# Nonlinear Model Reference Adaptive Control (NMRAC) for Multiple-Input, Multiple-Output 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

Christopher Onder
IDSC
ETH Zürich
Zürich, Switzerland
onder@idsc.mavt.ethz.ch

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

Abstract—Work in Alvaro's feedback

Index Terms—Neural Networks, Gradient Descent, Control, Lyapunov Method

## I. Introduction

### A. Literature Review

Model reference adaptive control (MRAC) was first introduced in the early 1970s by Whitaker et al. [1]. 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 [2], [3].

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 [4]. The MIT method does not come with stability guarantees [5], whereas the Lyapunov method is based on an adaptation rule derived from Lyapunov's second method [6]. 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 [3]:

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. [7], 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 [8], though formal guarantees are still an open question.

CMRACs have been extended to $n$-th order linear systems [9], [10]. Additionally, Lavretsky [11] 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

This method is a continuation of the work presented in wahby-NMRAC-FOS. The proposed method is based on the logic of linear MRAC, as presented in previous works. The novelty of the presented approach lies in the ability of learning a stabilizing controller, in spite of a nonlinear NN being present in the closed-loop system. Furthermore, the method is validated in simulation for a multiple-input multiple-output (MIMO) system and finally applied to a real world system.

## II. Definitions and Nomenclature

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 [12].

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

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

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 $\delta$ 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) $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is defined as:

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

where the activation functions are called $\varphi_i(\cdot)$, $\theta_i$ 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. Throughout this work, an activation function that is designed to model a smooth saturation function is used, as done in [13] and defined as follows:

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

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

Define LQR??

## III. Methodology

### A. Nonlinear Model Reference Adaptive Control

Theorem 1. The function constitutes a Lyapunov function:

$$V(e, \theta) = ||e||_P^2 + ||\alpha(e_x, \theta)||_{\Gamma^{-1}}^2 \tag{4}$$

The system is defined as in equation (5). Furthermore, we define a reference model that imposes a desired behavior on the system, as defined in equation (6). Note, that we assume $A, A_m \in \mathbb{R}^{n\times n}$, $B, B_m \in \mathbb{R}^{n\times m}$, $\theta, \theta_m \in \mathbb{R}^{m\times n}$, and $\phi, \phi_m$ are applied element-wise.

$$\dot{x} = Ax + B\phi(\theta e_x), \quad e_x = r - x \tag{5}$$

$$\dot{x}_m = A_m x_m + B_m \phi_m(\theta_m e_m) \quad e_m = r - x_m \tag{6}$$

We define the internal error of the system with respect to the model reference $e = x_m - x$, which results in the time derivative of the error to be as defined in equation (7).

$$\begin{aligned} \dot{e} =& \dot{x}_m - \dot{x} \\ =& A_m x_m + B_m \phi_m(\theta_m e_m) \\ & - [Ax + B\phi(\theta e_x)] \pm [A_m x + B_m \phi_m(\theta_m e_x)] \\ =& A_m e + \underbrace{B_m \phi_m(\theta_m e_m) - B_m \phi_m(\theta_m e_x)}_{:=\gamma_m(e_m, e_x)} \\ & + \underbrace{(A_m - A)x + B_m \phi_m(\theta_m e_x) - B\phi(\theta e_x)}_{:=\alpha(e_x, \theta)} \\ =& A_m e + \gamma_m(e_m, e_x) + \alpha(ex, \theta) \end{aligned} \tag{7}$$

Note, that the term $\alpha$ describes the difference in the dynamics.

The following Lyapunov candidate is defined:

$$V(e, \theta) = ||e||_P^2 + ||\alpha(e_x, \theta)||_{\Gamma^{-1}}^2 \tag{8}$$

Here $P, \Gamma^{-1} \in \mathbb{R}^{n\times n}$ are positive definite matrices. Note, that the Lyapunov function consists of the squared norm of the error $e$, and the squared norm of the difference in dynamics $\alpha(e_x, \theta)$, rendering the Lyapunov function to be positive definite. Hence, this function can only be equal to zero, when the internal error and the differnce in dynamics are exactly equal. Additionally, showing the negative definitness of its time derivative results in stability of the closed-loop system during the learning process.

Its time derivative is given by (9).

$$\begin{aligned} \dot{V}(e, \theta) =& \dot{e}^T P e + e^T P \dot{e} \\ & + \dot{\alpha}(e_x, \theta)^T \Gamma^{-1} \alpha(e_x, \theta) \\ & + \alpha(e_x, \theta)^T \Gamma^{-1} \dot{\alpha}(e_x, \theta) \\ =& \underbrace{-e^T(A_m^T P + P A_m)e^T}_{\text{(I)}} \\ & + \underbrace{2e^T P \gamma(e_m, e_x)}_{\text{(II)}} \\ & + \underbrace{2\alpha(e_x, \theta)^T (Pe + \Gamma^{-1}\dot{\alpha}(e_x, \theta))}_{\text{(III)}} \end{aligned} \tag{9}$$

Following the logic of linear MRAC, term (III) is used to define the update law through nullification. Hence, it remains to show that the sum of terms (I) and (II) are negative. The negative definiteness of these two terms are system dependent. In section IV, we will show that there exists at least one system, where the sum of terms remains negative. To continue with the derivation of the update law, we will for now assume that the sum of these two terms remain negative.

To derive the MIMO update law, we require term (III) in eq. (9) to be equal to zero. Since, $\Sigma = B\frac{\partial\phi(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta e_x} \in \mathbb{R}^{n\times m}$, the matrix is not invertible. Therefore, its Moore-Penrose inverse is defined as $\Sigma^\dagger$.

$$\Rightarrow \dot{\theta}e_x = \Sigma^\dagger\Big[\Gamma Pe + (A_m - A)\dot{x} + B_m\frac{\partial\phi_m(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta_m e_x}\theta_m\dot{e}_x - B\frac{\partial\phi(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta e_x}\theta\dot{e}_x\Big] \tag{10}$$

Henceforth, for shorter notation the term in big square brackets will now be called $\xi$.

Recall, $\dot{\theta} \in \mathbb{R}^{m\times n}$, $e_x \in \mathbb{R}^n$, $\Sigma^\dagger \in \mathbb{R}^{m\times n}$, $\xi \in \mathbb{R}^n$, and we desire to solve for $\dot{\theta}$. The solution of this equation is given by equation (11), which corresponds to the MIMO nonlinear MRAC update law. Note, $e_x^\dagger$ is the pseudo-inverse of $e_x$, which is defined as $e_x^\dagger = 0$, if $e_x = 0$, and $e_x^\dagger = e_x^T(e_x^T e_x)^{-1}$ otherwise.

$$\dot{\theta} = \Sigma^\dagger\xi e_x^\dagger \tag{11}$$

This system of equations always has an infinite number of solutions, except if $e_x = 0$, and simultaneously $\Sigma^\dagger\xi \neq 0$. Whenever, this case occurs, we propose to not update the weights of the neural network. However, when there is an infinite number of solutions, we can choose values for $\dot{\theta}$, that satisfy this equation. For example, we could choose the solution that minimizes the spectral norm of $\dot{\theta}$. The idea is based on [14], [15], where it was proposed to minimize the spectral norm of the Jacobian of the NN, to improve its robustness properties. In our case, this would correspond to taking the solution, where $\dot{\theta}$ has the smallest maximum singular value. By minimizing the norm, we ensure that no direction in the weight space undergoes excessively large changes during training. This can help in preventing sudden changes of $\theta$, and aid towards a more smooth learning process.

## B. System Model

The proposed method is simulated and implemented on a sea-saw-like system, as shown in fig. 1.

The dynamics of the system are given by the following:

$$\ddot{\theta} = \frac{1}{J}(L_1F_1 - L_2F_2 - f_r\dot{\theta} - d\cos(\theta)F_G) \tag{12}$$

with forces $F_1$, $F_2$ produced by the propellors, inertia $J$, friction $f_r$, and gravity term $F_G$. The input to the system are the thrust forces $F_1$ and $F_2$, which will henceforth

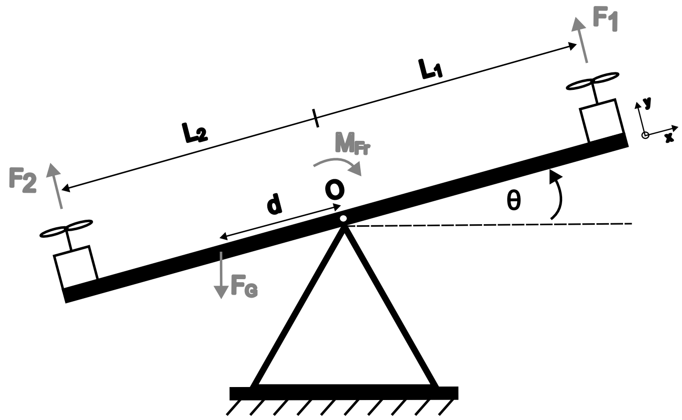



Fig. 1: Double flying arm

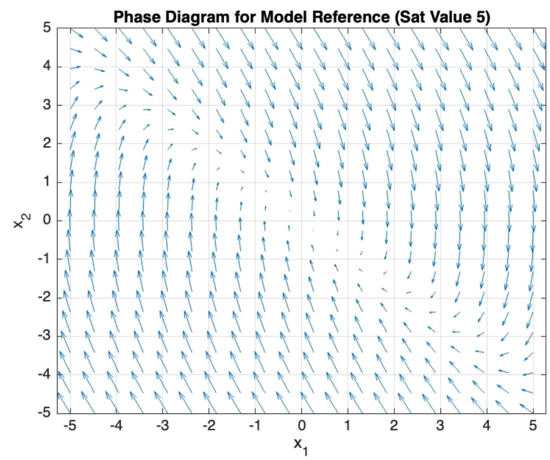



Fig. 2: Phase plot of the model reference

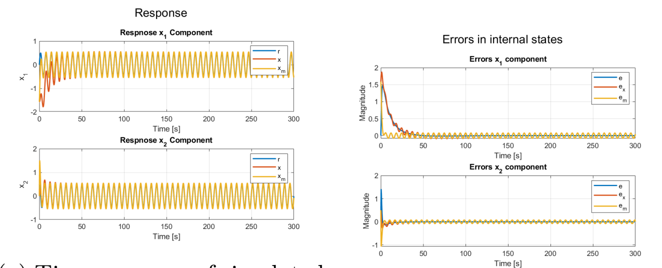
be called $u_1$ and $u_2$, respectively. Hence, the linearized dynamics around the origin are given by:

$$f(x,u) = \begin{bmatrix} 0 & 1 \\ -\frac{dmg}{J} & -\frac{f_r}{J} \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J} & -\frac{L_2}{J} \end{bmatrix} u \tag{13}$$

with states $x = [\theta, \dot{\theta}]^T$ and control input $u = [u_1, u_2]^T$.

## IV. Numerical Results

In this section the simulation results are presented. The simulated system corresponds to the system described in section III-B, where the center of gravity is chosen to be at the origin, meaning $d = 0$. The chosen model reference is a saturated LQR controller, using a smooth saturtation funciton as in eq. (3). A phase plot is provided in fig. 2 to get an idea of the behavior of the model reference, however, its stability is assumed and will not be formally shown in this work.




(a) Time response of simulated system

(b) Error in internal states

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

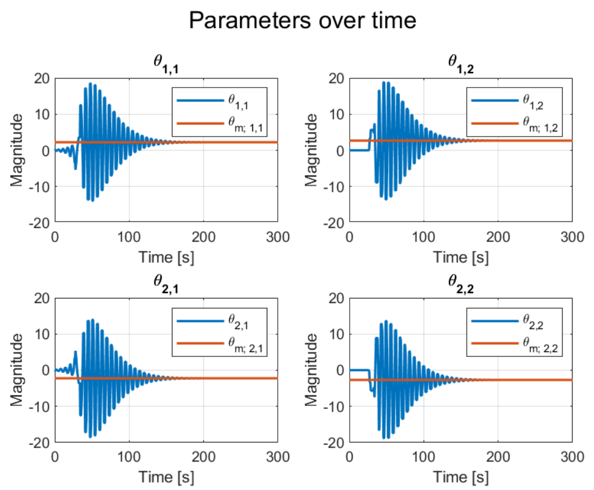
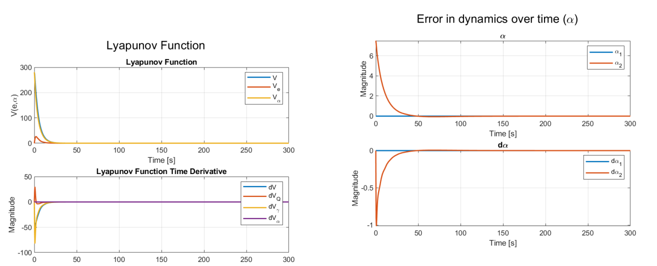



Fig. 4: Evolution of parameters over time




(a) Lyapunov function

(b) Time derivative of the Lyapunov function

Fig. 5: Lyapunov function

## V. Implementation on a System

## VI. Conclusion

## References


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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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/.

[8] 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/.

[9] 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.

[10] 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.

[11] 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/.

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

[13] 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.

[14] Y. Yoshida and T. Miyato, Spectral norm regularization for improving the generalizability of deep learning, 2017. arXiv: 1705.10941 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1705.10941.

[15] A. Johansson, N. Engsner, C. Strannegård, and P. Mostad, “Improved Spectral Norm Regularization for Neural Networks,” in Modeling Decisions for Artificial Intelligence, V. Torra and Y. Narukawa, Eds., Cham: Springer Nature Switzerland, 2023, pp. 181–201, isbn: 978-3-031-33498-6.