Nonlinear Model Reference Adaptive Control (NMRAC) for First-Order Systems

Dalim Wahby
Université Côte d'Azur
I3S/CNRS
Sophia Antipolis, France
wahby@i3s.unice.fr

Alvaro Detailleur
IDSC
ETH Zürich
Zürich, Switzerland
adetailleur@student.ethz.ch

Guillaume Ducard
Université Côte d'Azur
I3S/CNRS
Sophia Antipolis, France
ducard@i3s.unice.fr

***Abstract*—Lyapunov-based Model Reference Adaptive Control (MRAC) is a well-known and widespread control strategy, that by design comes with stability guarantees. However, it does not allow for nonlinearities in the closed-loop system, which limits its application to linear systems with linear feedback controllers. In this work, a Lyapunov-based update mechanism for first-order systems controlled by neural network-based controllers (NNC) is proposed. The update mechanism is validated in simulation for a constant reference signal, a sinusoidal reference signal, and a smooth pulse signal. Promising numerical results show that the proposed algorithm converges towards the desired values and that the convergence speed can be tuned by manipulating the learning rate. Additionally, this work identifies numerical instabilities as a potential issue, which can be reduced by decreasing the sampling time.**

***Index Terms*—Neural Networks, Model Reference Adaptive Control, Lyapunov Method, Nonlinear Control, Stability, Adaptive Control, Robotics, Machine Learning**

I. Introduction

Model reference control is a technique that imposes a behavior on a system, by equating the output of the model reference to the output of the controlled system. This way, the gains of a pre-defined controller structure can be found.

However, this only works if the system's dynamics are known exactly, which is often far from reality. If the system's dynamics are not fully known, the controller gains can be found adaptively by a technique called *model reference adaptive control* (MRAC). The adaptation law can be chosen either by 1) a gradient descent-based update law as first proposed by Whitaker et al. [1] and applied in for example [2], [3], or 2) a Lyapunov-based update law as proposed by Shackcloth et al. [4]. The latter enforces that in every update step, the control law is stabilizing, which is not guaranteed when using the former. In this work, a Lyapunov-based update law is used, to guarantee stability during the learning process.

Neural networks (NN) are ideal candidate functions for control laws since they can be considered to be universal approximators [5]. Since at least the 90s, research has been conducted on neural network-based controllers (NNC) [6], and it has been shown that they can learn a desired behavior, for example in [2], [7]. However, standard NNs are nonlinear, through the use of nonlinear activation functions, which makes their formal verification challenging. They often rely on post-training stability verification, as done in [8], [9], which does not support any type of online learning for improving the control strategy, while guaranteeing a stabilizing controller.

The algorithm, proposed in this paper, is derived from a Lyapunov-based, linear MRAC, as in [10]–[12] for example, and defines a stable learning strategy for first-order systems controlled by NNs.

This paper is organized as follows. Section I-A highlights where this work can be positioned within existing literature, and Section I-B states the contributions. Subsequently, required definitions are introduced in Section II. This is followed by the proposed framework in Section III, with numerical results in Section IV, and a discussion in Section V. The paper is concluded in Section VI and Section VII gives a brief outlook on future work.

A. Literature Review

Model reference adaptive control (MRAC) was first introduced in the early 1970s by Whitaker et al. [13]. Originally, it was thought to deal with process uncertainties and disturbance dynamics, however, the proposed techniques were also used in different contexts, including but not limited to auto-tuning, automatic construction of gain schedules, and adaptive filtering [14], [15].

The adaptation mechanism of parameters follows two main approaches, namely 1) the MIT method or gradient descent-based method, and 2) the Lyapunov-based method [12]. The MIT method does not come with stability guarantees [16], whereas the Lyapunov method is based on an adaptation rule derived from Lyapunov's second method [4]. It imposes stability since the adaptation rule is chosen in a way such that the decrease condition on the Lyapunov function is always satisfied, thus, implying system convergence.

Generally, MRACs can be categorized into the following three categories [15]:

1) *direct MRAC* updates the controller parameters directly, ensuring the closed-loop system behaves like the reference model,
2) *indirect MRAC* updates an estimated model of the plant, which is then used to compute the controller parameters. Unlike direct MRAC, where the controller is updated

directly, indirect MRAC continuously refines the plant model and adapts the control law accordingly, and
3) *Combined MRAC* (CMRAC), introduced by Duarte et al. [17], integrates both methods using two adaptation rules: (1) estimating unmatched model uncertainties and (2) identifying system parameters. Simulations suggest that CMRAC is more robust than either direct or indirect MRAC alone [18], though formal guarantees are still an open question.

CMRACs have been extended to n-th order linear systems [19], [20]. Additionally, Lavretsky [10] incorporated Radial Basis Function (RBF) NNs for system identification, enabling unmatched uncertainty estimation with stability guarantees, broadening its applicability to a larger class of nonlinear systems.

The proposed algorithm in this paper is a direct MRAC method, utilizing a Lyapunov-based learning mechanism, that includes nonlinearities in the form of a simple feedforward NN, with a nonlinear activation function.

B. Contributions

The method proposed in this work is based on the logic of the previously mentioned works and the novelty of this approach lies in the ability of learning a stabilizing controller, in spite of a nonlinear NN being present in the closed-loop system. Hence, this method is called *nonlinear MRAC* or *NMRAC*. Additionally, the proposed algorithm is validated in simulations, which show the convergence of the NN parameters to their desired values.

II. DEFINITIONS AND NOMENCLATURE

A. Systems and Stability

In order to define stable learning algorithms, it is imperative to first recall the definition of first-order systems and stability, which is taken from [21].

Definition 1 (First-order system)**.** A continuous-time first-order system is defined by the following map

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, (x(t), u(t)) \mapsto \dot{x}(t) = f(x(t), u(t)). \quad (1)$$

Additionally, if $u(t)$ is a function of x, the system is considered autonomous, with closed-loop dynamics $f(x(t))$. The trajectory of the system is defined by the evolution of $x(t)$. Furthermore, the system has an equilibrium point at $f(x(t)) = 0$.

Definition 2 (Stability)**.** Consider the equilibrium point $x = 0$ of (1). Then the system is:

- *stable*, if for each $\epsilon > 0$, there is a $\delta > 0$, such that

$$||x(0)|| < \delta \Rightarrow ||x(t)|| < \epsilon, \quad \forall t \geq 0$$

- *unstable*, if not stable, and
- *asymptotically stable* if it is stable and δ can be chosen such that

$$||x(0)|| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0$$

Henceforth, the dependency of the dynamics on time is considered to be implicit and will be neglected in the notation.

Definition 3 (Lyapunov function)**.** A Lyapunov function is a continuous function $V : \mathbb{R}^n \to \mathbb{R}$ with the following properties:

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

Definition 4 (Lyapunov stability)**.** A system of the form (1) is said to be *Lyapunov stable*, if there exists a function $V(\cdot)$ that satisfies the conditions of Definition 3a and 3b.

The ultimate goal of this work is to develop stable update laws for NNCs. Hence, the designed control law will be considered as a NN. Therefore, the following definition of NNs is used throughout this work.

Definition 5 (Neural network)**.** A feedforward neural network (NN) $\phi : \mathbb{R}^n \to \mathbb{R}^p$ is defined as:

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

where the activation functions are called $\varphi_i(\cdot)$, θ_i and b_i are the weight matrix and bias of layer i, respectively. Whenever a bias is not mentioned it is assumed to be zero.

NNs usually make use of nonlinear activation functions, which enable them to approximate nonlinear functions. Typically, functions such as the hyperbolic tangent, sigmoid, or ReLU are used in applications of machine learning. This work will utilize an activation function that is designed to model a smooth saturation function, as used in [2], [22] and defined as follows:

$$\sigma(x; a_{sat}) = \frac{2(1 - e^{-a_{sat}x})}{a_{sat}(1 + e^{-a_{sat}x})} \quad (3)$$

Note that the activation function saturates at $\pm\frac{2}{a_{sat}}$. Henceforth, the saturation function is denoted as $\sigma(x)$.

III. NONLINEAR MODEL REFERENCE ADAPTIVE CONTROL (NMRAC)

Direct MRAC is characterized by the controller parameters being directly updated through a mechanism, which takes into account the error of the system with respect to the desired output, as shown in Fig. 1. Furthermore, stability can be guaranteed through imposing that the conditions of Definition 3 are satisfied. In this section, the update law for NMRAC for first-order systems is derived.

The system, as in Definition 1, is parameterized as follows:

$$\dot{x} = -ax + b\sigma(\theta e_x) \quad (4)$$

where the NN $\phi : \mathbb{R} \to \mathbb{R}, \quad \phi(x) = \sigma(\theta e_x)$ defines the control input, with an adaptable parameter θ and the state-tracking error $e_x = x_r - x$. Note that x_r represents the reference signal and the activation function φ is defined to be a smooth saturation function, as in (3).

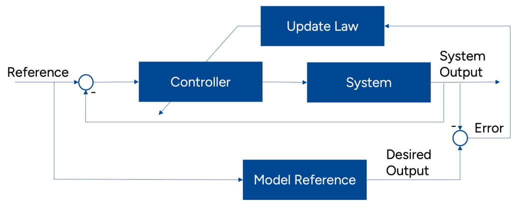


Fig. 1: General direct MRAC schematic

Next, a stable model reference of the same form is defined, as follows:

$$\dot{x}_m = -a_m x_m + b_m \sigma_m(\theta_m e_m) \tag{5}$$

where $e_m = x_r - x_m$, and $a_m > 0$.

Since the goal is that the system learns the behavior of the model reference, error dynamics are defined as follows:

$$e = x_m - x \tag{6}$$

Note that from this definition it follows that $e = e_x - e_m$, which is used as an alternative definition for the stability analysis of the update law later on in this work.

Following the error dynamics, a Lyapunov candidate is defined as follows:

$$V(e, \alpha) = ||e||_2^2 + c||\alpha||_2^2 \tag{7}$$

The factor $c > 0$ can be considered to be the learning rate, which is used to accelerate or decelerate the learning process. The squared norm of the error $||e||_2^2$ captures the distance between the internal states of the controlled system and the model reference, and the term $||\alpha||_2^2$ captures the difference in dynamics between the controlled system and the model reference. Note that both e and α should be 0 when the system follows the model reference and when the desired parameters are learned. This property renders our Lyapunov candidate in (7) to be positive definite.

To adhere to Definition 3, a negative time derivative of the Lyapunov function is required. Therefore, the behavior of the resulting function is analyzed, which is defined as follows:

$$\dot{V}(e, \alpha) = 2e\dot{e} + 2\alpha\dot{\alpha}c \tag{8}$$

Additionally, the time derivative of the error dynamics is defined as follows:

$$\begin{aligned} \dot{e} =& \dot{x}_m - \dot{x} \\ =& -a_m x_m + b_m \sigma_m(\theta_m e_m) - (-ax + b\sigma(\theta e_x)) \\ & \pm (a_m x + b_m \sigma_m(\theta_m e_x)) \\ =& -a_m \underbrace{(x_m - x)}_{=e} \\ & + \underbrace{(a - a_m)x + b_m \sigma_m(\theta_m e_x) - b\sigma(\theta e_x)}_{=\alpha(e_x, \theta)} \\ & + \underbrace{b_m(\sigma_m(\theta_m e_m) - \sigma_m(\theta_m e_x))}_{=\gamma_m(e_m, e_x)} \end{aligned} \tag{9}$$

TABLE I: Simulation summary

	Constant	Sinusoidal	Smooth pulse
Amplitude	-	1	0.25
Simulation time $[s]$	10	30	5
θ_m	4	4	4
Initial θ	0	0	0
Learning rate c	0.05	0.15	0.005
Convergence time $[s]$	5	25	2

Note the equation is extended by $\pm(a_m x b_m \sigma_m(\theta_m e_x))$, to construct the term α, which is now dependent on e_x, and θ. Two terms remain, namely, $-a_m e$, and $\gamma_m(e_m, e_x)$.

Through substitution of (9) into (8), (10) is obtained.

$$\begin{aligned} \dot{V}(e, \alpha) =& 2e(-a_m e + \alpha(e_x, \theta) + \gamma_m(e_m, e_x)) + 2\alpha c\dot{\alpha}(e_x, \theta) \\ =& -2a_m e^2 \\ & + 2e\gamma_m(e_m, e_x) \\ & + 2\alpha(e_x, \theta)(e + c\dot{\alpha}(e_x, \theta)) \end{aligned} \tag{10}$$

Note that the first term $-2a_m e^2$ is always negative, since $a_m > 0$, and $e^2 > 0$. The second term will always be negative, due to the property of $e = e_x - e_m$. It follows that if $e < 0 \Rightarrow \gamma_m(e_m, e_x) > 0$, which implies that the second term is negative, and if $e > 0 \Rightarrow \gamma_m(e_m, e_x) < 0$, which again implies that the second term is negative. Hence, the second term always stays negative.

It remains to show that the last term does not influence (10) such that it changes sign. Since θ is dependent on time, this implies $\dot{\alpha}(e_x, \theta)$ will include a $\dot{\theta}$ term. Therefore, $\dot{\theta}$ is chosen such that the third term is nullified. The resulting update law is constructed as follows:

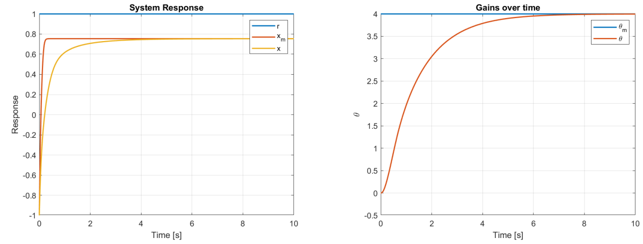
$$\dot{\theta} = \frac{(\frac{1}{c}e + (a - a_m)\dot{x} + b_m \frac{\partial \sigma_m(\tilde{x})}{\partial \tilde{x}}|_{\tilde{x}=\theta_m e_x} \theta_m \dot{e}_x)}{e_x b \frac{\partial \sigma(\tilde{x})}{\partial \tilde{x}}|_{\tilde{x}=\theta e_x}} - \frac{\theta}{e_x}\dot{e}_x \tag{11}$$

Note that due to the way the update law is constructed, it is guaranteed that (7) satisfies the conditions of Definition 3 and, therefore, implies stability according to Definition 4. Furthermore, upon closer examination of the update rule in (11), it follows that an update hold is required when $e_x = 0$. In practice, this means implementing a threshold ε below which the weights are not updated.

IV. Numerical Results

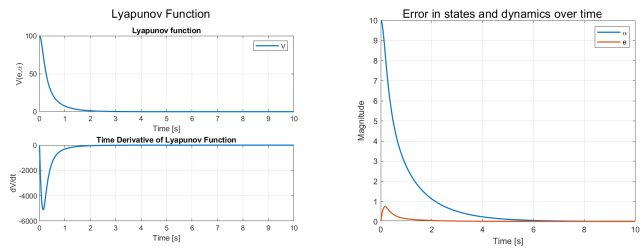
In this section, the numerical results of NMRAC for first-order systems are presented. The example is constructed to show that the parameters of the NN converge to the desired values, using the proposed algorithm. The NN consists of one neuron, with a nonlinear activation function, as defined in (3). Hence, the goal of the simulation is to show that θ converges to θ_m.

The dynamics of the system are characterized by $a_m = a = 10$, $b_m = b = 10$, $\theta_m = 4$, and the activation function saturates at a value of 1. The precision threshold ϵ is chosen to be computer precision. The sampling time for the simulations is $T_s = 0.01$s, and the learning rate differs in each simulation



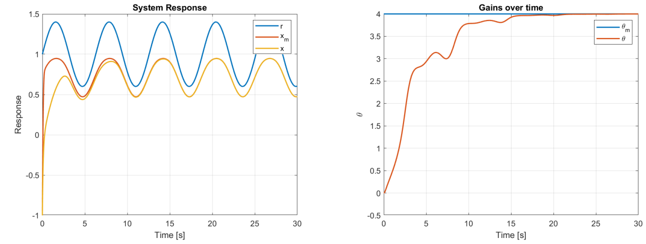
(a) System response (b) Evolution of the updated gain

Fig. 2: Simulation results of nonlinear MRAC with a constant input



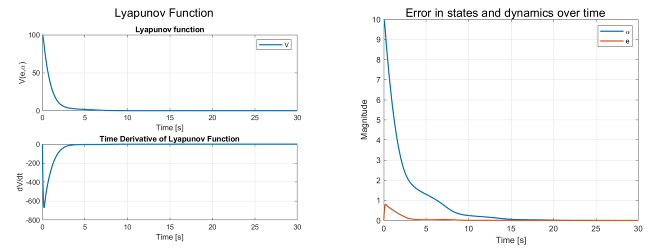
(a) Lyapunov function and its time derivate (b) Error in internal states and dynamics of the system

Fig. 3: Lyapunov function and convergence of the system w.r.t. internal states and dynamics



(a) System response (b) Evolution of the updated gain

Fig. 4: Simulation results of nonlinear MRAC with a sine input



(a) Lyapunov function and its time derivate (b) Error in internal states and dynamics of the system

Fig. 5: Lyapunov function and convergence of the system w.r.t. internal states and dynamics

to showcase faster and slower convergence. Different simulations for different reference signals are provided, including a constant signal, a sinusoidal signal, and a smooth pulse signal. A summary of the important simulation parameters and results can be taken from Table I.

A. *Constant Reference Signal*

Firstly, simulations for a constant reference signal are presented. The system's and model reference's initial conditions are chosen to be $x = x_m = 0$. The value of the constant reference signal is 1. This will result in a steady-state error for the model reference since only a feedback controller is taken into account. However, this steady-state error is ignored, since the learning ability of the controller is of interest and characterized by the response of the model reference. Additionally, the initial value of the weight $\theta = 0$ is chosen. Finally, the system response is simulated for 10 seconds and the learning rate is chosen to be $c = 0.05$.

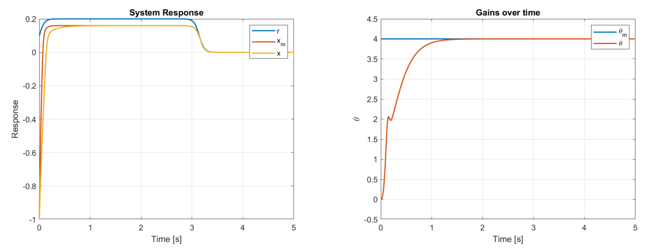
In Fig. 2a, a steady-state error can be observed, which is at around 0.2. As aforementioned, this will be disregarded, since the learning ability of the NNC is of interest. After 5 seconds system convergence is achieved. Furthermore, Fig. 2b depicts the convergence of the weight θ and the error e. Additionally, in Fig. 3a, the Lyapunov function is shown, which clearly converges to zero over time. Additionally, the time derivate of the Lyapunov function is negative and only reaches zero when the Lyapunov function itself converges to zero. This indicates that the proposed learning schematic is stable for both the desired reference signal and the desired parameter converges.

B. *Sine Signal*

The second reference is a sinusoidal signal, with an amplitude of 0.25, to stay in a region that is not too heavily affected by the saturation of the controller. Again, the initial θ and θ_m are chosen to be 0 and 4, respectively. The initial conditions of both the system and the model reference are at $x = x_m = -1$. Finally, the system response is simulated for 30 seconds and the learning rate is chosen to be $c = 0.15$, which corresponds to a less aggressive learning strategy compared to the case of the constant input signal.

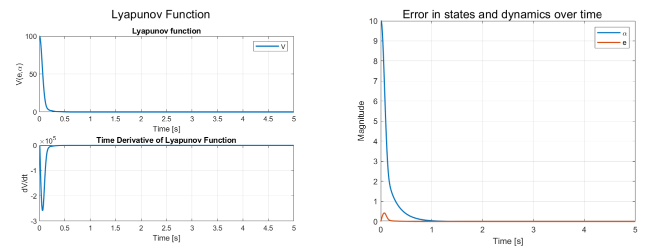
Fig. 4a shows the response of the system. It can be observed that the system state and the model reference state both converge in around 25 seconds. Simultaneously, the parameter θ converges towards the desired value $\theta_m = 4$, as shown in Fig. 4b. Even though the parameter converges, oscillations can be observed. This behavior is expected, due to the update law. Since the update law is dependent on $\dot{e}_x$, it is dependent on $\dot{r}$ and $\dot{x}$. Whenever $\dot{e}_x$ switches signs and simultaneously e is small, $\dot{\theta}$ changes sign, and hence the parameter starts oscillating. This behavior can be influenced by adapting the learning rate. A more aggressive learning strategy will be showcased in the next example.

In Fig. 5a the Lyapunov function is shown, which converges to zero over time. Additionally, the time derivative of the Lyapunov function is negative and only reaches zero when the Lyapunov function itself converges to zero. Again, the results of this simulation indicate that the proposed learning algorithm is stable for the desired reference signal.



(a) System response (b) Evolution of the updated gain

Fig. 6: Simulation results of nonlinear MRAC with a smooth pulse input



(a) Lyapunov function and its time derivate (b) Error in internal states and dynamics of the system

Fig. 7: Lyapunov function and convergence of the system w.r.t. internal states and dynamics

C. Smooth Pulse Signal

The third and last input signal is a pulse signal. More specifically, a smooth pulse signal is used, since the update law requires a continuous time derivative of the reference signal. The smooth pulse signal is constructed, by using a sine function and applying a smooth saturation function, as defined in (3). The signal is chosen to have an amplitude of 0.25 and a bias such that its minimum value touches upon 0. As before, the initial conditions of the system and plant are $x = x_m = -1$. The response is simulated for 5 seconds and a learning rate of $c = 0.005$ is chosen.

Fig. 6a shows the response of the system. It can be observed that the system converges towards the model reference within just under 1 second. The parameter θ converges towards the desired value within 2 seconds, as shown in Fig. 6b. Even though the parameter converges, there are oscillations present. They are present due to the same argument as for the sine signal, as explained in Section IV-B.

The Lyapunov function and its derivative, as shown in Fig. 7a indicate the convergence of the system and the parameters, since V converges to 0 and its time derivative is always negative and converges to zero as well.

V. Discussion

The simulation results show that the proposed update law enables the system to learn the desired parameters across different reference signals. The speed of convergence is influenced by the learning rate c, where smaller values accelerate convergence while larger values slow it down. However, a limitation of the method is its sensitivity to numerical instabilities, which may not be immediately apparent from the presented simulations. These instabilities can be mitigated by reducing the sampling time, thereby improving the precision of integral and derivative approximations and enhancing overall simulation accuracy.

VI. Conclusion

This paper presents a novel algorithm for learning NNCs with guaranteed stability properties. The proposed update law builds upon the theory of linear MRAC and extends it to handle nonlinearities within the controller. Simulation results demonstrate that the algorithm yields the expected results, leading the controlled system to converge towards the model reference system with stability guarantees. While numerical instabilities pose a limitation, appropriate tuning of the sampling time can mitigate their impact. These findings highlight the potential of the proposed approach for adaptive control applications involving nonlinear systems.

VII. Outlook

Future work should focus on refining the update law to enhance robustness against numerical instabilities, potentially through deriving direct discrete update laws. Additionally, extending the methodology to experimental validation on physical systems would provide deeper insights into its real-world applicability and further validate its effectiveness in practical control scenarios.

References

[1] H. Whitaker, “An adaptive system for control of the dynamics performances of aircraft and spacecraft,” *Inst. Aeronatical Sciences*, pp. 59–100, 1959.

[2] 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)*, Nov. 2024, pp. 205–210. DOI: 10.1109/ICRAE64368.2024.10851525. [Online]. Available: https://ieeexplore.ieee.org/document/10851525.

[3] 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)*, ISSN: 2767-9896, Nov. 2021, pp. 1–6. DOI: 10.1109/ICCAD52417.2021.9638766. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9638766.

[4] B. Shackcloth and R. L. But Chart, “Synthesis of Model Reference Adaptive Systems by Liapunov’s Second Method,” *IFAC Proceedings Volumes*, 2nd IFAC Symposium on the Theory of Self-Adaptive Control Systems, Teddington, UK, September 14-17, 1965, vol. 2, no. 2, pp. 145–152, Sep. 1965, ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)69028-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017690281.

[5] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, Jan. 1990, ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90005-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608090900056.

[6] Y. Jiang, C. Yang, J. Na, G. Li, Y. Li, and J. Zhong, "A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots," *Complexity*, vol. 2017, e1895897, Oct. 2017, Publisher: Hindawi, ISSN: 1076-2787. DOI: 10.1155/2017/1895897. [Online]. Available: https://www.hindawi.com/journals/complexity/2017/1895897/.

[7] 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)*, Mauritius, Mauritius: IEEE, Oct. 2021, pp. 1–6, ISBN: 978-1-66541-262-9. DOI: 10.1109/ICECCME52200.2021.9590912. [Online]. Available: https://ieeexplore.ieee.org/document/9590912/.

[8] 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, 2022, Conference Name: IEEE Control Systems Letters, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2022.3181806. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9792308.

[9] M. Revay, R. Wang, and I. R. Manchester, "A Convex Parameterization of Robust Recurrent Neural Networks," in *2021 American Control Conference (ACC)*, New Orleans, LA, USA: IEEE, May 2021, pp. 2824–2829, ISBN: 978-1-6654-4197-1. DOI: 10.23919/ACC50511.2021.9482874. [Online]. Available: https://ieeexplore.ieee.org/document/9482874/.

[10] E. Lavretsky, "Combined/Composite Model Reference Adaptive Control," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2692–2697, Nov. 2009, ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2009.2031580. [Online]. Available: http://ieeexplore.ieee.org/document/5289973/.

[11] S. Slama, A. Errachdi, and M. Benrejeb, "Model reference adaptive control for MIMO nonlinear systems using RBF neural networks," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Hammamet: IEEE, Mar. 2018, pp. 346–351, ISBN: 978-1-5386-4449-2. DOI: 10.1109/ASET.2018.8379880. [Online]. Available: https://ieeexplore.ieee.org/document/8379880/.

[12] K. J. Åström and B. Wittenmark, *Adaptive Control*. Courier Corporation, Jan. 2008, ISBN: 978-0-486-46278-3.

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

[14] K. J. Åström, "Theory and applications of adaptive control—A survey," *Automatica*, vol. 19, no. 5, pp. 471–486, Sep. 1983, ISSN: 0005-1098. DOI: 10.1016/0005-1098(83)90002-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/000510988390002X.

[15] K. Aström, "History of Adaptive Control," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds., London: Springer, 2014, pp. 1–9, ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_120-1. [Online]. Available: https://doi.org/10.1007/978-1-4471-5102-9_120-1.

[16] 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*, 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing 1986, Lund, Sweden, 30 June-2 July 1986, vol. 20, no. 2, pp. 161–166, Jul. 1987, ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)55954-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017559546.

[17] M. Duarte and K. Narendra, "Combined direct and indirect approach to adaptive control," *IEEE Transactions on Automatic Control*, vol. 34, no. 10, pp. 1071–1075, Oct. 1989, ISSN: 00189286. DOI: 10.1109/9.35278. [Online]. Available: http://ieeexplore.ieee.org/document/35278/.

[18] K. S. Narendra and M. A. Duarte, "Robust Adaptive Control Using Combined Direct and Indirect Methods," in *1988 American Control Conference*, Atlanta, GA, USA: IEEE, Jun. 1988, pp. 2429–2434. DOI: 10.23919/ACC.1988.4790133. [Online]. Available: https://ieeexplore.ieee.org/document/4790133/.

[19] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control: With Aerospace Applications* (Advanced Textbooks in Control and Signal Processing). London: Springer London, 2013, ISBN: 978-1-4471-4395-6 978-1-4471-4396-3. DOI: 10.1007/978-1-4471-4396-3. [Online]. Available: https://link.springer.com/10.1007/978-1-4471-4396-3.

[20] G. Tao, S. Chen, X. Tang, and M. J. Suresh, *Adaptive Control of Systems with Actuator Failures*. Springer London, Jun. 2013, ISBN: 978-1-4471-3758-0.

[21] H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002, ISBN: 978-0-13-067389-3.

[22] T. D. C. Thanh and K. K. Ahn, "Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network," *Mechatronics*, vol. 16, no. 9, pp. 577–587, Nov. 2006, ISSN: 0957-4158. DOI: 10.1016/j.mechatronics.2006.03.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415806000523.