Data-driven control of echo state-based recurrent neural networks with robust stability guarantees

William D'Amico[a], Alessio La Bella[a,*], Marcello Farina[a]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, Milan, 20133, Italy*

Abstract

In this work we propose a new data-based approach for robust controller design for a rather general class of recurrent neural networks affected by bounded measurement noise. We first present a new procedure to select the most suitable model class from input-output data via SM. Incremental input-to-state stability and desired performances for the closed loop system are enforced via a dedicated linear matrix inequality (LMI) optimization problem, exploiting on the virtual reference feedback tuning (VRFT) data-driven control design method. The numerical results show the effectiveness of both the proposed identification and control design approaches.

Keywords: Recurrent neural networks, linear matrix inequalities, data-based control.

1. Introduction

Neural networks (NNs) have recently gained great interest in many engineering fields, among which Automation and Control [1, 2]. This is due to the significant ability of NNs to reproduce nonlinear static or dynamical systems [3, 4, 5] as universal approximators and to the ever-growing amount of informative data that has paved the way to deep learning. The latter is a family of machine learning methods which relies on NNs to learn features

*Corresponding author

Email addresses: william.damico@polimi.it (William D'Amico), alessio.labella@polimi.it (Alessio La Bella), marcello.farina@polimi.it (Marcello Farina)

from data. The main classes of NNs used for control purposes are feedforward neural networks (FFNNs), to reproduce static functions, and recurrent neural networks (RNNs), characterized by the presence of internal loops representing state variables allowing them to replicate dynamical systems [6, 7]. The theoretical properties of RNNs have been marginally analysed in the literature. Sufficient conditions ensuring stability-related properties for RNNs are presented in [8] and in [9], the latter focusing on a specific class of RNNs, i.e., gated recurrent units. A small gain type stability condition, namely, finite gain l_2 stability, is provided in [10] for a class of RNNs with rectified linear unit (ReLU) activation functions. An extension of this condition, based on the integral quadratic constraint framework, is given in [11] for the same class of RNNs. Also, a global exponential stability condition is discussed in [12] for a class of RNNs, considering however the case with constant inputs. Other results, providing sufficient conditions guaranteeing contraction properties, are given in [13, 14] for echo-state networks (ESNs) and in [15, 16] for more general RNNs. Note also that the above-mentioned contributions address stability properties weaker than the incremental input-to-state stability (δISS) which, as shown, e.g., in [17, 18] can be directly enforced in the data-based RNN training phase. In particular, the δISS property plays a crucial role in RNN-based systems. It ensures that the state trajectories of a system, asymptotically, are independent on the initial conditions but they solely depend on the applied inputs [19]. As a consequence, the modelling performances of a δISS RNN are, asymptotically, independent of its initialization. Moreover, the δISS property implies also other stability properties, e.g., the global asymptotic stability (GAS) of the equilibria and the input-to-state stability (ISS) [19, 20]. It is worth noting that conditions for the δISS of open-loop RNN-based systems are established in [17, 18], where, however, the design of stabilizing RNN-based closed-loop controllers is not addressed. Regarding control systems, in [21, 22] the stability is analysed in case of FFNN controllers and assuming a linear controlled system with uncertainties. Design conditions for FFNN controllers are also provided in [23], considering specific classes of second-order nonlinear systems under control. Moreover, based on the recently introduced recurrent equilibrium networks [24, 25], some current works are devoted to learning nonlinear feedback policies with stability guarantees [26, 27]. Also, the input-to-state stability of an MPC-controlled neural nonlinear autoregressive exogenous system is discussed in [28]. Moreover, MPC regulation strategies for other RNN architectures are presented in [29], [30], and [31], ensuring closed-loop stability if the RNN-based model of the controlled system enjoys the δISS property. Finally, in [32], the reformulation of virtual reference feedback tuning (VRFT) [33] exploiting linear matrix inequalities (LMIs) is combined with LMI-based δISS constraints, derived in [34], to guarantee the asymptotic stability of the equilibria of the closed-loop system; in this case, the system to be controlled is assumed to belong to a class of RNNs and the controller is defined by an ESN [35].

RNNs are, by nature, employed in data-based and learning contexts. More specifically, in Automation and Control, they are commonly used for designing controllers applicable to dynamical systems in both *indirect* and in *direct* approaches. In a few words, indirect methods require a first identification of the system model based on which the controller is designed, whereas direct methods aim at directly identifying a controller lying in a previously-selected class. Both the latter approaches have advantages and disadvantages. For instance direct methods generally lack of stability guarantees for the closed-loop system. This has motivated the search of novel data-based design approaches, combining features of both direct and indirect ones. For instance, in [36], a unifying optimization problem is proposed in the linear setting, combining a direct LMI-based reformulation of a VRFT cost function with indirect LMI robust stability constraints based on a polytopic uncertainty set representation of the open-loop system, derived via set membership (SM) identification [37].

As far as RNN-based control schemes are concerned, despite the significant amount of works dedicated, in the current literature, to provide formal conditions for their stability, these conditions are not commonly used in comprehensive data-based design methodologies and procedures. In particular some challenges, which require to be addressed in a consistent fashion, have been so far disregarded in this context: (i) the selection of a suitable model class; (ii) the characterization of the model uncertainty resulting from data-based learning; (iii) the application of closed-loop stability conditions robustly with respect to model uncertainties; (iv) the development of data-based procedures that allow one to confer closed-loop performances. In particular, point (i) has not been considered extensively by the RNN-related literature; for instance, some guidelines for the selection procedure are given in [38] based on a trial-and-error procedure only for ESNs.

In this work, first, assuming the data to be affected by a bounded measurement noise, we propose a novel data-based SM approach for defining, at the same time, the most suitable model class (among a number of given model class candidates) and the uncertain system model consistent with the available data and with some prior knowledge, e.g., about the measurement noise bound. Interestingly, in this phase we will propose a novel model class selection criterion to be used in the place of the more widely used fitting index. Secondly, we propose a design approach, for ESN-based feedback controllers, that allows one to robustly confer δISS and desired performances to the closed loop system.
In this paper we restrict our attention to a rather general class of RNN systems that includes both ESNs and linear autoregressive terms, for this reason denoted nonlinear autoregressive with exogenous input echo state network (NARXESN). This choice allows us to define the open-loop system model uncertainty by means of a polytopic uncertainty set representation and to formulate the control design problem as an LMI-based one. Simulation examples validate the theoretical results presented in this work.

The paper is structured as follows. In Section 2 the main objectives of this work are stated. The novel method for the selection of the model class and the identification of the polytopic uncertainty set for the selected class of RNNs is introduced in Section 3, whereas Section 4 addresses the control design with robust δISS guarantees and VRFT-based performances. Section 5 shows the application of the methods in this paper to simulation examples and, finally, conclusions are drawn in Section 6. For clarity reasons, all proofs are postponed to the Appendix.

Notation
Given a matrix A, its transpose is $A^\top$, its inverse is A^{-1}, the transpose of its inverse is $A^{-\top}$, and its trace is $\mathrm{tr}(A)$. The entry in the ith row and jth column of a matrix A is denoted by a_{ij}, whereas the jth row of matrix A is indicated as $A_{(j,:)}$. The ith entry of a vector v is indicated by v_i. Given a symmetric matrix P, we use $P \succeq 0$, $P \succ 0$, $P \preceq 0$, and $P \prec 0$ to indicate that it is positive semidefinite, positive definite, negative semidefinite, and negative definite, respectively. We denote with $0_{n,m}$ (or 0) a zero matrix with n rows and m columns (or with a suitable number of rows and columns where clear from the context), I_n (or I) the identity matrix of dimension n (or with suitable dimensions where clear from the context) Given a sequence of square

matrices $A_1, A_2, \ldots, A_n$, $D = \mathrm{diag}(A_1, A_2, \ldots, A_n)$ is a block diagonal matrix having $A_1, A_2, \ldots, A_n$ as sub-matrices on the main-diagonal blocks. Moreover, $\|v\| = \sqrt{v^\top v}$ denotes the 2-norm of a column vector v. Given a vector $v \in \mathbb{R}^n$ and a convex closed set $\mathcal{O} \subset \mathbb{R}^n$, the point-to-set distance is defined as $\mathrm{dist}(v, \mathcal{O}) \coloneqq \min_{x \in \mathcal{O}} \|v - x\|^2$. Given a sequence $\vec{u} = u(0), u(1), \ldots$ or, equivalently, $\vec{u} = \{u(k)\}_{k=0}^{+\infty}$, we define its infinity norm as $\|\vec{u}\|_\infty = \sup_{k \in \mathbb{Z}_{\geq 0}} \|u(k)\|$. Also, $\mathrm{id}_n(\cdot)$ denotes a column vector of dimension n with all elements equal to the identity function $\mathrm{id}(\cdot)$ so that, for each $v \in \mathbb{R}^n$, $\mathrm{id}(v) = v$. The probability of an event E is denoted by $\mathcal{P}\{E\}$.

Preliminaries

Consider the following class of nonlinear discrete-time systems

$$x(k+1) = f(Ax(k) + Bu(k)) \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and $f(\cdot) = \begin{bmatrix} f_1(\cdot) & \ldots & f_n(\cdot) \end{bmatrix}^\top \in \mathbb{R}^n$ is a vector of scalar functions applied element-wise that satisfy the next assumption.

We firstly recall the notion of incremental input-to-state stability (δISS) next [19], which implies, e.g., the global asymptotic stability of the equilibria of (1).

Definition 1.1 (δISS)**.** *System* (1) *is* incrementally input-to-state stable *if there exists a function $\beta \in \mathcal{KL}$ and a function $\gamma \in \mathcal{K}_\infty$ [39] such that, for initial condition x_{01}, $x_{02} \in \mathbb{R}^n$ and input sequences $\vec{u}_1 = \{u_1(k)\}_{k=0}^{+\infty}$, $\vec{u}_2 = \{u_2(k)\}_{k=0}^{+\infty}$, the state trajectories $x(\cdot, x_{0i}, \vec{u}_i)$ resulting from* (1) *with initial condition x_{0i} and input $\vec{u}_i$, $i = 1, 2$, satisfy, for each $k \in \mathbb{Z}_{\geq 0}$,*

$$\begin{aligned} \|x(k, x_{01}, \vec{u}_1) - x(k, x_{02}, \vec{u}_2)\| \leq\ & \beta(\|x_{01} - x_{02}\|, k) + \\ +\ & \gamma(\|\vec{u}_1 - \vec{u}_2\|_\infty)\,. \end{aligned}$$

Now, let us introduce some concepts, derived from [34], which will be exploited to describe the approaches proposed in this paper.

Assumption 1.1. *The functions $f_i(\cdot)$, $i = 1, \ldots, n$, are globally Lipschitz continuous functions [40], each with Lipschitz constant L_{f_i}.*

Assumption 1.1 admits, as a special case, that some of the functions $f_i(\cdot)$, $i = 1, \ldots, n$, are the identity function $\mathrm{id}(\cdot)$ so we introduce the set

$$\mathcal{W}_f \coloneqq \{i \in \{1, \ldots, n\} \colon f_i(\cdot) \neq \mathrm{id}(\cdot)\}. \tag{2}$$

In view of Assumption 1.1, let $L_{f_i} := 1$ for each $i \notin \mathcal{W}_f$ and define $W := \text{diag}(L_{f_1}, \ldots, L_{f_n})$. For brevity and compactness, we introduce the following definition, thoroughly used in the following.

Definition 1.2 (**Property 1**). *Functions $f_i(\cdot)$, $i = 1, \ldots, n$, and matrix A enjoy **Property 1** with Lyapunov certificate P iff*
- Assumption 1.1 is verified by functions $f_i(\cdot)$, $i = 1, \ldots, n$, based on which the set $\mathcal{W}_f$ and the matrix W are derived as described;
- there exists $P = P^\top \succ 0$ where $p_{ij} = p_{ji} = 0$ for all $i \in \mathcal{W}_f$ and $j \in \{1, \ldots, n\}$, $j \neq i$, such that $\widetilde{A}^\top P \widetilde{A} - P \prec 0$, where $\widetilde{A} := WA$.

A sufficient condition for the δISS of (1) was provided in [34, Th. 2] and is recalled next.

Theorem 1.1 ([34]). *Let $f_i(\cdot)$, $i = 1, \ldots, n$, and matrix A enjoy **Property 1** with Lyapunov certificate P. Then, system (1) is δISS.*

2. Problem statement

We consider an unknown discrete-time system $\mathcal{S}$, displaying nonlinear dynamics, with input $u_{\mathcal{S}} \in \mathbb{R}^{m_{\mathcal{S}}}$, state $x_{\mathcal{S}} \in \mathbb{R}^{n_{\mathcal{S}}}$, output $y_{\mathcal{S}} \in \mathbb{R}^{l_{\mathcal{S}}}$, and measurement noise $w \in \mathbb{R}^{l_{\mathcal{S}}}$, defined as follows

$$\begin{cases} x_{\mathcal{S}}(k+1) &= f^o_{\theta^o}(x_{\mathcal{S}}(k), u_{\mathcal{S}}(k))\,, \\ y_{\mathcal{S}}(k) &= g^o_{\theta^o}(x_{\mathcal{S}}(k), u_{\mathcal{S}}(k)) + w(k)\,, \end{cases} \tag{3}$$

where $f^o_{\theta^o}$ and $g^o_{\theta^o}$ are generic function, whereas θ^o contains the unknown model parameters. The output measurement noise $w(k)$ is bounded, i.e., the following assumption holds.

Assumption 2.1. *Noise $w(k)$ is bounded, i.e., there exists a known $\bar{w}_j \in \mathbb{R}$ such that $|w_j(k)| \leq \bar{w}_j$, for all $k \in \mathbb{Z}$ and $j = 1, \ldots, l_{\mathcal{S}}$.* □

Assumption 2.1 is common in case of measurement noise affecting the output, and the noise bound can be evaluated, e.g., by using data or based on considerations about the sensor uncertainty.

The objective of this work is to derive a data-driven framework for robust control of system $\mathcal{S}$. In this work we assume that the following are available.

- A set of input-output data from system $\mathcal{S}$, i.e., $(u_{\mathcal{S}}(k), y_{\mathcal{S}}(k))$, for $k = -\bar{n}, \ldots, N_{\mathrm{t}} + N_{\mathrm{v}}$, where N_{t} is the number of training samples, N_{v} is the number of validation samples, and $\bar{n}$ is a parameter indicating the maximum considered input/output regressor order, as discussed later in (6). For the sake of compactness, we define sets containing the time indexes of the training and validation samples, i.e., $\mathcal{N}_{\mathrm{tr}} = \{-\bar{n}, \ldots, N_{\mathrm{t}}\}$ and $\mathcal{N}_{\mathrm{val}} = \{N_{\mathrm{t}}+1, \ldots, N_{\mathrm{t}}+N_{\mathrm{v}}\}$, respectively.

- A set of N_{class} candidate different *model classes* $\mathcal{M}_i$, with $i = 1, \ldots, N_{\mathrm{class}}$, defined as

$$\begin{cases} x^i_{\mathcal{S}}(k+1) &= f^i_{\theta^i}(x^i_{\mathcal{S}}(k), u_{\mathcal{S}}(k)) \\ y^i_{\mathcal{S}}(k) &= g^i_{\theta^i}(x^i_{\mathcal{S}}(k), u_{\mathcal{S}}(k)) + w(k) \end{cases} \tag{4}$$

where $x^i_{\mathcal{S}} \in \mathbb{R}^{n^i_{\mathcal{S}}}$, whereas θ^i contains generic identification parameters.
In this work we will assume that (3) and (4) lie in the class of echo-state based RNN systems (7) introduced in the next Section 3.

The first objective of this work is to define which *model class* $\mathcal{M}_i$, among the N_{class} available, is the most compliant with the available data and, at the same time, to identify, in the selected *model class*, the set of free parameters and the corresponding computed *optimal model set* which is compatible with the available data and noise bounds $\bar{w}_j$, $j = 1, \ldots, l_{\mathcal{S}}$.
In other words, here we first tackle the fundamental issue of model structure selection which, contrarily to the case of linear systems, has not been thoroughly addressed in case of RNNs. Indeed, the procedure aims to find $i^* \in \{1, \ldots, N_{\mathrm{class}}\}$ such that $\mathcal{S} \in \mathcal{M}_{i^*}$, i.e., the system belongs to a specific *model class*. However, this phase would in principle require that the selection is done with respect to an uncountably infinite set of *model classes*, while here, as it is customary, we are considering just a finite number N_{class} of them for tractability. Note that the number N_{class} of candidate model classes can be possibly defined based on scenario arguments, similarly to what is done in [41].

The second objective of this paper is to propose a novel control design approach which allows us to devise a controller for the system (3) such as to

- provide global asymptotic stability guarantees for the equilibria of the closed-loop system, robustly with respect to all models belonging to the *optimal model set*;
- achieve asymptotic tracking of constant reference signals;

- make the feedback control system as similar as possible to a given reference model of interest $\mathcal{M}_{\mathrm{CL}}$ to enforce the desired performances.

The proposed approach is inspired by [36], which however applies to linear systems, and combines the advantages of a direct approach (i.e., VRFT), used to define a cost function based on data for performance optimization, with robust stability constraints for the closed-loop requiring the *optimal model set* of the system, obtained in a data-based fashion, as typically done in indirect approaches.

3. Selection of the model class and identification of the optimal model set

3.1. The considered model classes

The proposed methodology is described considering N_{class} different candidate *model classes* $\mathcal{M}_i$ of the following type,

$$\chi^i(k+1) = \zeta^i(A^i_\chi \chi^i(k) + B^i_\phi \phi^i(k) + B^i_z z^i(k+1)), \tag{5a}$$

$$z^i(k+1) = H^i_1 \chi^i(k) + H^i_2 \phi^i(k)\,, \tag{5b}$$

$$y^i_{\mathcal{S}}(k) = z^i(k) + w(k)\,, \tag{5c}$$

where $i = 1, \ldots, N_{\text{class}}$, $z^i \in \mathbb{R}^{l_{\mathcal{S}}}$ is the noise-free output, the regressor

$$\begin{aligned} \phi^i(k) \;&\coloneqq [z^i(k)^\top \,\ldots\, z^i(k-n^i_z+1)^\top \, u_{\mathcal{S}}(k)^\top \,\ldots \\ &\quad u_{\mathcal{S}}(k-n^i_u+1)^\top]^\top \in \mathbb{R}^{n^i_\phi} \end{aligned} \tag{5d}$$

is the neural network input, having dimension $n^i_\phi = n^i_z l_{\mathcal{S}} + n^i_u m_{\mathcal{S}}$, $\chi^i \in \mathbb{R}^{n^i_\chi}$ is a vector of n^i_χ neurons, $\zeta^i(\cdot) \coloneqq \begin{bmatrix} \zeta^i_1(\cdot) & \ldots & \zeta^i_{n^i_\chi}(\cdot) \end{bmatrix}^\top \in \mathbb{R}^{n^i_\chi}$ is a vector of scalar functions applied element-wise. Also, let

$$\bar{n} \coloneqq \max\{n^1_z, n^1_u, \ldots, n^{N_{\text{class}}}_z, n^{N_{\text{class}}}_u\}\,. \tag{6}$$

The considered neural network models (5) are here denoted nonlinear autoregressive with exogenous input echo state networks (NARXESN), since they are a generalization of ESNs, in view of the presence of the linear autoregressive equation (5b). Consistently with the ESN training procedure [42], we consider the matrices $A^i_\chi \in \mathbb{R}^{n^i_\chi \times n^i_\chi}$, $B^i_\phi \in \mathbb{R}^{n^i_\chi \times n^i_\phi}$, $B^i_z \in \mathbb{R}^{n^i_\chi \times l_{\mathcal{S}}}$, the functions in $\zeta^i(\cdot)$ and the orders n_z^i, n_u^i, and n_χ^i as *hyperparameters*, previously selected for each *model class*. On the other hand, the matrices $H_1^i \in \mathbb{R}^{l_\mathcal{S} \times n_\chi^i}$ and $H_2^i \in \mathbb{R}^{l_\mathcal{S} \times n_\phi^i}$ are unknown free identification parameters. In Section 4, we will show that the considered model class (5a)-(5b) can be recast as (1). Moreover, besides showing remarkable modelling capabilities for nonlinear dynamics, this model class is extremely useful in our framework for a manifold reason. For instance, the learning procedure can leverage the linear-in-the-parameters structure of (5b); also, a simple observer, robust with respect to the unknown free parameter values, can be devised.

As indicated, the hyperparameters are user-generated. They must fulfill the following assumption.

Assumption 3.1. *For each $i = 1, \ldots, N_{\text{class}}$, the functions $\zeta_l^i(\cdot)$, $l = 1, \ldots, n_\chi^i$, and matrix A_χ^i enjoy* ***Property 1*** *with Lyapunov certificate P^i.* □

As stated next and consistently with Theorem 1.1, Assumption 3.1 allows us to guarantee that the first model equation in (5a) enjoys δISS [34] which, as discussed in [35, 42] and shown below, permits to provide a consistent estimation of the free parameters H_1^i and H_2^i regardless of the initialization of variable $\chi^i(k)$. Besides, as discussed in Section 4.1, this will also be fundamental for designing a suitable observer for the system state.

We assume that the dynamics of system $\mathcal{S}$ is compatible with a specific NARXESN model

$$\chi^o(k+1) = \zeta^o(A_\chi^o \chi^o(k) + B_\phi^o \phi^o(k) + B_z^o z(k+1)), \tag{7a}$$

$$z(k+1) = H_1^o \chi^o(k) + H_2^o \phi^o(k), \tag{7b}$$

$$y_\mathcal{S}(k) = z(k) + w(k), \tag{7c}$$

where

$$\begin{aligned}\phi^o(k) := \big[& z(k)^\top \ldots z(k - n_z^o + 1)^\top \\ & u_\mathcal{S}(k)^\top \ldots u_\mathcal{S}(k - n_u^o + 1)^\top \big]^\top \in \mathbb{R}^{n_\phi^o}.\end{aligned} \tag{7d}$$

where $n_\phi^o = n_z^o l_\mathcal{S} + n_u^o m_\mathcal{S}$.The functions in $\zeta^o(\cdot)$, the orders n_z^o, n_u^o, and n_χ^o (of the output and input regressors in vector ϕ^o and of the neurons vector χ^o, respectively), and the matrices $A_\chi^o \in \mathbb{R}^{n_\chi^o \times n_\chi^o}$, $B_\phi^o \in \mathbb{R}^{n_\chi^o \times n_\phi^0}$, $B_z^o \in \mathbb{R}^{n_\chi^o \times l_\mathcal{S}}$ are unknown. However, our assumption is that there exists an index $i^* \in \{1, \ldots, N_{\text{class}}\}$ such that $\zeta^o(\cdot) = \zeta^{i^*}(\cdot)$, $n_z^o = n_z^{i^*}$, $n_u^o = n_u^{i^*}$,

$n_\chi^o = n_\chi^{i^*}$, $A_\chi^o = A_\chi^{i^*}$, $B_\phi^o = B_\phi^{i^*}$, and $B_z^o = B_z^{i^*}$, i.e., such that $\mathcal{S} \in \mathcal{M}_{i^*}$. The procedure sketched in this section allows one, on the one hand, to define the index $i^* \in \{1, \ldots, N_{\text{class}}\}$ such that $\mathcal{S} \in \mathcal{M}_{i^*}$. Note that however, in practice, in view of the finite number N_{class} of available *model classes*, this step will unavoidably just lead to define the model class $\mathcal{M}_{i^*}$ which, according to a metrics defined later on, is closer to $\mathcal{S}$.

Also, since the available data are noisy, we will not estimate the "most likely" (in some sense) values of the free parameters $H_1^{i^*}$ and $H_2^{i^*}$ of the selected model class $\mathcal{M}_{i^*}$. Rather, using SM-based arguments, we will identify a set of values of the free parameters, i.e., the *feasible parameter set* (FPS) Θ_{i^*}, compatible with the available data and with the known bound on noise $w(k)$ such that $\begin{bmatrix} H_1^{i^*} & H_2^{i^*} \end{bmatrix} \in \Theta_{i^*}$. The set of models in class $\mathcal{M}_{i^*}$ satisfying the former inclusion will be denoted as *optimal model set*.

The general idea can be simply sketched as follows.

- Firstly, as described in Section 3.2, for each *model classes* $\mathcal{M}_i$, $i = 1, \ldots, N_{\text{class}}$ we perform the SM identification procedure leading to a candidate FPS Θ_i. This is the set of model parameters $H_\mathcal{S}^i \coloneqq \begin{bmatrix} H_1^i & H_2^i \end{bmatrix}$ compatible (according to the SM paradigm) with the available data, with the candidate *model class* $\mathcal{M}_i$, and with the known bound on the noise $w(k)$;

- Then, as described in Section 3.3, for each $i = 1, \ldots, N_{\text{class}}$ we compute, in a randomized way, the minimum distance $d_{\mathcal{S},i}^*$ between the outputs simulated by the models in the ith *optimal model set* and the tube of all possible output values, i.e., $y_{\mathcal{S},j}(k) \pm \bar{w}_j$ for all $j = 1, \ldots, l_\mathcal{S}$, and $\forall\, k \in \mathcal{N}_{\text{val}}$;

- Finally, we choose as best model class i^* the one corresponding with the minimum distance $d_{\mathcal{S},i}^*$, and the corresponding optimal model set is finally characterized by the FPS Θ_{i^*}.

3.2. Set membership model set identification

Consider the equations of the *model class* $\mathcal{M}_i$. First of all, we provide an estimate $\hat{\chi}^i(k)$ of variable $\chi^i(k)$ as

$$\hat{\chi}^i(k+1) = \zeta^i(A_\chi^i \hat{\chi}^i(k) + B_\phi^i \hat{\phi}^i(k) + B_z^i y_\mathcal{S}(k+1))\,, \tag{8}$$

where $\hat{\phi}^i(k) \coloneqq [y_\mathcal{S}(k)^\top \ldots y_\mathcal{S}(k - n_z^i + 1)^\top\, u_\mathcal{S}(k)^\top \ldots u_\mathcal{S}(k - n_u^i + 1)^\top]^\top \in \mathbb{R}^{n_\phi^i}$ is the available regressor measurement. Note that in (8), differently from (5), the output $y_\mathcal{S}$ is used in place of the state variable z, being the output measurable. In view of Assumptions 2.1 and 3.1, in the next proposition we obtain that, in case $\mathcal{S} \in \mathcal{M}_{i^*}$ for a certain $i^* \in \{1, \ldots, N_{\text{class}}\}$, the estimation error $\varepsilon_\chi(k) \coloneqq \chi^o(k) - \hat{\chi}^{i^*}(k)$ is bounded.

Proposition 3.1. *Let Assumption 2.1 hold. If functions $\zeta_j^i(\cdot)$, $j = 1, \ldots, n_\chi^i$ and matrix A_χ^i enjoy **Property 1** with Lyapunov certificate P^i then, if $\mathcal{S} \in \mathcal{M}_{i^*}$ where $i^* \in \{1, \ldots, N_{\text{class}}\}$, there exist a function $\beta \in \mathcal{KL}$ and a function $\gamma \in \mathcal{K}_\infty$ such that, for any $k \in \mathbb{Z}_{\geq 0}$ and any initial states $\chi^o(0), \hat{\chi}^{i^*}(0)$, it holds that $\|\varepsilon_\chi(k)\| \leq \bar{\varepsilon}_\chi(k)$, where*

$$\bar{\varepsilon}_\chi(k) \coloneqq \beta(\|\chi^o(0) - \hat{\chi}^{i^*}(0)\|, k) + \gamma\left(\sqrt{(n_z^{i^*} + 1)\sum_{j=1}^{l_{\mathcal{S}}} \bar{w}_j^2}\right).$$

The proof of Proposition 3.1 is reported in the Appendix. Proposition 3.1 allows us to define an adapted version of the SM algorithm [37, 36] for the estimation of the uncertainty set for the free parameters $H_{\mathcal{S}}^o \coloneqq \begin{bmatrix} H_1^o & H_2^o \end{bmatrix}$ of (7). In particular, note that (7b) is a linear equation, and that noisy estimates (but with bounded uncertainty) of the variables at both left and right hand sides are available. More specifically, we can rewrite (7b)-(7c) as

$$\begin{aligned} y_{\mathcal{S}}(k+1) = & H_1^o \hat{\chi}^{i^*}(k) + H_2^o \hat{\phi}^{i^*}(k) \\ & + \varepsilon(k) + w(k+1)\,, \end{aligned} \tag{9}$$

where $\varepsilon(k) = H_1^o \varepsilon_\chi(k) + H_2^o(\phi^o(k) - \hat{\phi}^{i^*}(k))$, which is bounded in view of Proposition 3.1 and Assumption 2.1, i.e., for $j = 1, \ldots, l_{\mathcal{S}}$, $|\varepsilon_j(k)| \leq \bar{\varepsilon}_j(k) \coloneqq \|H^o_{1\,(j,:)}\| \bar{\varepsilon}_\chi(k) + \|H^o_{2\,(j,:)}\| \sqrt{n_z^{i^*} \sum_{\mathrm{j}=1}^{l_{\mathcal{S}}} \bar{w}_{\mathrm{j}}^2}$. Therefore, denoting $\bar{\varepsilon}_j \coloneqq \bar{\varepsilon}_j(0)$, we have that $|\varepsilon_j(k)| \leq \bar{\varepsilon}_j$ for all $k \geq 0$. It is recalled that $H^o_{1\,(j,:)}$ and $H^o_{2\,(j,:)}$ are the jth rows of matrices H_1^o and H_2^o, respectively.

In view of this consideration, the SM theory can be readily applied, allowing us to compute Θ_i for each $i = 1, \ldots, N_{\text{class}}$: in Algorithm 1 the steps of the procedure for the estimation of the FPS of NARXESN models are shown.

Note that, to compensate for the uncertainty caused by the use of a finite number of measurements, the value $\underline{\lambda}_j^i$ is commonly inflated by a scalar $\alpha_j^i > 1$: the algorithm in [36] for the choice of α_j^i with probabilistic guarantees can be straightforwardly extended to the class of NARXESNs and to the multiple-input multiple-output (MIMO) case. This is not reported here for conciseness.

Algorithm 1 Estimation of Θ_i via SM

1. Collect an output/regressor dataset from the plant $(y_{\mathcal{S}}(k+1), \hat{\phi}^i(k))$, for each $k = 0, \ldots, N_{\mathrm{t}} - 1$.
2. Compute $\hat{\chi}^i(k)$, for each $k = 1, \ldots, N_{\mathrm{t}} - 1$, by using (8) with inputs $\hat{\phi}^i(k)$ and $y_{\mathcal{S}}(k+1)$, and by setting $\hat{\chi}^i(0) = 0$.
3. For all $j = 1, \ldots, l_{\mathcal{S}}$
   a. Solve the following optimization problem to obtain an estimate $\underline{\lambda}^i_j$ of the prediction error bound:
   $$\underline{\lambda}^i_j = \min_{H^i_{\mathcal{S},(j,:)} \in \Omega^i_j, \lambda \in \mathbb{R}_{\geq 0}} \lambda$$
   subject to, for all $k = K_0, \ldots, N_{\mathrm{t}} - 1$,
   $$|y_{\mathcal{S},j}(k+1) - H^i_{1\,(j,:)}\hat{\chi}^i(k) - H^i_{2\,(j,:)}\hat{\phi}^i(k)| \leq \lambda + \bar{w}_j,$$
   where $H^i_{\mathcal{S},(j,:)} := \begin{bmatrix} H^i_{1\,(j,:)} & H^i_{2\,(j,:)} \end{bmatrix}$, $H^i_{1\,(j,:)}$ and $H^i_{2\,(j,:)}$ are the jth row of H^i_1 and H^i_2, respectively, $y_{\mathcal{S},j}(k+1)$ is the jth element of $y_{\mathcal{S}}(k+1)$, K_0 is the "washout time", associated with the initial transient of the neural network, and $\Omega^i_j \subset \mathbb{R}^{1 \times n^i_\phi}$, $j = 1, \ldots, l_{\mathcal{S}}$, are arbitrarily large compact and convex polytopic sets.

   b. Compute the estimate of $\bar{\varepsilon}_j$, i.e.,
   $$\hat{\bar{\varepsilon}}^i_j = \alpha^i_j \underline{\lambda}^i_j, \tag{10}$$
   where $\alpha^i_j > 1$ is a design parameter as previously discussed.
4. Define the feasible parameter set (FPS) Θ_i, i.e., the tightest set of parameter values consistent with all the prior information and the available data, as
$$\begin{aligned} \Theta_i \;\; &:= \{H^i_{\mathcal{S}} \in \bar{\Omega}^i : |y_{\mathcal{S},j}(k+1) - H^i_{1\,(j,:)}\hat{\chi}^i(k) \\ &\quad -H^i_{2\,(j,:)}\hat{\phi}^i(k)| \leq \hat{\bar{\varepsilon}}^i_j + \bar{w}_j \, \forall k = K_0, \ldots, N_{\mathrm{t}} - 1, \\ &\quad \forall j = 1, \ldots, l_{\mathcal{S}}\}, \end{aligned} \tag{11}$$
where $\bar{\Omega}^i = \Omega^i_1 \times \ldots \times \Omega^i_{l_{\mathcal{S}}}$.

Finally, it is worth noting that the polytopic nature of the FPS Θ_i, as evident from (11), makes it possible to define any element $H_{\mathcal{S}}^i \in \Theta_i$ as a convex combination of the vertices of Θ_i, as discussed in [36]. Hence, assuming to know the n_V^i vertexes $H_{\mathcal{S}}^{i\,V_\nu}$, with $\nu = 1, \ldots, n_V^i$, of Θ_i, there exist $\sigma_1^i \geq 0$, $\ldots$, $\sigma_{n_V}^i \geq 0$ where $\sigma_1^i + \cdots + \sigma_{n_V^i}^i = 1$ such that

$$H_{\mathcal{S}}^i = \sum_{\nu=1}^{n_V^i} \sigma_\nu^i H_{\mathcal{S}}^{i\,V_\nu} \,. \tag{12}$$

3.3. Selection of the model class

The common criterion used for model class selection consists of chosing the model hyperparameters providing the smallest normalized root mean squared error (NRMSE) over the validation dataset $\mathcal{N}_{\text{val}}$, i.e.,

$$\text{NRMSE} = \frac{1}{\sigma_{y_{\mathcal{S}}}} \sqrt{\frac{\sum_{\forall k \in \mathcal{N}_{\text{val}}} (y_{\mathcal{S}}(k) - \hat{z}(k))^2}{N_{\text{v}}}},$$

where $y_{\mathcal{S}}$ are the actual system output data, $\hat{z}$ are the output data simulated by the identified model, and $\sigma_{y_{\mathcal{S}}}$ is the standard deviation of the validation output data $y_{\mathcal{S}}$. However, this approach may be numerically inaccurate and considers only one single model identified from data for each selection of hyperparameters. Also, the identified model may be far from the actual system due to the bias introduced by noise.

The alternative approach proposed in this paper consists of considering, for each model class $\mathcal{M}_i$, $i = 1, \ldots, N_{\text{class}}$, all models such that $H_{\mathcal{S}}^i \in \Theta_i$ (i.e., in the *optimal model set* for the class $\mathcal{M}_i$) to check if at least one of them allows us to obtain simulated output trajectories which are close to (or, ideally, inside of) the tube defined by the output components $y_{\mathcal{S},j}(k) \pm \bar{w}_j$, $k \in \mathcal{N}_{\text{val}}$, for all $j = 1, \ldots, l_{\mathcal{S}}$. For the sake of tractability, we draw a number M of scenarios $H_{\mathcal{S}}^{i,l} := \begin{bmatrix} H_1^{i,l} & H_2^{i,l} \end{bmatrix} \in \Theta_i$, for $l = 1, \ldots, M$, taken from the FPS. For each scenario $l = 1, \ldots, M$, the following system is simulated

$$\chi^{i,l}(k+1) = \zeta^i(A_\chi^i \chi^{i,l}(k) + B_\phi^i \phi^{i,l}(k) + B_z^i z^{i,l}(k+1)), \tag{13a}$$

$$z^{i,l}(k+1) = H_1^{i,l} \chi^{i,l}(k) + H_2^{i,l} \phi^{i,l}(k)\,, \tag{13b}$$

where $\phi^{i,l}(k) := [z^{i,l}(k)^\top \ldots z^{i,l}(k - n_z^i + 1)^\top u_{\mathcal{S}}(k)^\top \ldots u_{\mathcal{S}}(k - n_u^i + 1)^\top]^\top$. The minimum distance of the simulated output of the scenario l from the output data is computed as $d_{\mathcal{S}}^{i,l} := \sum_{\forall k \in \mathcal{N}_{\text{val}}} \tilde{d}_{\mathcal{S}}^{i,l}(k)$, where $\tilde{d}_{\mathcal{S}}^{i,l}(k) := \text{dist}(z^{i,l}(k),\, \mathcal{Y}(k))$ and

$$\mathcal{Y}(k) := \{\tilde{y}_{\mathcal{S}}(k) \in \mathbb{R}^{l_{\mathcal{S}}} : |\tilde{y}_{\mathcal{S},j}(k) - y_{\mathcal{S},j}(k)| \leq \bar{w}_j \ \forall j = 1, \ldots, l_{\mathcal{S}}\}.$$

The distance of the *model class* $\mathcal{M}_i$ with respect to the data is

$$d_{\mathcal{S}}^{i,*} := \min_{l=1,\ldots,M} d_{\mathcal{S}}^{i,l}\,. \tag{14}$$

Eventually, the most suitable model class $\mathcal{M}_{i^*}$ to be selected among the N_{class} considered is the one that provides the minimum distance $d_{\mathcal{S}}^{i,*}$.
The idea is grounded on the remark that, in case $\mathcal{S} \in \mathcal{M}_{i^*}$ for $i^* \in \{1, \ldots, N_{\text{class}}\}$, and $H_{\mathcal{S}}^{i^*,l^*} = H_{\mathcal{S}}^{o}$ for a scenario $l^* \in \{1, \ldots, M\}$, then $d_{\mathcal{S}}^{i^*,*} = 0$ (under the same choice of the initial conditions for (7) and (13)). However, we expect that $d_{\mathcal{S}}^{i,*} > 0$, even in case $\mathcal{S} \in \mathcal{M}_{i^*}$, since the probability that, for some $l = 1, \ldots, M$, $H_{\mathcal{S}}^{i^*,l} = H_{\mathcal{S}}^{o}$ is equal to zero considering that Θ_{i^*} has infinite cardinality. In Proposition 3.2 we provide, via scenario-based arguments [43], a minimum number M of scenarios guaranteeing that, for each $i = 1, \ldots, N_{\text{class}}$, the probability to extract a new scenario l^{new} leading to a distance $d_{\mathcal{S}}^{i,l^{\text{new}}} < d_{\mathcal{S}}^{i,*}$ is lower than a certain threshold. In the proposition, we use the notation $d_{\mathcal{S}}^{i}(H_{\mathcal{S}}^{i})$ to denote the distance $d_{\mathcal{S}}^{i,l}$ obtained with $H_{\mathcal{S}}^{i,l} = H_{\mathcal{S}}^{i}$, where $H_{\mathcal{S}}^{i}$ takes the role of a generic random matrix for class $\mathcal{M}_i$ with probability distribution $\mathbb{P}_{H_{\mathcal{S}}^i}$ over Θ_i. In this regard, some guidelines for the selection of $\mathbb{P}_{H_{\mathcal{S}}^i}$ are given next.

Proposition 3.2. *Let $H_{\mathcal{S}}^{i} \in \Theta_i$ be a random matrix with probability distribution $\mathbb{P}_{H_{\mathcal{S}}^i}$ over Θ_i. Let $\epsilon \in (0,1)$ and $\beta \in (0,1)$ be user-defined constants. For all $M \geq 1$ fulfilling*

$$M \geq \log_{1-\epsilon}(\beta)\,, \tag{15}$$

if $d_{\mathcal{S}}^{i,*}$ is computed as in (14), then, with probability at least $1-\beta$, it holds that $\mathcal{P}\{H_{\mathcal{S}}^{i} \in \Theta_i \,:\, d_{\mathcal{S}}^{i}(H_{\mathcal{S}}^{i}) < d_{\mathcal{S}}^{i,*}\} \leq \epsilon$. □

The proof of Proposition 3.2 is reported in the Appendix. The distribution $\mathbb{P}_{H_{\mathcal{S}}^i}$ over Θ_i can have different interpretations [43]. For example, it can be a descriptor of the relative importance given to the various uncertainty outcomes in Θ_i. Our suggestion is to make a similar choice for $\mathbb{P}_{H_{\mathcal{S}}^i}$ to the one commonly adopted in case of linear systems in absence of any a-priori information [44]. For instance, in the SISO case, $\mathbb{P}_{H_S^i}$ is equal to the normal distribution with mean vector $\hat{H}_S^i$ and covariance matrix $\Sigma_{H_S^i}$, where $\hat{H}_S^i$ are the parameters estimated via least squares (LS), whereas $\Sigma_{H_S^i}$ is the corresponding covariance matrix. In the MIMO case, one may proceed by defining such a distribution for each row of the parameter matrix H_S^i. In Algorithm 2 the overall procedure is summarized.

Algorithm 2 Selection of the *model class* $\mathcal{M}_{i^*}$

1. Define N_{class} candidate NARXESN model classes (5).
2. Collect an input/output dataset $(u_S(k), y_S(k))$, for $k \in \mathcal{N}_{\text{tr}} \cup \mathcal{N}_{\text{val}}$.
3. Choose a violation probability $\epsilon \in (0, 1)$ and a confidence parameter $\beta \in (0, 1)$.
4. For all $i = 1, \ldots, N_{\text{class}}$
   a. Define an output/regressor dataset, $(y_S(k+1), \hat{\phi}^i(k))$, for $k = 0, \ldots, N_{\text{t}} - 1$.
   b. Define the FPS Θ_i in (11) following Algorithm 1, by using the first $N_{\text{t}} - K_0$ training data.
   c. Define a probability distribution $\mathbb{P}_{H_S^i}$ over Θ_i (e.g., by using the first $N_{\text{t}} - K_0$ training data).
   d. Find the minimum integer $M \geq 1$ such that (15) holds.
   e. Generate a sample of scenarios $(H_S^{i,1}, H_S^{i,2}, \ldots, H_S^{i,M})$ of M independent random elements from $(\Theta_i, \mathbb{P}_{H_S^i})$.
   f. For each scenario $l = 1, \ldots, M$, simulate system (13), by using the whole dataset.
   g. Compute $d_S^{i,*}$ as in (14).
5. Select the model class $\mathcal{M}_{i^*}$ in such a way that

$$i^* = \underset{i \in \{1, \ldots, N_{\text{class}}\}}{\arg\min} \; d_S^{i,*} \tag{16}$$

4. Robust control design

In this section we address the control design problem. At this stage, we assume that the *model class* selection and the identification of the *optimal model set* are completed as discussed in Section 3, i.e., we have selected i^* and identified the feasible parameter set Θ_{i^*} in (11) such that $\mathcal{S} \in \mathcal{M}_{i^*}$ and that $H_{\mathcal{S}}^o \in \Theta_{i^*}$.

In this section, we start by providing a state-space representation of the system model (7) in Section 4.1. Secondly, we provide the overall equations of the control scheme in Section 4.2. Subsequently, Section 4.3 presents the condition for the robust δISS of the closed-loop, whereas Section 4.4 recalls the optimization problem for performance based on VRFT presented in [32], as well as a summarizing algorithm.

4.1. State-space representation of the system $\mathcal{S}$

To derive a state-space representation of (7), for $n_{\mathcal{S}} := n_\chi^o + n_z^o \cdot l_{\mathcal{S}} + (n_u^o - 1) \cdot m_{\mathcal{S}}$, consider $x_{\mathcal{S}}(k) := [\chi^o(k)^\top \; z(k)^\top \; \dots \; z(k-n_z^o+1)^\top \; u_{\mathcal{S}}(k-1)^\top \; \dots \; u_{\mathcal{S}}(k-n_u^o+1)^\top]^\top \in \mathbb{R}^{n_{\mathcal{S}}}$, $u_{\mathcal{S}}(k)$ and $y_{\mathcal{S}}(k)$ as the state, input, and output of (7), respectively. If we define matrices $B_{\phi,1}^o$, $B_{\phi,2}^o$, $B_{\phi,3}^o$, $H_{2,1}^o$, $H_{2,2}^o$, $H_{2,3}^o$ in such a way that $B_\phi^o \phi^o(k) = \begin{bmatrix} 0 & B_{\phi,1}^o & B_{\phi,3}^o \end{bmatrix} x_{\mathcal{S}}(k) + B_{\phi,2}^o u_{\mathcal{S}}(k)$ and $H_2^o \phi^o(k) = \begin{bmatrix} 0 & H_{2,1}^o & H_{2,3}^o \end{bmatrix} x_{\mathcal{S}}(k) + H_{2,2}^o u_{\mathcal{S}}(k)$, system (7) can be compactly rewritten as

$$x_{\mathcal{S}}(k+1) = f_{\mathcal{S}}^o(A_{\mathcal{S}}^o x_{\mathcal{S}}(k) + B_{\mathcal{S}}^o u_{\mathcal{S}}(k)) \,, \tag{17a}$$

$$y_{\mathcal{S}}(k) = C_{\mathcal{S}} x_{\mathcal{S}}(k) + w(k) \,, \tag{17b}$$

where $f_{\mathcal{S}}^o(\cdot) := \begin{bmatrix} \zeta^o(\cdot)^\top & \mathrm{id}(\cdot)^\top \end{bmatrix}^\top$,

$$A_{\mathcal{S}}^o := \begin{bmatrix} A_\chi^o + B_z^o H_1^o & B_{\phi,1}^o + B_z^o H_{2,1}^o & & B_{\phi,3}^o + B_z^o H_{2,3}^o & \\ H_1^o & H_{2,1}^o & & H_{2,3}^o & \\ 0 & I_{(n_z^o-1)\cdot l_{\mathcal{S}}} & 0 & 0 & \\ 0 & 0 & & 0 & \\ 0 & 0 & & I_{(n_u^o-2)\cdot m_{\mathcal{S}}} & 0 \end{bmatrix},$$

$$B_{\mathcal{S}}^o := \begin{bmatrix} B_{\phi,2}^o + B_z^o H_{2,2}^o \\ H_{2,2}^o \\ 0 \\ I_{m_{\mathcal{S}}} \\ 0 \end{bmatrix},$$

and $C_{\mathcal{S}} := \begin{bmatrix} 0_{l_{\mathcal{S}} \times n_{\chi}^o} & I_{l_{\mathcal{S}}} & 0 \end{bmatrix}$. As clear from Section 3, model (17) is partially unknown but, as a result of *model class* selection and *optimal model set* identification, the following can be stated.

- We assume to know i^* such that $\mathcal{S} \in \mathcal{M}_{i^*}$, meaning that $\zeta^o = \zeta^{i^*}$, $n_u^o = n_u^{i^*}$, $n_z^o = n_z^{i^*}$, $n_\chi^o = n_\chi^{i^*}$, $A_\chi^o = A_\chi^{i^*}$, $B_\phi^o = B_\phi^{i^*}$, and $B_z^o = B_z^{i^*}$. In view of this and in light of Proposition 3.1, we can use $\hat{\chi}^{i^*}(k)$, generated from (8) with $i = i^*$ as a reliable estimate of $\chi^o(k)$, and we can use $y_{\mathcal{S}}(k)$ as a reliable estimate of $z(k)$. This, in turn, permits to compute the estimate $\hat{x}_{\mathcal{S}}(k) \in \mathbb{R}^{n_{\mathcal{S}}}$ of $x_{\mathcal{S}}(k)$ as $\hat{x}_{\mathcal{S}}(k) := [\hat{\chi}^{i^*}(k)^\top \, y_{\mathcal{S}}(k)^\top \, \ldots \, y_{\mathcal{S}}(k - n_z^{i^*} + 1)^\top \, u_{\mathcal{S}}(k-1)^\top \, \ldots \, u_{\mathcal{S}}(k - n_u^{i^*} + 1)^\top]^\top$ with guaranteed bounded estimation error $\varepsilon_{\mathcal{S}}(k)$. Importantly, note that the computation of $\hat{x}_{\mathcal{S}}(k)$ can be done robustly for any value of the parameter $H_{\mathcal{S}}^o$, which is not known. Then, for all k,

$$\hat{x}_{\mathcal{S}}(k) = x_{\mathcal{S}}(k) + \varepsilon_{\mathcal{S}}(k)\,, \tag{18}$$

where we know a bound $\bar{\varepsilon}_{\mathcal{S}}$ on $\varepsilon_{\mathcal{S}}(k)$ by Proposition 3.1. Indeed, for all $k \in \mathbb{Z}_{\geq 0}$, it holds that $\|\hat{\chi}^{i^*}(k) - \chi^o(k)\| \leq \bar{\varepsilon}_\chi(0) = \beta(\|\hat{\chi}^{i^*}(0) - \chi^o(0)\|, 0) + \gamma\left(\sqrt{(n_z^{i^*} + 1)\sum_{j=1}^{l_{\mathcal{S}}} \bar{w}_j^2}\right)$. Under Assumption 2.1, we also know that $\|y_{\mathcal{S}}(k - h) - z(k-h)\| = \|w(k-h)\| \leq \bar{w} := \sqrt{\sum_{j=1}^{l_{\mathcal{S}}} \bar{w}_j^2}$ for all $h = 0, \ldots, n_z^{i^*} - 1$. Hence, for all $k \in \mathbb{Z}_{\geq 0}$, $\|\varepsilon_{\mathcal{S}}(k)\| \leq \bar{\varepsilon}_{\mathcal{S}} = \sqrt{n_z^{i^*} \cdot \bar{w}^2 + \bar{\varepsilon}_\chi^2(0)}$.

- Although $H_{\mathcal{S}}^o$ is unknown, a feasible parameter set Θ_{i^*} has been identified through SM such that $H_{\mathcal{S}}^o \in \Theta_{i^*}$. This implies that, although $A_{\mathcal{S}}^o$ and $B_{\mathcal{S}}^o$ in (17) are not available, as they depend on the unknown $H_{\mathcal{S}}^o$, through the knowledge of Θ_{i^*} we can compute a bounded set where $A_{\mathcal{S}}^o$ and $B_{\mathcal{S}}^o$ lie. In the following we will stabilize (17) with respect to all parameterizations of $A_{\mathcal{S}}^o$ and $B_{\mathcal{S}}^o$ compatible with the available data.

4.2. Control scheme

The proposed control scheme is depicted in Figure 1, where block $\int$ is an integral action with equation

$$\eta(k+1) = \eta(k) + e(k) \tag{19a}$$

$$v(k) = \eta(k) + e(k) \tag{19b}$$

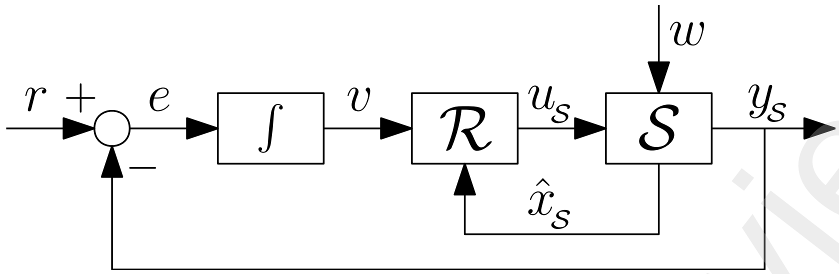

Figure 1: Closed-loop control scheme with explicit integral action and ESN controller: $\int$ is the discrete-time integrator, $\mathcal{R}$ is the ESN controller, and $\mathcal{S}$ is the system to be controlled.

where $e(k) \coloneqq r(k) - y_{\mathcal{S}}(k)$. For the sake of simplicity, the block $\mathcal{R}$ is selected as an ESN and has equation

$$\psi(k+1) = \xi(A_\psi \psi(k) + B_v v(k) + B_{x_{\mathcal{S}}} \hat{x}_{\mathcal{S}}(k) + B_{u_{\mathcal{S}}} u_{\mathcal{S}}(k)), \tag{20a}$$

$$u_{\mathcal{S}}(k) = H_1^\psi \psi(k) + H_2^v v(k) + H_2^x \hat{x}_{\mathcal{S}}(k), \tag{20b}$$

where $\psi \in \mathbb{R}^{n_\psi}$ is the ESN state, v and $\hat{x}_{\mathcal{S}}$ are the inputs to the network, the matrices in (20a) are fixed hyperparameters and the matrix $H_{\mathcal{R}} \coloneqq \begin{bmatrix} H_1^\psi & H_2^v & H_2^x \end{bmatrix} \in \mathbb{R}^{m_{\mathcal{S}} \times (n_\psi + l_{\mathcal{S}} + n_{\mathcal{S}})}$ in (20b) is the tunable controller parameter. Note that the NARXESN class in (5) is an extension of the ESN; this class could be also adopted for the controller at the price of additional design problem complexity. The hyperparameters of the regulator equations must be selected in order to fulfill the following assumption.

Assumption 4.1. *The functions $\xi_i(\cdot)$, $i = 1, \dots, n_\psi$ and matrix A_ψ enjoy* ***Property 1*** *with Lyapunov certificate $P_{\mathcal{R}}$.* $\square$

Assumption 4.1, in view of Theorem 1.1, allows us to guarantee the δISS and, in turn, the echo state property of (20a) with respect to the inputs v, $\hat{x}_{\mathcal{S}}$, and $u_{\mathcal{S}}$ (cf. [34]). This is useful to have a consistent estimation of the regulator parameters independently of the initialization of ψ.

4.3. Robust closed-loop δISS guarantees

In this section we provide a condition to guarantee robust closed-loop δISS guarantees for all the systems belonging to the FPS, under the assumption that $H_{\mathcal{S}}^o \in \Theta_{i^*}$. Before doing so, we derive the state equations of the closed loop by combining the equations of the system $\mathcal{S}$ in (17), the integrator in (19) and the ESN controller in (20). Let us define the overall control system state as $x \coloneqq [x_{\mathcal{S}}^\top \, \eta^\top \, \psi^\top]^\top \in \mathbb{R}^n$ and input $u \coloneqq [r^\top \, w^\top \, \varepsilon_{\mathcal{S}}^\top]^\top \in \mathbb{R}^{l_{\mathcal{S}}+l_{\mathcal{S}}+n_{\mathcal{S}}}$, where $n \coloneqq n_{\mathcal{S}} + l_{\mathcal{S}} + n_\psi$. In view of (17), (18), (19), and (20), we can write

$$x(k+1) = f(A^o x(k) + B^o u(k))\,, \tag{21a}$$

where $f(\cdot) = [f_{\mathcal{S}}^o(\cdot)^\top \;\, \mathrm{id}_{l_{\mathcal{S}}}(\cdot)^\top \;\, \xi(\cdot)^\top]^\top$ and

$$A^o = F^o + G^o J, \quad J = H_{\mathcal{R}} E, \tag{21b}$$

$$F^o \coloneqq \begin{bmatrix} A_{\mathcal{S}}^o & 0 & 0 \\ -C_{\mathcal{S}} & I_{l_{\mathcal{S}}} & 0 \\ B_{x_{\mathcal{S}}} - B_v C_{\mathcal{S}} & B_v & B_\psi \end{bmatrix}, \; G^o \coloneqq \begin{bmatrix} B_{\mathcal{S}}^o \\ 0 \\ B_{u_{\mathcal{S}}} \end{bmatrix}, \tag{21c}$$

$$E \coloneqq \begin{bmatrix} 0 & 0 & I_{n_\psi} \\ -C_{\mathcal{S}} & I_{l_{\mathcal{S}}} & 0 \\ I_{n_{\mathcal{S}}} & 0 & 0 \end{bmatrix}, \tag{21d}$$

Note that the definition of matrix B^o is not provided for conciseness since it is not used in the following. In view of Assumptions 3.1 and 4.1, functions $\zeta_j(\cdot)$, $j = 1, \ldots, n_\chi^i$ (for all i, including i^*) and $\xi_j(\cdot)$, $j = 1, \ldots, n_\psi$ are Lipschitz continuous and we can define

$$\mathcal{W}_f \coloneqq \{i \in \{1, \ldots, n\} \colon f_i(\cdot) \neq \mathrm{id}(\cdot)\}\,.$$

Let L_i^f be the Lipschitz constants of the functions $f_i(\cdot)$, $i = 1, \ldots, n$, with $L_i^f \coloneqq 1$ if $i \notin \mathcal{W}_f$. We define

$$W \coloneqq \mathrm{diag}(L_1^f, \ldots, L_n^f) \in \mathbb{R}^{n \times n}. \tag{22}$$

Note that, at this point, we know the n_V vertices $H_{\mathcal{S}}^{V_1}, \ldots, H_{\mathcal{S}}^{V_{n_V}}$ [45] of the convex and compact polytopic FPS Θ_{i^*}. Then, any element $H_{\mathcal{S}} \in \Theta_{i^*}$ can be written as a convex combination of these vertices, namely, there exist $\sigma_1 \geq 0$, $\ldots$, $\sigma_{n_V} \geq 0$ where $\sigma_1 + \cdots + \sigma_{n_V} = 1$ such that

$$H_{\mathcal{S}} = \sum_{\nu=1}^{n_V} \sigma_\nu H_{\mathcal{S}}^{V_\nu}\,. \tag{23}$$

For each vertex $\nu = 1, \ldots, n_V$, we can define from (21c)

$$F_\nu \coloneqq \begin{bmatrix} A_{\mathcal{S}}^{V_\nu} & 0 & 0 \\ -C_{\mathcal{S}} & I_{l_{\mathcal{S}}} & 0 \\ B_{x_{\mathcal{S}}} - B_v C_{\mathcal{S}} & B_v & B_\psi \end{bmatrix}, \; G_\nu \coloneqq \begin{bmatrix} B_{\mathcal{S}}^{V_\nu} \\ 0 \\ B_{u_{\mathcal{S}}} \end{bmatrix}, \tag{24}$$

where the matrices $A_{\mathcal{S}}^{V_\nu}$ and $B_{\mathcal{S}}^{V_\nu}$ are defined as $A_{\mathcal{S}}^o$ and $B_{\mathcal{S}}^o$, but with $H_{\mathcal{S}}^o = \begin{bmatrix} H_1^o & H_2^o \end{bmatrix}$ replaced by $H_{\mathcal{S}}^{V_\nu}$. For $\nu = 1, \dots, n_V$, we also define

$$\widetilde{F}_\nu \coloneqq W F_\nu, \quad \widetilde{G}_\nu \coloneqq W G_\nu \tag{25}$$

where W is defined in (22).
To make the systems in (21a) δISS with respect to input u for all $H_{\mathcal{S}} \in \Theta_{i^*}$, we state the next theorem.

Theorem 4.1. *Let $\mathcal{S} \in \mathcal{M}_{i^*}$, $H_{\mathcal{S}}^o \in \Theta_{i^*}$, and Assumptions 3.1 and 4.1 hold. Suppose that there exist $P_1 \in \mathbb{R}^{n \times n}, \dots, P_{n_V} \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{m_{\mathcal{S}} \times n}$, $Q \in \mathbb{R}^{n \times n}$ such that, for each $\nu = 1, \dots, n_V$, (i) $P_\nu = (P_\nu)^\top$, (ii) $(p_\nu)_{ij} = (p_\nu)_{ji} = 0$ for all $i \in \mathcal{W}_f$, $j \in \{1, \dots, n\}$, $j \neq i$, and (iii) for $\widetilde{F}_\nu$ and $\widetilde{G}_\nu$ defined in* (25)*,*

$$\begin{bmatrix} P_\nu & \widetilde{F}_\nu Q + \widetilde{G}_\nu L \\ (\widetilde{F}_\nu Q + \widetilde{G}_\nu L)^\top & Q + Q^\top - P_\nu \end{bmatrix} \succ 0\,. \tag{26}$$

Then, by setting

$$H_{\mathcal{R}} = L Q^{-1} E^{-1}\,, \tag{27}$$

the closed loop in (21) *is δISS with respect to u.* □

The proof of Theorem 4.1 is reported in the Appendix.

4.4. Performance

First of all, note that the static performance for tracking piecewise constant reference signals is guaranteed by δISS and by the explicit integral action included in the control scheme in Figure 1.
Dynamic performance will be enforced by relying on VRFT-based arguments [33]. Namely, we now propose an optimization problem to make the closed-loop system response similar to the one of a given reference model of interest $\mathcal{M}_{\mathrm{CL}}$. To obtain a cost function based on VRFT, we consider the collected dataset (cf. Section 2) and we perform the following steps.

1. Compute the *virtual reference* $r^*(k) = \mathcal{M}_{\mathrm{CL}}^{-1} y_{\mathcal{S}}(k)$, where $\mathcal{M}_{\mathrm{CL}}$ is an invertible stable (possibly non-causal) reference model with unitary input-output delay.
2. Compute the *virtual error* $e^*(k) = r^*(k) - y_{\mathcal{S}}(k)$.

3. Compute the *integrated virtual error* according to the recursive equation $v^*(k) = v^*(k-1) + e^*(k)$, with null initial condition for v^*.

4. Simulate the trajectory of $\hat{\chi}^{i^*}(k+1)$ by means of (8) with inputs $\hat{\phi}^{i^*}(k)$ and $y_{\mathcal{S}}(k+1)$, and null initial condition for $\hat{\chi}^{i^*}$. This allows us to compute the estimate $\hat{x}_{\mathcal{S}}(k)$.

5. Compute the evolution of $\psi(k)$ according to equation (20a) with inputs $v^*(k)$ (in place of $v(k)$), $\hat{x}_{\mathcal{S}}(k)$, $u_{\mathcal{S}}(k)$, and null initial condition for ψ. We denote this the virtual evolution of $\psi(k)$, later denoted as $\psi^*(k)$.

Now, we need to identify the ESN regulator model whose inputs are $v^*(k)$ and $\hat{x}_{\mathcal{S}}(k)$, and whose output is $u_{\mathcal{S}}(k)$. Since the regulator is an ESN, this boils down to a LS minimization problem, where the unknown is $H_{\mathcal{R}}$, as discussed in [46]. The VRFT problem consists of the minimization of

$$J_{VR}(H_{\mathcal{R}}) \coloneqq \frac{1}{N-K_0} \sum_{k=K_0}^{N-1} \left\| u_{\mathcal{S}}(k) - H_{\mathcal{R}} \begin{bmatrix} \psi^*(k) \\ v^*(k) \\ \hat{x}_{\mathcal{S}}(k) \end{bmatrix} \right\|^2, \tag{28}$$

where an initial instant K_0 is considered to discard the initial transient of the N-length trajectory. The previous cost function is data-based and does not depend on the system (unknown) parameters. However, since the δISS condition in (26) is defined in the free variables L, Q, and P_ν, $\nu = 1, \dots, n_V$, then we need to reformulate also the cost function (28) in the new optimazion variables to obtain a unifying LMI problem. To do this, define, for compactness, $\mathbf{U} \coloneqq \begin{bmatrix} u_{\mathcal{S}}(K_0) & \dots & u_{\mathcal{S}}(N-1) \end{bmatrix}^\top$, and

$$\mathbf{X} \coloneqq \begin{bmatrix} \psi^*(K_0)^\top & v^*(K_0)^\top & \hat{x}_{\mathcal{S}}(K_0)^\top \\ \vdots & \vdots & \vdots \\ \psi^*(N-1)^\top & v^*(N-1)^\top & \hat{x}_{\mathcal{S}}(N-1)^\top \end{bmatrix},$$

under the following assumption.

Assumption 4.2. *Matrix* $\mathbf{X}^\top\mathbf{X}$ *is invertible.*

Note that Assumption 4.2 is closely related to the identifiability properties of the system and to the persistency of excitation of the available data, see

the discussion provided about the Assumption 2 in [47], for a slightly different case. According to Assumption 4.2, we can also define $\widetilde{\mathbf{X}} := E^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ where E is defined as in (21d).
We can minimize (28) according to the following result.

Proposition 4.1. *Let Assumption 4.2 hold. The optimization problem*

$$\underset{L\in\mathbb{R}^{m_{\mathcal{S}}\times n},\Phi=\Phi^\top\in\mathbb{R}^{m_{\mathcal{S}}\times m_{\mathcal{S}}}}{\text{minimize}} \quad \mathrm{tr}(\Phi), \tag{29a}$$

subject to

$$\begin{bmatrix} \Phi - \mathbf{U}^\top\widetilde{\mathbf{X}}^\top Q\widetilde{\mathbf{X}}\mathbf{U} + L\widetilde{\mathbf{X}}\mathbf{U} + \mathbf{U}^\top\widetilde{\mathbf{X}}^\top L^\top & L \\ L^\top & Q \end{bmatrix} \succeq 0 \tag{29b}$$

is equivalent to minimize (28) *if, for any scalar* $\gamma_q > 0$*,*

$$Q = \gamma_q E^{-1}\mathbf{X}^\top\mathbf{X}E^{-\top}, \tag{30}$$

and (27) *holds.* □

The proof of Proposition 4.1 is reported in the Appendix. Finally, if the δISS constraint (26) is included in the optimization problem (29a)-(29b), setting Q as in (30) may lead to an infeasible problem or conservative results. Hence, condition (30) can be relaxed by defining the matrix Q as a free optimization variable and by replacing (30) with

$$Q - \gamma_q E^{-1}\mathbf{X}^\top\mathbf{X}E^{-\top} + \lambda_q I_n \succeq 0 \tag{31a}$$

$$-Q + \gamma_q E^{-1}\mathbf{X}^\top\mathbf{X}E^{-\top} + \lambda_q I_n \succeq 0 \tag{31b}$$

The scalar γ_q can be fixed by the user or, alternatively, left free as an optimization variable.
Based on the previous results and considerations, we are now in a position to summarize the overall control design procedure in Algorithm 3.

Algorithm 3 Data-based control design with δISS guarantees (polytopic uncertainty set)

1. Collect an input-output dataset from the plant $\mathcal{S}$, i.e., $(u_{\mathcal{S}}(k), y_{\mathcal{S}}(k))$, for $k = -\bar{n}, \ldots, N_{\mathrm{t}} + N_{\mathrm{v}}$.

2. Select the model class $\mathcal{M}_{i^*}$ and the FPS Θ_{i^*} in (11) by using Algorithms 1 and 2.

3. Compute the n_V vertices $H_{\mathcal{S}}^{V_1}, \ldots, H_{\mathcal{S}}^{V_{n_V}}$ of the FPS Θ_{i^*} in (11).

4. Define the matrices $W, E, \mathbf{U}, \mathbf{X}, \widetilde{\mathbf{X}}$, and, for all $\nu = 1, \ldots, n_V$, $F_\nu, G_\nu, \widetilde{F}_\nu, \widetilde{G}_\nu$.

5. Solve the following LMI optimization problem

$$\underset{\substack{Q=Q^\top, P_1=P_1^\top, \ldots, P_{n_V}=P_{n_V}^\top,\\ L, \Phi=\Phi^\top, \lambda_q \geq 0, \gamma_q > 0}}{\text{minimize}} \quad \mathrm{tr}(\Phi) + c\lambda_q \tag{32}$$

subject to (29b), (31) and, for all $\nu = 1, ..., n_V$, to (26), where $(p_\nu)_{ij} = (p_\nu)_{ji} = 0$ for all $i \in \mathcal{W}_f,\ j \in \{1, \ldots, n\},\ j \neq i$.

6. If the problem is feasible, then compute $H_{\mathcal{R}}$ according to (27).

5. Simulation example

In this section, to show the effectiveness of the theory and the algorithms described in the paper, we consider a numerical case study, where $\mathcal{S}$ is in a class of NARXESN. The system to be controlled is a nonlinear SISO dynamical system (i.e., $l_{\mathcal{S}} = m_{\mathcal{S}} = 1$) defined by a NARXESN as in (7) with $n_\chi^o = 3$ neurons, $n_u^o = n_z^o = 1$ input-output regressors, where, given the randomly generated matrices and the activation functions

$$\zeta^o(\cdot) = \begin{bmatrix} \tanh(\cdot) \\ \tanh(\cdot) \\ \mathrm{id}(\cdot) \end{bmatrix}, A^o_\chi = \begin{bmatrix} 0.2366 & 0.5763 & -0.8566 \\ 0 & 0 & 1.2831 \\ 1.2397 & 0 & 1.1902 \end{bmatrix}$$

$$B^o_\phi = \begin{bmatrix} -0.6261 & -0.1072 \\ -0.9963 & 0.8285 \\ -0.9612 & 0.7307 \end{bmatrix}, \; B^o_z = \begin{bmatrix} -0.3858 \\ 0.5741 \\ 0.9683 \end{bmatrix},$$

the corresponding nominal model unknown parameters are $H^o_1 = \begin{bmatrix} 0.3621 & 0.3397 & 0.5584 \end{bmatrix}$, and $H^o_2 = \begin{bmatrix} 0.309 & 0.8199 \end{bmatrix}$. For the random generation of the matrices A^o_χ and B^o_z, we set the spectral radius $\rho^o_\chi = 0.45$ and the feedback scaling $k^o_z = 1$, respectively.[1] Moreover, Assumption 3.1 is fulfilled by the definition of $\zeta^o(\cdot)$ and since $\|A^o_\chi\| = 0.9457 < 1$, see [46].

A noisy input-output dataset is collected from the system model with a sampling time $T_s = 1\ s$, where the input data consist of a multilevel pseudo-random signal (MPRS), whose amplitude is in the range $[-1.82, 1.54]$, whereas the measurement noise $w(k)$ is uniformly distributed, drawn from the range $[-\bar{w}, \bar{w}]$. Three different noise bound levels are considered, i.e., $\bar{w} = 0.0019$, $\bar{w} = 0.08$, and $\bar{w} = 0.4$, leading to the signal-to-noise ratios (SNRs) 66.34 dB, 33.85 dB, and 19.87 dB, respectively[2]. In Figure 2 the input-output data collected from the system with $\bar{w} = 0.0019$ are depicted. It is possible to notice that the nonlinear system has an oscillating output response to several constant inputs.

The proposed Algorithm 2 is validated for the three levels of noise with reference to the considered numerical example. For the application of the algorithm, $N_\mathrm{t} = 420$ and $N_\mathrm{v} = 180$, i.e., the 70% of the collected data are used to define the FPS and $\mathbb{P}_{H^i_\mathcal{S}}$, whereas the remaining 30% for the computation

[1]The random matrices are generated as follows: $A^o_\chi \in \mathbb{R}^{n^o_\chi \times n^o_\chi}$ is a sparse uniformly distributed random matrix with prescribed spectral radius ρ^o_χ and with density of nonzero elements equal to $\min\{1, 10/n^o_\chi\}$ (cf. the code provided in https://www.ai.rug.nl/minds/research/esnresearch/); $B^o_z := \widetilde{B}^o_z \,\mathrm{diag}(k^o_{z1}, \ldots, k^o_{zl_\mathcal{S}}) \in \mathbb{R}^{n^o_\chi \times l_\mathcal{S}}$ and $B^o_\phi \in \mathbb{R}^{n^o_\chi \times (n^o_z \cdot l_\mathcal{S} + n^o_u \cdot m_\mathcal{S})}$ are selected where $\widetilde{B}^o_z$ and B^o_ϕ are full uniformly distributed random matrices with elements in $[-1, 1]$, and $k^o_z := [k^o_{z1}, \ldots, k^o_{zl_\mathcal{S}}]^\top \in \mathbb{R}^{l_\mathcal{S}}$ is a feedback scaling vector.

[2]The SNR is computed as $\mathrm{SNR}_{dB} = 10\log_{10}(\sigma_z^2/\sigma_w^2)$, where σ_z and σ_w are the standard deviations of the noise-free output data z and of the measurement noise data w, respectively.

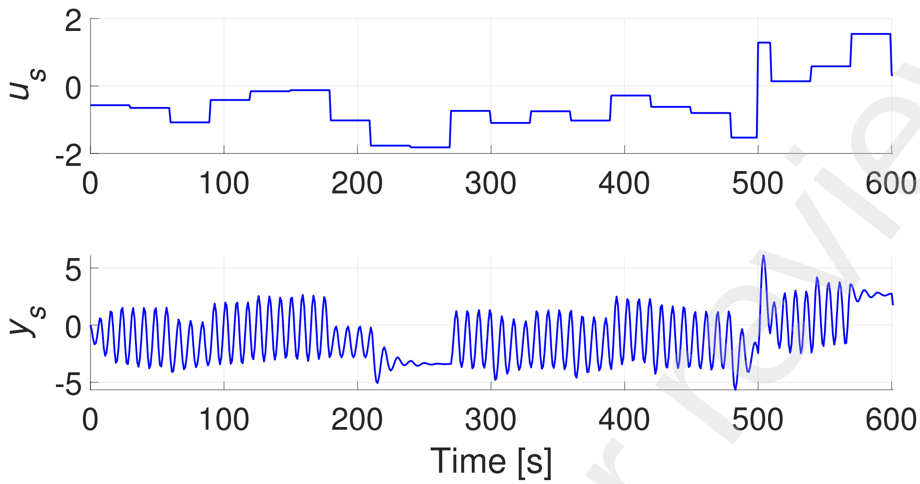

Figure 2: Input-output data collected from the open-loop system with $\bar{w} = 0.0019$. Upper plot: input $u_{\mathcal{S}}$ of the system $\mathcal{S}$; bottom plot: measured output $y_{\mathcal{S}}$ of the system $\mathcal{S}$.

of $d_{\mathcal{S}}^{i,*}$. Also, for $i = 1, \dots, N_{\text{class}}$, the compact and convex polytopic sets Ω_1^i are chosen as hypercubes defined by the ∞-norm $\|H_{\mathcal{S}}^i\|_\infty \leq 10^{10}$, whereas the inflation parameters α_1^i are always selected as the minimum values such that the parameter $\hat{H}_{\mathcal{S}}^i$, identified via LS by using the training data, belong to the FPS. A washout time $K_0 = 86$ is considered. Finally, we set the confidence parameter $\beta = 10^{-10}$ and the violation parameter $\epsilon = 0.05$, leading to a number $M = 449$ of scenarios.

In Table 1 the NRMSE of the identified models (computed over the validation dataset) and the minimum distance $d_{\mathcal{S}}^{i,*}$ are reported for different *model classes*, each one having different hyperparameters. In particular, the rows corresponding to the hyperparameters in $\mathcal{S}$ are in bold in the table. Moreover, for simplicity, $\zeta^i(\cdot)$ is modified by only varying the number of tanh functions, i.e., the first n_{tanh}^i activation functions are equal to $\tanh(\cdot)$, whereas the remaining ones are equal to $\mathrm{id}(\cdot)$. We can observe that the lowest $d_{\mathcal{S}}^{i,*}$ values are obtained in case $\mathcal{S} \in \mathcal{M}_i$ for all the noise levels, whereas the lowest NRMSEs are achieved by the actual hyperparameters only for low or medium noise levels, i.e., for $\bar{w} = 0.0019$ and $\bar{w} = 0.08$. On the other hand, for $\bar{w} = 0.4$, the selection of the *model class* based on NRMSE would lead to a wrong choice of the hyperparameters, corroborating the validity of Algorithm 2, especially in case of high noise levels.

Table 1: Selection of model hyperparameters (rows in bold correspond to the performances achieved when the hyperparameters of $\mathcal{S}$ are selected)

$\bar{w}$	n_χ^i	$n_{\tanh}^i$	n_u^i	n_z^i	ρ_χ^i	k_z^i	NRMSE	$d_{\mathcal{S}}^{i,*}$
0.0019	**3**	**2**	**1**	**1**	**0.45**	**1**	$\mathbf{5 \cdot 10^{-4}}$	$\mathbf{2.4 \cdot 10^{-8}}$
0.0019	30	20	1	1	0.45	1	0.2	9.27
0.0019	3	2	2	2	0.45	1	0.39	131.4
0.0019	3	2	1	1	0.35	1	0.09	5.44
0.0019	3	2	1	1	0.45	0.01	0.51	251.47
0.0019	15	5	2	2	0.5	1	0.29	20.92
0.08	**3**	**2**	**1**	**1**	**0.45**	**1**	**0.02**	$\mathbf{7.6 \cdot 10^{-5}}$
0.08	30	20	1	1	0.45	1	0.11	2.31
0.08	3	2	2	2	0.45	1	0.4	117.83
0.08	3	2	1	1	0.35	1	0.09	2.79
0.08	3	2	1	1	0.45	0.01	0.51	217.05
0.08	15	5	2	2	0.5	1	0.25	14.66
0.4	**3**	**2**	**1**	**1**	**0.45**	**1**	**0.24**	**0.02**
0.4	30	20	1	1	0.45	1	0.18	0.51
0.4	3	2	2	2	0.45	1	0.65	231.17
0.4	3	2	1	1	0.35	1	0.44	1.12
0.4	3	2	1	1	0.45	0.01	0.39	48.93
0.4	15	5	2	2	0.5	1	0.16	2.42

As a second step, we use the numerical example (with the actual hyperparameters) to validate Algorithm 3 for control design. While, in case of small and medium noise levels (i.e., $\bar{w} = 0.0019, 0.08$) Algorithm 3 provides a feasible solution, for $\bar{w} = 0.4$, the algorithm does not admit a feasible solution in view of its conservativeness. For all the noise levels, the desired reference model $\mathcal{M}_{\mathrm{CL}}$ is selected as a first-order asymptotically stable and unitary-gain LTI system with equation $y_{\mathcal{S}}(k) = -ay_{\mathcal{S}}(k-1) + (1+a)r(k-1)$, whose settling time is $10T_s = 10\ s$ under the choice $a = -0.6$.
In case $\bar{w} = 0.08$, we have that $\underline{\lambda}_1^{i^*} = 0.063$, $\alpha_1^{i^*} = 1.21$, and the unknown $H_{\mathcal{S}}^o \in \Theta_{i^*}$, where Θ_{i^*} has 134 vertices. The considered controller is the ESN (20)[3] with $n_\psi = 5$ neurons, where $\xi_j(\cdot) = \tanh(\cdot)$ for all $j = 1, \ldots, 5$, W_ψ is generated with spectral radius $\rho_\psi = 0.45$, and $\|W_\psi\| = 0.9622 < 1$, thus fulfilling Assumption 4.1 (see [46]). Moreover, $k_v = 0.0016$, and $k_{u_{\mathcal{S}}} = 1$. Assumption 4.2 holds, $W = I_{10}$ since all the Lipschitz constants are unitary, and we set $c = 10^{-3}$ in the LMI optimization problem (32). A feasible solution is obtained by solving the LMI optimization problem (32) with YALMIP [48] and MOSEK [49].

In Figure 3 the reference tracking results of the closed-loop control system are represented in case of application of Algorithm 3, where 10 different random initial conditions $\chi^o(0)$ are considered for the system neurons. We can see that the output trajectories converge one to each other according to the δISS property. Moreover, the desired model reference $\mathcal{M}_{\mathrm{CL}}$ is tracked with a fitting index $FIT(\%) = 88.84\%$ after discarding the initial 25 s, (cf. [36, Equation (40)] for the adopted definition of $FIT(\%)$).

[3] The random matrices in (20a) are generated as follows: $A_\psi \in \mathbb{R}^{n_\psi \times n_\psi}$ is a sparse uniformly distributed random matrix with prescribed spectral radius ρ_ψ and with density of nonzero elements equal to $\min\{1, 10/n_\psi\}$; $B_v := \widetilde{B}_v \operatorname{diag}(k_{v1}, \ldots, k_{vl_{\mathcal{S}}}) \in \mathbb{R}^{n_\psi, l_{\mathcal{S}}}$, $B_{u_{\mathcal{S}}} := \widetilde{B}_{u_{\mathcal{S}}} \operatorname{diag}(k_{u_{\mathcal{S}}1}, \ldots, k_{u_{\mathcal{S}}m_{\mathcal{S}}}) \in \mathbb{R}^{n_\psi \times m_{\mathcal{S}}}$, $B_{x_{\mathcal{S}}} \in \mathbb{R}^{n_\psi \times n_{\mathcal{S}}}$ are selected with $\widetilde{B}_v$, $\widetilde{B}_{u_{\mathcal{S}}}$, $B_{x_{\mathcal{S}}}$ being full uniformly distributed random matrices with elements in $[-1, 1]$ and $k_v := [k_{v1}, \ldots, k_{vl_{\mathcal{S}}}]^\top \in \mathbb{R}^{l_{\mathcal{S}}}$, $k_{u_{\mathcal{S}}} := [k_{u_{\mathcal{S}}1}, \ldots, k_{u_{\mathcal{S}}m_{\mathcal{S}}}]^\top \in \mathbb{R}^{m_{\mathcal{S}}}$ being user-defined feedback scaling vectors to modulate the effect of the integrator output v and of $u_{\mathcal{S}}$ in (20a), respectively.

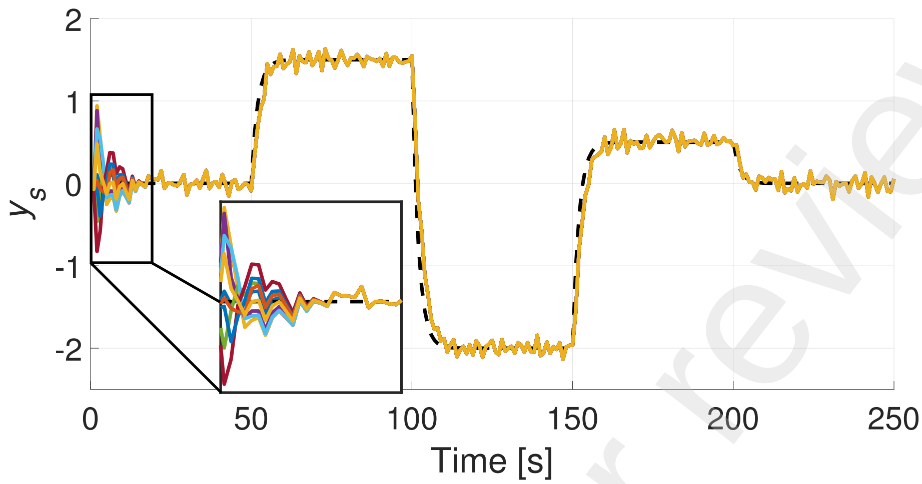

Figure 3: Output trajectories of the closed-loop system, with $\bar{w} = 0.08$, starting from different initial conditions, with zoom of the initial transient. Black dashed line: reference model output trajectory; colored lines: output trajectories for 10 different initial conditions $\chi^o(0)$.

6. Conclusions

In this work, we have proposed a new data-based approach to control unknown NARXESN systems affected by a bounded measurement noise, by means of an ESN feedback controller endowed with an explicit integral action and a state feedback term. Our algorithm allows us to robustly guarantee δISS and to enforce desired performances for the closed loop system via unifying LMI optimization problems based on VRFT. As a prior step, a novel model class selection phase based on scenario arguments and the definition of a polytopic uncertainty set representation of the open-loop system via SM identification must be carried out. Simulation examples have allowed us to validate the theory.

Future research works can take several directions. Since this work relies on the scenario approach, requiring the definition of probability distributions over a set, a further follow-up could be an in-depth investigation about the robustness of the scenario-based results with respect to mismatches on the assumed probability distributions due to possible inaccuracies in its definition. Secondly we will explore the possibility of using a less numerically complex characterization (e.g., ellipsoidal) of the uncertainty set. Finally, since the use of ESN-based systems entails the generation of random matrices for the neurons equations, a further extension of the proposed algorithms could be the possible multiple generation of the random matrices in order to choose the most suitable generation, e.g., via scenario-based arguments following the same lines of [41].

Acknowledgements

The authors thank Andrea Bisoffi for insightful and useful discussions.

Appendix

Proof of Proposition 3.1. First of all notice that, if we consider (7a) only, terms $\phi^o(k)$ and $z(k+1)$ can be regarded as exogenous terms. For simplicity, if we write $\mathbf{u}^o(k) = \begin{bmatrix}\phi^o(k)^\top & z(k+1)^\top\end{bmatrix}^\top$, we can rewrite (7a) as $\chi^o(k+1) = \zeta^o(A^o_\chi \chi^o(k) + \begin{bmatrix}B^o_\phi & B^o_z\end{bmatrix}\mathbf{u}^o(k))$, while, similarly, since i^* is such that $\mathcal{S} \in \mathcal{M}_{i^*}$ by assumption, we can write $\hat{\chi}^{i^*}(k+1) = \zeta^o(A^o_\chi \hat{\chi}^{i^*}(k) + \begin{bmatrix}B^o_\phi & B^o_z\end{bmatrix}\hat{\mathbf{u}}^{i^*}(k))$, where $\hat{\mathbf{u}}^{i^*}(k) := \begin{bmatrix}\hat{\phi}^{i^*}(k)^\top & y_{\mathcal{S}}(k+1)^\top\end{bmatrix}^\top$. Under Assumption 3.1, we can resort to Theorem 1.1 and prove that system (7a) is δISS with respect to the input $\mathbf{u}^o$ (and $\hat{\mathbf{u}}^{i^*}$). This implies that there exists a function $\beta \in \mathcal{KL}$ and a function $\gamma \in \mathcal{K}_\infty$ such that for any $k \in \mathbb{Z}_{\geq 0}$ and any initial states $\chi^o(0), \hat{\chi}^{i^*}(0)$, it holds that

$$\begin{aligned}\|\varepsilon_\chi(k)\| = \|\chi^o(k) - \hat{\chi}^{i^*}(k)\| &\leq \beta(\|\chi^o(0) - \hat{\chi}^{i^*}(0)\|, k) \\ &\quad + \gamma(\|\vec{\mathbf{u}}^o - \vec{\hat{\mathbf{u}}}^{i^*}\|_\infty)\,,\end{aligned} \tag{33}$$

where $\vec{\mathbf{u}}^o = \{\mathbf{u}^o(k)\}_{k=0}^{+\infty}$, and $\vec{\hat{\mathbf{u}}}^{i^*} = \{\hat{\mathbf{u}}^{i^*}(k)\}_{k=0}^{+\infty}$.

Note that $\mathbf{u}^o(k) - \hat{\mathbf{u}}^{i^*}(k) = \begin{bmatrix}\phi^o(k) - \hat{\phi}^{i^*}(k) \\ z(k+1) - y_{\mathcal{S}}(k+1)\end{bmatrix}$ for any $k \in \mathbb{Z}_{\geq 0}$, and that $z(k+1) - y_{\mathcal{S}}(k+1) = -w(k+1)$ and $\phi^o(k) - \hat{\phi}^{i^*}(k) =$

$$\begin{bmatrix} z(k) - y_{\mathcal{S}}(k) \\ \vdots \\ z(k-n^o_z+1) - y_{\mathcal{S}}(k-n^o_z+1) \\ u_{\mathcal{S}}(k) - u_{\mathcal{S}}(k) \\ \vdots \\ u_{\mathcal{S}}(k-n^o_u+1) - u_{\mathcal{S}}(k-n^o_u+1) \end{bmatrix} = - \begin{bmatrix} w(k) \\ \vdots \\ w(k-n^o_z+1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $n_z^o = n_z^{i^*}$ and $n_u^o = n_u^{i^*}$, since i^* is such that $\mathcal{S} \in \mathcal{M}_{i^*}$ by assumption. Hence, for all $k \in \mathbb{Z}_{\geq 0}$, in view of Assumption 2.1, $\|\mathbf{u}^o(k) - \hat{\mathbf{u}}^{i^*}(k)\| \leq \sqrt{(n_z^{i^*}+1)\sum_{j=1}^{l_\mathcal{S}} \bar{w}_j^2}$. Also, by the definition of infinity norm of a sequence and $\mathcal{K}_\infty$ function, we have that $\| \vec{\mathbf{u}}^o - \vec{\hat{\mathbf{u}}}^{i^*}\|_\infty \leq \sqrt{(n_z^{i^*}+1)\sum_{j=1}^{l_\mathcal{S}} \bar{w}_j^2}$ and $\gamma(\| \vec{\mathbf{u}}^o - \vec{\hat{\mathbf{u}}}^{i^*}\|_\infty) \leq \gamma\left(\sqrt{(n_z^{i^*}+1)\sum_{j=1}^{l_\mathcal{S}} \bar{w}_j^2}\right)$. From (33), it holds that $\|\varepsilon_\chi(k)\| \leq \beta(\|\chi^o(0) - \hat{\chi}^{i^*}(0)\|, k) + \gamma\left(\sqrt{(n_z^{i^*}+1)\sum_{j=1}^{l_\mathcal{S}} \bar{w}_j^2}\right)$. □

Proof of Proposition 3.2. We consider a sample $(H_\mathcal{S}^{i,1}, H_\mathcal{S}^{i,2}, \dots, H_\mathcal{S}^{i,M})$ of M independent random elements from $(\Theta_i, \mathbb{P}_{H_\mathcal{S}^i})$, where M fulfils (15). Without loss of generality, we consider D as a sufficiently large real number such that $d_\mathcal{S}^{i,l} < D$ for all $l = 1, \dots, M$. Then, we define the interval $\mathbb{M} \coloneqq [0, D]$. Let us consider the scenario program

$$\begin{aligned} &\underset{\mu \in \mathbb{M}}{\text{minimize}} \ -\mu \\ &\text{subject to } \mu \in \bigcap_{l=1,\dots,M} \mathbb{M}_{H_\mathcal{S}^{i,l}} \end{aligned} \tag{34}$$

where $\mathbb{M}_{H_\mathcal{S}^{i,l}} \coloneqq \{\mu \in \mathbb{M} : \mu \leq d_\mathcal{S}^{i,l}\}$. Note that the cost function is linear in μ. Moreover, $\mathbb{M}$ and $\mathbb{M}_{H_\mathcal{S}^i} \coloneqq \{\mu \in \mathbb{M} : \mu \leq d_\mathcal{S}^i(H_\mathcal{S}^i)\}$ with random variable $H_\mathcal{S}^i \in \Theta_i$, are convex and closed sets in μ, and the solution $\mu^* = d_\mathcal{S}^{i,*}$ to (34) exists and is unique.

In view of these facts and since there is only one optimization variable (i.e., μ), by using standard scenario-based arguments (e.g., see [43, Ch. 3]), if $M \geq 1$ fulfills $(1-\epsilon)^M \leq \beta$, we can state that, with probability at least $1-\beta$, it holds that $\mathcal{P}\left\{H_\mathcal{S}^i \in \Theta_i : d_\mathcal{S}^{i,*} \notin \mathbb{M}_{H_\mathcal{S}^i}\right\} = \mathcal{P}\{H_\mathcal{S}^i \in \Theta_i : d_\mathcal{S}^i(H_\mathcal{S}^i) < d_\mathcal{S}^{i,*}\} \leq \epsilon$. Note that $(1-\epsilon)^M \leq \beta$ can be easily rewritten as (15) by using basic logarithmic properties, since $\epsilon \in (0,1)$ and $\beta \in (0,1)$ by assumption.□

Proof of Theorem 4.1. Note that, any arbitrary $H_\mathcal{S} \in \Theta_{i^*}$ can be written as in (23), i.e., $H_\mathcal{S} = \sum_{\nu=1}^{n_V} \sigma_\nu H_\mathcal{S}^{V_\nu}$ for $\sigma_1 \geq 0, \dots, \sigma_{n_V} \geq 0$ and $\sigma_1 + \dots + \sigma_{n_V} = 1$. Consistently, we can write the corresponding F and G as

$$F = \sum_{\nu=1}^{n_V} \sigma_\nu F_\nu \text{ and } G = \sum_{\nu=1}^{n_V} \sigma_\nu G_\nu$$

for F_ν and G_ν in (24). Analogously to (25), define $\widetilde{F} \coloneqq WF$ and $\widetilde{G} \coloneqq WG$, where W is defined in (22). Define

$$P \coloneqq \sum_{\nu=1}^{n_V} \sigma_\nu P_\nu .$$

By the properties (i)-(ii) of each P_ν, $\nu = 1, \ldots, n_V$, we have $P = P^\top$, $p_{ij} = p_{ji} = 0$ for all $i \in \mathcal{W}_f$, $j \in \{1, \ldots, n\}$, $j \neq i$. Moreover, by property (iii),

$$\begin{aligned} &\begin{bmatrix} P & \widetilde{F}Q + \widetilde{G}L \\ (\widetilde{F}Q + \widetilde{G}L)^\top & Q + Q^\top - P \end{bmatrix} = \\ &\sum_{\nu=1}^{n_V} \sigma_\nu \begin{bmatrix} P_\nu & \widetilde{F}_\nu Q + \widetilde{G}_\nu L \\ (\widetilde{F}_\nu Q + \widetilde{G}_\nu L)^\top & Q + Q^\top - P_\nu \end{bmatrix} \succ 0. \end{aligned} \tag{35}$$

Note, in passing, that (35) implies that $P \succ 0$. This implies that $Q + Q^\top \succ 0$ and, hence, that $Q \in \mathbb{R}^{n \times n}$ is nonsingular [50, p. 433]. Therefore $J \coloneqq LQ^{-1}$ is well defined. Also E, as defined in (21d), is nonsingular, being full-row rank.

Since (35) holds, we can multiply it by the full-row-rank matrix $T = \begin{bmatrix} I_n & -(\widetilde{F} + \widetilde{G}J) \end{bmatrix}$ on the left and its transpose on the right to conclude that, since $L = JQ$,

$$T \begin{bmatrix} P & \widetilde{F}Q + \widetilde{G}(JQ) \\ (\widetilde{F}Q + \widetilde{G}(JQ))^\top & Q + Q^\top - P \end{bmatrix} T^\top \succ 0$$

or, equivalently, after algebraic computations,

$$P - (\widetilde{F} + \widetilde{G}J)P(\widetilde{F} + \widetilde{G}J)^\top \succ 0 \,. \tag{36}$$

Set $S = P^{-1}$. By the properties of P we have $S = S^\top \succ 0$ and $s_{ij} = s_{ji} = 0$ for all $i \in \mathcal{W}_f$, $j \in \{1, \ldots, n\}$, $j \neq i$. Moreover, since $A = F + GJ$ and in view of (36), $(WA)^\top S(WA) - S = (F + GJ)^\top W^\top SW(F + GJ) - S = (\widetilde{F} + \widetilde{G}J)^\top S(\widetilde{F} + \widetilde{G}J) - S = (\widetilde{F} + \widetilde{G}J)^\top P^{-1}(\widetilde{F} + \widetilde{G}J) - P^{-1} \prec 0$. Under Assumptions 3.1 and 4.1, we can conclude that the functions $f_i(\cdot)$ and matrix A enjoy **Property 1** with Lyapunov certificate S^{-1}. Therefore, by Theorem 1.1, the closed loop corresponding to $H_s \in \Theta_{i^*}$ and controller as in (27) is δISS with respect to u. Since $H_{\mathcal{S}}^o \in \Theta_{i^*}$ by assumption, the (unknown) closed loop in (21) is δISS with respect to u. □

Proof of Proposition 4.1. First, we rewrite (28) as

$$J_{VR}(H_{\mathcal{R}}) = \frac{1}{N-K_0}\,\mathrm{tr}((\mathbf{U}-\mathbf{X}H_{\mathcal{R}}^{\top})^{\top}(\mathbf{U}-\mathbf{X}H_{\mathcal{R}}^{\top}))\,. \tag{37}$$

If we consider L as optimization variable and we recall (27), we can rewrite (37) as

$$\begin{aligned}
J_{VR}(L) &= \frac{\mathrm{tr}((\mathbf{U}-\mathbf{X}E^{-\top}Q^{-1}L^{\top})^{\top}(\mathbf{U}-\mathbf{X}E^{-\top}Q^{-1}L^{\top}))}{N-K_0}\\
&= \frac{1}{N-K_0}\,\mathrm{tr}(\mathbf{U}^{\top}\mathbf{U} - LQ^{-1}E^{-1}\mathbf{X}^{\top}\mathbf{U} +\\
&- \mathbf{U}^{\top}\mathbf{X}E^{-\top}Q^{-1}L^{\top} + LQ^{-1}E^{-1}\mathbf{X}^{\top}\mathbf{X}E^{-\top}Q^{-1}L^{\top})\,,
\end{aligned}$$

where $\mathbf{U}^{\top}\mathbf{U}$ is constant with respect to the optimization variable L. If we set Q as in (30), under Assumption 4.2, we obtain

$$J_{VR}(L) = \frac{\mathrm{tr}(\gamma_q\mathbf{U}^{\top}\mathbf{U} + LQ^{-1}L^{\top} - L\widetilde{\mathbf{X}}\mathbf{U} - \mathbf{U}^{\top}\widetilde{\mathbf{X}}^{\top}L^{\top})}{(N-K_0)\gamma_q}.$$

In view of the properties of the trace, and since constant additive and strictly positive scaling terms do not take any role in the minimization of a cost function, minimizing $J_{VR}(L)$ is equivalent to minimizing

$$\begin{aligned}
\widetilde{J}_{VR}(L) &:= \mathrm{tr}((L^{\top} - Q\widetilde{\mathbf{X}}\mathbf{U})^{\top}Q^{-1}(L^{\top} - Q\widetilde{\mathbf{X}}\mathbf{U})) =\\
&= \mathrm{tr}(LQ^{-1}L^{\top} - L\widetilde{\mathbf{X}}\mathbf{U} - \mathbf{U}^{\top}\widetilde{\mathbf{X}}^{\top}L^{\top} + \mathbf{U}^{\top}\widetilde{\mathbf{X}}^{\top}Q\widetilde{\mathbf{X}}\mathbf{U})
\end{aligned}$$

which, in turn, is equivalent to solving

$$\begin{aligned}
&\underset{L\in\mathbb{R}^{m_{\mathcal{S}}\times n},\,\Phi=\Phi^{\top}\in\mathbb{R}^{m_{\mathcal{S}}\times m_{\mathcal{S}}}}{\mathrm{minimize}} \quad \mathrm{tr}(\Phi) \text{ subject to} \\
&\Phi - \mathbf{U}^{\top}\widetilde{\mathbf{X}}^{\top}Q\widetilde{\mathbf{X}}\mathbf{U} + L\widetilde{\mathbf{X}}\mathbf{U} + \mathbf{U}^{\top}\widetilde{\mathbf{X}}^{\top}L^{\top} - LQ^{-1}L^{\top} \succeq 0.
\end{aligned} \tag{38}$$

By using the Schur complement, (38) can be reformulated as (29a)-(29b). □

References

[1] J. C. Hoskins, D. M. Himmelblau, Process control via artificial neural networks and reinforcement learning, Computers & chemical engineering 16 (4) (1992) 241–251.

[2] K. J. Hunt, D. Sbarbaro, R. Żbikowski, P. J. Gawthrop, Neural networks for control systems—a survey, Automatica 28 (6) (1992) 1083–1112.

[3] A. M. Schäfer, H. G. Zimmermann, Recurrent neural networks are universal approximators, in: Artificial Neural Networks – ICANN 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 632–640.

[4] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366.

[5] M. Zakwan, M. d'Angelo, G. Ferrari-Trecate, Universal approximation property of Hamiltonian deep neural networks, IEEE Control Systems Letters 7 (2023) 2689–2694. doi:10.1109/LCSYS.2023.3288350.

[6] F. Bonassi, M. Farina, J. Xie, R. Scattolini, On recurrent neural networks for learning-based control: recent results and ideas for future developments, Journal of Process Control 114 (2022) 92–104.

[7] W. D'Amico, M. Farina, G. Panzani, Recurrent neural network controllers learned using virtual reference feedback tuning with application to an electronic throttle body, in: 2022 European Control Conference, 2022, pp. 2137–2142.

[8] J. Miller, M. Hardt, Stable recurrent models, in: International Conference on Learning Representations, 2019.
URL `https://openreview.net/forum?id=Hygxb2CqKm`

[9] D. M. Stipanović, M. N. Kapetina, M. R. Rapaić, B. Murmann, Stability of gated recurrent unit neural networks: Convex combination formulation approach, Journal of Optimization Theory and Applications 188 (1) (2021) 291–306.

[10] Y. Ebihara, H. Waki, V. Magron, N. H. A. Mai, D. Peaucelle, S. Tarbouriech, l_2 induced norm analysis of discrete-time LTI systems for nonnegative input signals and its application to stability analysis of recurrent neural networks, European Journal of Control 62 (2021) 99–104.

[11] Y. Ebihara, H. Waki, V. Magron, N. H. A. Mai, D. Peaucelle, S. Tarbouriech, Stability analysis of recurrent neural networks by IQC with copositive mutipliers, in: Conference on Decision and Control, IEEE, 2021, pp. 5098–5103.

[12] S. Hu, J. Wang, Global stability of a class of discrete-time recurrent neural networks, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 49 (8) (2002) 1104–1117.

[13] M. Buehner, P. Young, A tighter bound for the echo state property, IEEE Transactions on Neural Networks 17 (3) (2006) 820–824.

[14] I. B. Yildiz, H. Jaeger, S. J. Kiebel, Re-visiting the echo state property, Neural networks 35 (2012) 1–9.

[15] M. Revay, I. Manchester, Contracting implicit recurrent neural networks: Stable models with improved trainability, in: Learning for Dynamics and Control, Proceedings of Machine Learning Research, 2020, pp. 393–403.

[16] M. Revay, R. Wang, I. R. Manchester, A convex parameterization of robust recurrent neural networks, IEEE Control Systems Letters 5 (4) (2020) 1363–1368.

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

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

[19] F. Bayer, M. Bürger, F. Allgöwer, Discrete-time incremental ISS: A framework for robust NMPC, in: 2013 European Control Conference, 2013, pp. 2068–2073.

[20] D. Angeli, A lyapunov approach to incremental stability properties, IEEE Transactions on Automatic Control 47 (3) (2002) 410–421.

[21] H. Yin, P. Seiler, M. Arcak, Stability analysis using quadratic constraints for systems with neural network controllers, IEEE Transactions on Automatic Control 67 (4) (2021) 1980–1987.

[22] C. de Souza, A. Girard, S. Tarbouriech, Event-triggered neural network control using quadratic constraints for perturbed systems, Automatica 157 (2023) 111237.

[23] J. Vance, S. Jagannathan, Discrete-time neural network output feedback control of nonlinear discrete-time systems in non-strict form, Automatica 44 (4) (2008) 1020–1027.

[24] M. Revay, R. Wang, I. R. Manchester, Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness, IEEE Transactions on Automatic Control (2023) 1–16doi:10.1109/TAC.2023.3294101.

[25] M. Revay, R. Wang, I. R. Manchester, Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models, in: Conference on Decision and Control, IEEE, 2021, pp. 2282–2287.

[26] R. Wang, I. R. Manchester, Youla-REN: Learning nonlinear feedback policies with robust stability guarantees, in: American Control Conference, IEEE, 2022, pp. 2116–2123.

[27] L. Furieri, C. L. Galimberti, G. Ferrari-Trecate, Neural system level synthesis: Learning over all stabilizing policies for nonlinear systems, in: Conference on Decision and Control, IEEE, 2022, pp. 2765–2770.

[28] K. Seel, E. I. Grøtli, S. Moe, J. T. Gravdahl, K. Y. Pettersen, Neural network-based model predictive control with input-to-state stability, in: 2021 American Control Conference, 2021, pp. 3556–3563.

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

[30] F. Bonassi, A. La Bella, M. Farina, R. Scattolini, Nonlinear mpc design for incrementally iss systems with application to gru networks, Automatica 159 (2024) 111381.

[31] E. Terzi, F. Bonassi, M. Farina, R. Scattolini, Learning model predictive control with long short-term memory networks, International Journal of Robust and Nonlinear Control 31 (18) (2021) 8877–8896.

[32] W. D'Amico, A. La Bella, F. Dercole, M. Farina, Data-based control design for nonlinear systems with recurrent neural network-based controllers, IFAC-PapersOnLine 56 (2) (2023) 6235–6240.

[33] M. C. Campi, S. M. Savaresi, Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach, IEEE Transactions on Automatic Control 51 (1) (2006) 14–27.

[34] W. D'Amico, A. La Bella, M. Farina, An incremental input-to-state stability condition for a class of recurrent neural networks, IEEE Transactions on Automatic Control (2023) 1–16doi:10.1109/TAC.2023.3327937.

[35] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148 (34) (2001) 13.

[36] W. D'Amico, M. Farina, Virtual reference feedback tuning for linear discrete-time systems with robust stability guarantees based on set membership, Automatica 157 (111228) (2023).

[37] M. Milanese, J. Norton, H. Piet-Lahanier, É. Walter, Bounding approaches to system identification, Springer Science & Business Media, 2013.

[38] M. Lukoševičius, A practical guide to applying echo state networks, in: Neural Networks: Tricks of the Trade: Second Edition, Springer, 2012, pp. 659–686.

[39] H. K. Khalil, Nonlinear systems, 3rd ed., Prentice Hall, 2002.

[40] J. Cao, J. Wang, Absolute exponential stability of recurrent neural networks with Lipschitz-continuous activation functions and time delays, Neural networks 17 (3) (2004) 379–390.

[41] L. Bugliari Armenio, L. Fagiano, E. Terzi, M. Farina, R. Scattolini, Optimal training of echo state networks via scenario optimization, IFAC-PapersOnLine 53 (2) (2020) 5183–5188.

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

[43] M. C. Campi, S. Garatti, Introduction to the scenario approach, SIAM, 2018.

[44] M. C. Campi, A. Lecchini, S. M. Savaresi, Virtual reference feedback tuning (VRFT): a new direct approach to the design of feedback controllers, in: 39th IEEE Conference on Decision and Control, Vol. 1, 2000, pp. 623–629.

[45] D. Avis, D. Bremner, A. Deza, Polyhedral computation, Vol. 48, American Mathematical Soc., 2009.

[46] L. B. Armenio, E. Terzi, M. Farina, R. Scattolini, Echo state networks: analysis, training and predictive control, in: 2019 18th European Control Conference, 2019, pp. 799–804.

[47] A. Bisoffi, C. De Persis, P. Tesi, Data-driven control via Petersen's lemma, Automatica 145 (2022) 110537.

[48] J. Löfberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: IEEE International Symposium on Computer Aided Control Systems Design, 2004, pp. 284–289.

[49] M. ApS, The MOSEK optimization toolbox for MATLAB manual. Version 9.0. (2019).
URL http://docs.mosek.com/9.0/toolbox/index.html

[50] R. A. Horn, C. R. Johnson, Matrix Analysis, 2nd ed., Cambridge University Press, 2013.