Reconciling Deep Learning and Control Theory: Recurrent Neural Networks for Indirect Data-Driven Control

Fabio Bonassi

Abstract This Brief aims to discuss the potential of Recurrent Neural Networks (RNNs) for indirect data-driven control. Indeed, while RNNs have long been known to be universal approximators of dynamical systems, their adoption for system identification and control has been limited by the lack of solid theoretical foundations. We here intend to summarize a novel approach to address this gap, which is structured in two contributions. First, a framework for learning safe and robust RNN models is devised, relying on the Incremental Input-to-State Stability (δISS) notion. Then, after a δISS black-box model of the plant is identified, its use for the design of model-based control laws (such as Nonlinear MPC) with closed-loop performance guarantees is illustrated. Finally, the main open problems and future research directions are outlined.

1 Introduction

In recent decades, the control systems community has devoted an increasing research interest to data-driven control. This term relates to the approaches in which the control system is directly synthesized based on the data collected from the physical plant to be controlled (direct approaches), or designed relying on a dynamical model identified from such data (indirect approaches).

The common rationale behind these methods is that retrieving first-principle models of physical systems is generally a time-consuming task, and these models are often valid only in a neighborhood of the nominal operating conditions. Such limitations

This work has been partially supported by the European Union's Horizon 2020 programme under the Marie Skłodowska-Curie grant No. 953348.

F. Bonassi (✉)
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy
e-mail: fabio.bonassi@polimi.it

Department of Information Technology, Uppsala University, Lägerhyddsvägen 1, 75237 Uppsala, Sweden

F. Amigoni (ed.), *Special Topics in Information Technology*,
PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-031-51500-2_7

arise, e.g., from assumptions needed to obtain reasonable analytical models, or from the estimation of their unknown parameters, which is typically carried out by locally perturbing the operating conditions. Data-driven control aims to address these limitations, exploiting the information embedded in the measured data to synthesize control systems that are as accurate and global as possible, while minimizing the need for human intervention in the design phase.

In this context, researchers and engineers soon realized that many of the tools and methodologies developed within the deep learning community could find relevant applications for data-driven control. Neural Networks (NNs), and in particular Recurrent Neural Networks (RNNs), have been the object of many research efforts and engineering applications, see [21], owing to their advanced capabilities of modeling dynamical systems, for which they are known universal approximators. Although these modeling capabilities have been known for decades, the use of RNNs for learning dynamic systems has only recently been made effective by the increased availability of data, the development of RNN architectures less prone to vanishing and exploding gradient problems [20], and the development of open source software platforms for training them.

In light of their flexibility, there exists a multitude of different data-driven control strategies making use of NNs and RNNs [9]. In this work, we focus on (*i*) the use of RNNs for black-box nonlinear system identification and (*ii*) the design of theoretically-sound control laws based on the learned RNN models. By resorting to this indirect data-driven control paradigm, one can exploit RNNs' modeling capabilities while relying on the vast literature of nonlinear model-based control strategies, such as Nonlinear Model Predictive Control (NMPC).

Although RNN-based NMPC has been successful in numerous applications, such a strategy has often garnered criticism by the control systems community, due to the lack of solid theoretical foundations guaranteeing the accuracy of the learned models, let alone the closed-loop stability and performances. Despite these clear goals, only limited theoretical results, mainly related to the simplest RNN architecture (i.e. "vanilla" RNNs), had been obtained [25]. Researchers have indeed struggled to build a theoretical framework for the adoption of more advanced RNN architectures, such as Gated Recurrent Units (GRU) and Long Short Term-Memory (LSTM) networks, due to their structural complexity.

Contributions

This work is intended to outline some contributions given in [6] that aim to fill the above-mentioned methodological and theoretical gaps, establishing a novel and theoretically-sound framework for RNN-based indirect data-driven control. The approach is structured in two contributions.

Learning stable RNN models—A methodology for learning "safe" and "robust" RNN models is devised, by resorting to classical nonlinear stability notions such as the Incremental Input-to-State Stability (δISS, [1]). To this end, novel sufficient conditions on the weights of several RNN architectures, such as Neural NARXs (NNARXs), as well as more advanced RNNs like GRUs and LSTMs, have been proposed in [7, 8, 24], respectively. These conditions are leveraged to devise a training procedure for learning RNNs with δISS certification [9].

Control design—Based on the identified δISS RNN models, several control architectures with closed-loop guarantees are devised. One approach relies on the design of a state observer with exponential convergence guarantees to synthesize an NMPC law [6, 10]. Relying on the model's δISS, this control strategy can guarantee nominal closed-loop stability and recursively feasibility provided that the cost function is designed according to the proposed criterion, which makes the design procedure fairly easy. For the sake of compactness, only this control architecture is reported here, see Sect. 3.2. More involved NMPC architectures, able to attain asymptotic offset-free tracking of piecewise-constant reference signals, can also be synthesized, as shown in [12, 14]. These architectures rely on the enlargement of the RNN model with integral action and on the design of a state observer for the resulting enlarged system, which is provenly enabled by the δISS of the RNN model itself.

Remarkably, the proposed framework for learning δISS RNN models has also been shown to enable the design of a variety of other control architectures with closed-loop guarantees. To mention a few, in [22, 23] a disturbance estimation-based NMPC has been devised for LSTM models, whereas in [11] an Internal Model Control (IMC) architecture with local stability guarantees has been proposed. This latter control strategy has been shown to attain closed-loop performances close to those of NMPC laws at a fraction of their online computational burden, making it suitable for implementation on embedded control boards with limited computational resources.

The following notation is adopted here. Given a vector $v \in \mathbb{R}^n$, we denote by v' its transpose and by $\|v\|_p$ its p-norm. The Hadamard (element-wise) product between two vectors v and w of the same dimensions is denoted as $v \circ w$. Given a matrix A, $\|A\|_p$ is used to indicate its induced p-norm. Time-varying vectors are denoted by the time index k as a subscript. Sequences of vectors spanning from time k_1 to $k_2 \geq k_1$ are denoted as $v_{k_1:k_2} = \{v_{k_1}, v_{k_1+1}, ..., v_{k_2}\}$.

2 Learning Stable RNN Models

Let us consider an RNN in the state-space form

$$\Sigma(\Phi) : \begin{cases} x_{k+1} = f(x_k, u_k; \Phi) \\ y_k = g(x_k; \Phi) \end{cases}, \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, and $y_k \in \mathbb{R}^{n_y}$ denote the state, input, and output vectors, respectively, and $n_u = n_y$ for simplicity. The exact expression of the state transition function $f(\cdot)$ and of the output function $g(\cdot)$ depends on the specific RNN under consideration; see [6]. These functions are parametrized by the weights Φ, which are learned during the training procedure. In the following, let us compactly denote the state evolution of the system by $x_k(x_0, u_{0:k-1}; \Phi)$, obtained initializing (1) in x_0 and feeding it with the input sequence $u_{0:k-1}$.

Recalling the definitions of $\mathcal{K}_\infty$ and $\mathcal{KL}$ functions from [6, 9], we can now formulate the considered stability notion. Note that we here restrict the analysis to the δISS property, as it is the strongest and most useful property for the control synthesis. Other less strict stability notions are available in [6].

Definition 1 (δISS [1]) System (1) is regionally δISS in its invariant set $\mathcal{X}$ if there exist a $\mathcal{KL}$ function β and a $\mathcal{K}_\infty$ function γ such that, for any pair of initial states $x_{a,0} \in \mathcal{X}$ and $x_{b,0} \in \mathcal{X}$, and any pair of input sequences $u_{a,0:k} \in \mathcal{U}$ and $u_{b,0:k} \in \mathcal{U}$, at any time step $k \in \mathbb{Z}_{\geq 0}$ it holds that

$$\|x_{a,k} - x_{b,k}\|_p \leq \beta(\|x_{a,0} - x_{b,0}\|_p, k) + \gamma\Big(\max_{\tau \in \{0,\dots,k-1\}} \|u_{a,\tau} - u_{b,\tau}\|_p \Big), \tag{2}$$

where $x_{*,k}$ is short for $x_k(x_{*,0}, u_{*,0:k-1}; \Phi)$.

Note that the δISS implies, among other desirable properties, that (*i*) the effect of the initial conditions asymptotically vanishes, meaning that the modeling performances of the RNN are asymptotically independent of the random initial conditions; (*ii*) the RNN is robust against input perturbations, since closer input sequences imply a tighter asymptotic bound on the distance between the resulting state trajectories; (*iii*) the RNN is bounded-input bounded-state stable; (*iv*) the RNN admits exactly one equilibrium for any constant input $\bar{u} \in \mathcal{U}$ [22].

Theorem 1 (δISS sufficient conditions [6, 9]) *For each of the considered RNN architectures there exist sufficient conditions, in the form of nonlinear non-convex inequalities on the network's weights, compactly denoted as*

$$\nu(\Phi) < 0, \tag{3}$$

that guarantee the δISS in the sense specified by Definition 1.

Of course, each architecture has different expressions for condition (3). The interested reader is addressed to [7, 8, 13] for the exact expression in the case of NNARXs, LSTMs, and GRUs, respectively. Moreover, let us notice that for these architecture the δISS property is *exponential*, i.e., there exist $\mu > 0$ and $\lambda \in (0, 1)$ such that β can be expressed as $\beta(s, k) = \mu \lambda^k s$.

2.1 Training Procedure

Being a known function of the weights, the δISS condition (3) can be used not only to assess a-posteriori the stability of a trained model but can also be enforced during the training procedure, allowing one to learn RNN models with δISS certification. In the following, a training algorithm based on the Truncated Back-Propagation Through Time (TBPTT, [3]) is therefore outlined. A more detailed version of the algorithm can be found in [6].

Assume that N_{tr} pairs of input-output training subsequences of length T_s are available, and let them be denoted by $(u_{0:T_s}^{\{i\}}, y_{0:T_s}^{\{i\}})$, with $i \in \mathcal{I} = \{0, ..., N_{tr}\}$. Such subsequences are randomly extracted from the normalized[1] input-output sequences recorded from the plant during the experiment campaign. Note that N_{tr} and T_s are designed so that the subsequences are partially overlapping, which allows to mitigate the vanishing gradient phenomenon [20].

The training procedure is iterative, where at each iteration (known as epoch) the set $\mathcal{I}$ is randomly partitioned in B batches, denoted by $\mathcal{I}^{\{b\}}$. For each batch $b \in \{1, ..., B\}$, the training loss function is defined as

$$\mathcal{L}(\mathcal{I}^{\{b\}}; \Phi) = \sum_{i \in \mathcal{I}^{\{b\}}} \mathrm{MSE}\Big(y_{\tau_w:k}\big(x_0, u_{0:k}^{\{i\}}; \Phi\big), y_{\tau_w:k}^{\{i\}}\Big) + \rho\big(\nu(\Phi)\big), \tag{4}$$

where the first term penalizes the Mean Square Error (MSE) between the measured output sequence $y_{\tau_w:k}^{\{i\}}$ and the free-run simulation of the RNN (1) (starting from random initial conditions and fed by the input sequence $u_{0:k}^{\{i\}}$) after a *washout* period $\tau_w > 0$, which accommodates the initial transient. The second term is a regularizer that penalizes the violation of the δISS condition. The loss function gradient $\nabla_\Phi \mathcal{L}(\mathcal{I}^{\{b\}}; \Phi)$ is then backpropagated via gradient descent, or by accelerated gradient descent methods like ADAM and RMSProp [2].

At the end of each epoch, the performance metrics of the RNN on a validation dataset are computed. The training procedure is halted when the stability condition (3) is satisfied and the validation performance metrics stop improving, yielding the trained weights $\Phi^\star$. Finally, the modeling performances of the trained network are assessed on an independent test dataset.

3 Control Design

3.1 Definition of the Control Problem

At this stage, let us assume that an RNN model of the system, $\Sigma(\Phi^\star)$, has been trained and that it satisfies the δISS conditions described in Theorem 1. It is reminded that $\mathcal{X}$ denotes the invariant set with respect to the input set $\mathcal{U}$.

Under the Certainty Equivalence Principle (CEP), the control problem consists in synthesizing a control law that steers the model's output y_k to a piecewise-constant setpoint $\bar{y}$, while fulfilling the input constraint $u_k \in \mathcal{U}$. Letting $\mathrm{Int}(\mathcal{S})$ be the interior

[1] Input and output vectors are henceforth assumed to have zero mean and unity scale.

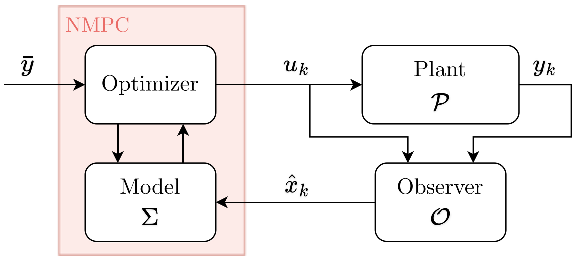

Fig. 1 Schematic of an RNN-based NMPC

part of set $\mathcal{S}$, the following assumption can be introduced to state the control problem more formally.

Assumption 1 Given the output setpoint $\bar{y}$, there exist $\bar{x} \in \text{Int}(\mathcal{X})$ and $\bar{u} \in \text{Int}(\mathcal{U})$ such that the triplet $\bar{z} = \bar{z}(\bar{y}) = (\bar{x}, \bar{u}, \bar{y})$ constitutes a feasible equilibrium of the RNN model (1), that is, $\bar{x} = f(\bar{x}, \bar{u}; \Phi^\star)$ and $\bar{y} = g(\bar{x}; \Phi^\star)$.

The control problem can now be formally stated.

Problem 1 Given the δISS RNN model $\Sigma(\Phi^\star)$ and the output setpoint $\bar{y}$, steer the system to the feasible equilibrium $\bar{z}(\bar{y})$ by means of a control action that satisfies the input constraint $\mathcal{U}$.

Leveraging the model's δISS, Problem 1 has been addressed with a variety of approaches [6], such as internal model control [11] and nonlinear model predictive control, see e.g. [23, 24], and [14]. In the following, one of the possible NMPC approaches is outlined for illustrative reasons.

3.2 NMPC Design

In this section, the synthesis procedure of the scheme depicted in Fig. 1 is summarized. Note that, since RNNs are generally black-box models and NMPC is a state-feedback control law, the model states need to be estimated by a suitably-designed state observer. The synthesis of the proposed control architecture is therefore structured in two steps, i.e., (*i*) the design of a state observer for the RNN model and (*ii*) the formulation of NMPC's underlying Finite Horizon Optimal Control Problem (FHOCP).

Weak Detector Design—In order to estimate the states of the black-box models from the plant's input and output data, a state observer with convergence guarantees should be designed. While nonlinear state observers can be designed with several different approaches, such as moving horizon estimators, we here consider Luenberger-like observers. Such observers are generally synthesized by including in the model dynamics a suitably designed innovation term.[2] In the following, we

[2] See [23, 24] for the design of observers for LSTMs, and [6] for GRUs.

denote such observer by

$$\mathcal{O}(\Phi_o): \quad \hat{x}_{k+1} = f_o(\hat{x}_k, u_k, y_k; \Phi_o), \tag{5}$$

parametrized by $\Phi_o = \Phi^\star \cup \Phi_L$, where Φ_L collects the observer's innovation gains.

Definition 2 (*Weak detector*) System (5) is said to be a weak detector of model (1) if there exist $\mu_o > 0$ and $\lambda_o \in (0, 1)$ such that, for any initial condition of the model $x_0 \in \mathcal{X}$, any initial guess $\hat{x}_0 \in \mathcal{X}$, and any input sequence $u_{0:k}$, it holds that $\|\hat{x}_k - x_k\|_2 \leq \mu_o \lambda_o^k \|\hat{x}_0 - x_0\|_2$.

Relying on the δISS property of the trained RNN model, in [6, 23, 24] sufficient conditions on the innovation gains Φ_L which guarantee the state observer to be a weak detector have been devised. A notable case is that of GRU models, where the devised conditions can be leveraged to formulate the observer design problem as a convex optimization program [6, Proposition 6.1].

Formulation of the FHOCP—According to the MPC paradigm, the control law is retrieved by solving, at every time-step k, the underlying FHOCP. Such an optimization problem relies on the RNN predictive model of the system, i.e. (1), to predict the future state trajectories throughout the prediction horizon N, given the current state estimate $\hat{x}_k$ yielded by the observer (5) and the applied control sequence. Let therefore $u_{k:k+N-1|k}$ be the control sequence applied throughout the prediction horizon, and let $x_{k:k+N|k}$ indicate the resulting state trajectories, where, of course, $x_{k|k} = \hat{x}_k$. Under this notation, letting $\mathcal{N} = \{0, ..., N-1\}$, the considered FHOCP can be stated as follows.

$$\min_{u_{k:k+N-1|k}} \sum_{\tau=0}^{N-1} \left(\|x_{k+\tau|k} - \bar{x}\|_Q^2 + \|u_{k+\tau|k} - \bar{u}\|_R^2 \right) + V_{\bar{z}}(x_{k+N|k}) \tag{6a}$$

$$s.t. \quad x_{k|k} = \hat{x}_k \tag{6b}$$

$$x_{k+\tau+1|k} = f(x_{k+\tau|k}, u_{k+\tau|k}; \Phi^\star) \qquad \forall \tau \in \mathcal{N} \tag{6c}$$

$$u_{k+\tau|k} \in \mathcal{U} \qquad \forall \tau \in \mathcal{N} \tag{6d}$$

Note that the predictive model is initialized at the observer state estimate in (6b), whereas its dynamics are embedded by means of constraint (6c). Input constraint satisfaction is ensured by (6d), while $x_{k+\tau|k} \in \mathcal{X}$ is guaranteed by the invariance of $\mathcal{X}$. The cost function (6a) is composed by two terms. The first term penalizes states' and inputs' deviations from their equilibrium values $\bar{x}$ and $\bar{u}$, respectively, by the weights $Q \succ 0$ and $R \succ 0$. The term $V_{\bar{z}}(x_{k+N|k})$ represents a terminal cost approximating the cost-to-go from the terminal state $x_{k+N|k}$ to the equilibrium $\bar{x}$ under the constant input $\bar{u}$. That is,

$$V_{\bar{z}}(x_{k+N|k}) = \sum_{h=0}^{M} \|x_{k+N+h|k} - \bar{x}\|_S^2, \tag{6e}$$

where $x_{k+N+h+1|k} = f(x_{k+N+h|k}, \bar{u}; \Phi^\star)$ for any $h \in \{0, ..., M-1\}$, $S \succ 0$ is the weight, and $M > 0$ the simulation horizon. According to the receding horizon principle, at every step k the FHOCP (6) is solved, and the first optimal control action is applied, i.e., $u_k = u^\star_{k|k}$. Then, at the next time step, the entire procedure is repeated, yielding an implicit state-feedback control law $u_k = \kappa_{\text{MPC}}(\hat{x}_k)$.

In this framework, the model's δISS property and the state observer's exponential convergence have been leveraged to propose conditions on the NMPC design parameters (Q, R, S, N, and M) that allow attaining nominal closed-loop stability and recursive feasibility [6, Theorem 6.2]. Such conditions boil down to inequalities on the singular values of the weight matrices Q and R, and to an explicit minimum value for the simulation horizon M [10]. Seen through these lenses, the devised NMPC scheme can be regarded as a constrained quasi-infinite horizon NMPC [5], where however a minimum prediction horizon is known explicitly.

3.3 Offset-Free NMPC

Albeit attaining desirable nominal closed-loop guarantees, applying the control scheme proposed in Sect. 3.2 to the plant may result in non-ideal tracking performances. The model may indeed be affected by plant-model mismatch, in which case zero-error output regulation might not be achieved. For the main RNN architectures, this problem has been addressed by resorting to the two traditional approaches for offset-free NMPC, namely

- *Integral action-based approaches*—In the spirit of [17], integrators can be placed on the output tracking error so that, as long as the closed-loop stability is preserved, robust asymptotic zero-error output regulation is achieved in virtue of the Internal Model Principle [16]. Applications of such strategy to RNN architectures are available in [12, 14, 19].
- *Disturbance estimation-based approaches*—Along the lines of [18], an approach guaranteeing offset-free static performances relies on the enlargement of the system model with the disturbance dynamics. This allows the disturbance to be estimated by means of a state observer and to be accounted for and compensated for in the NMPC formulation. Applications of this strategy to LSTM models are available in [22, 23].

4 Open Problems and Future Research Directions

Despite the great potential that RNNs have shown in the context of data-driven control, there are several open issues that have been only partially addressed, and whose resolution would lead to improved applicability of these strategies, even in safety-critical contexts. Below, these issues are briefly outlined, while for more details the interested reader is addressed to [6].

i. *Safety verification*—The problem of safety verification consists in assessing that the RNN model's output reachable set lies within a "safe" set, e.g., the set of physically-meaningful outputs. Safety verification hence allows certifying that the model does not generate unsafe or unexpected outputs. While this procedure is notoriously involved for nonlinear systems, especially for RNNs, their δISS certification allows for retrieval of an analytical expression of the output reachable set. Such an expression is however conservative in general, calling for numerical procedures to approximate the output reachable set and hence for RNNs' probabilistic safety verification algorithms [13].
ii. *Lifelong learning*—A common problem in the context of indirect data-driven control is ensuring that the identified model remains an accurate approximation of the plant throughout its lifespan. While in the case of plant's dramatic variations (e.g., due to faults) the common practice is to collect new data and learn a new RNN model of the system, in the case of moderate and slow variations it would be advisable to exploit the online data to adapt the model to these changes. This practice is commonly referred to as lifelong learning and should be conducted by averting the catastrophic forgetting phenomenon, i.e., the overfitting of the most recent data and the consequent forgetting of the past data. For black-box RNN models, this issue represents an open research topic, whereas for NNARX architectures preliminary results have been reported in [15], based on a moving horizon estimation approach.
iii. *Physics-based machine learning*—One of the research directions that have been recently considered to be most promising by the scientific community is that of physics-based machine learning. In summary, it consists in exploiting the available qualitative knowledge of the physical laws governing the plant in order to improve the consistency, interpretability, generalizability, and ultimately the accuracy of the model. As discussed in [9] and references therein, physical consistency can be achieved via a suitable design of the training loss function and of the NN architecture, so as to ensure—for example—a known dynamical structure or the satisfaction of known relationships between variables [4].
iv. *Robust control design*—A control architecture capable of ensuring robustness properties with respect to disturbances and plant-model mismatch while satisfying input and state constraints is one of the most challenging research directions when adopting RNN models, due to their structural complexity. A preliminary approach in this direction has been proposed in [26] for NNARX models and, more recently, in [22] for LSTM models.

5 Conclusions

In this Brief, we summarized a novel framework towards the training of black-box Recurrent Neural Network (RNN) models with Incremental Input-to-State Stability (δISS) certification. The proposed method thus allows learning RNNs that are safe and robust against input perturbations and mismatches in initial conditions and applies to a variety of RNN architectures, such as Neural NARXs, Gated Recurrent Units, and Long Short-Term Memory networks. Relying on the model's δISS, theoretically sound model-based control strategies can be synthesized. In particular, in this Brief the design of a nonlinear model predictive control law with nominal closed-loop stability guarantees has been outlined, discussing the extension of the scheme to also achieve asymptotic zero-error setpoint tracking. Finally, the main open problems and promising future research directions, such as safety verification of RNN models and physics-based machine learning, have been reported.

References

1. Bayer F, Bürger M, Allgöwer F (2013) Discrete-time incremental ISS: a framework for robust NMPC. In: 2013 European control conference (ECC), pp 2068–2073. IEEE (2013)
2. Bengio, Y., Goodfellow, I., Courville, A.: Deep learning, vol. 1. MIT Press Massachusetts, USA (2017)
3. Bianchi, F.M., Maiorino, E., Kampffmeyer, M.C., Rizzi, A., Jenssen, R.: Recurrent neural networks for short-term load forecasting: an overview and comparative analysis. Springer (2017)
4. Boca de Giuli L, La Bella A, Scattolini R (2023) Physics-informed neural network modelling and predictive control of district heating systems. arXiv e-prints, arXiv-2310
5. Boccia, A., Grüne, L., Worthmann, K.: Stability and feasibility of state constrained MPC without stabilizing terminal constraints. Syst Control Lett **72**, 14–21 (2014)
6. Bonassi F (2023) Reconciling deep learning and control theory: recurrent neural networks for model-based control design. Doctoral dissertation, Politecnico di Milano, Advisor: R. Scattolini
7. Bonassi, F., Farina, M., Scattolini, R.: On the stability properties of gated recurrent units neural networks. Syst Control Lett **157**, 105049 (2021)
8. Bonassi, F., Farina, M., Scattolini, R.: Stability of discrete-time feed-forward neural networks in NARX configuration. IFAC-PapersOnLine **54**(7), 547–552 (2021)
9. Bonassi, F., Farina, M., Xie, J., Scattolini, R.: On recurrent neural networks for learning-based control: recent results and ideas for future developments. J Process Control **114**, 92–104 (2022)
10. Bonassi, F., La Bella, A., Farina, M., Scattolini, R.: Nonlinear MPC design for incrementally ISS systems with application to GRU networks. Automatica **159**, 111381 (2024)
11. Bonassi F, Scattolini R (2022) Recurrent neural network-based internal model control of unknown nonlinear stable systems. Eur J Control 100632
12. Bonassi, F., Oliveira da Silva, C.F., Scattolini, R.: Nonlinear MPC for offset-free tracking of systems learned by GRU neural networks. IFAC-PapersOnLine **54**(14), 54–59 (2021)
13. Bonassi F, Terzi E, Farina M, Scattolini R (2020) LSTM neural networks: input to state stability and probabilistic safety verification. In: Learning for dynamics and control. PMLR, pp 85–94
14. Bonassi F, Xie J, Farina M, Scattolini R (2022) An offset-free nonlinear MPC scheme for systems learned by Neural NARX models. In: 2022 IEEE 61st conference on decision and control (CDC), pp 2123–2128

15. Bonassi F, Xie J, Farina M, Scattolini R (2022) Towards lifelong learning of recurrent neural networks for control design. In: 2022 European control conference (ECC), pp 2018–2023
16. Francis, B.A., Wonham, W.M.: The internal model principle of control theory. Automatica **12**(5), 457–465 (1976)
17. Magni, L., De Nicolao, G., Scattolini, R.: Output feedback and tracking of nonlinear systems with model predictive control. Automatica **37**(10), 1601–1607 (2001)
18. Morari, M., Maeder, U.: Nonlinear offset-free model predictive control. Automatica **48**(9), 2059–2067 (2012)
19. da Silva Oliveira CF (2021) Offset-free nonlinear MPC for systems learned by LSTM networks. Master thesis. Politecnico di Milano, Italy
20. Pascanu R, et al (2013) On the difficulty of training recurrent neural networks. In: International conference on machine learning. PMLR, pp 1310–1318
21. Pillonetto G, Aravkin A, Gedon D, Ljung L, Ribeiro AH, Schön TB (2023) Deep networks for system identification: a survey. arXiv:2301.12832
22. Schimperna I, Magni L (2023) Robust offset-free constrained model predictive control with long short-term memory networks–extended version. arXiv:2303.17304
23. Schimperna, I., Toffanin, C., Magni, L.: On offset-free model predictive control with long short-term memory networks. IFAC-PapersOnLine **56**(1), 156–161 (2023)
24. Terzi, E., Bonassi, F., Farina, M., Scattolini, R.: Learning model predictive control with long short-term memory networks. Int J Robust Nonlinear Control **31**(18), 8877–8896 (2021)
25. Wu, Z., Luo, J., Rincon, D., Christofides, P.D.: Machine learning-based predictive control using noisy data: evaluating performance and robustness via a large-scale process simulator. Chem Eng Res Design **168**, 275–287 (2021)
26. Xie, J., Bonassi, F., Farina, M., Scattolini, R.: Robust offset-free nonlinear model predictive control for systems learned by neural nonlinear autoregressive exogenous models. Int J Robust Nonlinear Control **33**(16), 9992–10009 (2023)