Nonlinear Model Reference Adaptive Control (NMRAC) for Miltiple-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

II. Literature Review and Contributions

A. Contributions

III. 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 [1].

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

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

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

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

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.

IV. Methodology

A. Nonlinear Model Reference Adaptive Control

The system is defined as in equation (3). Furthermore, we define a reference model that imposes a desired behavior on the system, as defined in equation (4). 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 ϕ, ϕ_m are applied element-wise.

$$\dot{x} = Ax + B\phi(\theta e_x), \quad e_x = r - x \quad (3)$$

$$\dot{x}_m = A_m x_m + B_m \phi_m(\theta_m e_m) \quad e_m = r - x_m \quad (4)$$

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 (5).

$$
\begin{aligned}
\dot{e} =& \dot{x}_m - \dot{x} \\
=& A_m x_m + B_m \phi_m(\theta_m e_m) \\
& - [Ax + B\phi(\theta e_x)] \pm [A_m x + B_m \phi_m(\theta_m e_x)] \\
=& A_m e + \underbrace{B_m \phi_m(\theta_m e_m) - B_m \phi_m(\theta_m e_x)}_{:=\gamma_m(e_m,e_x)} \\
& + \underbrace{(A_m - A)x + B_m \phi_m(\theta_m e_x) - B\phi(\theta e_x)}_{:=\alpha(e_x,\theta)} \\
=& A_m e + \gamma_m(e_m, e_x) + \alpha(ex_,\theta)
\end{aligned} \tag{5}
$$

Note, that the term α describes the difference in the dynamics.

The following Lyapunov candidate is defined:

$$
V(e,\theta) = ||e||^2_P + ||\alpha(e_x,\theta)||^2_{\Gamma^{-1}} \tag{6}
$$

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 (7).

$$
\begin{aligned}
\dot{V}(e,\theta) =& \dot{e}^T P e + e^T P \dot{e} \\
& + \dot{\alpha}(e_x,\theta)^T \Gamma^{-1} \alpha(e_x,\theta) \\
& + \alpha(e_x,\theta)^T \Gamma^{-1} \dot{\alpha}(e_x,\theta) \\
=& \underbrace{-e^T(A_m^T P + PA_m)e^T}_{\text{(I)}} \\
& + \underbrace{2e^T P \gamma(e_m, e_x)}_{\text{(II)}} \\
& + \underbrace{2\alpha(e_x,\theta)^T (Pe + \Gamma^{-1}\dot{\alpha}(e_x,\theta))}_{\text{(III)}}
\end{aligned} \tag{7}
$$

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 **??**, 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. (7) 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 $\Sigma^\dagger$.

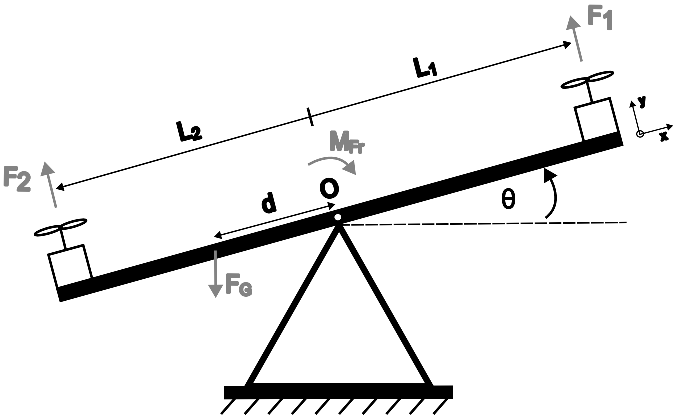


Fig. 1: Double flying arm

$$
\begin{aligned}
\Rightarrow \dot{\theta} e_x = \Sigma^\dagger \Bigg[& \Gamma P e \\
& + (A_m - A)\dot{x} + B_m \frac{\partial\phi_m(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta_m e_x} \theta_m \dot{e}_x \\
& - B\frac{\partial\phi(\tilde{x})}{\partial\tilde{x}}|_{\tilde{x}=\theta e_x} \theta \dot{e}_x \Bigg]
\end{aligned} \tag{8}
$$

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

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 (9), 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.

$$
\dot{\theta} = \Sigma^\dagger \xi e_x^\dagger \tag{9}
$$

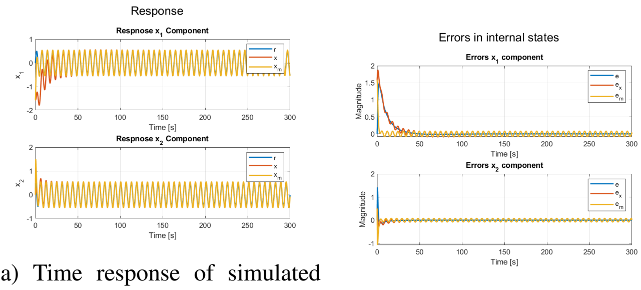
This system of equations always has an infinite number of solutions, except if $e_x = 0$, and simultaneously $\Sigma^\dagger\xi \neq 0$. Whenever, this case occurs, we propose to not update the weights of the neural network. However, when there is an infinite number of solutions, we can choose values 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 [2], [3], 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 θ, 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:

$$
\ddot{\theta} = \frac{1}{J}(L_1 F_1 - L_2 F_2 - f_r\dot{\theta} - d\cos(\theta) F_G) \tag{10}
$$



(a) Time response of simulated system

(b) Error in internal states

Fig. 2: Lyapunov function and convergence of the system w.r.t. internal states and dynamics

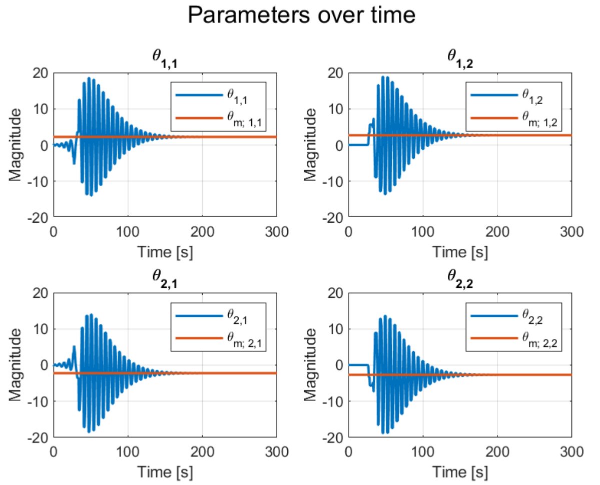


Fig. 3: Evolution of parameters over time

with forces F_1, F_2 produced by the propellors, inertia J, friction f_r, and gravity term F_{G1}.

The forces F_1 and F_2 are thrust forces produced by propellers, which can be characterized by [4]:

$$F_T = \rho D^4 C_{FT} n^2 \tag{11}$$

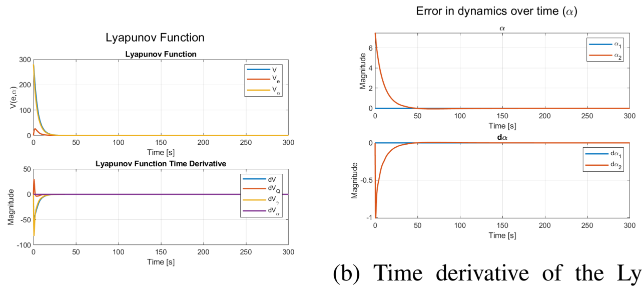
with air density ρ, propeller diameter D, thrust coefficient C_{FT}, and motor speed n.

We need a model that maps the pwm signal to the force!!! The resulting linearized system is given by:

$$f(x,u) = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{f_r}{J} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J} & -\frac{L_2}{J} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{12}$$

V. Numerical Results

In this section the simulation results are presented. The simulated system is a pure double integrator with two states and two inputs. The model reference is chosen to be



(a) Lyapunov function

(b) Time derivative of the Lyapunov function

Fig. 4: Lyapunov function

VI. Implementation on a System

VII. Conclusion

References

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

[2] 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.

[3] 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.

[4] G. J. J. Ducard, *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles*. Springer Science & Business Media, May 2009, Google-Books-ID: UkVIaMsQQZQC, ISBN: 978-1-84882-561-1.