https://doi.org/10.1590/jatm.v17.1361

ORIGINAL PAPER

Modelling and Neuro-Adaptive Robust Control Algorithms for Solid Fuel Rockets

Christian Danner Ramos de Carvalho[1,*], João Viana da Fonseca Neto[1,2]

1. Universidade Federal do Maranhão – Centro de Ciências Exatas e Tecnologia – Programa de Pós-Graduação em Engenharia Aeroespacial – São Luís/MA – Brazil.
2. Universidade Federal do Maranhão – Centro de Ciências Exatas e Tecnologia – Programa de Pós-Graduação em Engenharia Elétrica – São Luís/MA – Brazil

*Correspondence author: eng.chrisdanner@gmail.com

ABSTRACT

This study presents the development of a methodology for designing neuro-adaptive robust controllers based on a reference model associated with an artificial neural network of radial basis functions (ANN-RBF) for solid fuel suborbital rockets. The modelling and neuro-adaptive robust control algorithms for these rockets are presented. Initially, the methodology is evaluated for a robust controller based on a reference model with ANN-RBF for altitude control. The main objective of the control is to suppress the effect of non-linear uncertainties inherent in the process. The method involves mathematical and computational modelling, together with the design of adaptive controllers for stability and performance analysis. The controllers considered include model reference adaptive control (MRAC) techniques and a model reference neuro-adaptive control (MRNAC) approach. The analysis, carried out using computer simulations, evaluates the behavior of each controller in relation to system stability and performance. The final objective is to select the most suitable controller for the suborbital rocket, taking into account the system constraints, robust performance requirements, robust stability, and optimal adaptability. This research promotes the development of adaptive controllers for suborbital rockets, with possible applications in scientific research and commercial launches.

Keywords: Robust control; Rocket; Model reference adaptive control; Model reference neuro-adaptive control; Artificial neural networks; Radial basis function neural networks.

INTRODUCTION

Aerospace has experienced rapid technological evolution in recent decades, driven by a growing need to improve the efficiency, safety, and sustainability of airspace operations. In this context, control systems are key elements in ensuring the proper performance and reliable operation of various aircraft and space vehicles.

A reusable launch vehicle is an aerospace vehicle that can have its parts reused after launch, thus avoiding space debris. The stages of re-entry into the atmosphere, for the subsequent recovery of the parts, are managed by attitude control systems (Alves and Sica 2023), whose main objective is to minimize disturbances from outside the vehicle so that it can maintain its stability (Sampaio 2006).

Satellites play a crucial role in various applications, from telecommunications to environmental monitoring. Therefore, ensuring precise control of these spacecraft in orbit is essential to optimizing their performance and extending their lifespan. Ji and Shi (2023) propose a new velocity control strategy based on adaptive neural dynamics for stable subsatellite recovery of the tethered satellite system.

Received: May 20, 2024 | **Accepted:** Nov. 08, 2024
Peer Review History: Single Blind Peer Review.
Section editor: Luiz Martins-Filho

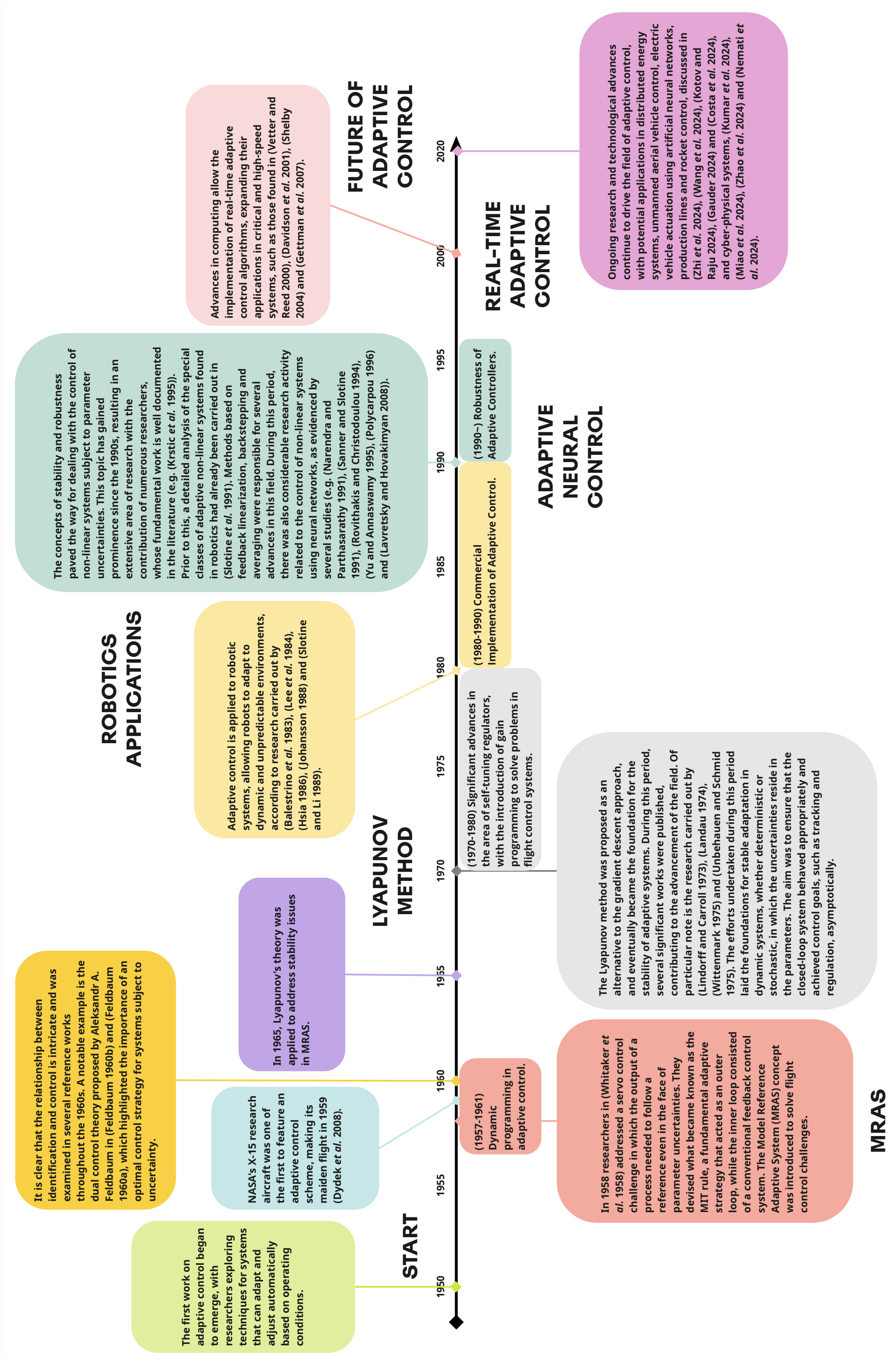
In Farsana *et al.* (2023), a study is presented on the adaptive control of electromechanical actuation systems for launch vehicles. The aim is to overcome the main challenges faced by these systems, such as parameter variations, disturbances, and sensor failures. The proposed adaptive control uses model reference adaptive control (MRAC), which does not require precise information about the location or amplitude of the fault. In addition, the use of the modified Massachusetts Institute of Technology (MIT) rule for the design of MRAC is discussed. The study addresses system modeling, including motor dynamics, load dynamics, and actuator dynamics. Non-linear mathematical models of the system and classical compensations for linear system control are presented. It is shown that both classical compensation and MRAC are capable of tracking the output of the non-linear model of the system. In addition, MRAC is shown to be tolerant of parameter variations and motor winding faults. In summary, MRAC is used to overcome system challenges such as parameter variations, disturbances, and sensor failures. The results show the effectiveness of the proposed control.

The development of suborbital rockets represents a fascinating challenge full of possibilities, propelling science and space exploration to new horizons. In this context, the motivation for this research arises, with the aim of designing an efficient altitude controller for solid-fuel suborbital rockets, guaranteeing the desired stability and performance in the face of the uncertainties inherent in the process. These uncertainties, arising from various sources such as modeling, environmental conditions, payload, engine behavior, and structural integrity, highlight the complexity of the challenge. The implementation of adaptive control systems is crucial to strengthening the rocket's ability to cope with unforeseen situations during flight. Through mathematical and computational modeling, this study explores adaptive control strategies, including direct and indirect MRAC and direct and indirect model reference neuro-adaptive control (MRNAC), as addressed by Lavretsky and Wise (2024). The detailed analysis of these controllers aims to identify their advantages and limitations in order to select the most suitable approach for altitude control, with a view to guaranteeing the stability and performance required for successful suborbital rocket missions.

According to Anavatti *et al.* (2015), driven by the need for highly efficient flight control systems, especially for experimental aircraft such as the X-15, the aeronautical industry saw a significant increase in interest in adaptive controls during the 1950s. In 1951, researchers reached an important milestone by successfully developing a self-optimizing controller for combustion engines, demonstrating its effectiveness in subsequent flight tests. Between 1957 and 1961, there were investigations into the use of dynamic programming in adaptive controls. In 1958, the Model Reference Adaptive System (MRAS) concept was introduced to solve flight control challenges, while in 1965 Lyapunov's theory was applied to address stability issues in MRAS. The following decades, from the 1970s to the 1980s, saw significant advances in the area of self-tuning regulators, with the introduction of gain programming to solve problems in flight control systems. From 1980 onwards, process control systems experienced major advances, leading to the commercial implementation of adaptive controls. In the early 1990s, the focus shifted to the robustness of adaptive controllers, seeking to make them more resilient in the face of uncertainties and variations in the system. Figure 1 describes the timeline of the evolution of robust neuro-adaptive control throughout history, together with the contributions of the main research to the present day.

The introduction of adaptive control was mainly motivated by the need to develop controllers capable of adjusting to variations in system dynamics and disturbance characteristics, as discussed by Aström and Wittenmark (2013). In addition, according to the authors, adaptive techniques have the potential to automatically tune controllers. According to Landau *et al.* (2011), adaptive control encompasses a set of techniques that provide a systematic approach for automatically tuning controllers in real time in order to achieve or maintain a desired level of control system performance when the parameters of the plant's dynamic model are unknown and/or change over time.

The general objective of this research is to develop a methodology to design a robust neuro-adaptive controller by reference model, using a neural network with radial basis functions (RBF), for the altitude control of a solid fuel suborbital rocket. The specific objectives are to obtain a mathematical model for the rocket based on its design characteristics and atmospheric conditions, to develop methods for designing adaptive controllers using direct and indirect approaches, to develop control algorithms for future applications and to validate the controller using simulations, including a comparison of robustness and stability between direct and indirect neuro-adaptive control techniques.



Source: Elaborated by the authors.

Figure 1. Neuro-adaptive robust control timeline.

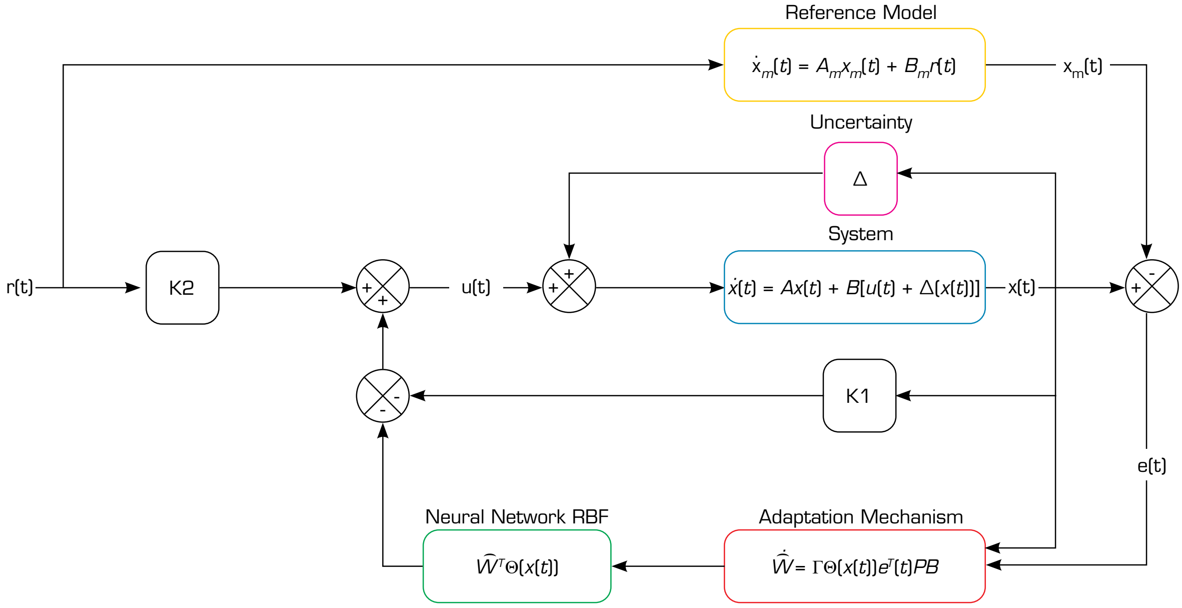
Direct MRNAC

The aim of developing the direct MRNAC (DMRNAC) for the solid-fuel suborbital rocket model stems from the need to cancel out unstructured (non-parametric) time-varying uncertainties that cannot be measured or determined precisely and which can significantly alter the vehicle's trajectory during flight. These uncertainties include unmodeled dynamics of the real system, wind forces, and variations of parameters such as propellant mass, air density, drag force, etc.

An ANN-RBF is implemented to approximate the dynamics of a complex time-varying uncertainty and estimate its parameters, so that later the adaptation mechanism cancels it out, making the system converge to the ideal reference model.

Due to the approximation of the uncertainty by the neural network, a residual error $\epsilon(x)$ is generated. The strategy is to implement a neural network with a sufficient number of neurons (nuclei) to make the residual error smaller quickly, thereby guaranteeing the accuracy of the approximation. Because of $\epsilon(x)$, it is not possible to obtain the derivative of the negative semi-definite Lyapunov candidate function $V \leq 0$, i.e., according to Khalil (2002), the stability of the system over time cannot be guaranteed. For this, an escape modification term is used in the adaptation law.

In the closed loop, in order to guarantee zero error in the permanent regime, a control regulator signal (correction signal) is implemented to recover the system's performance. Figure 2 illustrates the architecture diagram of the DMRNAC controller.



Source: Elaborated by the authors.

Figure 2. DMRNAC controller architecture.

Linearization of the nonlinear model

According to Hodel and Baginski (1995), a non-linear model that includes the elevation angle and the roll angle is proposed in their study, the approximate equations of rocket dynamics were derived, and the rocket dynamics is modeled according to Eqs. 1 and 2:

$$\dot{v}_z = \frac{T}{m}cos(\theta) - g - \frac{c_1}{m}|v|^2cos(\phi) - \frac{c_2 sin(\theta)}{m}|v|^2cos(\phi), \tag{1}$$

$$\dot{v}_x = \frac{T}{m}sin(\theta) - \frac{c_1}{m}|v|^2sin(\phi) - \frac{c_2 sin(\theta)}{m}|v|^2sin(\phi), \tag{2}$$

where m is the mass of the rocket, v is the speed of the rocket, g is the acceleration due to gravity, T is the thrust force, θ is the lift angle, ϕ is the roll angle, δ is the deflection angle of the motor nozzle, and c_1 and c_2 are aerodynamic constants given by Eq. 3:

$$c_1 = c_2 = \frac{C_d \rho S}{2}. \tag{3}$$

For the nominal trajectory studied, the elevation angle of the launch pad is 82°, the elevation angle θ = 8° is considered to be the angle between the initial launch direction and the vertical, in this case 90° - 82° = 8°, and to simplify the model the roll angle can be ϕ = 0°. In this case, the modified differential equation for the z component (altitude) is given by Eq. 4:

$$\dot{v}_z = \frac{T}{m} cos(\theta) - g - \frac{c_1}{m}|v|^2 - \frac{c_2 sin(\theta)}{m}|v|^2, \tag{4}$$

According to Wie *et al.* (2008), to add the δ angle of deflection of the motor nozzle, divide the thrust by cos(δ). In this case, the term $u(t) = 1/\cos(\delta)$ is the system's control input. The model equation results in Eq. 5:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{T\cos(\theta)}{m} u - g - \dfrac{c_1}{m} x_2^2 - \dfrac{c_2 \sin(\theta)}{m} x_2^2 \end{cases} \tag{5}$$

The non-linear model given by Eq.5 is linearized by Taylor series expansion, according to Ogata (2010), at the operating point (x_1 = 2,595 m, x_2 = 1,136 m/s), equivalent to the position and final velocity of the propelled phase of the suborbital rocket (t = 4 s). The values refer to the operating point at the end of the propelled phase, which for the model in question lasts approximately 4 s of propellant burn. The control will only act in the propelled phase, the reference trajectory is a set of points to be followed, and the operating point (x_1, x_2) is the end point of the reference trajectory, which the rocket will follow as a target before entering its ballistic (unpropelled) phase.

The linear model of the system to be controlled is given by Eq. 6:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -0.1190 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 373.31 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{6}$$

The controllability and observability requirements for the system were met, as discussed by Lavretsky and Wise (2024).

Problem formulation

According to Singh and Pal (2019), the MRAC method is an approach used in the design of an adaptive controller that modifies the parameters of the controller so that the output of the real system tracks the output of a reference model under the same reference input.

In line with the robust adaptive control formulations and structures found in the literature, such as Arabi *et al.* (2019), Glushchenko and Lastochkin (2022), and Gruenwald *et al.* (2017), this study addresses the following structure and mathematical modeling for a DMRNAC controller.

As discussed by Lavretsky and Wise (2024), the dynamic system with uncertainty is considered according to Eq. 7:

$$\dot{x} = Ax + B[u + \Delta(x)], \quad x(0) = x_0, \tag{7}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, $B \in \mathbb{R}^{n\times m}$ is the known control matrix. Furthermore, it is assumed that the pair (A, B) is controllable. In Eq. 7, the unknown vector function, which may be non-linear, $\Delta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, describes the intrinsic uncertainty of the system.

The parameterization of the unstructured uncertainty $\Delta(x)$ is given by Eq. 8:

$$\Delta(x) = W^T\Theta(x) + \varepsilon(x), \quad \|\varepsilon(x)\|_2 \leq \bar{\varepsilon}, \tag{8}$$

where W^T is a vector of unknown weights, to be estimated by the parameter update law, Θ(x) is a vector and includes RBF in its elements, and $\varepsilon : \mathbb{R}^n \to \mathbb{R}^m$ is the residual error.

It should be noted that the residual error ε(x) comes from the universal approximation theorem for RBF neural networks (Lavretsky and Wise 2024), and by including more neurons in the neural network it is possible to reduce this error. In addition, in second-order cases, as discussed in this research, or in high-order cases, it is necessary to include RBF for each x_i to cover each compact domain D_i.

The vector Θ(x) is given by Eq. 9:

$$\Theta(x) = [\theta_1(x_1), \ldots, \theta_n(x_1), \theta_{n+1}(x_2), \ldots, \theta_{2n}(x_2), 1]^T. \tag{9}$$

Therefore, the system is represented by Eq. 10:

$$\dot{x} = Ax + B[u + W^T\Theta(x) + \varepsilon(x)] \tag{10}$$

RBF neural network

For some control problems, an artificial neural network (ANN) can be trained to remember how to regulate a system by repeatedly providing examples of how to perform such a task. After this learning, the neural network can be used to retrieve the control input for each value of the detected output (Levine 2011).

According to Lavretsky and Wise (2024), an RBF neural network feedforward is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$, according to Eq. 11:

$$NN(x) = \widehat{W}_0^T \begin{bmatrix} exp(-\|x-c_i\|^2 w_i) \\ \vdots \\ exp(-\|x-c_n\|^2 w_n) \end{bmatrix} + b = \underbrace{\begin{bmatrix} \widehat{W}_0^T & b \end{bmatrix}}_{\widehat{W}^T} \underbrace{\begin{bmatrix} \theta_i(x) \\ \vdots \\ \theta_n(x) \\ 1 \end{bmatrix}}_{\Theta(x)} = \widehat{W}^T\Theta(x), \tag{11}$$

where $\widehat{W} = [\widehat{W}_0^T\ b]^T \in \mathbb{R}^{(n+1)\times m}$ is the vector of weights, $c_i \in \mathbb{R}^n$ is the center of the i-th receptive field, $b \in \mathbb{R}^m$ is the bias of the neural network, and $\Theta(x) = [\theta_i(x) \ldots \theta_n(x)\ 1]^T \in \mathbb{R}^{n+1}$ is the regressor vector, whose components are the basis (activation) functions, given by $\theta_i(x) = exp(-\|x - c_i\|^2 w_i)$, and the unit function. The terms w_i to w_n are the widths of the RBF and form a symmetric positive definite matrix, w_i is calculated by Eq. 12:

$$w_i = \frac{1}{2\sigma_i^2}, \quad (i = 1, \ldots, n). \tag{12}$$

The standard deviation σ of all components of the isotropic Gaussian RBF is set to:

$$\sigma = \frac{d_{\max}}{\sqrt{2n}}, \tag{13}$$

where n is the number of centers and d_{max} is the maximum distance between the chosen centers. This formula ensures that the individual RBFs are neither too sharp nor too flat, avoiding both extreme conditions (Lavretsky and Wise 2024).

Radial basis function networks offer a simple architecture, robust generalization, good noise tolerance, and online learning. In addition, from a generalization point of view, RBF networks must react effectively to patterns not used during training (Heidari *et al.* 2023).

Neuro-adaptive control law

According to Gruenwald *et al.* (2018), when the actuator dynamics are not present, this problem is solved by considering the control law given by Eq. 14:

$$u = u_n + u_a, \tag{14}$$

where u_n and u_a are the nominal and adaptive control laws, respectively. Therefore, the nominal control law is given by Eq. 15:

$$u_n = -K_1 x + K_2 r, \tag{15}$$

where $K_1 \in \mathbb{R}^{m\times n}$ and $K_2 \in \mathbb{R}^{m\times n}$ are the nominal feedback and feedforward gains, respectively, so that in closed loop, Eqs. 16 and 17 are obtained:

$$A_m = A - BK_1, \tag{16}$$

$$B_m = BK_2. \tag{17}$$

The (A, B) pair of the system is controllable, so the state feedback gain K_1 is calculated by pole allocation or by linear quadratic regulator (LQR). The direct-feedback gain K_2 is calculated according to Eq. 18:

$$K_2 = -(C(A - BK_1)^{-1}B)^{-1} \tag{18}$$

If $A_m = A - BK_1$ and $B_m = BK_2$, then the reference model of the system is described by Eq. 19:

$$\dot{x}_m = A_m x_m + B_m r \tag{19}$$

If the uncertainty can be parameterized as $\Delta(x) = W^T\Theta(x)$, then the adaptive control signal that cancels it is approximated by an ANN-RBF, according to Eq. 11. The adaptive control signal u_a is represented by Eq. 20:

$$u_a = -\widehat{W}^T\Theta(x). \tag{20}$$

Substituting Eqs. 15 and 20 in Eq. 14, Eq. 21 is obtained:

$$u = -K_1 x + K_2 r - \widehat{W}^T\Theta(x) \tag{21}$$

System error dynamics

According to Dogan *et al.* (2020), the system error is given by Eq. 22:

$$e = x - x_m, \tag{22}$$

and the dynamics of the error is given by Eq. 23:

$$\dot{e} = \dot{x} - \dot{x}_m. \tag{23}$$

Substituting Eqs. 10, 19, and 21 in Eq. 23, Eq. 24 is obtained:

$$\dot{e} = A_m e - B\widetilde{W}^T\Theta(x) + B\varepsilon(x), \tag{24}$$

where $\widetilde{W}$ is the weight update error matrix, represented by Eq. 25:

$$\widetilde{W} = \widehat{W} - W \tag{25}$$

Adaptation law

As presented by Yucelen and Calise (2010), an adaptation law must be chosen to bring the system error asymptotically to zero. A generic equation is written for the dynamics of weight estimation given by Eq. 26:

$$\dot{\widehat{W}} = \Gamma f_a(.), \quad \Gamma > 0, \tag{26}$$

where $\Gamma \in \mathbb{R}^+$ is the constant and positive learning rate and $f_a(.)$ is a generic adaptation function.

The dynamics of the weight update error is represented by Eq. 27:

$$\dot{\widetilde{W}} = \dot{\widehat{W}} - \dot{W}. \tag{27}$$

It should be noted that, assuming that the vector of real uncertainty weights Δ(x) are time invariant parameters, the variation of W is zero. Therefore, replacing Eq. 26 in Eq. 27, and taking the vector of real weights W as a constant, Eq. 28 is obtained:

$$\dot{\widetilde{W}} = \Gamma f_a(.) \tag{28}$$

According to Gruenwald *et al.* (2017), using the system error and the weight update error, it is possible to construct a candidate Lyapunov function described by Eq. 29:

$$V(e, \widetilde{W}) = e^T P e + tr\widetilde{W}^T\Gamma^{-1}\widetilde{W}. \tag{29}$$

It can be seen that $V(0, 0) = 0$ and $V(e, \widetilde{W}) > 0$ for all $(e, \widetilde{W}) \neq (0, 0)$, and its derivative is given by Eq. 30:

$$\dot{V}(e, \widetilde{W}) = 2e^T P\dot{e} + 2tr\widetilde{W}^T\Gamma^{-1}\dot{\widetilde{W}}. \tag{30}$$

Substituting Eqs. 24 and 28 in Eq. 30, Eq. 31 is obtained:

$$\dot{V}(e, \widetilde{W}) = 2e^T PA_m e + 2tr\widetilde{W}^T\left[f_a(.) - \Phi(x)e^T PB\right] \tag{31}$$

From Eq. 31, it can be seen that if a generic adaptation function such as that described by Eq. 32:

$$f_a(.) = \Phi(x)e^T PB, \tag{32}$$

The term $[f_a(.) - \Phi(x)e^T PB]$ will be zero, so the derivative of the candidate Lyapunov function is negative semidefinite, according to Eq. 33:

$$\dot{V}(e, \widetilde{W}) = -e^T Re. \tag{33}$$

According to Arabi *et al.* (2020), $P \in \mathbb{R}_+^{n\times n}$ is the solution of the Lyapunov equation, given by Eq. 34:

$$0 = A_m^T P + PA_m + R. \tag{34}$$

Therefore, substituting Eq. 32 into Eq. 26, the adaptation law is given by Eq. 35:

$$\dot{\widehat{W}} = \Gamma\Phi(x)e^T PB. \tag{35}$$

σ-modification

Because of the term $B\varepsilon(x)$ in Eq. 24, it is not possible to reach $\dot{V} \leq 0$. In this case, it is necessary to use an escape modification; in this work, the σ-modification is used. This approach is valid both when W is constant and when $W(t)$ is time-varying.

According to Lavretsky and Wise (2024), the adaptation law with the σ-modification is given by Eq. 36:

$$\dot{\widehat{W}} = \Gamma\left[\Theta(x)e^T PB - \sigma\widehat{W}\right], \quad \Gamma > 0, \tag{36}$$

where $\Gamma \in \mathbb{R}^+$ is a constant and positive learning rate and $-\sigma\widehat{W}$ is the σ-modification term, which guarantees the stability of the system over time.

The σ-modification term in Eq. 36 is used to damp possible oscillations in the control signal that could be induced in the case of high-gain adaptation (Fravolini *et al.* 2015).

According to Orłowski *et al.* (2022), the σ-modification does not guarantee, in general, that the state reaches an arbitrarily small neighborhood of the origin (for sufficiently small σ) and remains there indefinitely, even in a delay-free context.

In Stepanyan and Kalmanje (2010), a complete performance analysis is presented, including asymptotic and transient analysis of the modified MRAC (M-MRAC) architecture with σ-modification.

Stability

So, according to Khalil (2002), if $\dot{V}(e,\widetilde{W}) \leq 0$, there is a stable system according to Lyapunov, but the asymptotic stability of the system must be verified.

In the theory of ordinary differential equations, Barbalat's lemma is a mathematical result about the asymptotic properties of functions and their derivatives. When used correctly for dynamic systems, it can lead to the solution of many asymptotic stability problems, including compartmental epidemiological models. In general terms, it is the convergence to zero of a sufficiently well-behaved function whose integral is bounded (Zeraick Monteiro and Rodrigues Mazorche 2023). Barbalat's lemma is a useful technique for assessing the instability of non-autonomous systems. For non-autonomous systems with piecewise continuous dynamics, an extended adaptation of the lemma was presented by Su and Huang (2011).

According to Slotine and Li (1991) and as demonstrated by Lu *et al.* (2020), it is stated that:

- If $V(e,\widetilde{W})$ is lower bounded and $\dot{V}(e,\widetilde{W}) = -e^T Re \leq 0$, then $V(e,\widetilde{W})$ approaches a finite limit when $t \to \infty$.
- Moreover, if $\ddot{V}(e,\widetilde{W})$ is bounded, then $\dot{V}(e,\widetilde{W})$ is a uniformly continuous function of time.
- Then, by Barbalat's Lemma, as discussed by Lavretsky and Wise (2024), proved and exemplified by Hou *et al.* (2010), it follows that $\lim_{t\to\infty} \dot{V}(e,\widetilde{W}) = 0$, as long as the matrix R is positive definite, and therefore the error e converges asymptotically to zero.

Therefore, the state $x(t)$ of the system with uncertainties will approach the desired ideal state $x_m(t)$ (Eq. 37):

$$\lim_{t\to\infty} [x(t) - x_m(t)] = 0. \tag{37}$$

Command regulator signal

According to Yucelen and Johnson (2013), a command regulator signal is proposed to recover the steady-state performance of a closed-loop controlled system using adaptive methods. The methodology proposed by the authors guarantees both transient and steady-state performance in the closed loop, and can shape the transient response by adjusting the trajectory of the reference with the command regulator.

According to Yucelen and Johnson (2013), the control architecture by command regulator signal for the adaptive control problem with time-varying non-parametric uncertainty is given by Eq. 38:

$$r_v = r + Gv, \tag{38}$$

where $r_v \in \mathbb{R}^m$ is the total reference command, $r \in \mathbb{R}^m$ is the command given uniformly continuous and limited, and $Gv \in \mathbb{R}^m$ is the command regulator signal, with $G \in \mathbb{R}^{m \times n}$ being the matrix of the command regulator signal added to the total reference signal, defined by Eq. 39, and $v \in \mathbb{R}^n$ is the output of the command regulator:

$$G = K_2^{-1}(B^T B)^{-1} B^T. \tag{39}$$

The dynamic system of the control regulator is given by Eqs. 40 and 41:

$$\dot{\Psi} = -\lambda\Psi + \lambda e, \quad \Psi(0) = 0, \quad t \in \mathbb{R}^+, \tag{40}$$

$$v = \lambda\Psi + (A_r - \lambda I_n)e, \tag{41}$$

where $\Psi \in \mathbb{R}^n$ is the command regulator state vector and $\lambda \in \mathbb{R}^+$ is the command regulator gain. Therefore, substituting Eqs. 15 and 20 in Eq. 14 and adding the correction signal given by Eq. 38, the total control law is given by Eq. 42:

$$u = -K_1 x + K_2(r + Gv) - \widehat{W}^T \Theta(x). \tag{42}$$

Substituting the new reference command from Eq. 38 into Eq. 19 results in Eq. 43:

$$\dot{x}_m = A_m x_m + B_m r + B(B^T B)^{-1} B^T v. \tag{43}$$

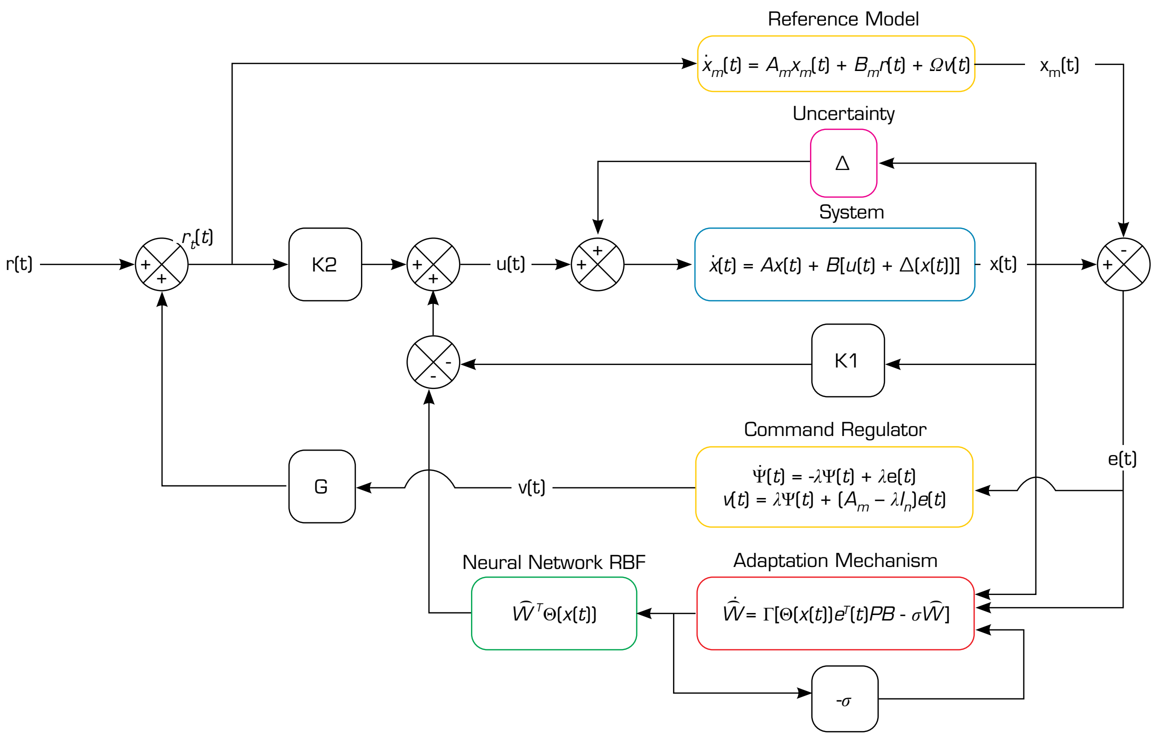
Therefore, the new reference model is represented by Eq. 44:

$$\dot{x}_m = A_m x_m + B_m r + \Omega v, \tag{44}$$

where Ω is the command regulator signal matrix added to the reference model, given by Eq. 45:

$$\Omega = B(B^T B)^{-1} B^T. \tag{45}$$

After implementing the dynamic system of the command regulator to recover system performance and adding the σ-modification term, the diagram of the complete control loop architecture is represented by Fig. 3.



Source: Elaborated by the authors.

Figure 3. DMRNAC control architecture with command regulator and σ-modification.

Indirect MRNAC

The indirect MRNAC (IMRNAC) represents a sophisticated and effective approach to controlling dynamic systems subject to uncertainty and variation, distinguished by its ability to dynamically adjust its parameters based on the system's actual response.

By employing the indirect methodology, the controller is based on identifying the dynamic model of the system, using the response to the control signal to adjust the adaptive parameters. This approach excels in environments where the system's characteristics are subject to change, offering the flexibility needed to deal with uncertainties, non-linearities, and time variations.

Problem formulation

According to the adaptive control frameworks in Arabi and Yucelen (2019) and Arabi *et al.* (2018), the indirect method approach consists of mathematically modeling the real system when its parameters are unknown, and a new state-space structure is set up in order to parameterize the complete uncertainty, containing not only the external disturbances inherent in the process but also the unknown parameters of the system. Therefore, all unknown parameters are estimated by the ANN-RBF and consequently canceled out as uncertainties. Through this new state space structure, other constant matrices A and B are used in the reference model.

Considering the system of Eq. 5 and replacing the parameters with uncertainties with the coefficients α_1, α_2, α_3, and α_4, Eq. 46 is obtained:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \underbrace{\frac{1}{m}}_{\Lambda} \left(\underbrace{T\cos(\theta)}_{\alpha_1} u - \underbrace{mg}_{\alpha_2} - \underbrace{c_1}_{\alpha_3} x_2^2 - \underbrace{c_2\sin(\theta)}_{\alpha_4} x_2^2 \right) \end{cases} \tag{46}$$

Assuming that none of the parameters of the real system is known, the system model can be represented by Eq. 47:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \Lambda \left(\alpha_1 u - \alpha_2 - \alpha_3 x_2^2 - \alpha_4 x_2^2 \right). \end{cases} \tag{47}$$

Note that a control law u can be chosen, which cancels out all the parameters with uncertainties present in the system (Eq. 48):

$$u = \frac{\alpha_2 + (\alpha_3 + \alpha_4)x_2^2}{\alpha_1}. \tag{48}$$

It is therefore possible to represent the system in state space with the model in Eq. 49:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\underbrace{\begin{bmatrix}\frac{1}{m}\end{bmatrix}}_{\Lambda}\left(u + \underbrace{\begin{bmatrix}\alpha_2/\alpha_1\\ \alpha_3/\alpha_1\\ \alpha_4/\alpha_1\end{bmatrix}^T}_{W_0^T}\underbrace{\begin{bmatrix}1\\ x_2^2\\ x_2^2\end{bmatrix}}_{\Theta_0(x)}\right) \tag{49}$$

Organizing the terms, the system in state space is represented by Eq. 50:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\Lambda\left[u + W_0^T\Theta_0(x)\right], \tag{50}$$

where W_0^T is the vector of initial weights and $\Theta_0(x)$ is the vector of initial basis functions. The equation for the nominal control law is the same as that described in Eq. 15, as are the calculations for the state feedback and direct supply gains, K_1 and K_2, respectively. In addition, the reference model used is the same as that presented in Eq. 19. Therefore, the system is represented by Eq. 51:

$$\dot{x} = Ax + B\Lambda[u + W_0^T\Theta_0(x)]. \tag{51}$$

Substituting the total control input from Eq. 14 and adding the terms Bu_n and $-Bu_n$ to Eq. 51, Eq. 52 is obtained:

$$\begin{aligned}\dot{x} &= Ax + B\Lambda[u_n + u_a + W_0^T\Theta_0(x)] + Bu_n - Bu_n\\ &= \underbrace{Ax + Bu_n}_{A_m x + B_m r} + B\Lambda[\underbrace{u_n - \Lambda^{-1}u_n}_{(I-\Lambda^{-1})u_n} + u_a + W_0^T\Theta_0(x)]\\ &= A_m x + B_m r + B\Lambda\left(u_a + \underbrace{\begin{bmatrix}W_0\\ I-\Lambda^{-1}\end{bmatrix}^T\begin{bmatrix}\Theta_0(x)\\ u_n\end{bmatrix}}_{W^T\Theta(x)}\right)\\ \dot{x} &= A_m x + B_m r + B\Lambda[u_a + W^T\Theta(x)].\end{aligned} \tag{52}$$

If the uncertainty $\Delta(x)$ is parameterized as $W^T\Theta(x)$, then the adaptive control signal, which cancels it, is approximated by an ANN-RBF, according to Eq. 11. The adaptive control signal u_a is represented by Eq. 53:

$$u_a = -\widehat{W}^T\Theta(x). \tag{53}$$

The dynamics of the error between the states of the system is given by Eq. 54:

$$\begin{aligned}e &= x - x_m\\ \dot{e} &= A_m e + B\Lambda\underbrace{[u_a + W^T\Theta(x)]}_{-\widetilde{W}^T\Theta(x)}\\ \dot{e} &= A_m e - B\Lambda\widetilde{W}^T\Theta(x) \quad \because \quad \widetilde{W} = \widehat{W} - W\end{aligned} \tag{54}$$

The adaptation law is given by Eq. 55:

$$\dot{\widehat{W}} = \Gamma\Theta(x)e^{T}PB, \quad \Gamma > 0, \tag{55}$$

where $\Gamma \in \mathbb{R}^{+}$ is a constant and positive learning rate and $P \in \mathbb{R}_{+}^{n \times n}$ is the solution of the Lyapunov equation according to Eq. 34.

RESULTS AND DISCUSSION

This section presents a detailed analysis of the response of the controlled system and the reference model over time, providing a visual comparison of the control's performance. In addition, graphs illustrating the tracking error between the controlled system and the reference model over time are displayed, allowing an accurate assessment of the control's precision and the system's ability to follow the reference. The control signals generated by the adaptive controller are plotted on separate graphs, offering insights into how the controller responds to changes in the system. Performance metrics such as mean absolute error (MAE), mean square error (MSE), Root MSE (RMSE), coefficient of determination (R-squared), and mean absolute percentage error (MAPE) are presented in tables or graphs, providing a quantitative assessment of the controller's performance. In addition, the results of the robustness of the controller in relation to uncertainties in the system are discussed, such as variations in system parameters or external disturbances, by means of specific graphs or tables. Finally, a comparison is made between the direct and indirect methods, highlighting their respective advantages and limitations in the context of adaptive control. The controllers and simulators developed in the study are presented in Algorithms 1 and 2 for the direct method (DMRNAC), and Algorithms 3 and 4 for the indirect method (IMRNAC).

Algorithm 1 implements DMRNAC robust control using the Lyapunov rule. The main blocks include: Block 1 – Setup of initial conditions with definitions of the dynamic system matrices and initial states; Block 2 – Design of the nominal controller with calculation of gains by pole allocation, where K_1 is determined by Ackerman's formula and K_2 by Eq. 18; Block 3 – Definition of reference model matching conditions, where Am and Bm are calculated according to Eqs. 16 and 17; Block 4 – Calculation of the correction signal for performance recovery, using G and Ω, according to Eqs. 39 and 45; Block 5 – Solving an algebraic Lyapunov equation to obtain the P matrix, given by Eq. 34; and Block 6 – Parameterization of an ANN for estimation and cancellation of system uncertainties, including the parameters $\widehat{W}$, *width*, Θ, and others, according to Eqs. 12 and 13.

Algorithm 1 DMRNAC Controller

Block 1 - Setup (Initial Conditions)
$A \leftarrow [0,1;0,-0.1190]$; $B \leftarrow [0;373.31]$; $C \leftarrow [1,0]$; $D \leftarrow 0$;
x; x_m; $x_{mi} \leftarrow [0;0]$; $\Psi \leftarrow [0;0]$;
Block 2 - Nominal Controller Design
$desired\_poles \leftarrow [-1-2]$;
$K_1 \leftarrow \text{ackerman}(A, B, desired\_poles)$;
$K_2 \leftarrow -(C*(A-B*K_1)^{-1}*B)^{-1}$;
Block 3 - Reference Model - Matching Conditions
$A_m \leftarrow A - B*K_1$; $B_m \leftarrow B*K_2$;
Block 4 - Correction Signal (Performance Recovery)
$G \leftarrow (K_2)^{-1}*(B^T*B)^{-1}*B^T$; $\Omega \leftarrow B*(B^T*B)^{-1}*B^T$;
Block 5 - Lyapunov Algebraic Equation
$Q_{lyap} \leftarrow I_{n\times n}$; $P \leftarrow A_m^T*P+P*A_m+Q_{lyap}=0$;
Block 6 - Artificial Neural Network - RBF
$n \leftarrow 25$; $b \leftarrow 5$; $\widehat{W} \leftarrow 0_{(2*n+1)\times 1}$; $\gamma \leftarrow 2$;
$\sigma \leftarrow 500$; $\lambda \leftarrow 5$; $c \leftarrow 2*b/(n-1)$;
Neurons' Centers Vector
$centers \leftarrow 0_{1\times n}$;
$centers(1) \leftarrow -b$;
    for i to $1:n-1$ **do**
        $centers(i+1) \leftarrow -b+i*c$;
    end for
$d_{max} \leftarrow 2*b$;
$width \leftarrow 1/(2*(d_{max}/(2*n)^{\frac{1}{2}})^2)$;
$\Theta \leftarrow 0_{(2*n+1)\times 1}$;
End of Algorithm 1

Algorithm 2 implements the simulator for the DMRNAC control system in discrete time, where the system is exposed to a non-linear uncertainty. Its main blocks include: Block 1 – Initialization of the simulation conditions with time and counter settings; Block 2 – Definition of a non-linear uncertainty function $\Delta(x)$ as a function of the current state; Block 3 – Definition of the reference signal r according to the current time; Block 4 - assembly of the regressor vector Θ of the ANN using RBFs and the bias according to Eq. Block 5 – Calculation of the correction signal for performance recovery with Ψ update and calculation of v according to Eqs. Block 6 – Calculation of the control signal u incorporating nominal control gains K_1 and K_2, reference signal r, correction signal Gv, and uncertainty estimate $\widehat{W}^T \Theta(x)$ according to Eq. 42, as well as updating x_m and applying the W adaptation law according to Eq. 55; and Block 7 – Updating the x state of the real system, taking into account the $\Delta(x)$ uncertainty and calculating the y output, repeating the iterations until the final simulation time.

Algorithm 2 DMRNAC Simulator

```
Block 1 - Setup - Initial Simulator Conditions
ft ← 20; dt ← 0.001;
   for k to 0 : dt : ft do
      Block 2 - Uncertainty
      Δ(x) ← 1 + x(1) + x(2) + x(1)^2 + sin(x(1)) + cos(x(1)) + sin(x(2)) + cos(x(2));
      Block 3 - Reference Signal
         if k ≤ 10 then
            r = 1;
         else
            if k > 10 then
               r = −1;
            end if
         end if
      Block 4 - NNA Regressor Vector Assembly
         for i to 1 : n do
            Θ(i) ← exp(−width ∗ |x(1) − centers(i)|^2);
            Θ(i+n) ← exp(−width ∗ |x(2) − centers(i)|^2);
         end for
      Θ(2∗n+1) ← 1;
      Block 5 - Correction Signal for Performance Recovery
      Ψ ← Ψ + dt ∗ (−λ ∗ (Ψ − (x − x_m)));   v ← λ ∗ Ψ + (A_m − λ ∗ I_{n×n} ∗ (x − x_m));
      Block 6 - Updates - Control Signal - Reference Model - Adaptation Law
      u ← −K_1 ∗ x + K_2 ∗ (r + G ∗ v) − Ŵ^T ∗ Θ;   x_m ← x_m + dt ∗ (A_m ∗ x_m + B_m ∗ r + Ω ∗ v);
      x_mi ← x_mi + dt ∗ (A_m ∗ x_mi + B_m ∗ r);   Ŵ ← Ŵ + dt ∗ (γ ∗ (Θ ∗ (x − x_m)^T ∗ P ∗ B − σ ∗ Ŵ));
      Block 7 - Real System with Uncertainty - Output Measurement
      x ← x + dt ∗ (A ∗ x + B ∗ (u + Δ(x)));
   end for
End of Algorithm 2
```

Algorithm 3 implements IMRNAC robust control. Its main blocks include: Block 1 – Initial configuration of the system with definition of the matrices and initial states; Block 2 – Definition of the actual system parameters and the initial vector W_0; Block 3 – Design of the nominal controller, where K_1 and K_2 are calculated; Block 4 – Setting the reference model matching conditions A_m and B_m are defined; Block 5 – Calculating the correction signal for performance recovery with G and Ω; Block 6 – Solving the Lyapunov algebraic equation (Eq. 34); and Block 7 – Defining the parameters of the RBF neural network, including the number of neurons, domain boundary, vector of centers, and width of neurons for estimating and compensating for uncertainties.

Algorithm 3 IMRNAC Controller

```
Block 1 - Setup (Initial Conditions)
A ← [0,1;0,0]; B ← [0;1]; C ← [1,0]; D ← 0;
x; x_m; x_mi ← [0;0]; Ψ ← [0;0];
Block 2 - Nominal Controller Design
a1 ← 25497; a2 ← 670.0230; a3 ← 0.0031; a4 ← 0.0004;
Λ ← 0.0146; W_0 ← [1/a1; −a1; −a2; −a3; −a4];
Block 3 - Nominal Controller Design
desired_poles ← [−1 −2];
K_1 ← ackerman(A, B, desired_poles);
K_2 ← −(C ∗ (A − B ∗ K_1)^−1 ∗ B)^−1;
Block 4 - Reference Model - Matching Conditions
A_m ← A − B ∗ K_1; B_m ← B ∗ K_2;
Block 5 - Correction Signal (Performance Recovery)
G ← (K_2)^−1 ∗ (B^T ∗ B)^−1 ∗ B^T; Ω ← B ∗ (B^T ∗ B)^−1 ∗ B^T;
Block 6 - Lyapunov Algebraic Equation
Q_lyap ← I_{n×n}; P ← A_m^T ∗ P + P ∗ A_m + Q_lyap = 0;
Block 7 - Artificial Neural Network - RBF
n ← 25; b ← 5; Ŵ ← 0_{(2∗n+2)×1}; γ ← 2;
σ ← 500; λ ← 5; c ← 2 ∗ b/(n − 1);
Neurons' Centers Vector
centers ← 0_{1×n};
centers(1) ← −b;
   for i to 1 : n − 1 do
      centers(i+1) ← −b + i ∗ c;
   end for
d_max ← 2 ∗ b; width ← 1/(2 ∗ (d_max/(2 ∗ n)^(1/2))^2); Θ ← 0_{(2∗n+1)×1};
End of Algorithm 3
```

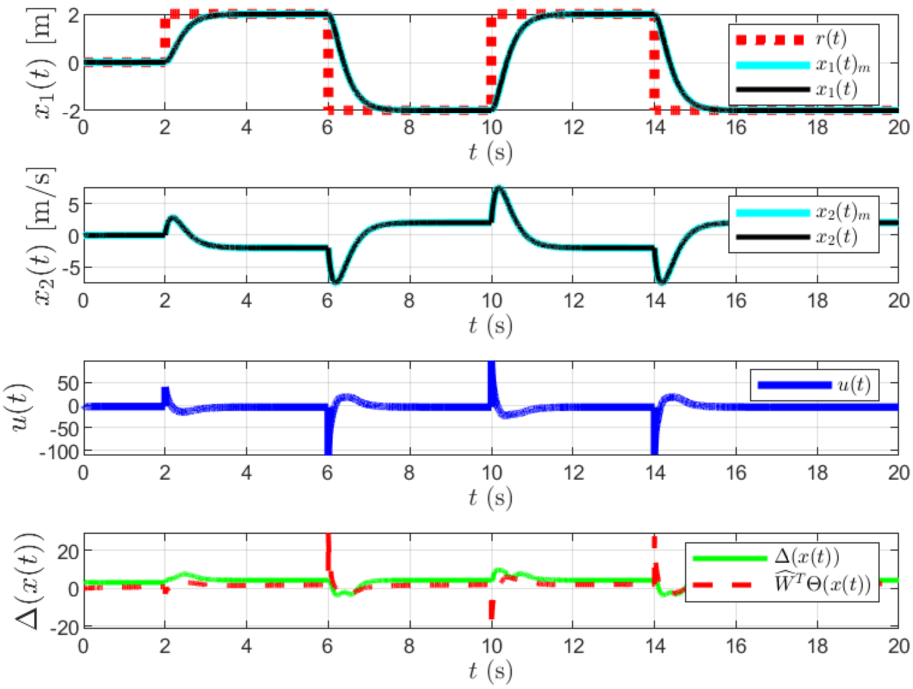
Algorithm 4 implements the IMRNAC control system simulator, where the system is exposed to a non-linear uncertainty. Its main blocks include: Block 1 – Initial configuration of the simulation, with definition of the total time and sampling interval; Block 2 – Definition of the non-linear uncertainty $\Delta(x)$, calculated using a regressor vector Θ_0 and a vector of weights W_0; Block 3 – Creation of the reference signal r; Block 4 – Assembly of the RBF regressor vector Θ, where elements are calculated with RBF applied to the state x and the centers of the neurons; Block 5 – Calculation of the correction signal, where the vector Ψ and the vector v are updated based on the difference between the state x and the desired state x_m; Block 6 – Calculating the control signal u, which includes the gains K_1 and K_2, the correction signal Gv, and the uncertainty estimate provided by the neural network $\widehat{W}^T\,\Theta(x)$; and Block 7 – Updating the real system with the uncertainty $\Delta(x)$, Where the state x is adjusted with the control u, and the output y is obtained from the matrix C, until the end of the simulation.

Algorithm 4 IMRNAC Simulator

```
Block 1 - Setup - Initial Conditions of the Simulator
ft ← 20; dt ← 0.001;
  for k to 0 : dt : ft do
    Block 2 - Uncertainty
    Θ0 ← [Λ^−1; Λ; 1; x(2)^2; x(2)^2];
    Δ(x) ← 1 + x(1) + x(2) + x(1)^2 + sin(x(1)) + cos(x(1)) + sin(x(2)) + cos(x(2));
    Block 3 - Reference Signal
      if k ≤ 10 then
        r = 1;
      else
        if k > 10 then
          r = −1;
        end if
      end if
    Block 4 - NNA Regressor Vector Assembly
    Θ(1) ← 1;
      for i to 1 : n do
        Θ(i+1) ← exp(−width|x(1) − centers(i)|^2); Θ(i+n+1) ← exp(−width|x(2) − centers(i)|^2);
      end for
    Block 5 - Correction Signal for Performance Recovery
    Ψ ← Ψ + dt ∗ (−λ ∗ (Ψ − (x − x_m)));   v ← λ ∗ Ψ + (A_m − λ ∗ I_n×n ∗ (x − x_m));
    Block 6 - Updates - Control Signal - Reference Model - Adaptation Law
    u ← −K1 ∗ x + K2 ∗ (r + G ∗ v) − Ŵ^T Θ;   x_m ← x_m + dt(A_m ∗ x_m + B_m ∗ r + Ω ∗ v);
    x_mi ← x_mi + dt ∗ (A_m ∗ x_mi + B_m ∗ r);   Ŵ ← Ŵ + dt ∗ (γ ∗ (Θ ∗ (x − x_m)^T P ∗ B − σŴ));
    Block 7 - Real System with Uncertainty - Output Measurement
    x ← x + dt ∗ (A ∗ x + B ∗ (u + Δ(x)));
  end for
End of Algorithm 4
```

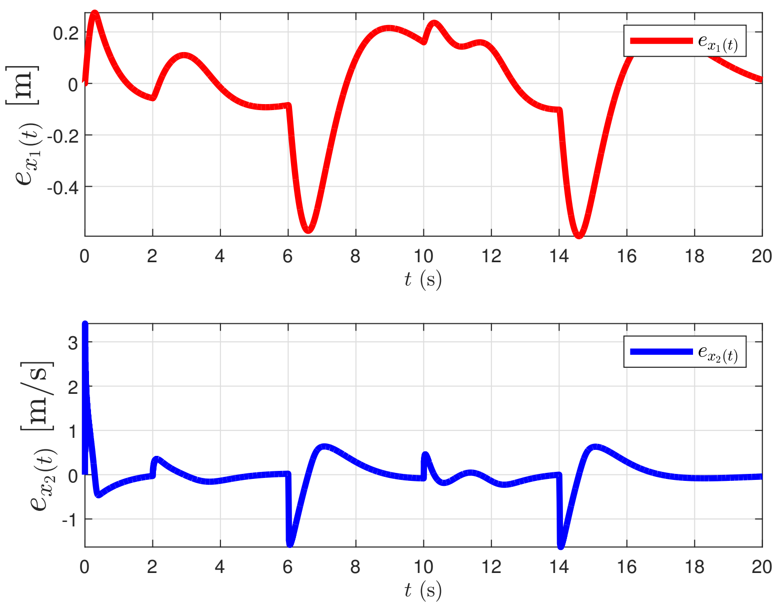
Regarding the controller tests, the following observations are made.

- Figure 4 describes the response of the closed-loop system with DMRNAC control for both states x_1 and x_2 that follow the reference model, the control effort $u(t)$ that presents limited oscillations, and the performance of the RBF neural network $\widehat{W}^T\,\Theta(x)$ to estimate the uncertainty $\Delta(x)$. The errors between the states of the real system and the reference model (e_{x_1} and e_{x_2}) are limited and converge to zero, $e_{x_1} \to 0$ and $e_{x_2} \to 0$ (Fig. 5). The estimation error of the RBF network converges to zero $e_{RBF} \to 0$ (Fig. 6).
- The metrics used to measure the estimation error of the RBF network in DMRNAC control (Fig. 7) have satisfactory results (Table 1).
- In DMRNAC control, for faster tracking of the reference signal, poles at [- 3 -2] are chosen resulting in the response of Fig. 8.
- Figure 9 describes the response of the closed-loop system with IMRNAC control for both states x_1 and x_2 that follow the reference model, the control effort $u(t)$ that presents limited oscillations, and the performance of the RBF neural network $\widehat{W}^T\,\Theta(x)$ to estimate the uncertainty $\Delta(x)$. The errors between the states of the real system and the reference model (e_{x_1} and e_{x_2}) are limited and converge to zero, $e_{x_1} \to 0$ and $e_{x_2} \to 0$ (Fig. 10). The estimation error of the RBF network converges to zero $e_{RBF} \to 0$, but with oscillations when the reference signal changes (Fig. 11).



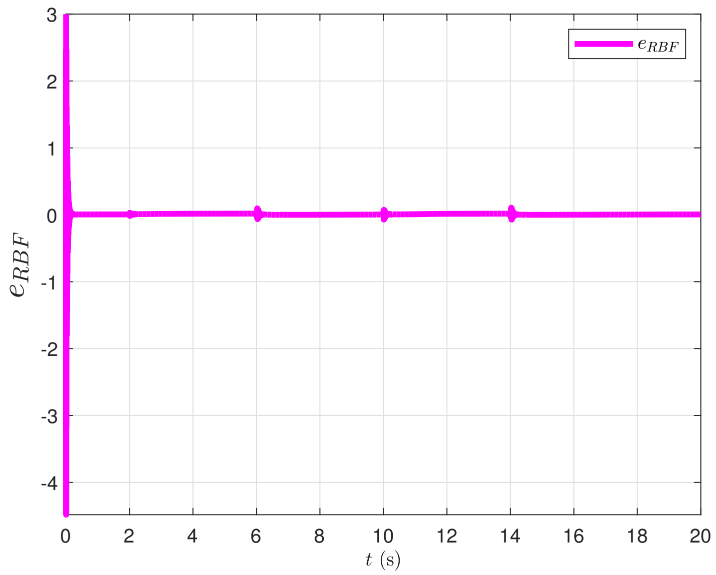
Source: Elaborated by the authors.

Figure 4. Controlled system response (DMRNAC).



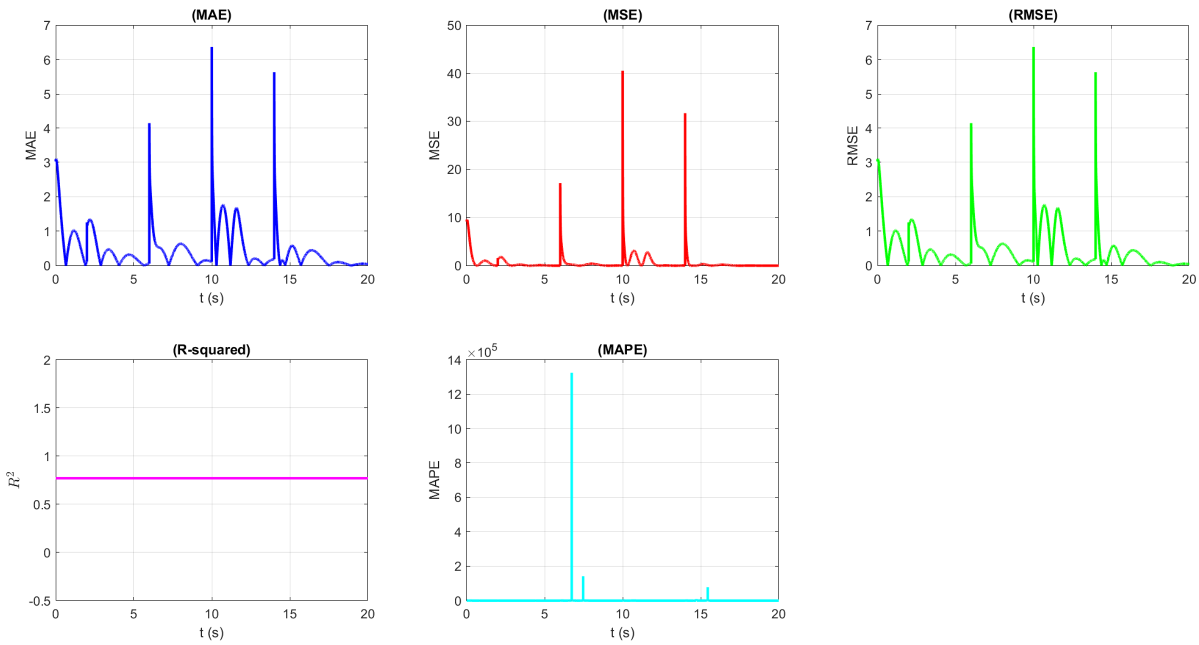
Source: Elaborated by the authors.

Figure 5. Reference model tracking error (DMRNAC).



Source: Elaborated by the authors.

Figure 6. RBF network estimation error (DMRNAC).



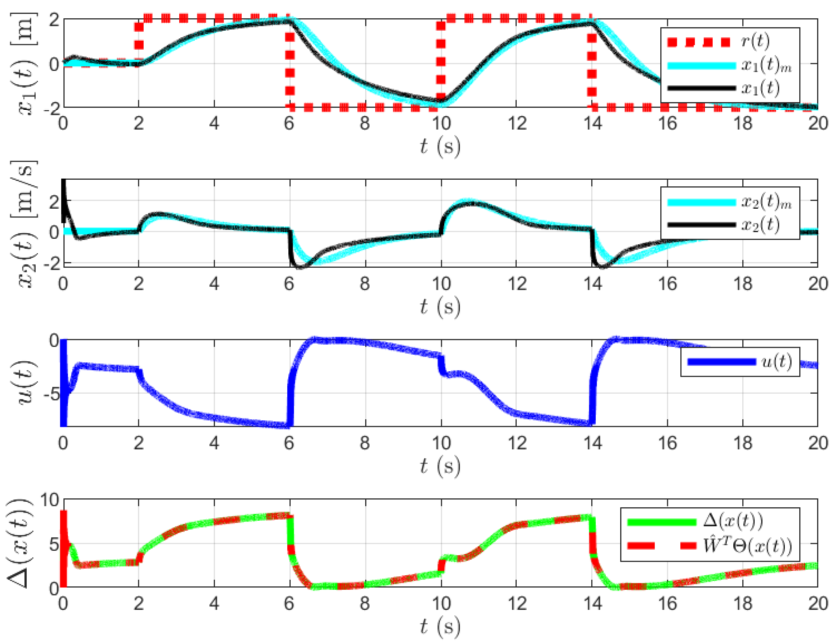
Source: Elaborated by the authors.

Figure 7. RBF network performance metrics (DMRNAC).

Table 1. Metrics for evaluating the RBF network in DMRNAC control.

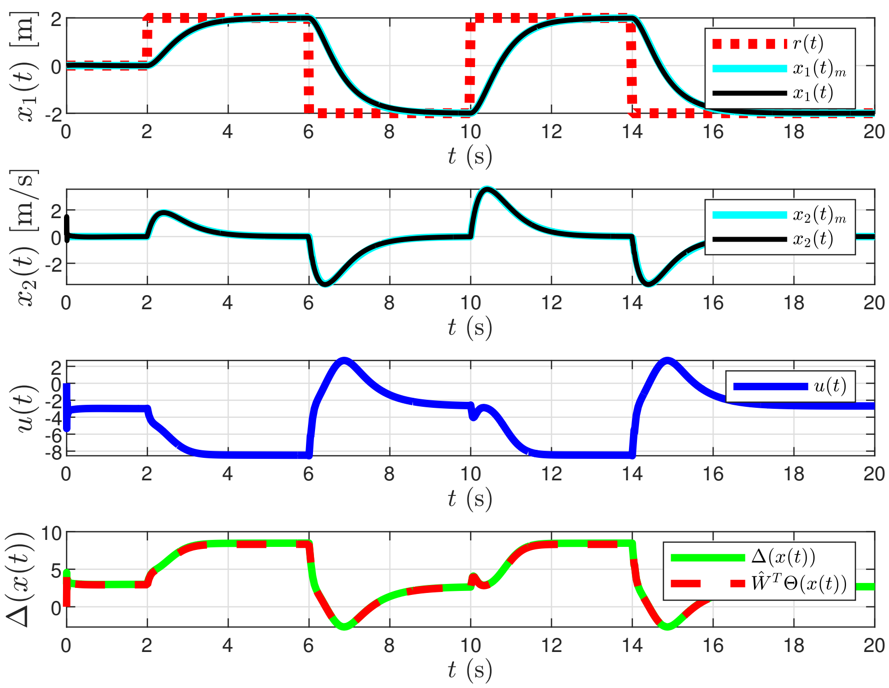
Metrics	Value
Mean Absolute Error (MAE)	0.00864
Mean Square Error (MSE)	0.00198
Root Mean Square Error (RMSE)	0.04459
Coefficient of Determination (R-squared)	0.99967
Mean Absolute Percentage Error (MAPE)	0.30919

Source: Elaborated by the authors.



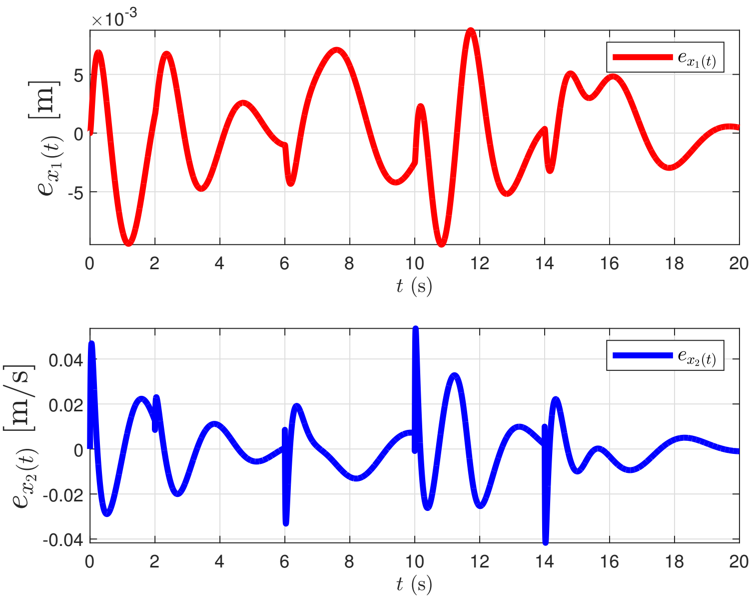
Source: Elaborated by the authors.

Figure 8. Faster reference signal tracking (DMRNAC).



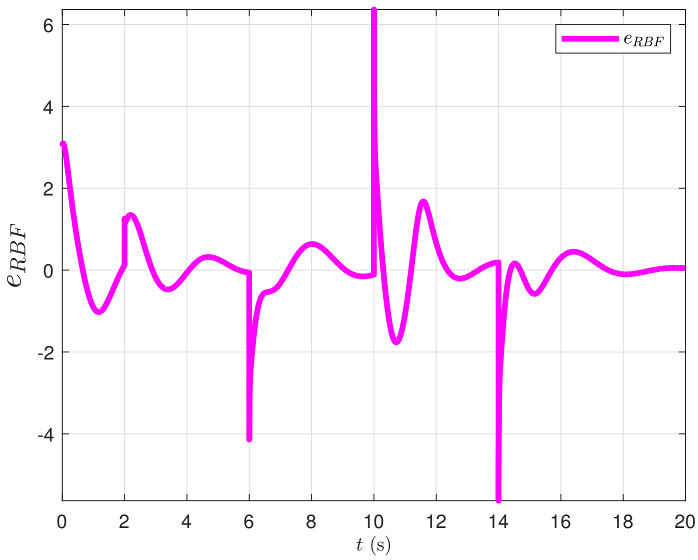
Source: Elaborated by the authors.

Figure 9. Controlled system response (IMRNAC).



Source: Elaborated by the authors.

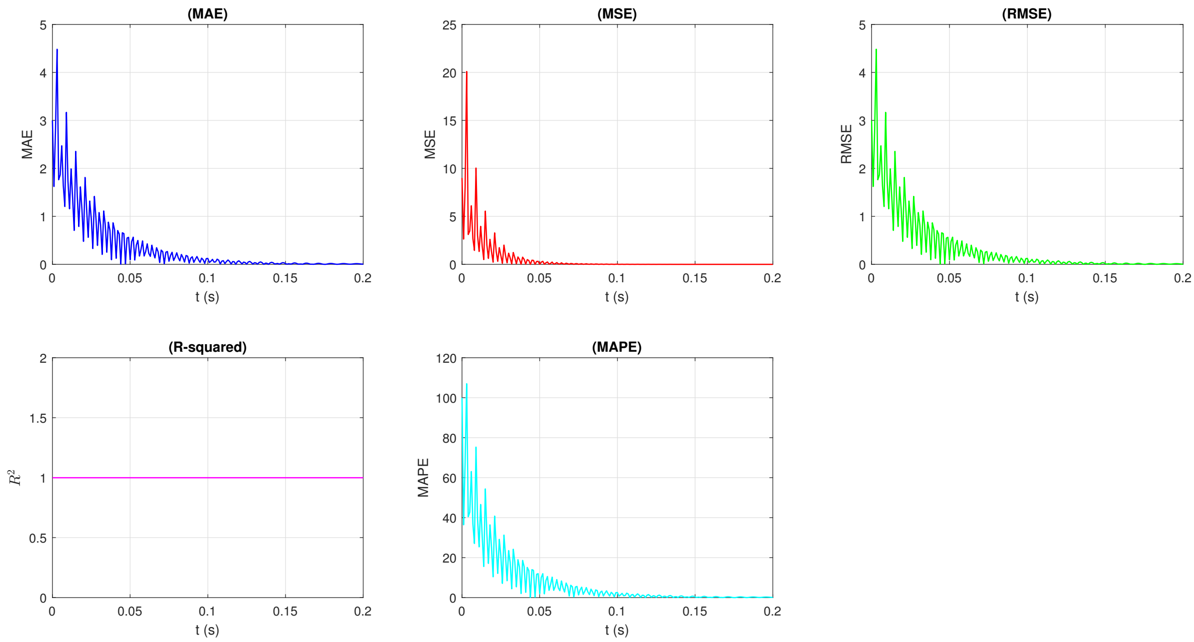
Figure 10. Reference model tracking error (IMRNAC).



Source: Elaborated by the authors.

Figure 11. RBF network estimation error (IMRNAC).

- The metrics used to measure the estimation error of the RBF network in IMRNAC control (Fig. 12), also showed satisfactory results (Table 2).



Source: Elaborated by the authors.

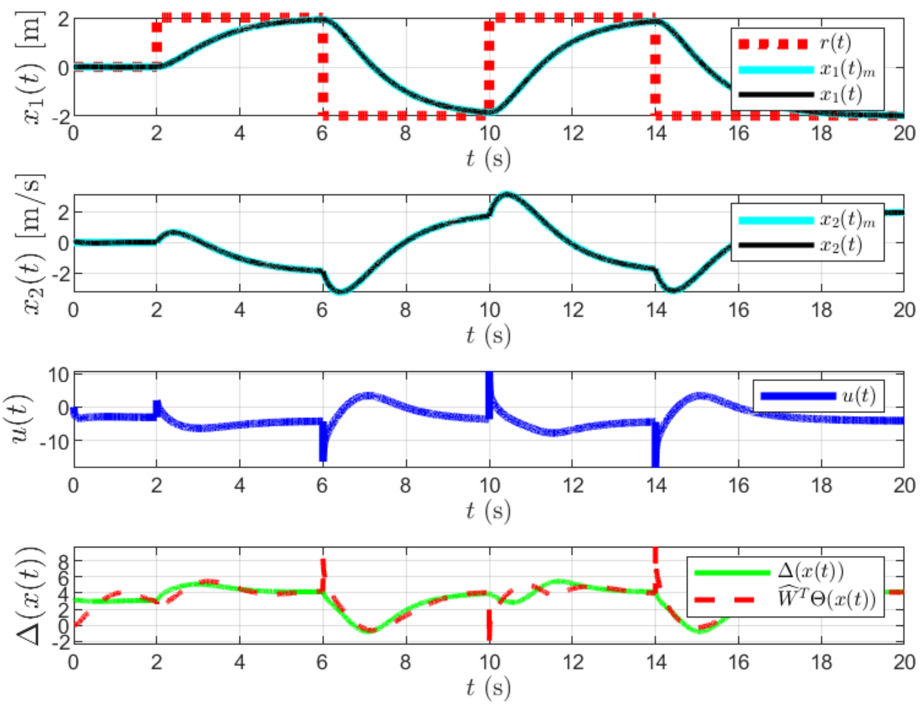
Figure 12. RBF network performance metrics (IMRNAC).

Table 2. Metrics for evaluating the RBF network in IMRNAC control.

Metrics	Value
Mean Absolute Error (MAE)	0.46402
Mean Square Error (MSE)	0.54524
Root Mean Square Error (RMSE)	0.73841
Coefficient of Determination (R-squared)	0.7692
Mean Absolute Percentage Error (MAPE)	110.7971

Source: Elaborated by the authors.

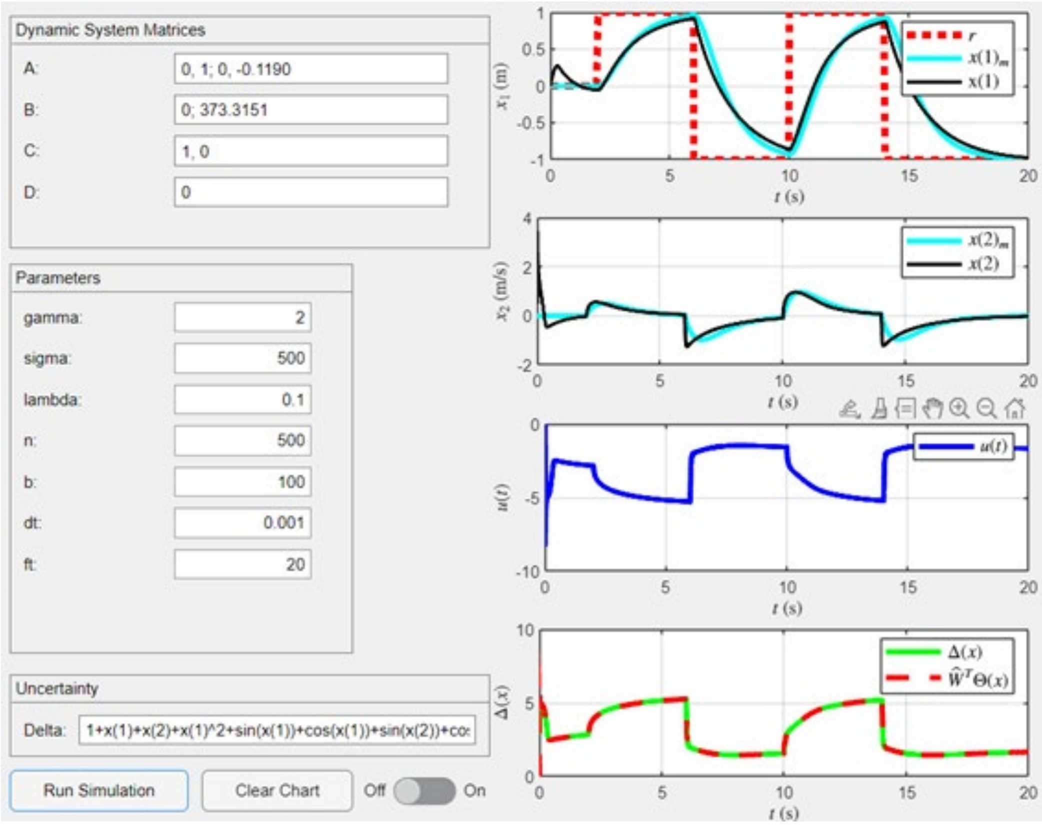
- In IMRNAC control, for faster tracking of the reference signal, poles at [-4 -5] are chosen, resulting in the response in Fig. 13.



Source: Elaborated by the authors.

Figure 13. Faster reference signal tracking (IMRNAC).

- The IMRNAC control showed robustness in all tests, with non-linear sinusoidal and quadratic uncertainties, isolated or summed.
- For all tests, both the DMRNAC and IMRNAC controls were stable, even in the presence of uncertainties.
- Based on Algorithms 1 and 2 an application with a graphical interface is developed, as shown in Fig. 14, for faster and more practical simulations of the controller, with information such as the performance of both states when following the reference model, the behavior of the control input and a comparison of the simulated uncertainty with the uncertainty estimated by the RBF neural network.



Source: Elaborated by the authors.

Figure 14. DMRNAC simulator graphical interface.

CONCLUSION

This article presents rocket modeling and a methodology for developing algorithms for the design of robust neuro-adaptive controllers. The algorithms for the proposed controllers were evaluated in simulators that are based on mathematical models and can be applied to a class of systems by changing the values of their matrices, configuration parameters, and the order of the models.

From the discussions presented, it can be concluded that this proposal offers a significant contribution to the aerospace sector, specifically, in the field of adaptive control in the context of altitude control of solid fuel rockets. Throughout this research, the modeling and design of controllers was based on an RBF neural network, meeting the specific requirements of this engineering challenge. The results obtained showed that the two controllers developed, DMRNAC and IMRNAC, demonstrated stability and robustness even in the presence of uncertainties.

The performance of the controllers in response to changes in the reference signal was evaluated, highlighting the ability to track and regulate the controlled systems. In addition, the practical implementation of the controller and simulator developed proved to be viable and promising for applications in real-world systems, offering a flexible and adaptable platform for future research and development.

As a general conclusion, this research provides a solid basis for future investigations in the field of adaptive control, paving the way for the continuous improvement of techniques and methodologies in this domain of control engineering and aerospace engineering.

CONFLICT OF INTEREST

Nothing to declare.

AUTHORS' CONTRIBUTION

Conceptualization: Carvalho CDR and Fonseca Neto JV; **Data curation**: Carvalho CDR and Fonseca Neto JV; **Acquisition of funding:** Carvalho CDR and Fonseca Neto JV; **Research**: Carvalho CDR and Fonseca Neto JV; **Methodology:** Carvalho CDR and Fonseca Neto JV; **Project administration:** Carvalho CDR and Fonseca Neto JV; **Resources:** Carvalho CDR and Fonseca Neto JV; **Supervision:** Carvalho CDR and Fonseca Neto JV; **Validation:** Carvalho CDR and Fonseca Neto JV; **Writing - Preparation of original draft:** Carvalho CDR and Fonseca Neto JV; **Writing – Proofreading and editing:** Carvalho CDR and Fonseca Neto JV; **Final approval:** Carvalho CDR and Fonseca Neto JV.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable.

FUNDING

Not applicable.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

Alves TCDS, Sica FC (2023) O estado da arte dos sistemas de controle de reentrada na atmosfera para veículos aeroespaciais reutilizáveis. Paper presented 2023 12° Workshop em Engenharia e Tecnologia. Instituto Nacional de Pesquisas Espaciais; São José dos Campos, Brazil.

Anavatti SG, Santoso F, Garratt MA (2015) Progress in adaptive control systems: past, present, and future. Paper presented 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation. Institute of Electrical and Electronics Engineers; Surabaya, Indonesia.

Arabi E, Yucelen T (2019) Set-theoretic model reference adaptive control with time-varying performance bounds. Int J Control 92(11):2509-2520. https://doi.org/10.1080/00207179.2018.1442026

Arabi E, Gruenwald BC, Yucelen T, Nguyen NT (2018) A set-theoretic model reference adaptive control architecture for disturbance rejection and uncertainty suppression with strict performance guarantees. Int J Control 91(5):1195-1208. https://doi.org/10.1080/00207179.2017.1312019

Arabi E, Panagou D, Yucelen T, Nguyen NT (2020) Model reference adaptive control of uncertain dynamical systems subject to high-order actuator dynamics with performance guarantees. Paper presented 2020 American Institute of Aeronautics and Astronautics Scitech Forum. AIAA; Orlando, USA.

Arabi E, Yucelen T, Gruenwald BC, Fravolini M, Balakrishnan S, Nguyen NT (2019) A neuroadaptive architecture for model reference control of uncertain dynamical systems with performance guarantees. Syst Control Lett 125:37-44. https://doi.org/10.1016/j.sysconle.2019.01.005

Aström KJ, Wittenmark B (2013) Adaptive control. 2nd ed. Boston: Courier Corporation.

Balestrino A, De Maria G, Sciavicco L (1983) An adaptive model following control for robotic manipulators. J Dyn Sys Meas Control 105(3):143-151. https://doi.org/10.1115/1.3140646

Costa BA, Parente FL, Belfo J, Somma N, Rosa P, Igreja JM, Belhadj J, Lemos JM (2024) A reinforcement learning approach for adaptive tracking control of a reusable rocket model in a landing scenario. Neurocomputing 577:e127377. https://doi.org/10.1016/j.neucom.2024.127377

Davidson JB, Lallman FJ, Bundick WT (2001) Real-time adaptive control allocation applied to a high performance aircraft. Paper presented 2001 5th Society for Industrial and Applied Mathematics Conference on Control and its Applications. SIAM; San Diego, USA.

Dogan KM, Yucelen T, Haddad WM, Muse JA (2020) Improving transient performance of discrete-time model reference adaptive control architectures. Int J Adapt Control Signal Process 34(7):901-918. https://doi.org/10.1002/acs.3114

Dydek ZT, Annaswamy AM, Lavretsky E (2008) Adaptive control and the Nasa x-15 program: a concise history, lessons learned, and a provably correct design. Paper presented 2008 American Control Conference. IEEE Control Systems Society; Seattle, USA.

Farsana N, Sebastian B, Priya PL, Kumar H (2023) Fault tolerant model reference adaptive controller for launch vehicle electromechanical actuation system. Paper presented 2023 International Conference on Control, Communication and Computing. Institute of Electrical and Electronics Engineers; Kerala, Índia.

Feldbaum A (1960a) Dual control theory. II. Avtomatika i Telemekhanika 21(11):1453-1464.

Feldbaum AA (1960b) Dual control theory. I. Avtomatika i Telemekhanika 21(9):1240-1249.

Fravolini ML, Yucelen T, Muse J, Valigi P (2015) Analysis and design of adaptive control systems with unmodeled input dynamics via multiobjective convex optimization. Paper presented 2015 American Control Conference. Institute of Electrical and Electronics Engineers; Illinois, USA.

Gauder DS (2024) Adaptive in-line qualitätsregelung in der mikro-verzahnungsfertigung.

Gettman D, Shelby SG, Head L, Bullock DM, Soyke N (2007) Data-driven algorithms for real-time adaptive tuning of offsets in coordinated traffic signal systems. Trans Res Rec, 2035(1):1-9. https://doi.org/10.3141/2035-01

Glushchenko A, Lastochkin K (2022) Neural network-based direct model reference adaptive control of quadrotor attitude. Paper presented 2022 16th International Conference on Stability and Oscillations of Nonlinear Control Systems. Institute of Electrical and Electronics Engineers; Moscow, Russia.

Gruenwald BC, Yucelen T, Muse JA (2017) Direct uncertainty minimization framework for system performance improvement in model reference adaptive control. Machines 5(1):9. https://doi.org/10.3390/machines5010009

Gruenwald BC, Yucelen T, Dogan KM, Muse JA (2018) An adaptive architecture for control of uncertain dynamical systems with unknown actuator bandwidths. IFAC-PapersOnLine 51(12):130-135. https://doi.org/10.1016/j.ifacol.2018.07.100

Heidari A, Navimipour NJ, Unal M (2023) A secure intrusion detection platform using blockchain and radial basis function neural networks for internet of drones. IEEE Internet Things J 10(10):8445-8454. https://doi.org/10.1109/JIOT.2023.3237661

Hodel AS, Baginski TA (1995) An interdisciplinary senior design course utilizing electronically guided model rockets. IEEE Trans Educ 38(4):321-327. https://doi.org/10.1109/13.473150

Hou M, Duan G, Guo M (2010) New versions of Barbalat's lemma with applications. J Control Theory Appl 8(4):545-547. https://doi.org/10.1007/s11768-010-8178-z

Hsia T (1986) Adaptive control of robot manipulators – A review. Paper presented 1986 International Conference on Robotics and Automation. Institute of Electrical and Electronics Engineers; San Francisco, USA.

Ji Z, Shi G (2023) Adaptive neural dynamics-based speed control strategy for stable retrieval of tethered satellite system. Adv Space Res 71(12):4987-4994. https://doi.org/10.1016/j.asr.2023.01.061

Johansson R (1988) Adaptive control of robot manipulator motion. IEEE Trans Robot 6(4):483-490. https://doi.org/10.1109/70.59359

Khalil H (2002) Nonlinear systems. Upper Saddle River: Prentice Hall.

Kotov EV, Raju KM (2024) Real-time adaptive control of electric vehicle drives using artificial neural networks. Paper presented 2024 MATEC Web of Conferences. EDP Sciences; Guilin, China.

Krstic M, Kokotovic PV, Kanellakopoulos I (1995) Nonlinear and adaptive control design. Hoboken: John Wiley & Sons.

Kumar R, Dey R, Guelton K, Bali A, Singh UP (2024) Adaptive control for cyber-physical systems under man-in-the-middle attacks with false data injections. J Frank Inst 361(4):106661. https://doi.org/10.1016/j.jfranklin.2024.106661

Landau I (1974) A survey of model reference adaptive techniques – Theory and applications. Automatica 10(4):353-379. https://doi.org/10.1016/0005-1098(74)90064-8

Landau ID, Lozano R, M'Saad M, Karimi A (2011) Adaptive control: algorithms, analysis and applications. New York: Springer Science & Business Media.

Lavretsky E, Hovakimyan N (2008) Adaptive dynamic inversion for nonaffine-in-control uncertain systems via time-scale separation. Part II. J Dyn Control Syst 14:33-41. https://doi.org/10.1007/s10883-007-9033-5

Lavretsky E, Wise K (2024) Robust and adaptive control: with aerospace applications. Advanced textbooks in control and signal processing. Cham: Springer International Publishing.

Lee C, Chung M-J, Lee BH (1984) An approach of adaptive control for robot manipulators. J Robot Syst 1(1):27-57. https://doi.org/10.1002/rob.4620010104

Levine WS (2011) Control system advanced methods. Boca Raton: CRC Press.

Lindorff D, Carroll R (1973) Survey of adaptive control using Liapunov design. Int J Control 18(5):897-914. https://doi.org/10.1080/00207177308932569

Lu M, Deng F, Liu L (2020) An extension of Barbalat's lemma with its application to synchronization of a class of switched networked nonlinear systems. IFAC-PapersOnLine 53(2):9778-9783. https://doi.org/10.1016/j.ifacol.2020.12.2659

Miao B, Li S, Liu L (2024) Nussbaum gain adaptive control for full-state constrained cyber-physical system under network attacks and actuator failures. Int J Robust Nonlinear Control 34(4):2515-2531. https://doi.org/10.1002/rnc.7093

Narendra KS, Parthasarathy K (1991) Gradient methods for the optimization of dynamical systems containing neural networks. IEEE Trans Neural Netw 2(2):252-262. https://doi.org/10.1109/72.80336

Nemati A, Mobayen S, Rouhani SH, Su C-L (2024) Design of nonsingular second-order terminal sliding mode controller for cyber-physical systems with time-delays and cyber-attack on actuators. Internat J Syst Sci 55(5):876-893. https://doi.org/10.1080/00207721.2023.2300717

Ogata K (2010) Engenharia de controle moderno. 5th ed. São Paulo: Pearson Education.

Orłowski J, Chaillet A, Destexhe A, Sigalotti M (2022) Adaptive control of Lipchitz time-delay systems by sigma modification with application to neuronal population dynamics. Syst Control Lett 159:105082. https://doi.org/10.1016/j.sysconle.2021.105082

Polycarpou MM (1996) Stable adaptive neural control scheme for nonlinear systems. IEEE Trans Autom Control 41(3):447-451. https://doi.org/10.1109/9.486648

Rovithakis GA, Christodoulou MA (1994) Adaptive control of unknown plants using dynamical neural networks. IEEE Trans Syst Man Cybern 24(3):400-412. https://doi.org/10.1109/21.278990

Sampaio RP (2006) Sistema de controle de atitude embarcado para voo autônomo de aviões em escala (dissertação de mestrado). Salvador: Universidade Federal da Bahia. In Portuguese. https://repositorio.ufba.br/handle/ri/21583

Sanner RM, Slotine J-JE (1991) Gaussian networks for direct adaptive control. Paper presented 1991 American Control Conference. Institute of Electrical and Electronics Engineers; Boston, USA.

Shelby SG (2004) Single-intersection evaluation of real-time adaptive traffic signal control algorithms. Trans Res Rec 1867(1):183-192. https://doi.org/10.3141/1867-21

Singh U, Pal NS (2019) Roll angle control of an aircraft using adaptive controllers. Paper presented 2019 International Conference on Automation, Computational and Technology Management. Institute of Electrical and Electronics Engineers; New Delhi, India.

Slotine J-JE, Li W (1989) Composite adaptive control of robot manipulators. Automatica 25(4):509-519. https://doi.org/10.1016/0005-1098(89)90094-0

Slotine J-JE, Li W (1991) Applied nonlinear control. Volume 199. Englewood Cliffs: Prentice Hall.

Stepanyan V, Kalmanje K (2010) Input and output performance of M-MRAC in the presence of bounded disturbances. Paper presented 2010 American Institute of Aeronautics and Astronautics Guidance, Navigation, and Control Conference. AIAA; Toronto, Canada.

Su Y, Huang J (2011) Stability of a class of linear switching systems with applications to two consensus problems. IEEE Trans Autom Control 57(6):1420-1430. https://doi.org/10.1109/TAC.2011.2176391

Unbehauen H, Schmid C (1975) Status and industrial application of adaptive control systems. Automat Control Theory Appl 3(1):1-2.

Vetter JS, Reed DA (2000) Real-time performance monitoring, adaptive control, and interactive steering of computational grids. Int J High Perform Comput Appl 14(4):357-366. https://doi.org/10.1177/109434200001400407

Wang Z, Han Z, Tayyaba S (2024) Adaptive control for uncrewed aerial vehicles based on communication information optimization in complex environments. PeerJ Comput Sci 10:e1920. https://doi.org/10.7717/peerj-cs.1920

Whitaker HP, Yamron J, Kezer A (1958) Design of model-reference adaptive control systems for aircraft. Massachusetts: Massachusetts Institute of Technology, Instrumentation Laboratory.

Wie B, Du W, Whorton M (2008) Analysis and design of launch vehicle flight control systems. Paper presented 2008 American Institute of Aeronautics and Astronautics Guidance, Navigation and Control Conference. AIAA; Honolulu, Hawaii.

Wittenmark B (1975) Stochastic adaptive control methods: a survey. Int J Control 21(5):705-730. https://doi.org/10.1080/00207177508922026

Yu S-H, Annaswamy AM (1995) Neural control for nonlinear dynamic systems. Neural Information Processing Systems 8.

Yucelen T, Calise A (2010) Derivative-free model reference adaptive control of a generic transport model. Paper presented 2010 American Institute of Aeronautics and Astronautics Guidance, Navigation and Control Conference. AIAA; Toronto, Canada.

Yucelen T, Johnson E (2013) A new command governor architecture for transient response shaping. Int J Adapt Control Signal Process 27(12):1065-1085. https://doi.org/10.1002/acs.2376

Zeraick Monteiro N, Rodrigues Mazorche S (2023) Limitations and applications in a fractional Barbalat's lemma. Fract Calc Appl Anal 26(1):253-275. https://doi.org/10.1007/s13540-022-00111-6

Zhao N, Tian Y, Zhang H, Herrera-Viedma E (2024) Fuzzy-based adaptive event-triggered control for nonlinear cyber-physical systems against deception attacks via a single parameter learning method. Inf Sci 657:119948. https://doi.org/10.1016/j.ins.2023.119948

Zhi Q, Xia L, Wang X, Zhang S, Zheng K (2024) Research on adaptive control strategy for distributed new energy. Paper presented 2024 Ninth International Conference on Energy Materials and Electrical Engineering. International Society for Optics and Photonics; Guilin, China.