

Nonlinear MPC for Offset-Free Tracking of systems learned by GRU Neural Networks

Fabio Bonassi* Caio Fabio Oliveira da Silva*

Riccardo Scattolini

*Politecnico di Milano, Dipartimento di Eletttronica, Informazione e Bioingegneria, Via Ponzio 34/5, 20133 Milano, Italy
(e-mail: name.surname@polimi.it)*

Abstract: The use of Recurrent Neural Networks (RNNs) for system identification has recently gathered increasing attention, thanks to their black-box modeling capabilities. Albeit RNNs have been fruitfully adopted in many applications, only few works are devoted to provide rigorous theoretical foundations that justify their use for control purposes. The aim of this paper is to describe how stable Gated Recurrent Units (GRUs), a particular RNN architecture, can be trained and employed in a Nonlinear MPC framework to perform offset-free tracking of constant references with guaranteed closed-loop stability. The proposed approach is tested on a pH neutralization process benchmark, showing remarkable performances.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Machine Learning, Nonlinear Model Predictive Control, Model and Identification of Nonlinear Systems, Offset-free Tracking

1. INTRODUCTION

In recent years, the advances in machine learning techniques have motivated the control systems community to investigate the use of Recurrent Neural Networks (RNNs) for the identification of nonlinear dynamical systems. Among the various RNN architectures, Gated Recurrent Units (GRUs, Chung et al. (2014)) and Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) networks have proven to be particularly suited for system identification tasks (Forgione and Piga, 2020), as their stateful structure is able to retain long-term memory of past inputs and states.

In light of their superior modeling capabilities, these gated RNNs have been widely adopted by control practitioners, in conjunction with nonlinear model-based control strategies, such as Model Predictive Control (MPC), especially in process control domain (Wong et al., 2018). For example, in Terzi et al. (2020) a complex nonlinear system is identified by means of an LSTM network, which is then used as a prediction model in a Nonlinear MPC (NMPC) regulator. A similar strategy is adopted in Lanzetti et al. (2019), where a GRU network is used to identify and control a paper machine.

Despite the adoption of RNNs in many data-driven control applications, only limited theoretical foundations are nowadays available to justify the use of these NN architectures in systems control domain. The available theoretical results are mainly focused on achieving provably stable models. In Bonassi et al. (2020a) and Stipanović et al. (2020), the authors derive conditions that should be satisfied by network's weights in order to guarantee its stability. Similar conditions have been derived for LSTMs networks

in Bonassi et al. (2020b) and Terzi et al. (2021), where an NMPC control strategy with guaranteed closed-loop stability is also formulated.

While the stability of RNN architectures can be used to design standard closed-loop-stable MPC laws, see Terzi et al. (2021), the tracking performances of these control strategies mainly depend on the accuracy of the trained networks. To address this problem, in the literature several offset-free NMPC formulations have been proposed, which allow to achieve perfect tracking of output references with stability guarantees (Pannocchia et al., 2015). In Morari and Maeder (2012) offset-free tracking is achieved by enlarging the system with a disturbance model. A state observer is then designed for the enlarged system, so that the disturbance is estimated and compensated by the NMPC control system. Another approach is the one proposed by Magni et al. (2001c), where the system is augmented with an integral action on the output tracking error, a state observer is designed and a stabilizing NMPC law is synthesized for the augmented system.

The aim of this work is to further elaborate on Bonassi et al. (2020a) and Terzi et al. (2021), and to describe an unitary approach for the identification and off-set free control of an unknown dynamical system. Specifically, we herein propose to use a stable GRU network to identify the unknown plant from data collected from the real system. Once trained, this network is used as predictive model for an NMPC controller which ensures offset-free tracking of constant references with closed-loop stability guarantees, adopting the control strategy proposed by Magni et al. (2001c). Therefore, sufficient conditions under which it is possible to design a converging state observer for the augmented system are devised. The stabilizing NMPC law is thus formulated for the enlarged system. The approach is tested on a pH neutralization process benchmark system,

* Indicates equal contribution to this work.

showing remarkable results in terms of reference tracking capabilities under control constraints.

1.1 Notation

If v is a vector, v' is its transpose and $\|v\|$ its Euclidean norm. For simplicity, vectors whose components are all equal to one and to zero are compactly denoted as $\mathbf{1}$ and $\mathbf{0}$. By boldface fonts we indicate sequences of vectors, i.e. $\mathbf{v} = \{v(0), v(1), \dots\}$, and $\|\mathbf{v}\|_p = \max_{k \geq 0} \|v(k)\|_p$. For compactness, for time-varying quantities the generic time index k is omitted if it can be inferred from the context. Superscript $+$ indicates some quantity at the successive time instant, i.e. at $k + 1$. Therefore, $x = x(k)$ and $x^+ = x(k + 1)$. The Hadamard product between matrices or vectors is denoted by \circ . By σ and ϕ we indicate the sigmoid and tanh activation functions, respectively, i.e. $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\phi(x) = \tanh(x)$.

2. MODEL

In this section we introduce GRUs, discussing how these networks can be used to identify nonlinear dynamical systems and how stability properties can be enforced. For simplicity we focus on single-layer GRU networks, but the approach can be easily extended to multi-layer networks.

Let us now introduce a single-layer GRU expressed in state-space form (Bonassi et al., 2020a)

$$\Sigma : \begin{cases} x^+ = z \circ x + (1 - z) \circ \phi(W_r u + U_r f \circ x + b_r) \\ z(x, u) = \sigma(W_z u + U_z x + b_z) \\ f(x, u) = \sigma(W_f u + U_f x + b_f) \\ y = U_o x + b_o \end{cases}, \quad (1)$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ is the output, and $m = p$ is assumed. Functions $z = z(x, u)$ and $f = f(x, u)$ are known as update and forget gates, respectively. The matrices W_* , U_* , b_* , are the weights of the network, and must be tuned during the training procedure in such a way that (1) approximates sufficiently well the unknown plant from which the data was collected. For compactness, Σ may be re-written as

$$\Sigma : \begin{cases} x^+ = \varphi(x, u) \\ y = \eta(x) \end{cases}, \quad (2)$$

where $\varphi(x, u)$ and $\eta(x)$ can be easily retrieved from (1).

In the following we assume that the input u is unity-bounded, i.e. $u \in \mathcal{U} = [-1, 1]^m$ or, equivalently, $\|u\|_\infty \leq 1$. Note that this is a quite customary assumption when working with neural networks, and it can be easily satisfied by means of a suitable normalization (Terzi et al., 2021). It can be then shown that in finite time the state of the GRU enters a unity-boxed invariant set.

Lemma 1. (Bonassi et al. (2020a)). For any initial state \bar{x} and any input sequence \mathbf{u} , there exists a finite time instant $\bar{k} \geq 0$ at which the state enters its invariant set $\mathcal{X} = [-1, 1]^n$, i.e. $x(k) \in \mathcal{X} \forall k \geq \bar{k}$.

We remind that some function $\psi(s)$ is said to be a \mathcal{K}_∞ function if $\psi(0) = 0$, it is strictly increasing and $\psi(s) \rightarrow \infty$ as $s \rightarrow +\infty$. A function $\psi(s, t)$ is of class \mathcal{KL} if it is \mathcal{K}_∞ with respect to s and, in addition, $\psi(s, t) \rightarrow 0$ as $t \rightarrow 0$. Under these premises, we can summarize the main stability results for GRU networks. For more details, the interested reader is addressed to Bonassi et al. (2020a).

Definition 1. (δ ISS). System (2) is said to be Incrementally Input-to-State Stable (δ ISS) if there exists functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that, for any pair of initial states \bar{x}_a and \bar{x}_b , and any pair of input sequences \mathbf{u}_a and \mathbf{u}_b it holds that

$$\begin{aligned} & \|x(k, \bar{x}_a, \mathbf{u}_a) - x(k, \bar{x}_b, \mathbf{u}_b)\| \\ & \leq \beta(\|\bar{x}_a - \bar{x}_b\|, k) + \gamma(\|\mathbf{u}_a - \mathbf{u}_b\|), \end{aligned} \quad (3)$$

where $x(k, \bar{x}, \mathbf{u})$ is the state trajectory at time k , when (2) is initialized in \bar{x} and it is fed by the input sequence \mathbf{u} .

Note that, since $\beta \in \mathcal{KL}$, this stability property implies that the effects of initial conditions asymptotically vanish, guaranteeing that the modeling performances of the network is independent of the initialization.

As δ ISS plays a key role in the proposed control design, see Section 3, the following sufficient condition for the δ ISS of GRU networks proposed by Bonassi et al. (2020a) is enforced during the training of the network.

Theorem 1. (Bonassi et al. (2020a)). A sufficient condition for the δ ISS of system (2) is that

$$\|U_r\|_\infty \left(\frac{1}{4} \|U_f\|_\infty + \bar{\sigma}_f \right) < 1 - \frac{1}{4} \frac{1 + \bar{\phi}_r}{1 - \bar{\sigma}_z} \|U_z\|_\infty, \quad (4)$$

where $\bar{\sigma}_z$, $\bar{\phi}_r$, and $\bar{\sigma}_f$ are defined as

$$\begin{aligned} \bar{\sigma}_z &= \sigma(\|W_z \quad U_z \quad b_z\|_\infty), \\ \bar{\phi}_r &= \phi(\|W_r \quad U_r \quad b_r\|_\infty), \\ \bar{\sigma}_f &= \sigma(\|W_f \quad U_f \quad b_f\|_\infty). \end{aligned} \quad (5)$$

Note that this sufficient condition boils down to an inequality on the weights of the network. The fulfillment of (4) can be enforced as a constraint during the training procedure of the network. In case of use of unconstrained training algorithms, this condition may be relaxed, penalizing the violation of (4) in the loss function, as outlined in Section 4 and discussed in Bonassi et al. (2020a).

3. CONTROL ARCHITECTURE DESIGN

Thus far, a black-box model with guaranteed stability properties has been presented. In this Section, we assume that the Σ has been trained to model the unknown plant, and that the weights of this network satisfy condition (4), so that Theorem 1 guarantees the δ ISS of the network.

The goal of this section is to employ such model for offset-free tracking of constant reference signals. In particular, we adopt the control strategy proposed by Magni et al. (2001c) which, informally speaking, consists in the following steps: (i) augmenting the system model with an integrator on the output tracking error, (ii) designing a weak detector for the augmented plant, and (iii) synthesizing a stabilizing NMPC law for the augmented system.

The resulting control architecture is depicted in Figure 1. Under mild assumptions on the model's smoothness, and provided that the state observer is a weak detector for the augmented system and that the NMPC law stabilizes the reference equilibrium, Magni et al. (2001c) guarantees offset-free tracking of piece-wise constant references.

Remark 1. Differently from Magni et al. (2001c), in general $(x^0, u^0, y^0) = (0, 0, 0)$ is not an equilibrium of Σ , i.e. $\varphi(0, 0) \neq 0$ and $\eta(0) \neq 0$. However, since the designed

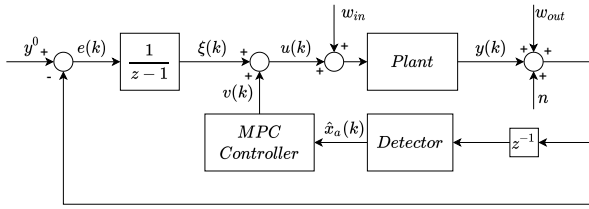


Fig. 1. A representation of the controller architecture.

observer is a weak detector with respect to any equilibrium, it is enough to design the NMPC law stabilizing the generic equilibrium (x^0, u^0, y^0) of Σ .

3.1 Model augmentation and state observer design

To achieve offset-free tracking of the output reference $y^0 \in \mathbb{R}^p$, the system model is augmented with the discrete-time integral of the tracking error. The enlarged system Σ_a reads as

$$\Sigma_a : \begin{cases} x^+ = \varphi(x, u) \\ y = \eta(x) \\ \xi^+ = \xi + y^0 - y \\ u = v + \xi \end{cases}, \quad (6)$$

where $\xi \in \mathbb{R}^p$ is the state of the integrator, and v is the exogenous input computed by the NMPC. Denoting by $x_a = [x', \xi']'$ the state of the augmented system and by $y_a = [y', \xi']'$ its output, (6) can be compactly denoted by

$$\Sigma_a : \begin{cases} x_a^+ = \varphi_a(x_a, v, y^0) \\ y_a = \eta_a(x_a, y^0) \end{cases}. \quad (7)$$

Then, a state observer needs to be designed for the enlarged system. This observer is required to be a weak detector of (7) for the equilibrium $x_a^0 = [x^{0r}, u^{0r}]'$, $v^0 = 0$, and $y_a^0 = [y^{0r}, u^{0r}]'$, in the sense specified by the following definition (Magni et al., 2001b).

Definition 2. System (7) is weakly detectable with respect to the equilibrium $(x_a^0, 0, y_a^0)$ if there exists a function $g : \mathbb{R}^{n+p} \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^{2p} \rightarrow \mathbb{R}^{n+p}$ and a function $V_e \in \mathcal{C}^1$ such that the following properties hold:

- (1) $g(\hat{x}_a, v, y^0, y_a)$ is of class \mathcal{C}^1 and $\hat{x}_a^0 = g(\hat{x}_a^0, 0, y^0, y_a^0)$
- (2) There exist functions a_e , b_e , and c_e of class \mathcal{K}_∞ such that, for any $x, \hat{x} \in \mathcal{X}$, any ξ and v such that $\|\xi + v\|_\infty \leq 1$, and any $\hat{\xi}$ such that $\|\hat{\xi} + v\|_\infty \leq 1$, it holds

$$a_e(\|x_a - \hat{x}_a\|) \leq V_e(x_a, \hat{x}_a) \leq b_e(\|x_a - \hat{x}_a\|), \quad (8a)$$

$$V_e(x_a^+, \hat{x}_a^+) - V_e(x_a, \hat{x}_a) \leq -c_e(\|x_a - \hat{x}_a\|), \quad (8b)$$
 where $x_a^+ = \varphi_a(x_a, v, y^0)$ and $\hat{x}_a^+ = g(\hat{x}_a, v, y^0, y_a)$.

Then, $\hat{x}_a^+ = g(\hat{x}_a, v, y^0, y_a)$ is called *weak detector* of Σ_a .

The proposed state observer for the augmented system reads as follows

$$\mathcal{O}_a : \begin{cases} \hat{x}^+ = \hat{z} \circ \hat{x} + (1 - \hat{z}) \circ \phi(W_r(v + \hat{\xi}) + U_r \hat{f} \circ \hat{x} + b_r) \\ \hat{z}(\hat{x}_a, v) = \sigma(W_z(v + \hat{\xi}) + U_z \hat{x} + b_z + L_{z\xi}(\xi - \hat{\xi}) + L_{zy}(y - \hat{y})) \\ \hat{f}(\hat{x}_a, v) = \sigma(W_f(v + \hat{\xi}) + U_f \hat{x} + b_f + L_{f\xi}(\xi - \hat{\xi}) + L_{fy}(y - \hat{y})) \\ \hat{\xi}^+ = \hat{\xi} + y^0 - \hat{y} + L_{\xi y}(y - \hat{y}) + L_{\xi\xi}(\xi - \hat{\xi}) \\ \hat{y} = U_o \hat{x} + b_o \end{cases} \quad (9)$$

The following results allow to provide guidelines for the tuning of the observer gains L_\star so as to guarantee that \mathcal{O}_a is a weak detector of Σ_a . The proofs are reported in the extended version of this paper (Bonassi et al., 2021).

Theorem 2. A sufficient condition under which \mathcal{O}_a is a weak detector of system (7), in the sense specified by Definition 2, is that there exists $\delta > 0$ such that

$$\|U_r\|_\infty \left(\frac{1}{4} \|U_f - L_{fy} U_o\|_\infty + \bar{\sigma}_f \right) + \frac{1}{4} \frac{1 + \bar{\phi}_r}{1 - \bar{\sigma}_z} \|U_z - L_{zy} U_o\|_\infty \leq 1 - \delta \quad (10a)$$

and that the matrix A_δ defined as

$$A_\delta = \begin{bmatrix} 1 - \delta & \\ \|U_o\|_\infty \|I + L_{\xi y}\|_\infty & \|I - L_{\xi\xi}\|_\infty \end{bmatrix} \quad (10b)$$

is Schur stable, where

$$\alpha = \frac{1}{4} \|W_z - L_{z\xi}\|_\infty (1 + \|W_r\|_\infty + \frac{1}{4} \|U_r\|_\infty \|W_f - L_{f\xi}\|_\infty). \quad (10c)$$

Proposition 1. The observer \mathcal{O}_a satisfies Theorem 2 if its gains L_\star are a feasible solution of the following optimization problem

$$\begin{aligned} & \min_{L_\star} \|A_\delta\| \\ & \text{s.t.} \quad (10a) \\ & \delta(1 - \|I - L_{\xi\xi}\|_\infty) > \alpha \|U_o\|_\infty \|I + L_{\xi y}\|_\infty \\ & (1 - \delta) \|I - L_{\xi\xi}\|_\infty < 1 + \alpha \|U_o\|_\infty \|I + L_{\xi y}\|_\infty \\ & \delta > 0 \end{aligned} \quad (11)$$

Note that Proposition 1 involves solving a non-linear optimization problem to spot suitable weights of \mathcal{O}_a . The constraints of (11) enforce the nominal convergence of the observed states \hat{x}_a to the real states x_a . Furthermore, recalling that the spectral radius of the matrix is bounded by its norm, i.e. $\rho(A_\delta) < \|A_\delta\|$, by minimizing $\|A_\delta\|$ one minimizes the maximum eigenvalue, likely making the observer convergence faster.

Corollary 1. If the model satisfies the δ ISS condition (4), the optimization problem (11) admits a feasible solution, corresponding to $L_{fy} = L_{zy} = 0_{n,p}$, $L_{z\xi} = W_z$, $L_{\xi\xi} < \lambda I_{p,p}$ with $\lambda \in (0, 1)$, and any $L_{f\xi}$ and $L_{\xi y}$.

Therefore, the δ ISS property of the model allows to guarantee the weak detectability of the augmented system, and the designed observer \mathcal{O}_a enjoys an estimation error nominally converging to zero.

3.2 MPC control design

Having designed a weak detector for the augmented system Σ_a , an NMPC stabilizing control law is now synthesized. As discussed, we consider the equilibrium $(x_a^0, 0, y_a^0)$, such that $x_a^0 = \varphi_a(x_a^0, 0, y^0)$ and $y_a^0 = \eta_a(x_a^0, y^0)$.

Let A_a , B_a , and C_a be the matrices of the linearized augmented system around this equilibrium

$$A_a = \left. \frac{\partial \varphi_a}{\partial x_a} \right|_{(x_a^0, 0, y^0)}, \quad B_a = \left. \frac{\partial \varphi_a}{\partial v} \right|_{(x_a^0, 0, y^0)}, \quad C_a = \left. \frac{\partial \eta_a}{\partial x_a} \right|_{(x_a^0, y^0)}. \quad (12)$$

Henceforth it is assumed that the triplet (A_a, B_a, C_a) is stabilizable, observable and does not have transmission at point one in the complex plane. These conditions guarantee the existence of an open neighborhood of y^0 where any reference \tilde{y}^0 can define an equilibrium of the system (De Nicolao et al. (1997), Theorem 1).

According to the MPC approach, at each time-step a Finite Horizon Optimization Control Problem (FHOCP) is solved to spot the control sequence $\mathbf{v} = [v'(0), \dots, v'(N_c - 1)]'$ minimizing a cost function J throughout the horizon $\{1, \dots, N_p\}$. The parameter N_p is called prediction horizon,

and it is the time window throughout which the system is simulated and the cost function is evaluated. In addition, we define the parameter $N_c \leq N_p$, called control horizon. From the end of the control horizon onward, the control v is assumed to be computed by the auxiliary law $v_{lq} = -K_{lq}(x_a - x_a^0)$, where K_{lq} is the LQ $_{\infty}$ gain stabilizing the linearized system, i.e. (A_a, B_a, C_a) , and computed with the state and control weight matrices $Q \succ 0$ and $R \succ 0$.

According to the Receding Horizon technique, once the optimal solution of the FHOCP \mathbf{v}^* is retrieved, the first optimal control move, $v^*(0)$, is applied to the system, i.e. $v(k) = v^*(0)$. At the following time instant, the model is re-initialized in the observed state \hat{x}_a , and the entire procedure is repeated.

The FHOCP is stated as follows

$$\begin{aligned} \min_{\mathbf{v}} J(\mathbf{x}_a, \mathbf{v}) & \quad (13a) \\ \text{s.t. } x_a(0) = \hat{x}_a(k) & \quad (13b) \\ \tilde{\xi}(0) = \xi(k) & \quad (13c) \\ x_a(i+1) = \varphi_a(x_a(i), v(i), y^0) \quad \forall i \in \{1, \dots, N_c - 1\} & \quad (13d) \\ x_a(i+1) = \varphi_a(x_a(i), v_{lq}(i), y^0) \quad \forall i \in \{N_c, \dots, N_p - 1\} & \quad (13e) \\ v_{lq}(i) = -K_{lq}(x_a(i) - x_a^0) \quad \forall i \in \{N_c, \dots, N_p - 1\} & \quad (13f) \\ \tilde{\xi}(i+1) = \tilde{\xi}(i) + y^0 - E_p y_a(i) \quad \forall i \in \{0, \dots, N_p - 1\} & \quad (13g) \\ \tilde{\xi}(i) + v(i) \in [-1, 1] \quad \forall i \in \{0, \dots, N_p - 1\} & \quad (13h) \\ x_a(N_p) \in \Omega_{\omega} & \quad (13i) \end{aligned}$$

where a standard quadratic cost function is adopted

$$\begin{aligned} J(\mathbf{x}_a, \mathbf{v}) = & \sum_{i=0}^{N_c-1} (\|x_a(i) - x_a^0\|_Q^2 + \|v(i)\|_R^2) \\ & + \sum_{i=N_c}^{N_p-1} \|x_a(i) - x_a^0\|_{Q_{lq}}^2 + V_f(x_a(N_p)), \end{aligned} \quad (14)$$

and Q and R are the matrices used to compute K_{lq} . Note that after the control horizon N_c the LQ $_{\infty}$ auxiliary control law (13f) is assumed to be adopted, thus $Q_{lq} = Q + K'_{lq} R K_{lq}$. The term $V_f(x_a(N_p))$ is the terminal cost, defined as a non-quadratic approximation of the LQ $_{\infty}$ cost-to-go, as discussed later this section.

The system model is embedded in the FHOCP by means of (13d) and (13e), where the state x_a is initialized in the values estimated by the weak detector \mathcal{O}_a (13b). Moreover, in addition to the integrator model embedded in Σ_a , a further integrator model with state $\tilde{\xi}$ is included. This integrator is initialized in the true (known) integrator state $\xi(k)$, see (13c), and it is used to ensure that the input sequence v satisfies the input constraints (13h) throughout the prediction horizon. The state $\tilde{\xi}$ is then updated according to the integrator model (13g), where E_p is the selection matrix extracting y from y_a , i.e. $E_p y_a = y$.

Eventually, the terminal constraint (13i) is imposed, by means of which the state at the end of the prediction horizon is forced to lie within the terminal set Ω_{ω} . A suitable design of the terminal cost $V_f(x_a(N_p))$ and of the terminal set Ω_{ω} is paramount to guarantee the closed-loop stability and the recursive feasibility of the NMPC scheme. Herein we consider the approach proposed in Magni et al. (2001a) for the computation of these terminal ingredients.

By definition, the terminal set Ω_{ω} is the set of states that can be steered to the equilibrium $(x_a^0, 0)$ by the stabilizing

LQ $_{\infty}$ control law $v_{lq} = -K_{lq}(x_a - x_a^0)$. In practice, this set is approximated by an inner hyper-ellipsoid in which a Lyapunov function defined by $V_l = (x_a - x_a^0)' \Pi (x_a - x_a^0)$ is decreasing, that is

$$\begin{aligned} \Omega_{\omega} = \{x_a \in \mathbb{R}^{n+p} \mid \|x_a - x_a^0\|_{\Pi}^2 \leq \omega, \\ v_{lq} = -K_{lq}(x_a - x_a^0) \in [-1, 1], \\ \|\varphi_a(x_a, v_{lq}, y^0) - x_a^0\|_{\Pi}^2 - \|x_a - x_a^0\|_{\Pi}^2 + \|x_a - x_a^0\|_{\gamma I}^2 \leq 0\}, \end{aligned} \quad (15)$$

where Π is the solution of the LQ $_{\infty}$ Lyapunov equation $(A_a - B_a K_{lq})' \Pi (A_a - B_a K_{lq}) - \Pi = -\tilde{Q}$, with $\tilde{Q} \succ 0$ and $\gamma > 0$ being design parameters.

Concerning the terminal cost $V_f(x_a(N_p))$, we define it as a finite-step approximation of the cost required to steer the system from $\chi_a(0) = x_a(N_p)$ to the equilibrium x_a^0 by means of the LQ $_{\infty}$ auxiliary control law. Specifically, being $\chi_a(j+1) = \varphi_a(\chi_a(j), -K_{lq}(\chi_a(j) - x_a^0), y^0)$ the state of the system subject to the auxiliary control law, the terminal cost can be approximated by

$$V_f(x_a(N)) = \sum_{j=0}^{N_f-1} \|\chi(j)\|_{Q_{lq}}^2, \quad (16)$$

where it is reminded that $Q_{lq} = Q + K'_{lq} R K_{lq}$, and N_f is the index that determines the truncation of the infinite horizon cost while ensuring an accurate approximation.

Remark 2. Let $v(k) = \kappa^{RH}(\hat{x}_a(k))$ be the Receding Horizon control law. According to Ohno (1978), this control law is Lipschitz continuous, since the constraints of (13) are regular. Thus, provided that a weak detector such as (9) is employed, by Theorem 3 of Magni et al. (2001c) the closed-loop system $x_a^+ = \varphi_a(x_a, \kappa^{RH}(\hat{x}_a), y^0)$ is guaranteed to converge to the target equilibrium $(x_a^0, 0, y_a^0)$, i.e. offset-free tracking is achieved.

4. NUMERICAL RESULTS

In this Section, the pH neutralization process described by Hall and Seborg (1989) is used as a test-bed for evaluating both the modeling capabilities of GRU networks and the performance of the proposed control architecture.

As depicted in Figure 2, the pH process consists of two tanks. Tank 2 is fed by an acid flow rate q_1 , and its output is q_{1e} . Since hydraulic dynamics are faster than the others involved, it is assumed that $q_1 = q_{1e}$. Tank 1, named reactor tank, receives three flows: the acid stream q_{1e} , a buffer flow q_2 and an alkaline flow q_3 . The only control variable is q_3 , while the other two are considered as disturbances. The goal is to set the pH concentration of the output flow rate, q_4 , to the desired level. Hence, the process has only one input, $u = q_3$, and one output, $y = pH(q_4)$. The underlying SISO nonlinear model can be described by the following equations

$$\begin{aligned} \dot{x}(t) = f_1(x(t)) + f_2(x(t))u(t) + f_3(x(t))d(t), \\ 0 = c(x(t), y(t)). \end{aligned} \quad (17)$$

For space reasons, the exact expression of functions f_1 , f_2 , and f_3 is here omitted, and the interested reader is addressed to the extended version (Bonassi et al., 2021). The input of the system is constrained to be in the range $u \in [11.2, 17.2]$. In the following we assume that the only information available for identification and control purposes is the input-output data. This data-set was generated by simulating (17) as discussed below.

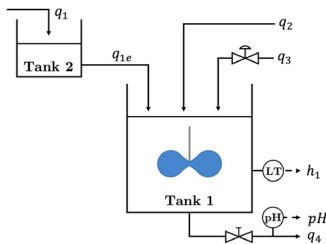


Fig. 2. A representation of the pH neutralization process.

4.1 Identification

To build the data-set, the simulated process is excited by a Multilevel Pseudo-Random Signal (MPRS) with a sampling time of $\tau_s = 10s$, so that at least 30 input-output data points are collected during the settling time of each input step. White noise is added both to the input and output with standard deviations of 10^{-3} and $7.5 \cdot 10^{-3}$, respectively. The resulting training set consists of an experiment of 5060 input-output pairs (u, y) , corresponding to 14 hours of operation. In order to facilitate training and satisfy the assumptions made in Section 2, both the input and output need to be normalized, so that $y \in [-1, 1]$ and $u \in [-1, 1]$.

The GRU network described in Section 2 is used to identify the unknown plant. In light of the recurrent nature of the GRU, the so-called Truncated Back-Propagation Through Time (T-BPTT) is adopted, which consists in splitting the experiment in N_s partially overlapping sequences of length $T_s = 1000$, denoted by $(\mathbf{u}^{\{i\}}, \mathbf{y}^{\{i\}})$. More specifically, the i -th input and output sequence are obtained as follows

$$\mathbf{u}^{\{i\}} = \{u(i\tau + 1), \dots, u(i\tau + L_s)\}, \quad \mathbf{y}^{\{i\}} = \{y(i\tau + 1), \dots, y(i\tau + L_s)\},$$

where $\tau = 5$ is the number of non-overlapping samples.

At each training step, a random subset \mathcal{I} of these sequences is extracted. For each $i \in \mathcal{I}$, a random initial state \bar{x} of the network is generated, and the network (2) is simulated in open loop with the input sequence $\mathbf{u}^{\{i\}}$. This operation artificially augments the training set and causes the network to minimize the T_s -step-ahead prediction error, which is desirable to ensure that the prediction model of the NMPC is accurate throughout the horizon.

The loss function L adopted for training is the Mean-Squared Error (MSE) between the measured output sequence $\mathbf{y}^{\{i\}}$ and the output trajectory predicted by the network, $y(k, \bar{x}, \mathbf{u}^{\{i\}})$, obtained initializing (2) in the random state \bar{x} and feeding it with the input sequence $\mathbf{u}^{\{i\}}$. A washout period of T_w steps is considered, meaning that the simulation error of the first T_w steps is not penalized, to accommodate the network's initial transitory due to the random initialization. Hence, the adopted loss functions is

$$L = \sum_{i \in \mathcal{I}} \left[\frac{1}{T_s - T_w} \sum_{k=T_w}^{T_s} \|y(k, \bar{x}, \mathbf{u}^{\{i\}}) - \mathbf{y}^{\{i\}}(k)\|^2 \right] + \rho(\nu), \quad (18)$$

where $\rho(\nu)$ is a term penalizing the violation of the δ ISS condition (4), which reads as

$$\nu = \|U_\tau\|_\infty \left(\frac{1}{4} \|U_f\|_\infty + \bar{\sigma}_f \right) + \frac{1}{4} \frac{1 + \bar{\phi}_r}{1 - \bar{\sigma}_z} \|U_z\|_\infty - 1. \quad (19)$$

Steering ν to negative values one obtains, according to Theorem 1, a provably δ ISS network. A strictly increasing

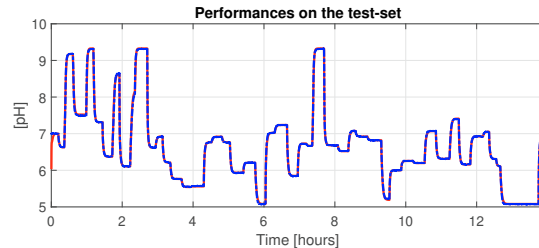


Fig. 3. Performances of the GRU network on the test set: network output (red continuous line) versus the real measured output (blue dashed-dotted line).

$\rho(\nu)$ function, such as a piece-wise linear function, should be hence adopted (Bonassi et al., 2020a). Here we consider

$$\rho(\nu) = \rho^+ \max(0, \nu) + \rho^- \min(0, \nu), \quad (20)$$

with $\rho^+ = 10^{-2}$ and $\rho^- = 10^{-6}$.

The implemented GRU network (2) has $n = 10$ neurons, and it has been trained for 200 epochs, using the ADAM optimization algorithm to minimize the loss function (18).

The training has been carried out in Python 3.7 with the PyTorch 1.5 deep learning library. At the end of the training procedure, the performances of the network have been tested on an independent test set, obtaining the results depicted in Figure 3. The MSE on the test set is approximately 10^{-3} . The accuracy is also quantified by the FIT index [%], defined as

$$\text{FIT} = 100 \left(1 - \frac{\|\mathbf{y}^{\{ts\}} - \mathbf{y}\|}{\|\mathbf{y}^{\{ts\}} - y_{avg}\|} \right)$$

where $\mathbf{y}^{\{ts\}}$ is the test set output, y_{avg} is its average, and \mathbf{y} is the output of the network, initialized in the random state \bar{x} and fed by the test input sequence $\mathbf{u}^{\{ts\}}$. The trained network scores FIT = 97.34%, which confirms yet again the remarkable modeling capability of GRU networks. Moreover, at the end of training, the residual (19) is negative, $\nu = -0.0284$, meaning that the condition for the δ ISS is successfully enforced.

4.2 Control

This section aims to test the ability of the NMPC controller to robustly track constant set-points. The state estimates are provided by a detector \mathcal{O}_a , defined as (9), whose gains are determined by the solution of the optimization problem (11). The control and prediction horizons are selected as $N_c = 20$ and $N_p = 75$, while the adopted weighting matrices are $Q = I$ and $R = 1$. For the computation of the terminal region, we take $\bar{Q} = 10I$ and $\gamma = 0.01$. The terminal cost V_f (16) is adopted, with $N_f = 10^3$. A moving average filter is applied to the reference signal to smooth out the transitions between different references.

In order to test the controller robustness, three different disturbances are also applied on the system. The first disturbance, applied between $t = 0.5h$ and $t = 1.5h$, is a constant additive disturbance on the plant output with amplitude $w_{out} = -0.5$, which represents about 10% of the output range. Then, from $t = 2.5h$ to $t = 3.5h$, the flow q_2 is changed from 0.55 to 0.4. Eventually, a constant additive input disturbance is introduced from $t = 4.5h$ to $t = 5.5h$, with amplitude $w_{in} = 0.6$, corresponding to 10% of the input range.

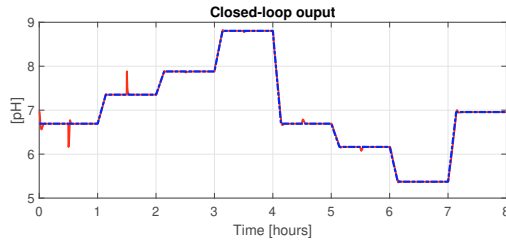


Fig. 4. Closed-loop performances: plant output (red solid line) versus reference (blue dashed-dotted line).

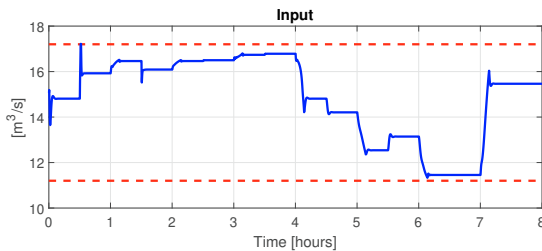


Fig. 5. The denormalized input $u = v + \xi$ (blue line) is bounded within its minimum and maximum values.

The output of the closed-loop system is shown Figure 4. It is apparent that the controller successfully compensates the modeling errors and external unknown disturbances, achieving offset-free tracking, while fulfilling the constraints on the input variable u , see Figure 5. Even though the control system is based on the nominal system, the performance translates well into the real plant, owing to integral action and accurate modeling.

Remark 3. If the set-point changes, the equilibrium associated with the new set-point must be re-computed, as well as (in principle) the auxiliary control law and the terminal set Ω_ω . While this computation may be time consuming for on-line operation, Ω_ω can be computed in advance for any target equilibrium $(x_a^0, 0, y_a^0)$ by applying (15).

5. CONCLUSIONS

In this paper we addressed the task of stable offset-free tracking of constant references for nonlinear stable dynamical systems learned by Gated Recurrent Units (GRU). To this end, a GRU network was firstly trained to identify the unknown plant, ensuring the stability of the model by fulfilling the stability conditions provided in Bonassi et al. (2020a). Then, to ensure offset-free tracking, the model was augmented with an integrator, and a converging state observer was setup for the augmented model. The control strategy provided by Magni et al. (2001c), guaranteeing stable offset-free tracking of constant references, was applied. The proposed approach was tested on the pH neutralization process benchmark, witnessing remarkable performances and good robustness to disturbances.

REFERENCES

- Bonassi, F., da Silva, C.F.O., and Scattolini, R. (2021). Nonlinear MPC for Offset-Free Tracking of systems learned by GRU Neural Networks. *arXiv preprint arXiv:2103.02383*.
- Bonassi, F., Farina, M., and Scattolini, R. (2020a). On the stability properties of gated recurrent units neural networks. *arXiv preprint arXiv:2011.06806*.
- Bonassi, F., Terzi, E., Farina, M., and Scattolini, R. (2020b). LSTM neural networks: Input to state stability and probabilistic safety verification. In *Learning for Dynamics and Control*, 85–94.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- De Nicolao, G., Magni, L., and Scattolini, R. (1997). Stabilizing predictive control of nonlinear arx models. *Automatica*, 33(9), 1691–1697.
- Forgione, M. and Piga, D. (2020). Model structures and fitting criteria for system identification with neural networks. In *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 1–6. IEEE.
- Hall, R.C. and Seborg, D.E. (1989). Modelling and self-tuning control of a multivariable ph neutralization process part i: Modelling and multiloop control. In *1989 American Control Conference*, 1822–1827.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Lanzetti, N. et al. (2019). Recurrent neural network based MPC for process industries. In *2019 18th European Control Conference (ECC)*, 1005–1010. IEEE.
- Magni, L., De Nicolao, G., Magnani, L., and Scattolini, R. (2001a). A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37(9), 1351–1362.
- Magni, L., De Nicolao, G., and Scattolini, R. (2001b). Model predictive control: output feedback and tracking of nonlinear systems. In *Non-linear Predictive Control: theory and practice*, chapter 3, 61–80. Institution of Engineering and Technology.
- Magni, L., De Nicolao, G., and Scattolini, R. (2001c). Output feedback and tracking of nonlinear systems with model predictive control. *Automatica*, 37(10), 1601–1607.
- Morari, M. and Maeder, U. (2012). Nonlinear offset-free model predictive control. *Automatica*, 48(9), 2059–2067.
- Ohno, K. (1978). A new approach to differential dynamic programming for discrete time systems. *IEEE Transactions on Automatic Control*, 23(1), 37–47.
- Pannocchia, G., Gabiccini, M., and Artoni, A. (2015). Offset-free mpc explained: novelties, subtleties, and applications. *IFAC-PapersOnLine*, 48(23), 342–351.
- Stipanović, D.M., Kapetina, M.N., Rapaić, M.R., and Murmann, B. (2020). Stability of gated recurrent unit neural networks: Convex combination formulation approach. *Journal of Optimization Theory and Applications*, 1–16.
- Terzi, E., Bonetti, T., Saccani, D., Farina, M., Fagiano, L., and Scattolini, R. (2020). Learning-based predictive control of the cooling system of a large business centre. *Control Engineering Practice*, 97, 104348.
- Terzi, E., Bonassi, F., Farina, M., and Scattolini, R. (2021). Learning model predictive control with long short-term memory networks. *International Journal of Robust and Nonlinear Control*, 1–20. doi: 10.1002/rnc.5519.
- Wong, W. et al. (2018). Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. *Mathematics*, 6(11), 242.