An Offset-Free Nonlinear MPC scheme for systems learned by Neural NARX models

Fabio Bonassi[1,*] , Jing Xie[1], Marcello Farina[1], and Riccardo Scattolini[1]

Abstract— **This paper deals with the design of nonlinear MPC controllers that provide offset-free setpoint tracking for models described by Neural Nonlinear AutoRegressive eXogenous (NNARX) networks. The NNARX model is identified from input-output data collected from the plant, and can be given a state-space representation with known measurable states made by past input and output variables, so that a state observer is not required. In the training phase, the Incremental Input-to-State Stability (δISS) property can be forced when consistent with the behavior of the plant. The δISS property is then leveraged to augment the model with an explicit integral action on the output tracking error, which allows to achieve offset-free tracking capabilities to the designed control scheme. The proposed control architecture is numerically tested on a water heating system and the achieved results are compared to those scored by another popular offset-free MPC method, showing that the proposed scheme attains remarkable performances even in presence of disturbances acting on the plant.**

Index Terms— **Predictive control for nonlinear systems, Neural Networks, Output Regulation**

I. INTRODUCTION

With the availability of large and informative data sets and increasing computation power, learning-based methods for nonlinear system identification have become popular in the control community [1], [2], see for example the control design algorithms based on set membership ide ntification [3] and Koopman-based system identification [4].

Among the most popular machine learning approaches for control, the ones relying on Recurrent Neural Networks (RNN) have been proven to provide significant results [5]. Among the many model architectures proposed in the literature, it is worth mentioning here Neural NARXs (NNARX) [6], Echo State Networks (ESN) [7], Long Short Term Memory networks (LSTM) [8], and Gated Recurrent Units (GRU) [9]. In particular, it has been shown that these RNNs can be recast as state-space dynamical systems that can be trained to identify unknown systems provided that enough input-output measured data is available [10], [11].

How to guarantee stability properties of these RNNs architectures, in terms of Input to State Stability (ISS) and Incremental Input-to-State stability (δISS), has been recently studied [10], see [12] for LSTMs, [13] for GRUs, [14] for ESNs, and [15] for NNARXs. In these works, sufficient conditions for the ISS and δISS of RNNs are stated as nonlinear inequalities on the networks' parameters. In [10] these stability properties have been shown to be useful to

* Corresponding author
[1] The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133, Milano, Italy. E-mail: `name.surname@polimi.it`

address the interpretability, safety and robustness issues. Furthermore, δISS has proven to be a fundamental tool for the design of provenly stabilizing Model Predictive Control (MPC) laws for several RNN architectures, see [12], [14].

A major limitation of these stabilizing MPC strategies, however, is that their static performances, i.e. their capability of steering the system's output towards a constant setpoint, is tightly related to the magnitude of the plant-model mismatch and to the presence of disturbances that affect the system. In many applications, however, ensuring that the controller can track asymptotically-constant reference signals with zero offset might be a requirement. In this context, several offset-free nonlinear MPC strategies have been proposed in the literature, see [16] for a review on the topic. Among them, one of the most popular is the one described in [17], in which the authors propose to augment the system model with a disturbance model, and then to design an observer to estimate its state. The observed state is then used to synthesize a stabilizing nonlinear MPC law. This approach relies on the possibility to suitably model and estimate such disturbance.

An alternative approach is described in [18], which proposes to augment the system with the output tracking error integrator, and to use a state observer to reconstruct the state of such system. Notably, this scheme can be adopted to solve the tracking problem even for time-varying references that are generated by stable exogenous systems, such as ramps or sinusoids. This solution has been adopted in [19] for the design of an offset-free MPC controller for systems learned by GRU networks.

In this paper, we focus on systems learned by NNARX models, which are quite popular owing to their simple structure and training. Indeed, in NARX models the output at the future time instant is computed as a nonlinear function of past input and output data. In particular, Neural NARX models are those that feature a feed-forward neural network as nonlinear regression function. The advantage of NNARXs is that, since their state boils down to a vector of past input-output data, when these models are operated in closed-loop the state is known [15], which makes the control design procedure significantly easier.

In this context, the goal of this paper is to design a control strategy for NNARX models that guarantees offset-free tracking of constant references, as well as the nominal stability of the closed-loop system. Unlike the aforementioned approaches, the proposed strategy does not rely upon a state observer. Along the lines of [18], [19], we propose to include two elements in the control system: (*i*) an output tracking error integrator, which allows to attain offset-free

tracking capabilities; (*ii*) a derivative action, which ensures that – at steady state – the regulation of the system relies entirely upon the integral action, whereas the goal of the MPC is to improve the dynamic performances and to ensure constraint satisfaction during the transient.

To provide sound guarantees, we show that the δISS of the model, under mild assumptions, ensures that such integral action can be designed to preserve the local asymptotic stability of the closed-loop system. The proposed approach has been tested on a water heating benchmark system, and the achieved closed-loop performances have been compared to those achieved by the strategy proposed in [17]. The simulation results show that, unlike this latter, the proposed approach attains offset-free tracking of constant references even in presence of significant plant perturbations.

The paper is structured as follows. In Section II the NNARX models and their stability properties are presented. In Section III the proposed control architecture is detailed, which is then tested on the water heating benchmark system in Section IV. Finally, conclusions are drawn in Section V.

A. Notation

The following notation is adopted. Given a vector v, we indicate by v' its transpose and by $\|v\|_p$ its p-norm. Moreover, given a matrix Q, we denote $\|v\|_Q^2 = v'Qv$. For compactness, the time instant associated to time-varying vectors is reported as a subscript, e.g. v_k. Sequences of vectors are indicated by bold-face fonts, i.e. $\boldsymbol{v}_k = \{v_0, ..., v_k\}$, and their $\ell_{p,q}$ norm is defined as $\|\boldsymbol{v}_k\|_{p,q} = \big\|[\|v_0\|_p, ..., \|v_k\|_p]\big\|_q$. Notably, $\|\boldsymbol{v}_k\|_{p,\infty} = \max_{t \in \{0,...,k\}} \|v_t\|_p$.

II. NEURAL NARX MODEL

NNARX models [15] are nonlinear, time-invariant, discrete-time models with input u, assumed to lie in a compact set $\mathcal{U} \subseteq \mathcal{R}^m$, and output $y \in R^p$. In this paper, a square system is assumed, i.e., $p = m$. Letting k be the discrete time index, at time k the future output y_{k+1} is computed as a nonlinear regression function η on past N input and output samples:

$$y_{k+1} = \eta(y_k, y_{k-1}, ..., y_{k-N+1}, u_k, u_{k-1}, ...u_{k-N}; \Phi), \quad (1)$$

where Φ indicates the model's parameters. It is easy to rewrite model (1) in state space form by defining, for $i \in \{1, ..., N\}$,

$$z_{i,k} = \begin{bmatrix} y_{k-N+i} \\ u_{k-N-1+i} \end{bmatrix}, \quad (2)$$

and by denoting the state vector $x_k = [z'_{1,k}, ..., z'_{N,k}]' \in \mathbb{R}^n$, the model can be compactly rewritten as

$$\begin{cases} x_{k+1} = Ax_k + B_u u_k + B_x \eta(x_k, u_k; \Phi) \\ y_k = Cx_k \end{cases} \quad (3)$$

where A, B_u, B_x, and C are fixed matrices with known structure and elements equal to zero or one, see [15].

In NNARX models, the regression function η in (1) is a Feed-Forward Neural Network (FFNN), i.e. a static map of M layers of neurons. Each layer is a linear combination of

its inputs, squashed by a suitable nonlinear function, named activation function. A compact formulation of η is

$$\eta(x_k, u_k) = U_0 \eta_M(\eta_{M-1}(...\eta_1(x_k, u_k), u_k), u_k) + b_0 \quad (4)$$

where η_l is the nonlinear relation between l-th and the previous layer, which can be stated as

$$\eta_l(\eta_{l-1}, u_k) = \psi_l(W_l u_k + U_l \eta_{l-1} + b_l), \quad (5)$$

where ψ_l is a Lipschitz-continuous activation function, applied element-wise on its argument, having Lipschitz constant $L_{\psi l}$ and satisfying $\psi_l(0) = 0$. The matrices W_l, U_l and b_l are the weights of the layer, which constitute the network's parameters $\Phi = \{U_0, b_0, \{U_l, W_l, b_l\}_{l=1,...,M}\}$. An example of activation function is the tanh function, see [15]. For compactness, the NNARX model is henceforth denoted as,

$$\Sigma : \begin{cases} x_{k+1} = f(x_k, u_k) \\ y_k = Cx_k \end{cases}, \quad (6)$$

where the dependency on Φ is omitted for compactness.

The weights Φ are learned from the input-output data collected from the system during the so-called training procedure, in which the parameters that best explain the measured data are sought. Generally, one seeks the set of weights minimizing the simulation error, i.e. the open-loop prediction error between the model and the real system. Entering into the details of this procedure is not among the aims of this article: the interested reader is referred to [15].

Under the assumption that the motion of the plant to be identified displays stability properties[1], as discussed in [15] it is possible to include an additional term in the training loss function that allows to learn a provenly ISS and δISS NNARX model. The definition of δISS for a generic state-space nonlinear system, such as (6), is reported below.

Definition 1 (δISS): A system is δISS if there exist functions β of class $\mathcal{KL}$ and γ of class $\mathcal{K}_\infty$ such that, for any pair of initial states $x_{a,0}$ and $x_{b,0}$, and any pair of input sequences $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$, where $u_{a,k} \in \mathcal{U}$ and $u_{b,k} \in \mathcal{U}$, such that

$$\|x_{a,k} - x_{b,k}\|_2 \leq \beta(\|x_{a,0} - x_{b,0}\|_2, k) + \gamma(\|\boldsymbol{u}_{a,k} - \boldsymbol{u}_{b,k}\|_{2,\infty}) \quad (7)$$

for any $k \geq 0$, where $x_{*,k}$ denotes the state trajectory of the system initialized in $x_{*,0}$ and fed by the sequence $\boldsymbol{u}_{*,k}$.

Henceforth, it is assumed that the NNARX is trained according to the prescriptions detailed in [15] in order to ensure its δISS, allowing to verify the following assumption.

Assumption 1: The NNARX model (3) is δISS.

III. CONTROLLER DESIGN

The main goal of this paper is, given the NNARX model (6) of the system, to propose a solution for problem of offset-free tracking of constant references. Specifically, for some constant output reference $\bar{y}$, we want to design an

---

MPC law which guarantees that the output error converges asymptotically to zero, i.e.

$$e_k = \bar{y} - y_k \xrightarrow[k \to \infty]{} 0. \tag{8}$$

A. Linearization

To solve this problem, we will rely upon the linearization of model (6) around an equilibrium point $(\bar{x}, \bar{u}, \bar{y})$ satisfying

$$\begin{cases} \bar{x} = f(\bar{x}, \bar{u}) \\ \bar{y} = C\bar{x} \end{cases} \tag{9}$$

where $\bar{u}$ is assumed to belong to $\mathcal{U}$. Let us first denote by

$$A_\delta = \left. \frac{\partial f(x, u)}{\partial x} \right|_{(\bar{x}, \bar{u})}, \quad B_\delta = \left. \frac{\partial f(x, u)}{\partial u} \right|_{(\bar{x}, \bar{u})}, \tag{10}$$

the matrices of the linearized system around the equilibrium $(\bar{x}, \bar{u}, \bar{y})$. To characterize the stability properties of A_δ, the following result is provided.

Proposition 1: Consider a nonlinear system in the form of (6). Assume that it is δISS in the sense specified by Definition 1, and that function β admits an exponential form, i.e. that there exist constants $\rho > 0$ and $\lambda \in (0, 1)$ such that $\beta(\|x_{a,0} - x_{b,0}\|_2, k) \leq \rho \|x_{a,0} - x_{b,0}\|_2 \lambda^k$. Then, for each equilibrium $(\bar{x}, \bar{u}, \bar{y})$ satisfying (9), the matrix A_δ (10) is Schur stable.

Proof: See the Appendix. ∎

Let us remark that the exponential form of function β is indeed enjoyed by δISS RNNs, see [12], [13], and specifically [15] for NNARXs. Moreover, let us introduce the following Assumption.

Assumption 2: The tuple (A_δ, B_δ, C) is reachable, observable, and does not have invariant zeros at $z = 1$.

Under Assumption 2, in light of Theorem 1 in [20], one can guarantee the existence of an open neighborhood of $\bar{y}$, denoted by $\Gamma(\bar{y}) \subseteq \mathcal{R}^m$, where, for any $\tilde{y} \in \Gamma(\bar{y})$, there exists an equilibrium $(\tilde{x}(\tilde{y}), \tilde{u}(\tilde{y}), \tilde{y})$, where

$$\begin{cases} \tilde{x}(\tilde{y}) = f(\tilde{x}(\tilde{y}), \tilde{u}(\tilde{y})) \\ \tilde{y} = C\tilde{x}(\tilde{y}) \end{cases}. \tag{11}$$

This local result allows to conclude that it is possible to move the output reference signal in a neighborhood of the output equilibrium $\bar{y}$ and still guarantee that a feasible solution to the tracking problem exists.

B. The control architecture

Once the conditions for the existence of a solution to the output tracking problem have been established, we are in the position to describe the main elements of the adopted control architecture, depicted in Figure 1, listed below

   i. The system is augmented with the integral of the output tracking error $e_k = \bar{y} - y_k$. Indeed, in light of the Internal Model Principle [21], such integral action guarantees robust asymptotic zero-error regulation for constant reference signals, i.e. $e_k \xrightarrow[k \to \infty]{} 0$, and plant's parametric uncertainties, provided that the closed-loop stability guarantees are maintained.
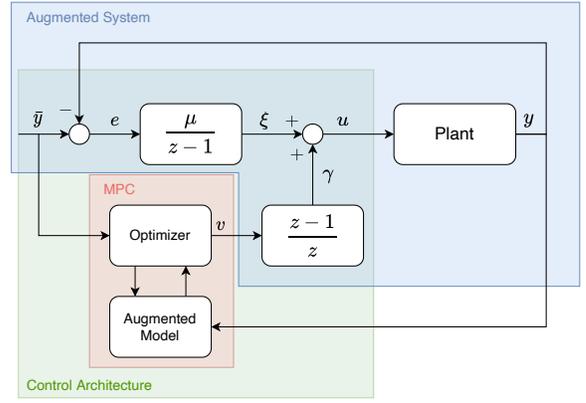
Fig. 1. Schematic of the proposed control architecture

   ii. The model is augmented with a derivative action on MPC's control variable v. This guarantees that, at steady state, the MPC contribution is null and the control variable entirely relies on the integral action. This approach, later detailed, is useful in the definition of suitable terminal constraints to be used in the formulation of the stabilizing MPC algorithm. The aim of the MPC regulator is that of performance enhancement and constraint handling during transients.

As clear from Figure 1, the control action u is composed of two terms

$$u_k = \xi_k + \gamma_k, \tag{12}$$

where $\xi_k \in \mathbb{R}^m$ and $\gamma_k \in \mathbb{R}^m$ are the integral and derivative actions, respectively. More specifically, the integral action is ruled by

$$\xi_{k+1} = \xi_k + \mu(\bar{y} - Cx_k), \tag{13}$$

where μ denotes the gain of the integral action. Also, the derivative action γ_k is defined as

$$\begin{cases} \theta_{k+1} = v_k \\ \gamma_k = v_k - \theta_k \end{cases}. \tag{14}$$

Thus, the augmented system is obtained combining (6), (12), (13), and (14), and it reads as

$$\begin{cases} x_{k+1} = f(x_k, u_k) \\ \xi_{k+1} = \xi_k + \mu(\bar{y} - Cx_k) \\ \theta_{k+1} = v_k \\ \gamma_k = v_k - \theta_k \\ u_k = \xi_k + \gamma_k \\ y_k = Cx_k \end{cases}, \tag{15}$$

which will henceforth be compactly denoted as

$$\Sigma_a : \begin{cases} \chi_{k+1} = f_a(\chi_k, v_k, \bar{y}) \\ \zeta_k = g_a(\chi_k) \end{cases}, \tag{16}$$

where $\chi_k = [x_k', \xi_k', \theta_k']'$ denotes the state of the augmented system and $\zeta_k = [y_k', u_k']'$ its output.

The first step of the design procedure consists of tuning the gain μ, following the guidelines of [22], such that

the enlarged system (16) displays stability properties. The following proposition can be stated.

Corollary 1: Assume that A_δ is Schur stable, and that Assumption 2 holds. Then, there exists $\check{\mu} > 0$ such that, for any $\tilde{\mu} \in (0, \check{\mu})$, the integrator gain

$$\mu = \tilde{\mu}\big[C_\delta(I - A_\delta)^{-1}B_\delta\big]^{-1} \tag{17}$$

makes the enlarged system (16), linearized around $(\bar{\chi}, \bar{v}, \bar{\zeta})$, asymptotically stable, where $\bar{\chi} = \big[\bar{x}', \bar{\xi}', \bar{\theta}'\big]' = \big[\bar{x}', \bar{u}', \bar{v}'\big]'$, $\bar{\zeta} = [\bar{y}', \bar{u}']'$, and $\bar{v}$ is any constant value.

Proof: In light of Proposition 1, the matrix A_δ is Schur stable. Then, thanks to Assumption 2, the results shown in [22] can be applied to prove the corollary. ∎

C. MPC design

Having defined the augmented system model Σ_a, a stabilizing nonlinear MPC law can be designed. Letting $\bar{\chi}$ and $\bar{\zeta}$ be the target state and output introduced in Corollary 1, the stabilizing MPC can be stated as follows[2].

$$\min_{v_{0|k}, \ldots, v_{N_p-1|k}} \sum_{i=0}^{N_p} \left[\big\|\chi_{i|k} - \bar{\chi}\big\|_Q^2 + \big\|\zeta_{i|k} - \bar{\zeta}\big\|_R^2 \right] \tag{18a}$$

$$\text{s.t.} \quad \forall i \in \{0, \ldots, N_p - 1\}$$

$$\chi_{0|k} = \chi_k \tag{18b}$$

$$\chi_{i+1|k} = f_a(\chi_{i|k}, v_{i|k}, \bar{y}) \tag{18c}$$

$$\zeta_{i|k} = g_a(\chi_{i|k}) \tag{18d}$$

$$E_u \zeta_{i|k} \in \mathcal{U} \tag{18e}$$

$$\chi_{N_p|k} = \bar{\chi} \tag{18f}$$

The adopted cost function (18a) penalizes the deviation of the augmented system's state and output from their value at equilibrium. Note that the target equilibrium for the integrator state ξ is $\bar{u}$, as the integral action is assumed to provide the input's equilibrium at steady state. The equilibrium value of the derivator state, $\bar{v}$, is arbitrary but constant, so that $\bar{\gamma} = 0$, i.e. the contribution of the derivative action converges to zero. The weight matrices are defined as $Q = \mathrm{diag}(Q_x, Q_\xi, Q_\theta)$ and $R = \mathrm{diag}(R_e, R_u)$, where $\mathrm{diag}(\cdot)$ indicates the block-diagonal operator. The weights R_e and R_u penalize the output error and control effort, respectively, while Q penalizes the distance of the augmented state from its equilibrium value. Note that the deviation of θ from its arbitrary equilibrium is only penalized for numerical reasons. It is hence advisable to select $Q_\theta \ll Q_x, Q_\xi$.

The augmented model Σ_a is used as predictive model, see (18c) and (18d), and it is initialized in the known state χ_k, see (18b). The inputs saturation constraints are also embedded via constraint (18e), where E_u is a selection matrix that extracts $u_{i|k}$ out of $\zeta_{i|k} = [y_{i|k}', u_{i|k}']'$. Lastly, as customary in MPC, the terminal constraint (18f) is imposed.

According to the Receding Horizon principle, at time k the optimization problem (18) is solved, retrieving the optimal

[2]To reduce the computational burden of the optimization problem, a control horizon smaller than the prediction horizon could be adopted [23].
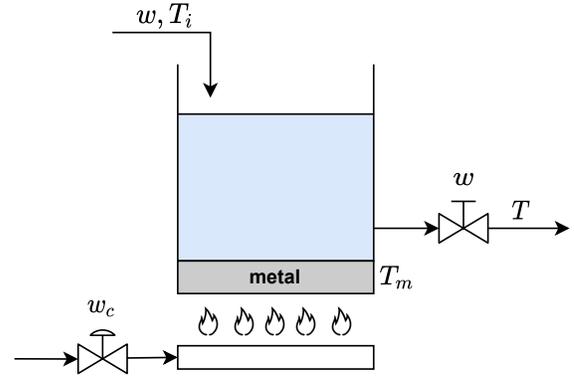
Fig. 2.   Water-heating system illustration

control sequence $v_{0|k}^*, \ldots, v_{N_p-1|k}^*$, and only the first control move, i.e. $v_{0|k}^*$, is applied. At the successive time step the procedure is repeated, based on the measured state χ_{k+1}.

Remark 1: The MPC law formulated in (18) is a standard MPC with terminal constraint. Hence, its nominal recursive feasibility and closed-loop stability can be guaranteed [23].

IV. NUMERICAL EXAMPLE

A. Benchmark system description

The proposed control architecture has been tested on the water-heating benchmark system depicted in Figure 2. The objective of this system is to control the temperature of the water in a reservoir so as to provide the users with the required flow of water at the desired temperature. Specifically, the water is heated through a metal plate placed under the tank, which is heated by means of a gas burner.

The water demand w, expressed in kg/s, represent a disturbance. For simplicity, it is assumed that the water flow rate at the inlet matches the demand, so that the level dynamics are neglected. We indicate by T_i the temperature of the water at the inlet, and by T the temperature of the water served to the users. Both temperatures are expressed in K, and the water temperature is assumed to be uniform throughout the tank. The water is heated by the metal plate, having temperature T_m, which is radiated by the flames resulting from the combustion of the gas, whose flow rate is denoted by w_c. Assuming the absence of heat loss, and that the flame heat is exchanged only via radiation, the following model of the system can hence be formulated:

$$\mathcal{P}: \begin{cases} \dot{T} = \dfrac{1}{\rho_w A_t z_w} \left[w\left(T_i - T\right) + \dfrac{k_{lm} A_t}{c_w}\left(T_m - T\right) \right] \\ \dot{T}_m = \dfrac{1}{M_m c_m}\left[-k_{lm} A_t\left(T_m - T\right) + \sigma k_f w_c\left(T_f^4 - T_m^4\right) \right] \end{cases}. \tag{19}$$

This model has one controllable input $u = [w_c]$, one output $y_p = [T]$, and two states $x_p = [T, T_m]'$. Moreover, system (19) is affected by two disturbances, $d_p = [w, T_i]'$, whose nominal values are reported, alongside the other parameters of the model, in Table I. The gas flow rate w_c is also subject

Parameter	Description	Value	Units
A_t	Tank's cross-section	$\frac{\pi}{4}$	m^2
ρ_w	Water's density	997.8	$\frac{kg}{m^3}$
c_w	Water's specific heat	4180.0	$\frac{J}{kg \cdot K}$
M_m	Metal plate's mass	617.32	kg
c_m	Metal's specific heat	481.0	$\frac{J}{kg \cdot K}$
σ	Radiation coefficient	5.67×10^{-8}	$\frac{W}{m^2 \cdot K^4}$
k_{lm}	Heat exchange coefficient	3326.4	$\frac{kg}{s^3 \cdot K}$
T_f	Flame's temperature	1200	K
k_f	Heat exchange coefficient	8.0	$\frac{m^2 \cdot s}{kg}$
z_w	Water level	2.0	m
$\bar{w}$	Nominal water demand	1.0	$\frac{kg}{s}$
$\bar{T}_i$	Nominal inlet water temperature	298	K

to saturation, i.e.

$$w_c \in [0.05, 0.18]. \qquad (20)$$

A simulator of the benchmark system has been implemented in Simulink, so as to collect the training data and to test the proposed control architecture.

B. NNARX model training

In order to collect the data used to train the NNARX model of the plant (19), the simulator has been forced with a Multilevel Pseudo-Random Signal (MPRS), so as to properly excite the system. The input-output data has been recorded with sampling time $\tau_s = 120\,s$, for a total of 2500 time steps. According to the Truncated Back-Propagation Through Time (TBPTT) principle, $N_t = 120$ random subsequences of length $T_s = 400$, denoted by $(\boldsymbol{u}_{T_s}^{\{i\}}, \boldsymbol{y}_{p,T_s}^{\{i\}})$, with $i \in \mathcal{I}_t = \{1, ..., N_t\}$, have been extracted from the experiment data. Other two shorter experiments have been performed, from which $N_v = 30$ and $N_f = 1$ subsequences have been extracted as validation and independent test datasets, respectively. These two datasets are denoted by the set of indices $\mathcal{I}_v = \{N_t + 1, ..., N_t + N_v\}$ and $\mathcal{I}_f = \{N_t + N_v + 1, ..., N_s\}$, respectively, where $N_s = N_t + N_v + N_f$.

The training procedure has been conducted using PyTorch 1.9. The adopted NNARX model features a single-layer ($M = 1$) FFNN regression function with 30 neurons, and the past $N = 5$ input-output data has been used as regressors. Following the guidelines of [10], [15], the model has been trained by minimizing the simulation Mean Square Error (MSE) over the training set $\mathcal{I}_t$, i.e. the MSE between the NNARX open-loop prediction (given the input sequence $\boldsymbol{u}_{T_s}^{\{i\}}$) and the measured output sequence $\boldsymbol{y}_{p,T_s}^{\{i\}}$. The δISS property of the NNARX model has been enforced during training by including a suitable regularization term in the loss function [15], so that Assumption 1 is satisfied. The NNARX model has been trained for 1288 epochs, when its modeling performances on the validation set stopped improving.

Eventually, the model performances have been assessed on the independent test set. In Figure 3 the test input sequence, $\boldsymbol{u}_{T_s}^{\{\iota\}}$, with $\iota \in \mathcal{I}_f$, is depicted, while in Figure 4 the corresponding open-loop simulation of the NNARX model is
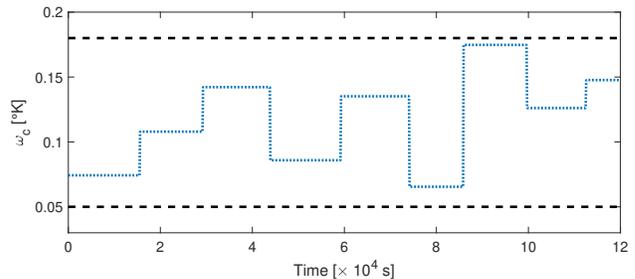
Fig. 3. Input sequence used for testing the performances of the trained NNARX model.
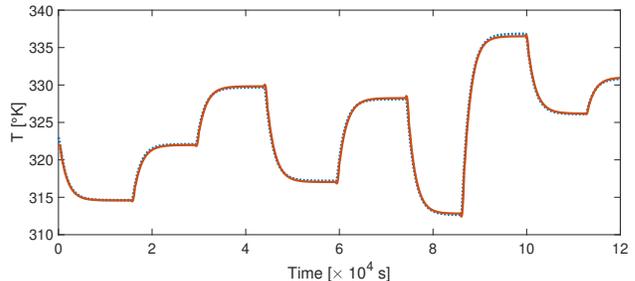
Fig. 4. Modeling performances of the trained NNARX model: open-loop prediction (red line) versus ground truth (blue dotted line).

compared to the ground truth $\boldsymbol{y}_{p,T_s}^{\{\iota\}}$. The performance have been quantified using the FIT [%] index, defined as

$$\text{FIT} = 100 \left(1 - \frac{\sum_{k=0}^{T_s} \|y_k^{\{\iota\}} - y_{p,k}^{\{\iota\}}\|_2}{\sum_{k=0}^{T_s} \|y_{p,k}^{\{\iota\}} - y_{avg}\|_2} \right), \qquad (21)$$

where $y_k^{\{\iota\}}$ indicates the output of the NNARX model (6) fed by the input sequence $\boldsymbol{u}_k^{\{\iota\}}$ and initialized in a random initial state, and y_{avg} is the average of $y_{p,k}^{\{\iota\}}$ over time. The FIT scored by the trained model is 92.8%, which indicates fair modeling performances of the NNARX model.

Having trained a δISS NNARX model of the system, the control architecture proposed in Section III can be tested.

C. Control synthesis

The proposed control architecture has been implemented, with the primary goal of tracking piecewise-constant water temperature references, while asymptotically rejecting possible (unmeasured) disturbances associated to variations of the water demand w or the water temperature at the inlet T_i. To assess the offset-free tracking capabilities and the robustness of the proposed architecture, the temperature reference depicted in Figure 5 and the disturbances illustrated in Figure 6 have been considered.

It should be noted that, since the output reference is piecewise constant, at every change of the setpoint $\bar{y}$, the nominal equilibrium triplet $(\bar{x}, \bar{u}, \bar{y})$ of the corresponding nominal equilibrium of the augmented system $(\bar{\chi}, \bar{v}, \bar{\zeta})$ needs to be computed by solving (9). Moreover, if the new setpoint is not in the neighborhood of the previous equilibrium, Assumption 2 should also be verified, and the integral action's gain μ should be recomputed according to Corollary 1. The
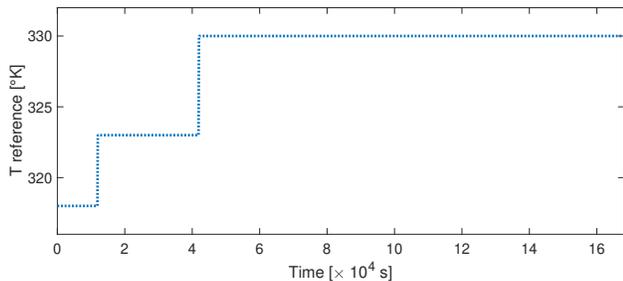
Fig. 5.   Piecewise-constant output reference trajectory.
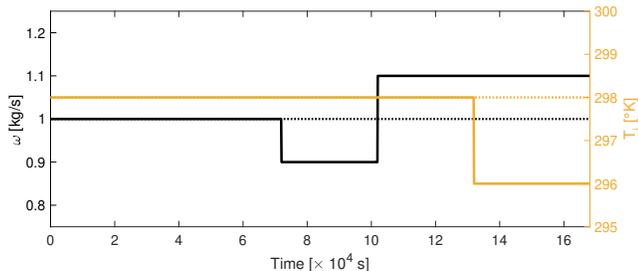
Fig. 6.   Disturbance applied to the system. Water demand w (black continuous line) compared to its nominal value (black dotted line), and inlet water temperature T_i (yellow continuous line) compared to its nominal value (yellow dotted line).

Fig. 7.   Closed-loop output tracking performances of the proposed approach (red continuous line) compared to that of the DEB-MPC (green dashed-dotted line). The setpoint is represented by the blue dotted line.

Fig. 8.   Control action of the proposed approach (red continuous line) compared to that of the DEB-MPC (green dashed-dotted line).

prediction horizon of the adopted MPC law is $N_p = 50$, while the weights are chosen as $R_e = 10$, $R_u = 0.1$, $Q_x = \text{diag}(R, R, R, R, R)$, $Q_\xi = 1$, and $Q_\theta = 10^{-5}$.

Concerning the tuning of the integral action gain μ, Corollary 1 has been verified numerically. For the three setpoints considered in Figure 5, any gain $\mu \in (0, 0.251)$ results in a locally asymptotically stable augmented system. Specifically, $\mu = 0.14$ has been chosen.

Alternative approach for comparison

In order to evaluate the performances of the proposed control architecture, the popular offset-free MPC strategy described in [17] has also been implemented. In brief, this control strategy, henceforth named Disturbance Estimation Based MPC (DEB-MPC), requires to augment the NNARX model with a fictitious matched disturbance on the input, for the estimation of which a Moving Horizon Estimator (MHE) is designed. Then, a standard state-feedback nonlinear MPC law is designed to stabilize the augmented model, featuring a prediction horizon $N_p = 50$, and weights in line with those used for the proposed control architecture.

Closed-loop results

The closed-loop output tracking performances achieved by the proposed approach are compared to those of DEB-MPC in Figure 7. It is apparent that, while initially the DEB-MPC scheme is able to compensate the plant-model mismatch thanks to a reliable estimate of the fictitious matched disturbance, after the instant $t = 7 \cdot 10^4 s$ – when disturbances occur, see Figure 6 – the tracking performances of such control scheme are lost. In contrast, the proposed control architecture is able to attain zero tracking error, even in the presence of the severe disturbances that affect the system. In Figure 8, the control variable requested by the two schemes is compared. In both schemes the input constraints
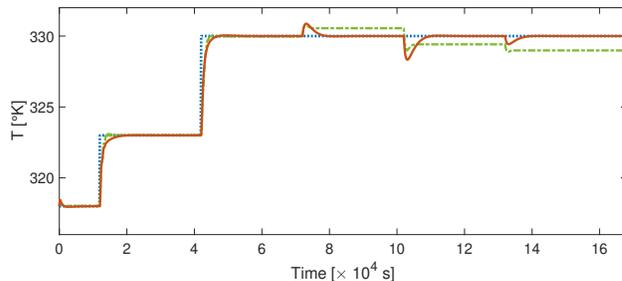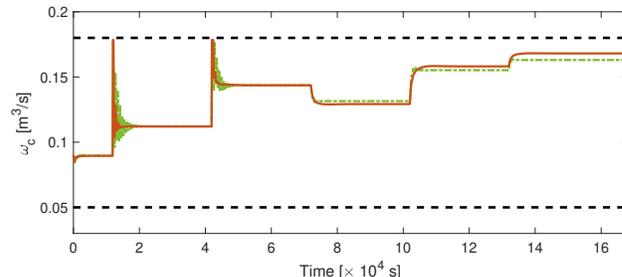
are fulfilled, although it can be observed that the control action issued by DEB-MPC is less moderate, mainly due to the transients of the disturbance estimator. It should be noted that in DEB-MPC, the choice of the disturbance model is crucial to achieve satisfactory performance. Recent works proposing alternative disturbance estimation-based strategies, see [24], will thus be object of future investigations.

V. CONCLUSIONS

In this paper, a nonlinear Model Predictive Control (MPC) strategy is proposed to achieve offset-free tracking of constant references for system learned by Neural NARX (NNARX) models. To this end, we proposed to augment the NNARX model with the integrator of the output tracking error and with a derivative action, and then to design a stabilizing MPC law for such augmented model. The proposed control scheme attains nominal closed-loop stability and offset-free tracking capabilities. The control law was tested on a water heating benchmark system, demonstrating satisfactory closed-loop performance and a good degree of robustness to the disturbances affecting the plant.

REFERENCES

[1] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[2] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[3] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini, "Learning-based predictive control for linear systems: A unitary approach," *Automatica*, vol. 108, p. 108473, 2019.

[4] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[5] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020.

[6] A. Levin and K. Narendra, "Control of nonlinear dynamical systems using neural networks: controllability and stabilization," *IEEE Transactions on Neural Networks*, vol. 4, no. 2, pp. 192–206, 1993.

[7] H. Jaeger, "Echo state network," *Scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

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

[11] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent neural networks for short-term load forecasting: an overview and comparative analysis*. Springer, 2017.

[12] E. Terzi, F. Bonassi, M. Farina, and R. Scattolini, "Learning model predictive control with long short-term memory networks," *International Journal of Robust and Nonlinear Control*, 2021.

[13] F. Bonassi, M. Farina, and R. Scattolini, "On the stability properties of gated recurrent units neural networks," *System & Control Letters*, vol. 157, p. 105049, 2021.

[14] L. B. Armenio, E. Terzi, M. Farina, and R. Scattolini, "Model predictive control design for dynamical systems learned by echo state networks," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 1044–1049, 2019.

[15] F. Bonassi, M. Farina, and R. Scattolini, "Stability of discrete-time feed-forward neural networks in NARX configuration," in *19th IFAC Symposium on System Identification (SYSID 2021)*, 2021.

[16] G. Pannocchia, "Offset-free tracking mpc: A tutorial review and comparison of different formulations," in *2015 European control conference (ECC)*. IEEE, 2015, pp. 527–532.

[17] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 2012.

[18] L. Magni, G. De Nicolao, and R. Scattolini, "Output feedback and tracking of nonlinear systems with model predictive control," *Automatica*, vol. 37, no. 10, pp. 1601–1607, 2001.

[19] F. Bonassi, C. F. Oliveira da Silva, and R. Scattolini, "Nonlinear MPC for Offset-Free Tracking of systems learned by GRU Neural Networks," in *3rd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON 2021)*, 2021.

[20] G. De Nicolao, L. Magni, and R. Scattolini, "Stabilizing predictive control of nonlinear ARX models," *Automatica*, vol. 33, no. 9, pp. 1691–1697, 1997.

[21] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[22] R. Scattolini and N. Schiavoni, "A parameter optimization approach to the design of structurally constrained regulators for discrete-time systems," *International Journal of Control*, vol. 42, no. 1, pp. 177–192, 1985.

[23] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, 2017, vol. 2.

[24] P. Tatjewski and M. Ławryńczuk, "Algorithms with state estimation in linear and nonlinear model predictive control," *Computers & Chemical Engineering*, vol. 143, p. 107065, 2020.

[25] H. K. Khalil, *Nonlinear systems; 3rd ed.* Prentice-Hall, 2002.

[26] N. Bof, R. Carli, and L. Schenato, "Lyapunov theory for discrete time systems," *arXiv preprint arXiv:1809.05289*, 2018.

APPENDIX

A. Proof of Proposition 1

Let δx_k and δu_k be the displacement from the equilibrium point $(\bar{x}, \bar{u})$, i.e. $x_k = \bar{x} + \delta x_k$ and $u_k = \bar{u} + \delta u_k$. The nonlinear system (6) can be rewritten as

$$\delta x_{k+1} + \bar{x} = f(\bar{x} + \delta x_k, \bar{u} + \delta u_k). \tag{22}$$

Since the goal is to analyze the asymptotic stability of the linearized system, for simplicity it is assumed that $\delta u_k = 0$. It is worth noticing, however, that this proof could be easily extended to consider $\delta u_k \neq 0$, at the price of more involved computations. Under this simplification, (23) reads

$$\delta x_{k+1} + \bar{x} = f(\bar{x} + \delta x_k, \bar{u}). \tag{23}$$

System (23) can be recast as its linearization plus the linerization error ε

$$\delta x_{k+1} = A_\delta \delta x_k + \varepsilon(\delta x_k), \tag{24}$$

where

$$A_\delta = \left. \frac{\partial f}{\partial x_k} \right|_{\bar{x}, \bar{u}}. \tag{25}$$

The goal is to show that the linear system

$$\delta x_{k+1} = A_\delta \delta x_k \tag{26}$$

is asymptotically stable. Along the lines of [25], the linearization error is first bounded as follows.

Consider the i-th state component, with $i \in \{1, ..., n\}$. In light of the Mean Value Theorem there exists $\tilde{x}$ between $\bar{x}$ and $\bar{x} + \delta x_k$ such that

$$f_i(\bar{x} + \delta x_k, \bar{u}) - f_i(\bar{x}, \bar{u}) = \left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\tilde{x}, \bar{u}} \delta x_k$$
$$= \left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\bar{x}, \bar{u}} \delta x_k + \left[\left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\tilde{x}, \bar{u}} - \left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\bar{x}, \bar{u}} \right] \delta x_k$$
$$= A_{\delta i} \delta x_k + \tilde{\varepsilon}_i(\delta x_k) \delta x_k \tag{27}$$

Under the customary assumption that the gradient of $f(x, \bar{u})$ with respect to x is Lipschitz continuous with Lipschitz constant L_1, it holds that

$$\|\tilde{\varepsilon}_i(\delta x_k)\|_2^2 \leq \left\| \left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\tilde{x}, \bar{u}} - \left. \frac{\partial f_i(x, \bar{u})}{\partial x} \right|_{\bar{x}, \bar{u}} \right\|_2^2 \tag{28}$$
$$\leq L_1^2 \|\tilde{x} - \bar{x}\|_2^2 \leq L_1^2 \|\delta x_k\|_2^2.$$

Hence, being $\varepsilon(\delta x_k) = \tilde{\varepsilon}(\delta x_k) \delta x_k$, the linearization error can be bounded as

$$\|\varepsilon(\delta x_k)\|_2 \leq \|\delta x_k\|_2 \sqrt{\sum_{i=1}^n \|\tilde{\varepsilon}_i(\delta x_k)\|_2^2 \|\delta x_k\|_2^2} \tag{29}$$
$$\leq L_\varepsilon \|\delta x_k\|_2^2,$$

where $L_\varepsilon = L_1 \sqrt{n}$.

At this stage, let us recall that the δISS property implies the Global Asymptotic Stability (GAS) of any equilibrium. Indeed, recalling that $\delta u_k = 0$, from (7) it follows that

$$\|x_k - \bar{x}\|_2 \leq \beta(\|x_0 - \bar{x}\|_2, k).$$

Moreover, in light of the assumption on β, the exponential GAS property of any equilixbrium can be shown, since

$$\|\delta x_k\|_2 \leq \rho\|\delta x_0\|_2\lambda^k. \tag{30}$$

This allows to invoke Theorem 5.8 of [26] which, under the assumption of exponential GAS, guarantees the existence of a quadratic Lyapunov function $V(\delta x)$ for the nonlinear system (23). That is, there exist positive constants c_1, c_2, c_3, c_4, such that

$$c_1\|\delta x_k\|_2^2 \leq V(\delta x_k) \leq c_2\|\delta x_k\|_2^2, \tag{31a}$$

$$V(A_\delta\delta x_k + \varepsilon(\delta x_k)) - V(\delta x_k) \leq -c_3\|\delta x_k\|_2^2 \tag{31b}$$

$$\left\|\frac{\partial V(A_\delta\delta x_k + \varepsilon(\delta x_k))}{\partial \varepsilon}\right\|_2 \leq c_4\|\delta x_k\|_2. \tag{31c}$$

The goal is to show that $V(\delta x_k)$ is also a Lyapunov function for the linear system (26). To this end, let us add and subtract $V(A_\delta\delta x_k)$ from the left-hand side of (31b), leading to

$$V(A_\delta\delta x_k) - V(\delta x_k) + \left[V(A_\delta\delta x_k + \varepsilon(\delta x_k)) - V(A_\delta\delta x_k)\right]$$
$$\leq -c_3\|\delta x_k\|_2^2 \tag{32}$$

In light of (31c) and (29), $V(A_\delta\delta x_k + \varepsilon(\delta x_k)) - V(A_\delta\delta x_k)$ can be bounded as

$$\left\|V(A_\delta\delta x_k + \varepsilon(\delta x_k)) - V(A_\delta\delta x_k)\right\|_2$$
$$\leq \left\|\frac{\partial V(A_\delta\delta x_k + \varepsilon(\delta x_k))}{\partial \varepsilon}\right\|_2 \|\delta x_k\|_2 \tag{33}$$
$$\leq c_4 L_\varepsilon\|\delta x_k\|_2^3.$$

Owing to the bound (33) and to the exponential GAS (30), recalling that $\lambda \in (0, 1)$, from (32) it holds that

$$V(A_\delta\delta x_k) - V(\delta x_k) \leq -c_3\|\delta x_k\|_2^2 - c_4 L_\varepsilon\|\delta x_k\|_2^3$$
$$\leq -c_3\|\delta x_k\|_2^2 - \rho c_4 L_\varepsilon\|\delta x_0\|_2\|\delta x_k\|_2^2$$
$$\leq -\left(c_3 - \rho c_4 L_\varepsilon\|\delta x_0\|_2\right)\|\delta x_k\|_2^2. \tag{34}$$

Hence, there exist constants $c_5 > 0$ and $r_0 > 0$ such that, $\forall \delta x_0 \in \{\delta x_0 : \|\delta x_0\|_2 \leq r_0\}$,

$$V(A_\delta\delta x_k) - V(\delta x_k) \leq -c_5\|\delta x_k\|_2^2.$$

Hence, the asymptotic stability of the linear system (26) is proven by using $V(\delta x_k)$ as Lyapunov function, which implies that A_δ is Schur stable. ∎