, *22*, 7064–7079.
Rawlings, J. B., Mayne, D. Q., & Diehl, M. (2017). *Model predictive control: theory, computation, and design*: *vol. 2*, Nob Hill Publishing Madison, WI.
Riverso, S., Farina, M., & Ferrari-Trecate, G. (2013). Plug-and-play decentralized model predictive control for linear systems. *IEEE Transactions on Automatic Control*, *58*(10), 2608–2614.
Rüffer, B. S. (2010). Monotone inequalities, dynamical systems, and paths in the positive orthant of Euclidean n-space. *Positivity*, *14*, 257–283.
Scattolini, R., Doan, M. D., Keviczky, T., & De Schutter, B. (2009). Hierarchical and distributed model predictive control of large-scale systems. D2.1: Report on Literature Survey and Preliminary Definition of the Selected Methods for the Definition of System Decomposition and Hierarchical Control Architectures. URL https://hdmpc.dsrepo.eu/deliverables/hd_mpc_D_2_1.pdf.
Tang, W., & Daoutidis, P. (2018). Network decomposition for distributed control through community detection in input–output bipartite graphs. *Journal of Process Control*, *64*, 7–14.
Terzi, E., Bonassi, F., Farina, M., & Scattolini, R. (2021). Learning model predictive control with long short-term memory networks. *International Journal of Robust and Nonlinear Control*, *31*(18), 8877–8896.
Terzi, E., Bonetti, T., Saccani, D., Farina, M., Fagiano, L., & Scattolini, R. (2020). Learning-based predictive control of the cooling system of a large business centre. *Control Engineering Practice*, *97*, Article 104348.
Varga, R. S. (2011). *Geršgorin and his circles*: *vol. 36*, Springer Science & Business Media.
Vidyasagar, M. (2002). *Nonlinear systems analysis*. SIAM.
Xie, J., Bonassi, F., & Scattolini, R. (2025). Learning control affine neural NARX models for internal model control design. *IEEE Transactions on Automation Science and Engineering*, *22*, 8137–8149.
Xu, H., Duan, Z., Wang, Y., Feng, J., Chen, R., Zhang, Q., & Xu, Z. (2021). Graph partitioning and graph neural network based hierarchical graph matching for graph similarity computation. *Neurocomputing*, *439*, 348–362.

Laura Boca de Giuli received the B.Sc. and M.Sc. degrees cum laude in Automation and Control Engineering from Politecnico di Milano in 2020 and 2023, respectively. Her Master's thesis received the "Si può fare di più" award from Fondazione Cogeme, the "Nicola Schiavoni" award from Politecnico di Milano, and the "Outstanding master thesis in control" award from the IEEE Italian Control System Society. In 2021, she spent a semester as exchange student at École Polytechnique Fédérale de Lausanne (Switzerland). Since June 2023, she has been a Ph.D. researcher in Systems and Control at Politecnico di Milano within the Information Technology programme. In 2025, she was a visiting researcher at the Institute for Dynamic Systems and Control, ETH Zürich (Switzerland). Her research interests focus on learning-based control of industrial and energy plants, with an emphasis on physics-informed neural networks, lifelong learning, and model predictive control.

Alessio La Bella received the M.Sc. in Automation Engineering cum laude from Politecnico di Milano in 2015 and the M.Sc. in Mechatronics Engineering cum laude from Politecnico di Torino in 2016. He received the Ph.D. Degree cum laude in Information Technology - Systems and Control at Politecnico di Milano in 2020. From 2020 to 2022, he worked as Research Engineer at Ricerca sul Sistema Energetico - RSE SpA, designing predictive control architectures for district heating networks and large-scale battery plants with industrial companies and energy utilities. In 2022, he joined Politecnico di Milano as Assistant Professor at Dipartimento di Elettronica, Informazione e Bioingegneria. He was visiting researcher at École Polytechnique Fédérale de Lausanne (Switzerland), in 2018, at Technische Universiteit Delft (The Netherlands) in 2024, and at Imperial College London (United Kingdom) in 2024 and 2025. He is Associate Editor for the International Journal of Adaptive Control and Signal Processing, for the EUCA Conference Editorial Board, and for the Technology Conference Editorial Board of the IEEE Control Systems Society. His research interests concern the theory and design of predictive, multi-agent and learning-based control systems, with particular emphasis on practical challenges arising from the upcoming energy transition. He was recipient of the Dimitris N. Chorafas Prize in 2020.

Marcello Farina received the M.Sc degree in Electronic Engineering in 2003 and the Ph.D. degree in Information Engineering in 2007, both from Politecnico di Milano. In 2005 he was visiting student at the Institute for Systems Theory and Automatic Control, Stuttgart, Germany. Presently, he is Associate Professor at Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. His theoretical research interests include model predictive control, distributed and decentralized state estimation and control, data-based methods for direct and indirect design of control systems, and the use recurrent neural network models in controller design. His applicative research includes the development of assistive autonomous devices.

Riccardo Scattolini was born in Milano, Italy, in 1956. He is Full Professor of Automatic Control at the Politecnico di Milano. During the academic year 1984/85 he was a visiting researcher at the Oxford University. He has also spent one year working in industry on the simulation and control of chemical plants. He was awarded the Quazza Premium and the Heaviside Premium of the Institution of Electrical Engineers, U.K. He has been Associate Editor of the IFAC journal Automatica and of the International Journal of Adaptive Control and Signal Processing. He is author of more than 130 papers published in the main international journals on control, and of more than 150 papers presented at the main international conferences. His main research interests include modeling, identification, simulation, diagnosis, and control of industrial plants and electrical systems, with emphasis on the theory and application of Model Predictive Control and fault detection methods.