2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

NAW-NET: neural anti-windup control for saturated nonlinear systems

Valentina Breschi*, Daniele Masti†, Simone Formentin*, Alberto Bemporad†

Abstract— **One major issue in industrial control applications is how to handle input constraints due to physical limitations of the actuators. Such constraints introduce nonlinearities in the feedback loop, that are commonly tackled through anti-windup or model predictive control schemes. Since these techniques might result into poor closed-loop performance when an accurate model of the plant is not available, in this work we present an off-line strategy to learn a neural anti-windup control scheme (NAW-NET) from a set of open-loop data collected from an unknown nonlinear process. The proposed scheme, that includes a feedback controller and an anti-windup compensator, is trained to reproduce the desired closed-loop behavior while simultaneously accounting for actuator limits. The effectiveness of the approach is illustrated on a simulation example, involving the control of a Hammerstein-Wiener process with saturated inputs.**

I. Introduction

Due to the increasing complexity of control systems and thanks to the ever-growing availability of large sets of data, classical control techniques are now commonly paired with data-driven strategies to either compute/update a model of the plant to be controlled or to directly tune the parameters of the controller. Indeed, several *System Identification* (SI) techniques have been proposed over the years to learn approximate laws that describe the behavior of otherwise unknown processes [1]. However, these approaches usually aim at obtaining models that achieve the best open-loop simulation accuracy, while control techniques can attain satisfactory performance even with a crude approximation of the real plant dynamics. This creates quite a large discrepancy between the target of most identification techniques and the actual requirements of control design problems, often creating an undue burden during the modeling phase. Moreover, the quest for models minimizing the simulation error usually results in large-dimensional models, which in turn requires the use of model-reduction techniques for model-based control synthesis.

As an alternative to the classical two-stage design paradigm, where a model of the plant is first identified and then a model-based controller is designed, *direct data-driven control* strategies have been proposed. In this case, data collected from the plant are directly used to synthesize a controller, without the involvement of an explicit model of the system. These methods range from data-driven optimal control approaches such as *Reinforcement Learning* (RL) [2]–[5] to schemes exploiting the *model reference* paradigm, such as the *Virtual Reference Feedback Tuning* (VRFT) approach [6], [7]. The latter approaches have the advantage of being one-shot techniques, in that they do not require additional data to refine the control law, although they provide little to no flexibility to ease the integration of constraint handling methods. Since in practice the authority of actuators is always limited, the use of these techniques in industrial applications is restricted to cases in which these components operate far from their operational bounds. On the other hand, one should account for such limits when high control performance is sought.

Traditionally, input saturations are managed either by using predictive control strategies [8] or by pairing a controller with an *anti-windup compensator*, which aims at preserving performance within actuator bounds and at guaranteeing the asymptotic recovery of unconstrained behavior after saturation occurs [9]–[11]. As an alternative to classical model-based strategies, some attempts have been recently made to incorporate constraint-handling methods within the data-driven control framework. For example, in [12] a reference governor is used on top of a direct data-driven controller for *Linear Parameter Varying* (LPV) systems [13] to impose constraint satisfaction. Despite its effectiveness, the above approach involves distinct design phases, so that one has to renounce to the advantages of a one-shot learning procedure. This limitation is overcome in [14], where the reference governor and a data-driven controller are jointly designed. However, both approaches still require the solution an optimization problem in real-time for each control step.

Inspired by *anti-windup* architectures, in this work we propose a *one-shot* learning scheme to design a *Neural NETwork for Anti-Windup control* (NAW-NET) directly from data. Although preliminary, the proposed approach already allows us to retrieve both a controller and an anti-windup compensator for an *unknown* nonlinear plant from a set of input/output data, that can handle actuator constraints effectively. The learning procedure is fully carried out off-line. Due to the nonlinear nature of the plant to be controlled, we parameterize the controller and the compensator through two different *Artificial Neural Networks* (ANNs), due to their flexibility and excellent function approximation properties [15].

The paper is organized as follows. Section II formally states the control problem of interest. Section III is devoted to the formulation of the NAW-NET learning task, with

*Valentina Breschi and Simone Formentin are with Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy {valentina.breschi, simone.formentin}@polimi.it

†Daniele Masti and Alberto Bemporad are with IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy {daniele.masti, alberto.bemporad}@imtlucca.it

The first two authors equally contributed to the work.

This project was partially supported by the Italian Ministry of University and Research under the PRIN'17 project "Data-driven learning of constrained control systems", contract no. 2017J89ARP.

the structure proposed in this paper for the controller and the anti-windup compensator are described in Section III-A. A strategy to handle non-informative data is proposed in Section III-C. Simulation results are shown in Section IV, while conclusions and possible directions for future work are discussed in Section V.

II. Setting and Goal

Let $\mathcal{P}$ be a discrete-time nonlinear *Single-Input Single-Output* (SISO) plant, whose dynamics is

$$\mathcal{P}: \quad \begin{cases} x_{k+1} = f(x_k, u_k), \\ y_k^{\text{o}} = h(x_k), \end{cases} \tag{1}$$

where $u_k \in \mathbb{R}$, $x_k \in \mathbb{R}^{n_x}$ and $y_k^{\text{o}} \in \mathbb{R}$ are the commanded input, the state and the noiseless output of the plant at the instant $k \in \mathbb{N}$, respectively, and $f: \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$, $h: \mathbb{R}^{n_x} \to \mathbb{R}$ are *unknown* maps in the input and the state of $\mathcal{P}$. Suppose that the system is *Bounded-Input Bounded-Output* (BIBO) stable and that the actuator has a limited range, so that the actual input received by the plant is

$$\tilde{u}_k = g(u_k) = \begin{cases} \underline{u} & \text{if } u_k < \underline{u}, \\ u_k & \text{if } u_k \in [\underline{u},\ \bar{u}], \\ \bar{u} & \text{if } u_k > \bar{u}, \end{cases} \tag{2}$$

with $\underline{u},\ \bar{u} \in \mathbb{R}$ being the *known* limits of the actuator. Accordingly, the dynamics (1) can be recast as

$$\mathcal{P}: \quad \begin{cases} x_{k+1} = \tilde{f}(x_k, \tilde{u}_k), \\ y_k^{\text{o}} = h(x_k). \end{cases} \tag{3}$$

where $\tilde{f}(x, u) = f(x, g(u))$. Since no prior knowledge is available on $\mathcal{P}$, other than the actuator bounds, and output sensors are noisy, we assume that open-loop experiments are carried out by feeding a sequence of inputs u_k to the plant and record

$$y_k = y_k^{\text{o}} + v_k$$

for $k = 1, \dots, N$, where $v_k \in \mathbb{R}$ is measurement noise. Given the *open-loop* data collection $Z_N = \{u_k, y_k\}_{k=1}^N$, our aim is to design a *nonlinear* feedback controller C_θ belonging to a pre-defined class $\mathcal{C}_\theta$ so that: (i) some desired reference-to-output tracking performance is attained, if allowed by the control authority limitations; (ii) a safe behavior of the controlled system is ensured even if the actuator saturates. Throughout the rest of the paper, we focus on an input-output parametrization $\mathcal{C}_\theta$ of the controller, described as

$$\mathcal{C}_\theta: \ \nu_{k+1} = C_\theta(\nu_k, \dots, \nu_{k-Q}, e_{k+1}, \dots, e_{k-Q}), \tag{4}$$

where $e_k = r_k - y_k$ is the tracking error attained at time k and $Q > 0$ is the order of the controller, which is fixed a priori by the designer. Following the direct data-driven control philosophy, we design the controller without identifying a model of $\mathcal{P}$ first. As a design specification, we consider a *model reference* architecture, so that the desired closed-loop performance is characterized through an a priori selected

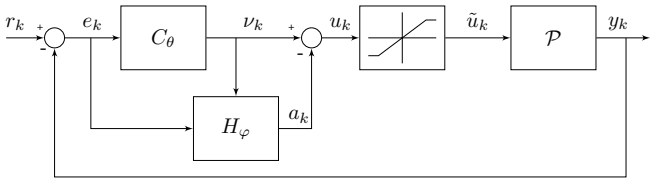


Fig. 1. The proposed direct data-driven anti-windup control scheme.

reference model $\mathcal{M}$. Namely, given the set point $r_k \in \mathbb{R}$, the output we aim at attaining in closed-loop is described as

$$\mathcal{M}: \quad y_{k+1}^{\mathcal{M}} = f_{\mathcal{M}}(y_k^{\mathcal{M}}, \dots, y_{k-M}^{\mathcal{M}}, r_k, \dots, r_{k-M}), \tag{5}$$

where $M > 0$ is the order of the closed-loop reference model $\mathcal{M}$. To use the additional knowledge on the saturation bound, we further resort to a control architecture inspired by standard *anti-windup* schemes [9]. Accordingly, we introduce an anti-windup block $H_\varphi \in \mathcal{H}_\varphi$ that embeds the known saturation, defined as

$$\mathcal{H}_\varphi: \ a_{k+1} = H_\varphi(\nu_{k+1}, \nu_k, \dots, \nu_{k-H}, e_{k+1}, \dots, e_{k-H}), \tag{6}$$

whose output is subtracted to the one generated by the controller C_θ, so that the overall control architecture can be represented as in Figure 1.

In this work, without loss of generality, we assume that both the controller C_θ and the anti-windup compensator are parameterized via two distinct *Artificial Neural Networks* (ANNs) and that they share the same order $H = Q$. We stress that the knowledge of the saturation is embedded within the chosen parametrization of C_θ and H_φ, as explained in Section III-A.

Remark 1: To guarantee a clear distinction between C_θ and H_φ, one can slightly modify the structure of the compensator as follows:

$$\tilde{\mathcal{H}}_\varphi: \ a_{k+1} = H_\varphi(\nu_{k+1}, \nu_k, \dots, \nu_{k-H}),$$

so that the anti-windup action depends only on the collection of corrected and past outputs of C_θ. This leads to a structure more similar to the one adopted in standard anti-windup schemes [9]. On the other hand, it reduces the generality in the structure of the compensator. ■

III. NAW-NETs: training and practical hints

To avoid modeling the plant $\mathcal{P}$, we rely only on the open-loop dataset Z_N and exploit the rationale of the *Virtual Reference* approach, originally described in [6]. Therefore, we construct a *virtual* closed-loop using the open-loop data collection, such that the reference-to-output relationship is exactly $\mathcal{M}$, as shown in Figure 2.

Starting from the available input and output measurements, we compute what would have been the *virtual* reference $\tilde{r}_k$ and the *virtual* error signal $\tilde{e}_k = \tilde{r}_k - y_k$ of such a fictitious closed-loop, by noticing that the former corresponds to the signal that would produce the measured output y_k when feeding the reference model $\mathcal{M}$, *i.e.,*

$$y_{k+1} = f_{\mathcal{M}}(y_k, \dots, y_{k-M}, \tilde{r}_k, \dots, \tilde{r}_{k-M}).$$

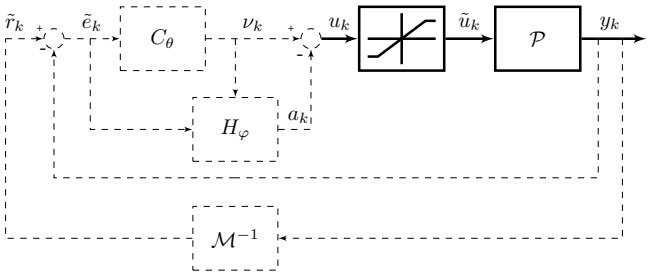


Fig. 2. The *Virtual Reference* rationale of [6] used within our setting. The thick line denotes the real plant on which the experiment has been performed, while the dashed lines illustrate the "virtual" remainder of the loop.

The computation of $\tilde{r}_k$ requires the inversion of $\mathcal{M}$, which can be explicitly obtained for relatively simple classes of reference models, such as linear transfer functions with asymptotically stable zeros [12]. In case of more complex reference models $\mathcal{M}$, more generally the fictitious reference can be obtained by solving the fitting problem

$$\begin{aligned} \min_{\{\tilde{r}_k\}_{k=M}^{N}} \quad & \frac{1}{N-M}\sum_{k=M}^{N} \ell_M(\hat{y}_k^M, y_k) \\ \text{s.t.} \quad & \hat{y}_{k+1}^{\mathcal{M}} = f_{\mathcal{M}}(y_k, \dots, y_{k-M}, \tilde{r}_k, \dots, \tilde{r}_{k-M}), \end{aligned} \tag{7}$$

where $\ell_M : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is a properly defined loss function. In the example reported in this paper we will focus on a linear reference model $\mathcal{M}$, for which solving (7) is not required.

Once the fictitious reference sequence $\{\tilde{r}_k\}_{k=M}^{N}$ is computed, the latter can be used to design the desired blocks (see equations (4) and (6), respectively), by trying to match the measured input sequence $\{u_k\}_{k=1}^{N}$ with the predicted input from the controller C_θ and the anti-windup compensator H_φ, while weighting unfeasible control actions according to the limits dictated by the known actuator bounds. Formally, we solve the following learning problem

$$\begin{aligned} \min_{\substack{C_\theta \in \mathcal{C}_\theta \\ H_\varphi \in \mathcal{H}_\varphi}} \quad & \mathcal{J}(\hat{\nu}_k(\theta), u_k, \hat{a}_k(\varphi)), \\ \text{s.t.} \quad & \hat{\nu}_{k+1}(\theta) = C_\theta(u_k, \dots, u_{k-Q}, \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}), \\ & \hat{a}_{k+1}(\varphi) = H_\varphi(\hat{\nu}_{k+1}, u_k, \dots, u_{k-Q}, \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}), \\ & \qquad\qquad k = 1, \dots, N-Q, \end{aligned} \tag{8}$$

where

$$\begin{aligned} &\mathcal{J}(\hat{\nu}(\theta), u, \hat{a}(\varphi)) = \\ &\quad \frac{1}{N-Q}\sum_{k=Q}^{N} \ell_C(\hat{\nu}_k(\theta), u_k) + \gamma \ell_S(\hat{\nu}_k(\theta) - \hat{a}_k(\varphi)), \end{aligned} \tag{9}$$

and $\gamma > 0$ is a hyper-parameter to be tuned that trades off between fitting the input samples and violate saturation limits. Note that the order of the controller Q dictates the number of samples that are discarded in the learning phase.

We adopt the *Mean Absolute Error* (MAE) loss figure

$$\ell_C(\hat{\nu}_k(\theta), u_k) = \|\hat{\nu}_k(\theta) - u_k\|_1,$$

to weight the fitting error on the input (other losses could be used too), while the error due to unfeasible control actions is accounted for by defining ℓ_S as follows:

$$\ell_S(\hat{u}_k(\theta,\varphi)) = \|\hat{u}_k(\theta,\varphi) - g(\hat{u}_k(\theta,\varphi))\|_1,$$

with $\hat{u}_k(\theta,\varphi) = \hat{\nu}_k(\theta) - \hat{a}_k(\varphi)$, for $k = Q, \dots, N$.

Remark 2: The learning problem in (8), along with the proposed anti-windup control scheme, can be readily adapted to a 1-DOF architecture, by setting $\hat{a}_k = 0$, $\forall k$, so that the learning problem becomes

$$\begin{aligned} \min_{C_\theta \in \mathcal{C}_\theta} \quad & \frac{1}{N-Q}\sum_{k=Q}^{N} \ell_C(\hat{\nu}_k(\theta), u_k) + \gamma \ell_S(\hat{\nu}_k(\theta)) \\ \text{s.t.} \quad & \hat{\nu}_{k+1}(\theta) = C_\theta(u_k, \dots, u_{k-Q}, \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}), \\ & \qquad\qquad k = 1, \dots, N-Q. \end{aligned}$$

Although not explicitly exploiting an anti-windup block, this optimization problem still allows one to train a controller that accounts for saturation limits, thanks to the structure of C_θ (see Section III-A).

Remark 3: Since the solution of problem (8) allows us to concurrently train the controller and the anti-windup block, a possible strategy for the selection of γ is described next. By solving problem (8) multiple times iteratively, one can initially choose γ low to promote better fitting performance. The value of γ can then be gradually increased so to enforce satisfaction of the actuator bounds.

Remark 4: Alternatively to the cost in (9), one can also minimize the following objective:

$$\mathcal{J}(\hat{u}(\theta,\varphi), \tilde{u}) = \frac{1}{N-Q}\sum_{k=Q}^{N} \ell_C(\hat{u}_k(\theta,\varphi), \tilde{u}_k) + \gamma \ell_S(\hat{u}_k(\theta,\varphi)),$$

with $\hat{u}_k(\theta,\varphi) = \hat{\nu}_k(\theta) - \hat{a}_k(\varphi)$. In this case, C_θ and H_φ are designed so that the input to $\mathcal{P}$ corresponds to the saturated control actions $\{\tilde{u}_k\}_{k=1}^{N}$. We stress that ℓ_S has still to be included in the cost if one wants the compensator to act only when the control action ν_k exceeds the saturation bounds, as in standard anti-windup schemes.

A. NAW-NET parameterization

It is clear that the chosen parameterizations for the maps $C_\theta \in \mathcal{C}_\theta$ and $H_\varphi \in \mathcal{H}_\varphi$ are crucial to achieve good control performance, namely to attain the desired closed-loop behavior $\mathcal{M}$ while capitalizing on the whole operating range of the actuators. Among possible alternatives for both the controller and the anti-windup block parameterizations, in this work we use *Artificial Neural Networks* (ANNs) [16] for their well-known effectiveness as maps approximators and the availability of well-maintained tools for easily training them.

Specifically, the controller is parameterized as the input/output law

$$\nu_{k+1} = \mathcal{NN}_\theta(\mathcal{I}_k^C)'\mathcal{I}_k^C, \tag{10}$$

where $\mathcal{I}_k^C \in \mathbb{R}^{n_\mathcal{I}}$ is the feature vector fed to the parametric part of the controller, namely the ANN $\mathcal{NN}_\theta : \mathbb{R}^{N_\mathcal{I}} \to \mathbb{R}$, whose internal structure is schematically represented in Figure 3. The feature vector $\mathcal{I}_k^C$ is extracted via a map

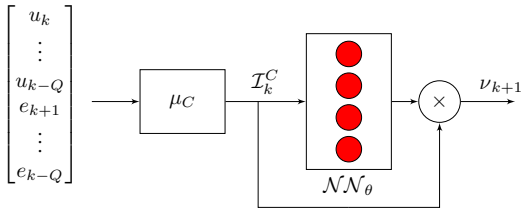


Fig. 3. The selected controller structure.

$\mu_C : \mathbb{R}^{2Q+1} \to \mathbb{R}^{n_\mathcal{I}}$ on the measured inputs and tracking errors, namely

$$\mathcal{I}_k^C = \mu_C(u_k, \dots, u_{k-Q}, \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}), \tag{11}$$

through which we embed in our controller any available prior knowledge on $\mathcal{P}$. Since in our setting the only prior information on the plant resides in the known saturation function g in (2), we select μ_C as

$$\begin{aligned}\mu_C(u_k, \dots, u_{k-Q}, \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}) = \\ = \left[u_k, \cdots, u_{k-Q}, g(u_k), \cdots, g(u_{k-Q}), e_{k+1}, \cdots, e_{k-Q}\right],\end{aligned} \tag{12}$$

so to also include the projections $g(u_{k-j})$ on the admissible input range of past control samples u_{k-j} as additional regressive terms, $j = 0, \dots, Q$. We remark that, in principle, μ_C can be learned from data along with $\mathcal{NN}_\theta$. However, this may require a much larger amount of training data and would result in a more complex architecture, eventually making the approach too demanding for simple embedded applications. From a practical stand point, C_θ is implemented through a feed-forward neural network fed by $\mathcal{I}_k^C$. Nonetheless, the network is used to predict a set of intermediate coefficients that produce the control action once they are multiplied by $\mathcal{I}_k^C$.

The chosen structure of C_θ allows us to trade-off between the expressiveness of the parameterization and the ease of inspection of the learned controller. Moreover, if a *linear time invariant* (LTI) parameterization is sufficient, it is fairly simple to downgrade the neural controller to an LTI one, by exploiting parameter shrinkage strategies on the non-bias weights of the network in the training phase [17]. Indeed, if after training these weights they all result zero, this implies that the network itself can be replaced by its (constant) output.

Differently from the controller, for the anti-windup block we resort to a standard input-output feed-forward structure, with the ANN parameterizing H_φ directly mapping its input into the corrective action a_k. This choice is due to the fact that the anti-windup block is nonlinear, thus making more complex architectures like the one chosen for C_θ quite useful, at the price of a more complex optimization problem to be solved during the training phase. For the compensator to share similar characteristics to standard anti-windup blocks, we do still exploit a preliminary feature extraction map $\mu_H = \mu_C$.

B. Improving NAW-NET performance via Truncated BPTT

Despite its appealing simple structure, a design problem based on one-step-ahead predictions like the one defined in (8) might result in a data-driven control scheme that, albeit good over short horizons, may not be suited to be repeatedly iterated over time. A possible approach to overcome this limitation is to exploit *Back Propagation Through Time* (BPTT) in the learning scheme, which has already been used for direct control applications in [15], [18]. This approach allows one to evaluate the controller+anti-windup block *simulation* accuracy, by concatenating predictions at two successive sampling steps. However, it concurrently involves the optimization of loss functions that are computationally expensive to evaluate and hard to minimize [19].

To avoid over-complicating the learning scheme, while retaining some of the desirable features of a BPTT architecture, in this work we resort to the so-called *truncated* BPTT approach [20]. Accordingly, predictions are propagated for a limited number of steps $F \ll N$, with $F > 0$ being an additional tuning parameter of the approach. Let v_{k+h} be defined as

$$v_{k+h} = \begin{cases} u_{k+h} \text{ if } h \leq 0, \\ \hat{\nu}_{k+h} \text{ otherwise,} \end{cases} \tag{13a}$$

and

$$\begin{aligned}\tilde{\mathcal{J}}(\hat{\nu}(\theta), u, \hat{a}(\varphi)) = \\ \frac{1}{N-Q} \sum_{k=Q}^{N} \sum_{j=1}^{F} \ell_C(\hat{\nu}_k(\theta), u_k) + \gamma \ell_S(\hat{\nu}_k(\theta) - \hat{a}_k(\varphi)).\end{aligned} \tag{13b}$$

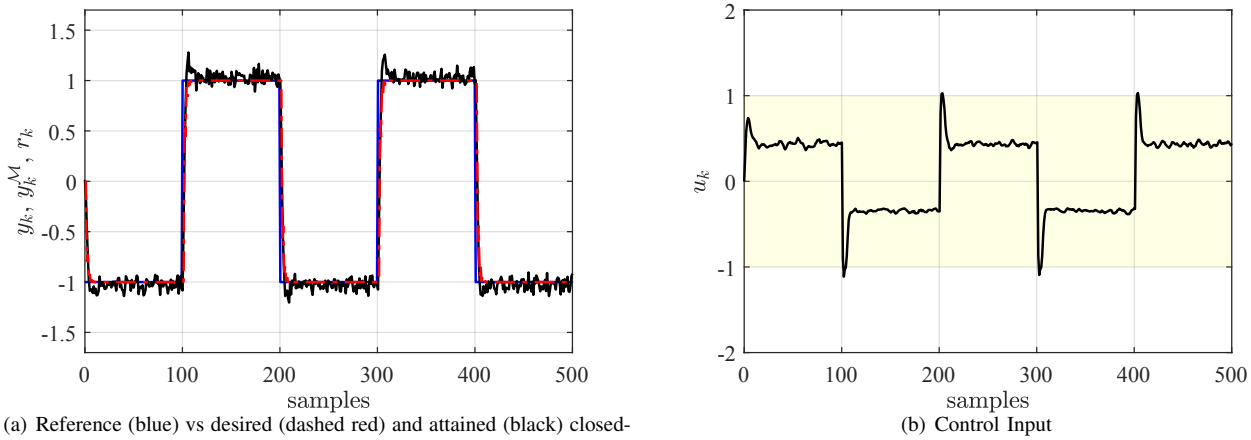
The design problem in (8) is thus recast as:

$$\begin{aligned}\min_{\substack{C_\theta \in \mathcal{C}_\theta \\ H_\varphi \in \mathcal{H}_\varphi}} \ & \tilde{\mathcal{J}}(\hat{\nu}(\theta), u, \hat{a}(\varphi)) \\ \text{s.t. } & \hat{\nu}_{k+j}(\theta) = C_\theta(v_{k+j}, \dots, v_{k-Q+j}, \tilde{e}_{k+j}, \dots, \tilde{e}_{k-Q+j}), \\ & \hat{a}_{k+j}(\varphi) = H_\varphi(\hat{\nu}_{k+j}, v_{k+j-1}, \dots, v_{k-Q+j}, \\ & \qquad \tilde{e}_{k+1}, \dots, \tilde{e}_{k-Q}), \\ & \qquad j = 1, \dots, F, \ k = 1, \dots, N-Q-F,\end{aligned} \tag{13c}$$

It is clear that the quality of the control scheme obtained via the solution of the optimization problem in (13c) heavily depends on the choice of the length of the prediction horizon F. We stress that F should be chosen so as to compromise of the available computing capabilities, the simulation accuracy of the learned controller, and the ability of the optimization solver to reach a good quality solution of (13).

C. Data augmentation

Suppose that the dataset available to design the controller+anti-windup compensator is given, so that one cannot arbitrarily select the inputs that excite the plant. In this case, it might be possible that the data contain pairs of saturated inputs and corresponding outputs only. This might be good for the identification of $\mathcal{P}$ in (3). However, having a training set that only consists of input/output sequences that already belong to the feasible region, namely $u_k \in [\underline{u}, \bar{u}] \ \forall k$, the controller and constraint enforcement module obtained by solving (13c) are likely to lead to poor closed-loop performance, since the available dataset is not informative enough for the considered learning task. An effective and rather general strategy to handle this problem is to augment



(a) Reference (blue) vs desired (dashed red) and attained (black) closed-loop response

(b) Control Input

Fig. 4. [Left panel] Desired and achieved closed-loop response when tracking the set point in (16). The reference signal and the desired output are almost always overlapped. [Right panel] Control input (black) and linear operating region of the actuator (yellow area).

the original dataset with *artificial samples* that violate the saturation bounds. The generation of these data is quite a delicate task, which could also deteriorate the quality of the original dataset with samples that the actual plant could never generate.

To bypass these limitations, by accounting for the fact that the saturation bounds act on the input only, we rely on the following heuristic. We augment the dataset Z_N, by superimposing a suitably defined white noise sequence on available inputs. A possible choice for the latter is an additive noise sequence with Gaussian distribution, so that the available dataset becomes $\tilde{Z}_n = \{u_k, \rho(u_k), y_k\}_{k=1}^N$, with $\rho : \mathbb{R} \to \mathbb{R}$ being the following stochastic map:

$$\rho(u_k) = u_k + w_k, \quad w_k \sim \mathcal{N}(0, \sigma^2), \tag{14}$$

and where the variance $\sigma^2 \in \mathbb{R}$ is a new parameter to be tuned. We stress that the artificially generated inputs are used solely to train the anti-windup module and that the quality of the results obtained by exploiting this approach strongly depends on the characteristics of the chosen noise sequence.

IV. Simulation results

To show the effectiveness of the proposed data-driven scheme, we consider the problem of controlling the following two-dimensional Hammerstein-Wiener plant:

$$\begin{cases} x_{k+1} = \begin{bmatrix} 0.755 & 0.250 \\ -0.199 & 0 \end{bmatrix} x_k + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \operatorname{sign}(\tilde{u}_k)\sqrt{|\tilde{u}_k|}, \\ z_k = \begin{bmatrix} 0.699 & -0.443 \end{bmatrix} x_k, \\ y_k^{\circ} = z_k + 5\sin(z_k), \end{cases}$$

where $\tilde{u}_k$ is generated according to (2), with $\underline{u} = -1$ and $\bar{u} = 1$. In order to attain the closed-loop behavior described by the reference model

$$\mathcal{M}: \quad y_{k+1}^{\mathcal{M}} = -0.1 y_k^{\mathcal{M}} + 0.3 y_{k-1}^{\mathcal{M}} + 0.8 r_{k-1},$$

we learn a controller and an anti-windup compensator of order $Q = 4$ by using a set Z_N of $N = 10000$ samples. These are generated by exciting the system with a sequence u_k obtained by superimposing a normally distributed white noise sequence with a periodic step signal with variable amplitude and period equal to 7. The output is corrupted by a white noise sequence $\{v_k\}_{k=1}^N$ with zero-mean Gaussian distribution and variance equal to 0.2, and the resulting signal is normalized using the empirical mean and standard deviation computed on the available data before starting the learning procedure. All the involved ANNs are chosen as compact networks featuring 4 nonlinear hidden layers of 15 neurons each and a final output layer of proper dimensions. We resort to the well-known *Rectified Linear Unit* (ReLU) maps [21] as the activation function of the neurons. The implementation of all the networks is carried out in Keras [22] with $F = 5$. In the learning phase, we initially set $\gamma = 0$ in (13c). Once this optimization procedure converges, we retrain both C_θ and H_φ by imposing $\gamma = 6$ and using the weights resulting from the previous phase as initial guesses for the new instance of the learning scheme. All the computations were carried out using a laptop equipped with a 2.8-GHz Intel Core i7 with 16 GB of RAM.

Figure 4 shows the response of the designed scheme when we consider a reference sequence $\{r_k\}_{k=1}^{\tilde{N}}$ of length $\tilde{N} = 500$, where r_k is the piecewise-constant signal

$$r_k = \begin{cases} -1, \text{ if } 100(i-1)+1 \le k \le 100 \cdot i, \\ 1, \text{ otherwise,} \end{cases} \tag{16}$$

for $i = 1, 3, 5, \ldots$, and the output is corrupted by a noise with the same distribution of that acting on the training set. It is clear that the output obtained in closed-loop tracks the desired response and that the control input feeding the plant rarely exceeds the saturation bounds. Since our aim is to track the set point as similarly as possible to the reference model $\mathcal{M}$, while devising control inputs that exceeds as little as possible the saturation limits, we quantitatively assess the performance of the scheme by introducing the following indexes:

$$\text{RMSE}_{\mathcal{M}} = \sqrt{\frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{N}} (y_k - y_k^{\mathcal{M}})^2}, \tag{17a}$$

TABLE I

PERFORMANCE INDEXES OBTAINED WITH DIFFERENT CONTROLLERS

	unbounded actuator (ideal case)	controller without anti-windup action ($\gamma = 0$)	controller+anti-windup compensator (2-DOF, $\gamma = 6$)
$\text{RMSE}_{\mathcal{M}}$	0.154	0.110	0.098
$\mathcal{OB}_{\%}$ [%]	-	4.8	2.0

TABLE II

INDEXES OBTAINED WHEN USING LESS INFORMATIVE TRAINING DATA

	$w_k = 0$, $\forall k$ in (14)	$\sigma^2 = 0.4$ in (14)
$\text{RMSE}_{\mathcal{M}}$	0.092	0.097
$\mathcal{OB}_{\%}$ [%]	1.4	0.4

$$\mathcal{OB}_{\%} = \frac{\#\{k \in [1,\ \tilde{N}] : u_k \notin [\underline{u},\ \bar{u}]\}}{\tilde{N}} \cdot 100\%, \qquad (17b)$$

with $\text{RMSE}_{\mathcal{M}}$ and $\mathcal{OB}_{\%}$[1] respectively indicating how well the desired closed-loop behavior is tracked and how many times the actuator bounds are exceeded. The obtained results are reported in Table I, along with the ones retrieved by considering the ideal case of unbounded actuators and a scheme designed by neglecting input saturation, *i.e.,* by setting $\gamma = 0$ in (13b) and not introducing the saturated inputs as regressive terms. It is clear that the achieved performances are comparable when considering the tracking capabilities of the different configurations, with the proposed 2-DOF architecture allowing us to reduce the number of times the actuator bounds are exceeded and slightly improve tracking performance.

We finally test the proposed approach in the presence of less informative data, by assuming that only saturated inputs are available to learn the controller with anti-windup. In this case, we compare the performance attained when the proposed 2-DOF architecture is designed with and without the additional data augmentation presented in Section III-C. The obtained quality indexes are reported in Table II, showing that the exploited strategy still allows us to obtain comparable closed-loop performance in terms of tracking, while resulting in a control input that exceeds the saturation bounds even less than the one obtained by learning the controller+anti-windup block with unsaturated inputs.

V. CONCLUSIONS

In this paper we have presented a preliminary data-driven design method of neural controllers with anti-windup, by relying on a model-reference architecture approach. The proposed learning strategy only requires the knowledge of the actuator limits, and does not involve the identification of a full open-loop model of the plant to be controlled. As shown by the promising initial numerical results, the approach allows us to attain satisfactory performance in terms of reference tracking, in spite of bounded actuators. Future research will be devoted to study data-driven strategies for the selection of proper reference models and to assess the performance of the approach in more challenging scenarios.

[1]Given a set $\mathcal{A}$, $\#\mathcal{A}$ indicates its cardinality.

REFERENCES

[1] L. Lennart, "System identification: theory for the user," *PTR Prentice Hall, Upper Saddle River, NJ*, pp. 1–14, 1999.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[3] B. Recht, "A Tour of Reinforcement Learning: The View from Continuous Control," 2018.

[4] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using Reinforcement Learning: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 2042–2062, Jun 2018.

[5] L. Ferrarotti and A. Bemporad, "Synthesis of optimal feedback controllers from data via stochastic gradient descent," in *2019 18th European Control Conference (ECC)*, pp. 2486–2491, 2019.

[6] M. Campi, A. Lecchini, and S. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002.

[7] S. Formentin, M. C. Campi, A. Carè, and S. M. Savaresi, "Deterministic continuous-time virtual reference feedback tuning (VRFT) with application to PID design," *Systems & Control Letters*, vol. 127, pp. 25–34, 2019.

[8] A. Bemporad, "Model-based predictive control design: New trends and tools," in *Proceedings of 45th IEEE Conference on Decision and Control*, pp. 6678–6683, 2006.

[9] L. Zaccarian and A. Teel, *Modern Anti-windup Synthesis: Control Augmentation for Actuator Saturation.* Princeton University Press, 2011.

[10] S. Formentin, F. Dabbene, R. Tempo, L. Zaccarian, and S. M. Savaresi, "Robust linear static anti-windup with probabilistic certificates," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1575–1589, 2016.

[11] F. Todeschini, S. Formentin, G. Panzani, M. Corno, S. M. Savaresi, and L. Zaccarian, "Nonlinear pressure control for bbw systems via deadzone and antiwindup compensation," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1419–1431, 2015.

[12] D. Piga, S. Formentin, and A. Bemporad, "Direct data-driven control of constrained systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1422–1429, 2017.

[13] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi, "Direct learning of LPV controllers from data," *Automatica*, vol. 65, pp. 98–110, 2016.

[14] D. Masti, V. Breschi, S. Formentin, and A. Bemporad, "Direct data-driven design of neural reference governors," in *Proc. 59th IEEE Conf. on Decision and Control*, 2020. To appear.

[15] P. Yan, D. Liu, D. Wang, and H. Ma, "Data-driven controller design for general MIMO nonlinear systems via virtual reference feedback tuning and neural networks," *Neurocomputing*, vol. 171, pp. 815–825, 2016.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[18] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[19] M. Forgione and D. Piga, "Model structures and fitting criteria for system identification with neural networks," *arXiv preprint arXiv:1911.13034*, 2019.

[20] G. Puskorius and L. Feldkamp, "Truncated backpropagation through time and Kalman filter training for neurocontrol," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 4, pp. 2488–2493, 1994.

[21] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[22] F. Chollet *et al.*, "Keras." https://github.com/keras-team/keras, 2015.