Nonlinear MRAC for MIMO Linear Time-Invariant Systems under Neural Network Control

Dalim Wahby, Alvaro Detailleur, Guillaume Ducard

***Abstract*— Neural network controllers (NNCs) can represent highly nonlinear control laws; however, ensuring closed-loop stability remains a non-trivial task. State-of-the-art approaches that can guarantee stability typically either verify stability post-learning or embed stability guarantees directly by design. The goal of this work is to develop stable adaptation laws for feedforward NNCs that guarantee stability during learning. To this end, stability is embedded by design through leveraging concepts from adaptive control. A direct model reference adaptive control (MRAC) strategy is proposed to derive an adaptation mechanism for multiple-input multiple-output (MIMO) linear time-invariant (LTI) systems controlled by NNCs. The algorithm, referred to as nonlinear model reference adaptive control (NMRAC), ensures closed-loop stability through a Lyapunov-based adaptation mechanism for updating the NNC parameters. Additionally, a new generalized and highly interpretable Lyapunov candidate function is introduced, allowing desired control behavior to be imposed by selecting an appropriate model reference system. The value of our contributions is demonstrated through two numerical examples: a regulation task for an initially unstable system controlled by a NNC, and a reference-tracking task.**

I. Introduction

Since at least the 1980s, it has been recognized that neural networks used as controllers, so-called neural network controllers (NNCs), offer significant potential due to their universal approximation capabilities [1] and the ability to store complex function parameterizations, whilst retaining low computational requirements [2], [3]. Nevertheless, their inherent nonlinearities make closed-loop stability verification of systems controlled by NNCs a non-trivial task [4], [5]. This has motivated research into both the verification of stability and the design of systems that are inherently stable by design.

State-of-the-art approaches can be broadly classified into post-learning verification methods, which analyze system stability after training, and design-based methods, which provide stability guarantees by construction. Post-learning stability verification methods consider the NNC to have fixed weights and they typically involve posing the search for a Lyapunov function as an optimization problem. Noteworthy works utilize (integral) quadratic constraints [6] and incremental quadratic constraints [7], mixed integer program (MIP)-based approaches [8] [9], and semialgebraic set-based modeling techniques [4] [10]. In contrast, design-based methods construct controllers that guarantee closed-loop stability by design. A prominent subclass of these methods comes from adaptive control, which models the closed-loop system as a time-varying system with adapting NNC parameters. In this paper, we focus on a method from this subclass, namely model reference adaptive control (MRAC).

D. Wahby and G. Ducard are with Université Côte d'Azur, i3S, CNRS, 06903 Sophia Antipolis, France. (E-mail: dalim.wahby@univ-cotedazur.fr; guillaume.ducard@univ-cotedazur.fr).

A. Detailleur and C. Onder are with the Institute for Dynamic Systems and Control (IDSC), Department of Mechanical and Process Engineering, ETH Zurich, Leonhardstrasse 21, 8092 Zürich, Switzerland. (E-mail: adetailleur@student.ethz.ch; Telephone: +41 44 632 87 96)

This work was supported by the French government through the France 2030 investment plan managed by the National Research Agency (ANR), as part of the Initiative of Excellence Université Côte d'Azur under reference number ANR- 15-IDEX-01.

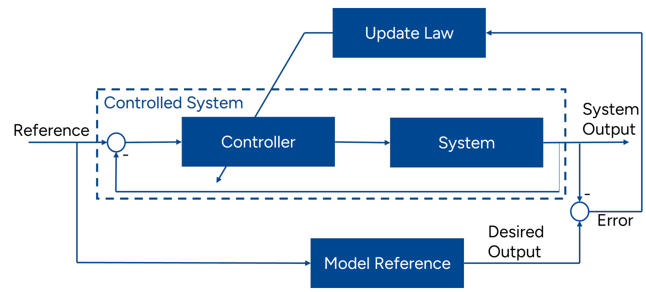

Fig. 1: General structure of MRAC, consisting of three main components: 1) the model reference system, 2) the controlled system, and 3) the adaptation mechanism, to adapt the controller parameters.

A. Model Reference Adaptive Control

MRAC, introduced in the 1950s [11], enables a controlled system to match the behavior of a stable model reference system. As illustrated in Fig. 1, the architecture uses an adaptation mechanism to update controller parameters, ensuring performance under uncertain or time-varying dynamics. While gradient-based methods, also known as the MIT rule, lack stability guarantees during learning [12], Lyapunov-based approaches ensure stability by selecting adaptation laws that minimize a defined error function [13].

Standard MRAC strategies include [14]:

- *Direct MRAC*: Updates controller parameters directly to minimize tracking error.
- *Indirect MRAC*: Estimates plant parameters to compute the control law.
- *Combined MRAC (CMRAC)*: Integrates both tracking and estimation errors into a composite adaptation law to improve parameter convergence and robustness [15], [16].

While these methods have been applied to n-th order linear time-invariant (LTI) systems [17], [18] and augmented with NNs for system identification [19], a direct, Lyapunov-based adaptation law for multiple-input multiple-output (MIMO)

LTI systems controlled by NNCs remains largely unexplored. Deriving such a law is challenging because the stability analysis must explicitly account for the interaction between the adaptive law and the network's nonlinear activation functions. This paper introduces a nonlinear model reference adaptive control (NMRAC) framework to address this gap.

B. Contributions

The aim of this work is to develop adaptation laws for feedforward NNCs that are used to control MIMO LTI systems, while guaranteeing stability. To this end, the proposed algorithm builds on the work presented in [20] and is a direct, Lyapunov-based MRAC strategy. This paper's specific contributions can be summarized as follows:

1) novel approach to Lyapunov-based MRAC for linear MIMO LTI systems controlled by feedforward NNCs, called nonlinear model reference adaptive control (NMRAC), is introduced and
2) a new, generalized, highly interpretable candidate Lyapunov function for Lyapunov-based MRAC, that guarantees stability during the adaptation of the NNC parameters.

The remaining paper is organized as follows: in Section I-C the notation and important concepts to this work are introduced. Subsequently, in Section II the proposed method of this work is presented. To highlight the value of our contributions, in Section III two numerical examples are provided and finally, in Section IV the paper is concluded and a brief outlook is discussed.

C. Notation and Definitions

This work uses the following notation:

- $n, m \in \mathbb{N}^+$ define the dimensionality of the state and input, respectively.
- Given a vector $x \in \mathbb{R}^n$, its components are denoted by $x = [x_1, x_2, \dots, x_n]^\top$.
- A function evaluated at x will be denoted as $f(x)$, whereas a function that is not evaluated at a specific point will be denoted as $f(\cdot)$.
- Lyapunov candidates and functions are represented with the variable name V.
- The norm $||\cdot||_A$ is a matrix-induced norm by matrix A and is defined as $||x||_A = \sqrt{x^\top A x}$, for a positive definite matrix A.
- Consider a matrix A, then $A^\dagger$ denotes its pseudo-inverse in the form of the Moore-Penrose inverse. Furthermore, the pseudo-inverse of a vector x is defined as $x^\dagger = x(x^\top x)^{-1}$ and the pseudo-inverse of the null vector is defined to be 0.

The following definitions are used in this work:

Definition I.1 (LTI System)**.** A continuous-time LTI system is defined by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1}$$

with state vector $x \in \mathbb{R}^n$, control input vector $u \in \mathbb{R}^m$, internal dynamics matrix $A \in \mathbb{R}^{n\times n}$ and control matrix $B \in \mathbb{R}^{n\times m}$.

If $u(t)$ is a function of x, the system is considered to be in closed-loop. The trajectory of the system is defined by the evolution of $x(t)$ over time. Furthermore, the system is assumed to have an equilibrium point at $\dot{x} = 0$. Henceforth, the dependency of the system's dynamics on time is considered to be implicit and will be neglected in the notation.

Definition I.2 (Lyapunov function)**.** A Lyapunov function for system (1) is a continuous function $V : \mathcal{D} \to \mathbb{R}$, defined on some domain $\mathcal{D} \subseteq \mathbb{R}^n$ containing the origin, with the following properties:

(a) *positive definiteness*, $V(x) > 0 \quad \forall x \in \mathcal{D} \setminus \{0\}$ and $V(0) = 0$,
(b) *decrease condition*, $\dot{V}(x) \leq 0 \quad \forall x \in \mathcal{D}$.

Definition I.3 (Lyapunov stability)**.** A system of the form (1) is said to be *Lyapunov stable* if there exists a function $V(\cdot)$ that satisfies Definition I.2. If, in addition, $\dot{V}(x) < 0$ for all $x \in \mathcal{D} \setminus \{0\}$, the system is *asymptotically stable*, implying that $x(t) \to 0$ as $t \to \infty$.

Throughout this work, stability and Lyapunov stability are used interchangeably.

The goal of this work is to develop a stable adaptation mechanism for feedforward NNCs.

Definition I.4 (Feedforward neural network)**.** A feedforward neural network (NN) $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$\begin{aligned} \phi(x) &= \varphi_H \circ L_H \dots \varphi_2 \circ L_2 \circ \varphi_1 \circ L_1(x) \\ L_i(x) &= \theta_i x + b_i \quad \forall i \in \{1, ..., H\}, \end{aligned} \tag{2}$$

where $\varphi_i(\cdot)$ are the activation functions applied element-wise, θ_i and b_i are the weight matrix and bias of layer i, respectively.

Henceforth, we use the term NN for the mathematical object and NNC for a NN used as a controller. Note that in this work only smooth activation functions are considered to ensure the existence of their derivatives over the entire domain.

NNs usually make use of nonlinear activation functions, which enable them to approximate nonlinear functions. The activation function used in this work is designed to model a smooth saturation function, which is taken from [21], [22] and defined as

$$\sigma(x; a) = \frac{2(1 - e^{-ax})}{a(1 + e^{-ax})}. \tag{3}$$

Note that the activation function saturates at $\pm\frac{2}{a}$. Henceforth, the saturation function is denoted as $\sigma(\cdot)$ and is applied element-wise.

II. Nonlinear Model Reference Adaptive Control

The aim is to find an adaptation mechanism for the NNC parameters that satisfies sufficient conditions to prove stability, using a Lyapunov-based MRAC. Stability is demonstrated by the convergence of the controlled system to the model reference system through a Lyapunov function defined on the error between the two systems, which leads us to the following assumption.

Assumption 1 (Stable model reference system)**.** *The model reference system is chosen to be stable according to Definition I.3.*

This convergence is achieved when the two systems become equal to one another. Hence, the equality between the controlled system and the model reference system can be characterized by two conditions: 1) their internal states are identical, expressed by the state error being zero, and 2) they evolve according to the same dynamics, expressed by the dynamic error being zero. This leads to the following assumption on the existence of a solution to the problem formulation.

Assumption 2 (Matching conditions)**.** *The matching conditions are satisfied if there exists a controller that makes the controlled system equal to the model reference system.*

Remark 1. Assumption 2 is satisfied if the model reference is chosen such that the required control effort to match the dynamics falls within the range of the control matrix B, and the NNC possesses sufficient expressive power to make the controlled system match the model reference.

Note that Assumption 1 and Assumption 2 ensure the existence of a Lyapunov function for the closed-loop, controlled system.

To develop the proposed adaptation law for the NNC, consider a MIMO LTI system (1). Furthermore, we define the input to this system to be a feedforward NNC, as per Definition I.4, with

$$u = \phi(e_x) = \sigma(\theta e_x), \tag{4}$$

with the tracking error $e_x = r - x$ and reference signal r. Note that the activation function of the NNC is a smooth saturation function, as defined in (3) and parameter $\theta \in \mathbb{R}^{m\times n}$ is the adaptable parameter of the NNC.

Furthermore, we define the model reference system to be of the same form as (1) and denote it as

$$\dot{x}_m = A_m x_m + B_m u_m. \tag{5}$$

The control input u_m is defined with the same functional structure as the control input u. Specifically, the control input is generated by the NNC $\phi_m(\cdot)$:

$$u_m = \phi_m(e_m) = \sigma_m(\theta_m e_m) \tag{6}$$

with the tracking error $e_m = r - x_m$ and parameters $\theta_m \in \mathbb{R}^{m\times n}$.

Next, a Lyapunov candidate is proposed that encodes the error in internal states

$$e = x_m - x, \tag{7}$$

and the dynamical error

$$\alpha = (A_m - A)x + B_m\phi_m(e_x) - B\phi(e_x). \tag{8}$$

Note that since e is defined to be the error in internal states between the model reference system and the controlled system, $e = 0$ means that the two system have the same state. Secondly, since α is defined as the difference in dynamics, $\alpha = 0$ implies that system evolves over time in the same way. Hence, if both e and α are equal to zero, this effectively implies that the systems can be considered equivalent.

The candidate Lyapunov function proposed in the following theorem is constructed for the entire closed-loop system, including both the plant and the model reference system, to assess the stability of the overall adaptive system. If this candidate satisfies the conditions of Definition I.2, it follows that (7) and (8) converge to zero over time. Consequently, the controlled system converges to the model reference system, and stability is guaranteed by Assumption 1. Note that, although not explicitly indicated in the notation, the dependence of the Lyapunov function V on the system state x is implicitly captured, since both e and α are functions of x.

Theorem 1. *Consider the function $V : \mathcal{D} \to \mathbb{R}$ defined as*

$$V(e,\alpha) = ||e||_P^2 + ||\alpha||_{\Gamma^{-1}}^2, \tag{9}$$

with positive definite matrices P and Γ^{-1}. Additionally, suppose that the following condition is satisfied

$$2e^\top P A_m e + 2e^\top P\big(B_m(\phi_m(e_m) - \phi_m(e_x))\big) < 0 \tag{10}$$

over the domain $\mathcal{D} \subseteq \mathbb{R}^n$.

Then $V(e,\alpha)$ constitutes a Lyapunov function, as per Definition I.2, for system (1)*, controlled by a NNC whose parameters are adapted according to*

$$\dot{\theta} = \Sigma^\dagger\left[\Gamma P e + (A_m - A)\dot{x} + \Sigma - B\frac{\partial\sigma(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta e_x}\theta\dot{e}_x\right]e_x^\dagger, \tag{11}$$

where $\Sigma = B_m\dot{\phi}_m(e_x)$.

Proof: The proof consists in demonstrating the properties of Definition I.2, i.e. (a) positive definiteness and (b) the decrease condition.

The positive definiteness of (9) follows directly from the matrix-induced norms $||\cdot||_P$ and $||\cdot||_{\Gamma^{-1}}$ and positive definiteness of matrices P and Γ^{-1}.

It remains to establish the decrease condition. To this end, the time derivative of the Lyapunov function is given by

$$\dot{V}(e,\alpha) = 2e^\top P\dot{e} + 2\alpha^\top\Gamma^{-1}\dot{\alpha}. \tag{12}$$

The time derivative of the error in internal states yields

$$\dot{e} = \dot{x}_m - \dot{x}. \tag{13}$$

After expanding (13) by $\pm(A_m x + B_m\phi_m(e_x))$, the following equation is obtained

$$\begin{aligned}\dot{e} =& A_m e \\ &+ B_m(\phi_m(e_m) - \phi_m(e_x)) \\ &+ \underbrace{(A_m - A)x + B_m\phi_m(e_x) - B\phi(e_x)}_{=\alpha},\end{aligned} \tag{14}$$

which leads to

$$\begin{aligned}\dot{V}(e,\alpha) = &\underbrace{2e^\top P A_m e}_{\text{(I)}} \\ &+ \underbrace{2\alpha^\top (Pe + \Gamma^{-1}\dot{\alpha})}_{\text{(II)}} \\ &+ \underbrace{2e^\top P\big(B_m(\phi_m(e_m) - \phi_m(e_x))\big)}_{\text{(III)}}.\end{aligned} \tag{15}$$

Term (II) is used to define the adaptation law through nullification. The remaining stability condition requires that the sum of terms (I) and (III) be negative, which is inherently system dependent. Hence, under the condition in (10) and the adaptation law (11), the function (9) constitutes a Lyapunov function for the closed-loop system, and it is therefore guaranteed to be stable in the region $\mathcal{D} \subseteq \mathbb{R}^n$. ■

Remark 2. The intuition behind this law is the "nullification" of the indefinite terms in the Lyapunov derivative $\dot{V}$. By selecting $\dot{\theta}$ such that it cancels out the dynamical error components, the closed-loop system is forced to inherit the stable manifold of the model reference.

III. Numerical Results

The contribution of Section II is demonstrated in two examples, 1) regulation task of an initially unstable-closed-loop MIMO LTI system controlled by a NNC and 2) a reference tracking task for the same system as used in the first example.

The presented examples serve to demonstrate that the proposed adaptation law (11) achieves convergence of the controlled system to the model reference system in both a regulation and a reference-tracking task. While theorem 1 requires condition (10) to be negative definite for stability, its feasibility is demonstrated numerically in the following example. For every sampled initial condition, the stability term was verified to be negative throughout the adaptation process. This indicates that the proposed NMRAC framework operates within a stable regime for the investigated system dynamics.

Consider the system of the form (1), defined by the following two matrices

$$A = \begin{bmatrix} -0.0125 & 1.0232 \\ 0.0451 & 0.0099 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{16}$$

Note that this system is unstable, since the real part of one of the poles of A is positive. Next, a stable model reference system is to be chosen, to adhere with Assumption 1. Therefore, we choose a double integrator system

$$A_m = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B_m = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{17}$$

and θ_m of (6) is obtained from the LQR solution and matrix P is the matrix satisfying the associated algebraic Riccati equation. This guarantees stability of the model reference system under the control law u_m within the non-saturated region of the controller.

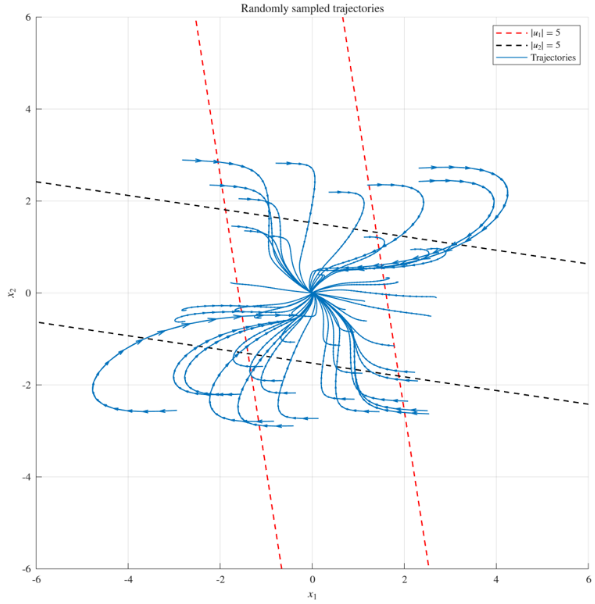

Fig. 2: Trajectories of 50 randomly sampled initial conditions, where the black and red lines indicate the saturation of inputs u_1 and u_2, respectively.

The method derived in Section II is in continuous-time, the closed-loop dynamics are discretized using an Euler forward discretization method during simulation. Additionally, for all simulations the initial controller parameter θ is chosen to be a matrix of zeros of the same dimensions as u_m. The remaining simulation parameters can be found in Table I.

TABLE I: Parameters used for simulation

Parameter	Symbol	Value	Unit
Regulation			
Sampling time	T_s	0.01	$[s]$
Simulation time	T_{sim}	100	$[s]$
Saturation value	a	0.4	-
Learning rate	Γ	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	-
Reference Tracking			
Sampling time	T_s	0.001	$[s]$
Simulation time	T_{sim}	200	$[s]$
Saturation value	a	0.4	-
Learning rate	Γ	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$	-

A. Regulation Task

The first example demonstrates that the proposed method stabilizes system (16). Fifty initial conditions are randomly sampled from the domain $[-3,3]^2 \subset \mathbb{R}^2$. As shown in Fig. 2, all simulated trajectories converge to the origin, confirming that the NNC, adapted via the proposed adaptation law (11), stabilizes the system across the sampled initial conditions. This demonstrates that the proposed adaptation mechanism achieves stabilization of an initially unstable system.

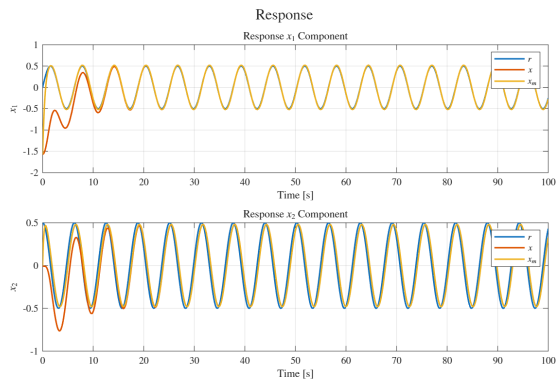

Fig. 3: System response for the reference-tracking task. The reference signal is defined to be sinusoidal.

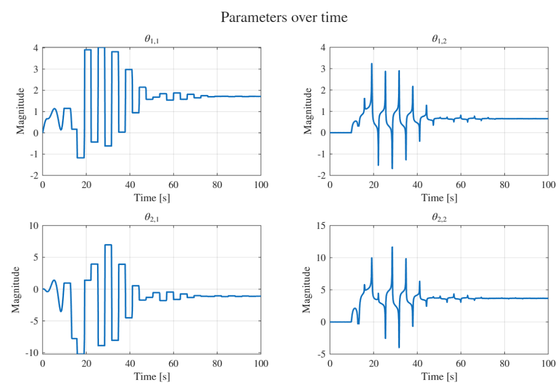

Fig. 4: Evolution of the NNC parameters over time for the reference-tracking task.

B. Reference-Tracking Task

For the reference-tracking task, the initial conditions of both the system and model reference system are set to $x = x_m = [-\frac{\pi}{2}, 0]^\top$. The reference r is a sinusoidal signal of amplitude 0.5.

Fig. 3 shows that the system response converges to that of the model reference system, with the state error e vanishing within approximately 10 seconds. Fig. 4 illustrates that the NNC parameters converge to a stationary value within roughly 80 seconds, implying convergence of the dynamics error α. Fig. 5 presents the Lyapunov function and its time derivative, which remains negative throughout the simulation. Hence, both the state and dynamics errors decrease over time, resulting in convergence of the closed-loop system to the reference behavior. This example demonstrates the effectiveness of the proposed adaptation mechanism for reference-tracking tasks.

IV. Conclusion

In this work, we extend the NMRAC method for first-order systems proposed in [20] by introducing a more general Lyapunov candidate function. This formulation enables the derivation of an adaptation mechanism for the NNC parameters, thereby allowing nonlinearities to be incorporated into

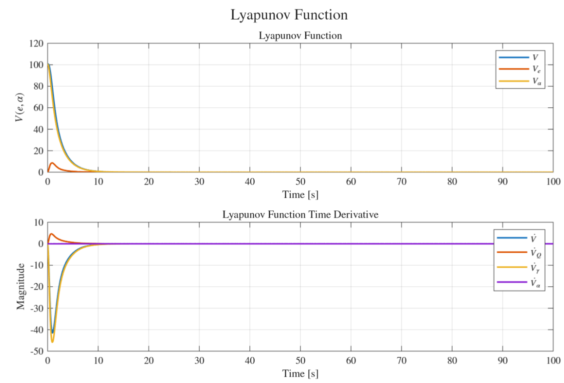

Fig. 5: Lyapunov function over time, where V_e and V_α represent the component of V defined by e and α. Furthermore, $\dot{V}$ defines the time derivative of the Lyapunov function and $\dot{V}_Q$, $\dot{V}_\alpha$, and $\dot{V}_\gamma$ represent terms (I), (II), and (III) of (15), respectively.

the closed-loop system. The value of the proposed method is demonstrated through two numerical examples, 1) a regulation task and 2) a reference-tracking task, performed on an initially unstable MIMO LTI system, achieving the desired behavior. Although formal stability proofs are not provided for these examples, the results illustrate the potential of the proposed method to achieve stable and well-behaved closed-loop control.

Furthermore, the presented examples are academic. Future research should investigate the proposed algorithm's application to real-world systems. Additionally, the NNC used here is small and relatively simple. Scaling to larger networks would enable representation of more complex control laws. For instance, the method could be used to approximate an MPC controller offline using a ReLU NNC, since it is known that an MPC controller can be efficiently represented using a ReLU network [23]. The learned network could then be deployed directly to the system, eliminating the need to solve the MPC optimization online and potentially enabling high-frequency evaluation of an approximately optimal controller with stability guarantees on real hardware.

Finally, this method is sensitive to discrete numerical approximation issues. In the examples, this was mitigated by reducing the sampling time of the continuous-time system, however, further reducing the sampling time may be impractical in real applications. Therefore, future work includes developing a discrete-time version of the method.

Acknowledgement

We acknowledge Mikael Johansson for fruitful discussions and meaningful inputs to this work.

References

[1] B. Hanin, "Universal function approximation by deep neural nets with bounded width and ReLU activations," en, *Mathematics*, vol. 7, no. 10, p. 992, Oct. 2019.

[2] A. Detailleur, D. Wahby, G. Ducard, and C. Onder, *Synthesis and SOS-based Stability Verification of a Neural-Network-Based Controller for a Two-wheeled Inverted Pendulum*, arXiv preprint, 2025. [Online]. Available: `https://arxiv.org/abs/2508.15616`.

[3] G. Ducard and G. Carughi, "Neural Network Design and Training for Longitudinal Flight Control of a Tilt-Rotor Hybrid Vertical Takeoff and Landing Unmanned Aerial Vehicle," *Drones*, vol. 8, no. 12, 2024.

[4] M. Korda, "Stability and Performance Verification of Dynamical Systems Controlled by Neural Networks: Algorithms and Complexity," *IEEE Control Systems Letters*, vol. 6, pp. 3265–3270, Jun. 2022.

[5] G. Norris, G. J. J. Ducard, and C. Onder, "Neural networks for control: A tutorial and survey of stability-analysis methods, properties, and discussions," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2021, pp. 1–6.

[6] H. Yin, P. Seiler, and M. Arcak, "Stability Analysis Using Quadratic Constraints for Systems With Neural Network Controllers," *IEEE Transactions on Automatic Control*, vol. 67, pp. 1980–1987, Apr. 2022.

[7] M. Revay, R. Wang, and I. R. Manchester, "Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness," *IEEE Transactions on Automatic Control*, vol. 69, pp. 2855–2870, May 2024.

[8] M. Dubach and G. Ducard, "A comparison of verification methods for neural-network controllers using mixed-integer programs," in *2022 7th International Conference on Robotics and Automation Engineering (ICRAE)*, Singapore, Nov. 2022, pp. 43–48.

[9] R. Schwan, C. N. Jones, and D. Kuhn, "Stability verification of neural network controllers using mixed-integer programming," *IEEE Transactions on Automatic Control*, vol. 68, pp. 7514–7529, Jun. 2023.

[10] A. Detailleur, D. Wahby, G. Ducard, and C. Onder, *Contributions to Semialgebraic-Set-Based Stability Verification of Dynamical Systems with Neural-Network-Based Controllers*, arXiv preprint, 2025. [Online]. Available: `https://arxiv.org/abs/2510.24391`.

[11] H. P. Whitaker and A. Kezer, *Design of Model Reference Adaptive Control Systems for Aircraft*. M.I.T. Instrumentation Laboratory, 1958.

[12] I. M. Y. Mareels, B. D. O. Anderson, R. R. Bitmead, M. Bodson, and S. S. Sastry, "Revisiting the Mit Rule for Adaptive Control," *IFAC Proceedings Volumes*, 1987.

[13] B. Shackcloth and R. L. But Chart, "Synthesis of Model Reference Adaptive Systems by Liapunov's Second Method," *IFAC Proceedings Volumes*, no. 2, 1965.

[14] K. Aström, "History of Adaptive Control," in *Encyclopedia of Systems and Control*, Springer, 2014.

[15] M. Duarte and K. Narendra, "Combined direct and indirect approach to adaptive control," *IEEE Transactions on Automatic Control*, vol. 34, Oct. 1989.

[16] K. S. Narendra and M. A. Duarte, "Robust Adaptive Control Using Combined Direct and Indirect Methods," in *1988 American Control Conference*, Atlanta, GA, USA, Jun. 1988.

[17] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control: With Aerospace Applications*. Springer, 2013.

[18] G. Tao, S. Chen, X. Tang, and M. J. Suresh, *Adaptive Control of Systems with Actuator Failures*. Springer, Jun. 2013.

[19] E. Lavretsky, "Combined/Composite Model Reference Adaptive Control," *IEEE Transactions on Automatic Control*, Nov. 2009.

[20] D. Wahby, A. Detailleur, and G. Ducard, "Nonlinear Model Reference Adaptive Control (NMRAC) for First-Order Systems," in *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, Barcelona, Spain, Jul. 2025, pp. 1–6.

[21] D. Wahby and G. Ducard, "Enhanced PID Neural Network Control (EPIDNNC) - A Model Reference Approach," in *2024 9th International Conference on Robotics and Automation Engineering (ICRAE)*, Singapore, Nov. 2024, pp. 205–210.

[22] M. Bosshart, G. Ducard, and C. Onder, "Comparison between two pid neural network controller approaches - simulation and discussion," in *2021 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2021.

[23] B. Karg and S. Lucia, "Efficient Representation and Approximation of Model Predictive Control Laws via Deep Learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.