https://doi.org/10.3311/PPee.40289

Periodica Polytechnica Electrical Engineering and Computer Science

Constrained Neural Network-based Model Predictive Control for Quadrotors Using the Sine Cosine Algorithm

Mohamed Benrabah[1*], Mohamed Lamine Fas[2], Chafea Stiti[2], Kamel Kara[2]

[1] Laboratory of Robotics, Parallelism and Embedded Systems, Department of Automatic, Faculty of Electrical Engineering, University of Science and Technology Houari Boumediene, P. O. B. 32, Bab Ezzouar, 16111 Algiers, Algeria
[2] Laboratory of Electrical Systems and Remote Control, Department of Automatic and Electrical Engineering, Faculty of Technology, Blida 1 University, P. O. B. 270, 09000 Blida, Algeria
* Corresponding author, e-mail: mbenrabah@usthb.dz

Received: 24 February 2025, Accepted: 21 June 2025, Published online: 01 July 2025

Abstract

In this paper, an efficient nonlinear control algorithm, called Constrained Neural Networks based Model Predictive control using Sine Cosine Algorithm (CNNMPC-SCA) is developed to control the dynamics of quadrotors. The main objective is to design an efficient controller for quadrotors that ensures satisfactory performance while minimizing the gap between the quadrotor positions and the reference trajectories. Indeed, a novel dynamic model architecture of the quadrotor is developed using several Nonlinear Autoregressive Exogenous (NARX) neural networks, this model aims to accurately predict the future behavior of the quadrotor within a short and acceptable time frame, making it suitable for implementation in the control process. The designed model was validated and then integrated into the CNNMPC-SCA algorithm. Furthermore, the metaheuristic algorithm known as the Sine Cosine Algorithm (SCA) was modified and employed to solve the non-convex, nonlinear optimization problem of the proposed predictive controller. To assess the efficiency of the proposed CNNMPC-SCA algorithm, a comparative study was conducted using the Adaptive Fuzzy PID controller and the hybrid Fuzzy PID controller. The obtained results demonstrate that the proposed control algorithm achieves better control performances compared to those obtained using the other considered controllers.

Keywords

quadrotor control, nonlinear model predictive control, metaheuristic, NARX, sine cosine algorithm

1 Introduction

Unmanned Aerial Vehicle (UAVs) have attracted significant attention from researchers due to their versatility, ease of maneuverability, safety, adaptability to various environments, and their ability to be customized for a wide range of applications. Due to their architecture and advanced control algorithms, UAVs can operate effectively in challenging and confined spaces. Moreover, their deployment in harsh or hazardous environments enhances safety by minimizing risks to human operators. Nowadays, UAVs are widely utilized in numerous fields, including agriculture, surveillance, transportation, military operations, and more [1–5].

Quadrotors represent a specific class of UAVs, distinguished by their simple yet efficient architecture, consisting of four rotors, typically powered by brushless DC motors. This design enables them to perform highly complex maneuvers, making them a subject of significant interest among researchers [6, 7]. Over the years, various control strategies have been developed to manage the dynamics of these systems [8–17]. For instance, PID-based methods have been employed, with optimized parameters to ensure a desired level of control performance [10, 11, 16]. Additionally, Sliding Mode Control techniques have been widely studied and used to control quadrotors for their robustness against parameter variations and different disturbances [12, 13, 17]. Furthermore, control algorithms leveraging the principles of the Linear Quadratic Regulator (LQR), such as the classical LQR and Observer-Based LQR, have been explored [14, 15]. On another front, some researchers have utilized metaheuristic algorithms to optimize the parameters of nonlinear controllers, including Type-1 and Type-2 Fuzzy Logic Controllers (FLC) [8, 9], aiming to enhance control performance.

Despite the variety of control algorithms optimized to manage the dynamics of quadrotors, achieving optimal

Cite this article as: Benrabah, M., Fas, M. L., Stiti, C., Kara, K. "Constrained Neural Network-based Model Predictive Control for Quadrotors Using the Sine Cosine Algorithm", Periodica Polytechnica Electrical Engineering and Computer Science, 2025. https://doi.org/10.3311/PPee.40289

control performance remains a challenge due to the complexity and nonlinearity of these systems. Therefore, this work proposes a novel control strategy that aims to improve control performances, such as response speed, stability, and precision in tracking desired reference trajectories. Specifically, a constrained neural network-based model predictive control (NN-MPC) approach is employed, where a metaheuristic algorithm is used to solve the associated optimization problem.

Model-based Predictive Control (MPC) is a highly effective approach that gives satisfactory control performance and can handle various types of constraints for both linear and nonlinear systems [18–23]. This technique relies on the principle of utilizing a prediction model to determine the future behavior of the controlled system over a prediction horizon and computing a control sequence over the control horizon by minimizing a specified cost function through an optimization technique. Therefore, designing an MPC algorithm involves two key aspects: defining the prediction model and selecting the optimization technique for the cost function [24, 25].

The developed prediction model for the quadrotor is based on several Nonlinear Autoregressive Exogenous (NARX) neural networks, along with some integrators and two blocks used to calculate specific parameters. The NARX model consists of an input layer, one or more hidden layers, and an output layer. Its architecture is typically a feedforward network, except that its outputs are recurrent through delay operators connected to the inputs. Thus, the inputs to this network include both the exogenous inputs and the delayed exogenous inputs and outputs. This network is known for its ability to model nonlinear dynamical systems [26, 27]. The proposed prediction model was optimized to be both simple and capable of accurately predicting the future behavior of the quadrotor within a short and acceptable time frame.

The optimization problem of the proposed CNNMPC-SCA was solved using a metaheuristic approach known as the Sine Cosine Algorithm (SCA). The SCA technique is a powerful stochastic, population-based, iterative, global search, non-nature-inspired, memory-based metaheuristic algorithm [28–30]. The SCA algorithm begins by initializing a population of random candidate solutions, then adapts the position of each candidate by moving it toward or outward the best solution using sine and cosine functions. This algorithm involves four main stochastic parameters and one specific parameter. The superiority of the SCA algorithm in solving complex, non-convex, and nonlinear optimization problems has been demonstrated through various tests using unimodal, multi-modal, and composite benchmark functions.

To enhance the performance of the proposed CNNMPC-SCA algorithm, the SCA optimization technique was modified by adding a greedy selection step within the SCA optimizer. Specifically, a new step was incorporated into the SCA algorithm after computing the new positions of candidate solutions. In this step, the fitness of the cost function is evaluated using the new candidate solutions. If the new solutions yield a better fitness than the previous ones, the new solutions are adopted; otherwise, they are discarded, and the old solutions are retained.

The rest of this paper is organized as follows: Section 2 details the algorithms of the SCA optimizer and its proposed modified version. Section 3 presents the quadrotor dynamical model and the proposed prediction model. In Section 4, the proposed CNNMPC-SCA algorithm is introduced. Section 5 provides and discusses the simulation results for various cases. Finally, Section 6 summarizes the conclusions.

2 Sine cosine algorithm

2.1 Original sine cosine algorithm

Sine Cosine Algorithm leverages trigonometric functions, specifically sine and cosine, to update the positions of candidate solutions. The SCA excels at exploiting the search space while maintaining a strong balance between exploration and exploitation phases of metaheuristic algorithms, due to the periodic property of sine and cosine functions.

The SCA uses a set of candidate solutions to search for the minimum of a given cost function. During the initial phase of the SCA algorithm, candidate solutions called search agents, or the population are generated using Eq. (1):

$$X_{ij} = X\min_j + r \cdot \left(X\max_j - X\min_j \right) \tag{1}$$

where $i = 1{:}N_{pop}$, $j = 1{:}D$, X_{ij} is the j^{th} dimension of the i^{th} candidate solution, $X\max_j$ and $X\min_j$ represent the upper bound and the lower bound of the search space of X_{ij}, r is a random number between 0 and 1, N_{pop} and D denote the size of the population and the dimension of the optimization problem respectively.

The second step of the SCA algorithm consists of evaluating the objective function for all search agents and selecting the best candidate solution as the destination point.

In the third step, the SCA updates the position of each candidate solution using Eq. (2) as follows:

$$\boldsymbol{X}_i(\boldsymbol{k}+1) = \boldsymbol{X}_i(\boldsymbol{k}) + r_1 \cdot f(r_2) \cdot \left| r_3 \cdot \boldsymbol{P}_g(\boldsymbol{k}) - \boldsymbol{X}_i(\boldsymbol{k}) \right| . \tag{2}$$

Where

$$f(r_2) = \sin(r_2) \text{ if } r_4 < 0.5$$
$$f(r_2) = \cos(r_2) \text{ if } r_4 \geq 0.5$$

$\boldsymbol{X}_i(\boldsymbol{k}+1)$ is the new solution for the i^{th} candidate solution $\boldsymbol{X}_i(\boldsymbol{k})$, and the destination point $\boldsymbol{P}_g(\boldsymbol{k}) = [P_{g,1}(k), P_{g,2}(k), \ldots, P_{g,D}(k)]$ represents the best solution obtained so far. The parameters r_1, r_2, r_3 and r_4 are four control variables of the SCA algorithm. Where, r_1 is an algorithm-specific parameter, while the other three are randomly generated. The parameter r_1 determines the movement direction of the new solution, either toward or outward the destination point. Its value is linearly updated from a to zero according to Eq. (3):

$$r_1 = a - k \cdot \frac{a}{k_{\max}} \tag{3}$$

where a is a specific parameter defined by the user, k is the current iteration, $k_{\max}$ is the maximum number of iterations.

The r_2, r_3 and r_4 are randomly generated as follows:

$$r_2 = 2 \cdot \pi \cdot \text{rand}(\ ),\ r_3 = 2 \cdot \text{rand}(\ ),\ r_4 = \text{rand}(\ )$$

where rand( ) generates uniformly distributed random numbers in the range [0,1].

In the final step of the SCA algorithm, the termination criteria are evaluated. If any of them is satisfied, the optimization process ends; otherwise, the algorithm returns to the second step. The flowchart of the SCA is shown in Fig. 1.

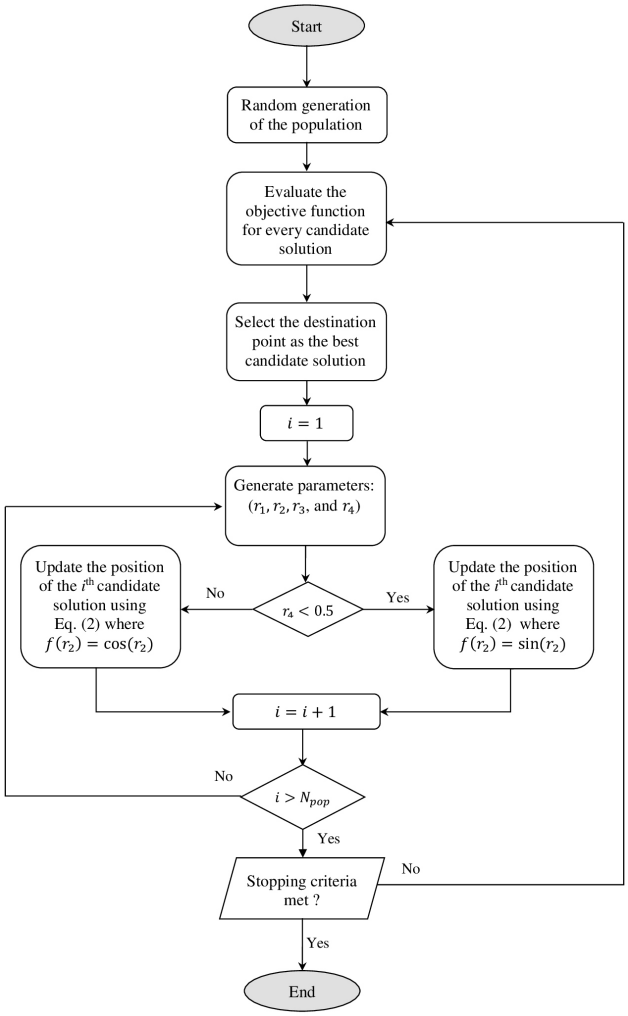

Fig. 1 Flowchart of the Sine Cosine Algorithm

2.2 Modified sine cosine algorithm

The proposed modified version of the SCA was developed to enhance the precision and speed of the optimization process of the original SCA algorithm. Specifically, the evaluation process of the second step of the original SCA, which involves evaluating the objective function and selecting the best solution, was modified to occur only at the first iteration instead of at every iteration. After the initial evaluation, the objective function does not need to be recalculated at every iteration. The fitness required for selecting the best solution is already available from previous steps of the algorithm. By avoiding redundant computations, this modification helps reduce the overall execution time without compromising performance. In addition, after the third step of the original SCA, a greedy selection process is implemented. After calculating the new candidate solutions, the objective function is evaluated using these new solutions and compared to the ones obtained with the previous solutions. If a new solution provides better fitness, it is retained; otherwise, it is discarded. The flowchart of the modified SCA is given in Fig. 2.

3 Dynamic model of the quadrotor

Developing a reliable prediction model is a crucial step in designing a numerically optimized MPC algorithm [25]. The model must satisfy two essential criteria:

1. The developed model should accurately predict the future behavior of the controlled system.
2. The prediction model must be computationally efficient, allowing the MPC algorithm to perform all necessary calculations within a time frame shorter than the fixed sampling time.

To fulfill these criteria, a prediction model based on nonlinear autoregressive exogenous neural networks is proposed, which balances accuracy and computational simplicity.

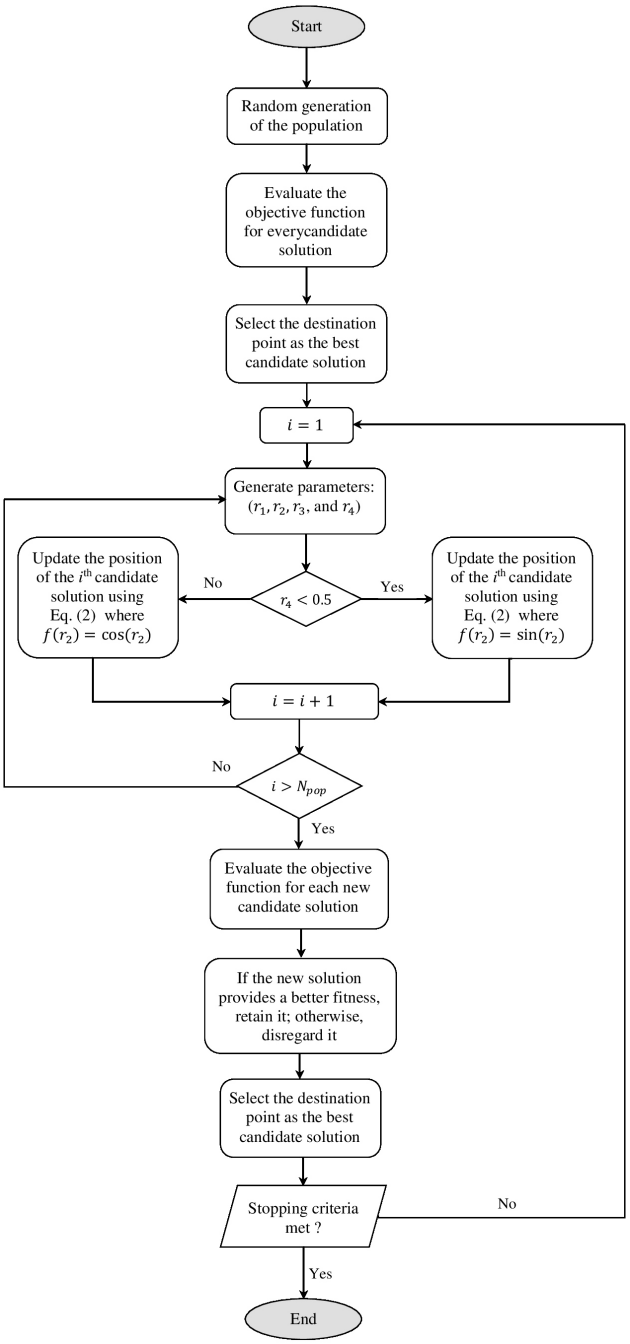

Fig. 2 Flowchart of the modified Sine Cosine Algorithm

3.1 Developed dynamic model

The proposed prediction model for the quadrotor combines multiple Nonlinear Autoregressive Exogenous (NARX) neural networks, integrators, and two functional blocks. The first block computes (u_1, u_2, u_3, u_4, $\bar{w}$), while the second calculates (u_x, u_y).

In the initial stage of the model, four NARX neural networks are employed to model the behavior of the rotors. These networks have the same architecture: they take two inputs—one representing the voltage applied to the rotor and the other is the delayed output of the model. Where, all delays in the proposed model are fixed at 10 ms, matching the sampling time. The hidden layer consists of eight neurons, and the output represents the rotor's velocity.

In the second stage of the model, the first functional block is designed to compute the variables (u_1, u_2, u_3, u_4, $\bar{w}$) based on the outputs of the four NARX networks from the first stage, as follows:

$$\begin{aligned} u_1 &= K_p\left(w_1^2 + w_2^2 + w_3^2 + w_4^2\right) \\ u_2 &= K_p\left(w_4^2 - w_2^2\right) \\ u_3 &= K_p\left(w_3^2 - w_1^2\right) \\ u_4 &= C_d\left(-w_1^2 + w_2^2 - w_3^2 + w_4^2\right) \\ \bar{w} &= w_1 - w_2 + w_3 - w_4 \end{aligned} \tag{4}$$

where $K_p = 0.000029842$; $C_d = 0.0000002232$ are the thrust and drug coefficients of the quadrotor [31]. The w_1, w_2, w_3 and w_4 are the outputs of the four NARX networks from the first stage.

The third stage of the proposed model consists of a single NARX network with seven inputs, one hidden layer containing twenty neurons, and three outputs that predict the derivatives ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) of the quadrotor angles. The outputs of this network are integrated using the Heun method to obtain the predicted angles (ϕ, θ, ψ). The inputs to this network include u_2, u_3, u_4, $\bar{w}$, the variables calculated in the second stage, and the delayed angles.

In the fourth stage, the second functional block is designed to compute the variables (u_x and u_y) based on the predicted angles that were calculated in the previous stage, as follows:

$$\begin{aligned} u_x &= \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ u_y &= \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi). \end{aligned} \tag{5}$$

The final stage of the proposed model consists of two NARX networks. The first network is designed with ten inputs, one hidden layer containing twenty neurons, and two outputs that predict the derivatives ($\dot{X}$ and $\dot{Y}$) of the quadrotor's positions. These outputs are then integrated using the Heun method to calculate the predicted positions (X and Y). The inputs to this network include (u_1, u_x, u_y, ϕ, θ, ψ), the variables calculated in the previous stages, as well as the delayed (u_x and u_y) and the delayed outputs from the network.

The second network has one output, which predicts the derivative ($\dot{Z}$) of the quadrotor's altitude. This

network consists of six inputs (u_1, cos(ϕ), cos(θ), cos($q^{-1}\cdot\phi$), cos($q^{-1}\cdot\theta$)), along with the delayed variable of its own output. It has one hidden layer containing ten neurons. The output of this network is also integrated using the Heun method to calculate the predicted altitude. The block diagram of the proposed model is given in Fig. 3.

3.2 Quadrotor state model

In order to train and test the developed prediction model, the dynamic model of the quadrotor given by Eq. (6) and that of the rotors, as described by Eq. (7), were used [32]. This model serves as a benchmark for generating system data under various conditions, which is essential for both training and validation of the developed model.

The constants are provided in Table 1, while the total thrust (u_1), the roll torque (u_2), the pitch torque (u_3), the yaw torque (u_4), and the average rotor angular velocity ($\bar{w}$) are defined by Eq. (2). Where, $w_{i=1:4}$ represent the angular velocities of the rotors, which are determined by integrating the rotors' dynamic model described in Eq. (7):

$$\begin{aligned}
\ddot{\phi} &= \frac{1}{I_x}\left(\dot{\theta}\dot{\psi}\left(I_y - I_z\right) + d\cdot u_2 - k_{fax}\dot{\phi}^2 - J_r\bar{w}\dot{\theta}\right)\\
\ddot{\theta} &= \frac{1}{I_y}\left(\dot{\phi}\dot{\psi}\left(I_z - I_x\right) + d\cdot u_3 - k_{fay}\dot{\theta}^2 + J_r\bar{w}\dot{\phi}\right)\\
\ddot{\psi} &= \frac{1}{I_z}\left(\dot{\phi}\dot{\theta}\left(I_x - I_y\right) + u_4 - k_{faz}\dot{\psi}^2\right)\\
\ddot{x} &= \frac{1}{m}\left(\left[\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)\right]\cdot u_1 - k_{ftx}\dot{x}\right)\\
\ddot{y} &= \frac{1}{m}\left(\left[\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)\right]\cdot u_1 - k_{fty}\dot{y}\right)\\
\ddot{z} &= \frac{1}{m}\left(\left[\cos(\phi)\cos(\theta)\right]\cdot u_1 - k_{ftz}\dot{z}\right) - g
\end{aligned} \tag{6}$$

$$\dot{w}_i = \frac{k_m}{R_m\cdot J_r}\cdot V_i - \frac{C_s}{J_r} - \frac{k_e\cdot k_m}{R_m\cdot J_r}\cdot w_i - \frac{k_r}{J_r}\cdot w_i^2 \tag{7}$$

where i = 1:4.

The parameters values of the rotors are given in Table 2.

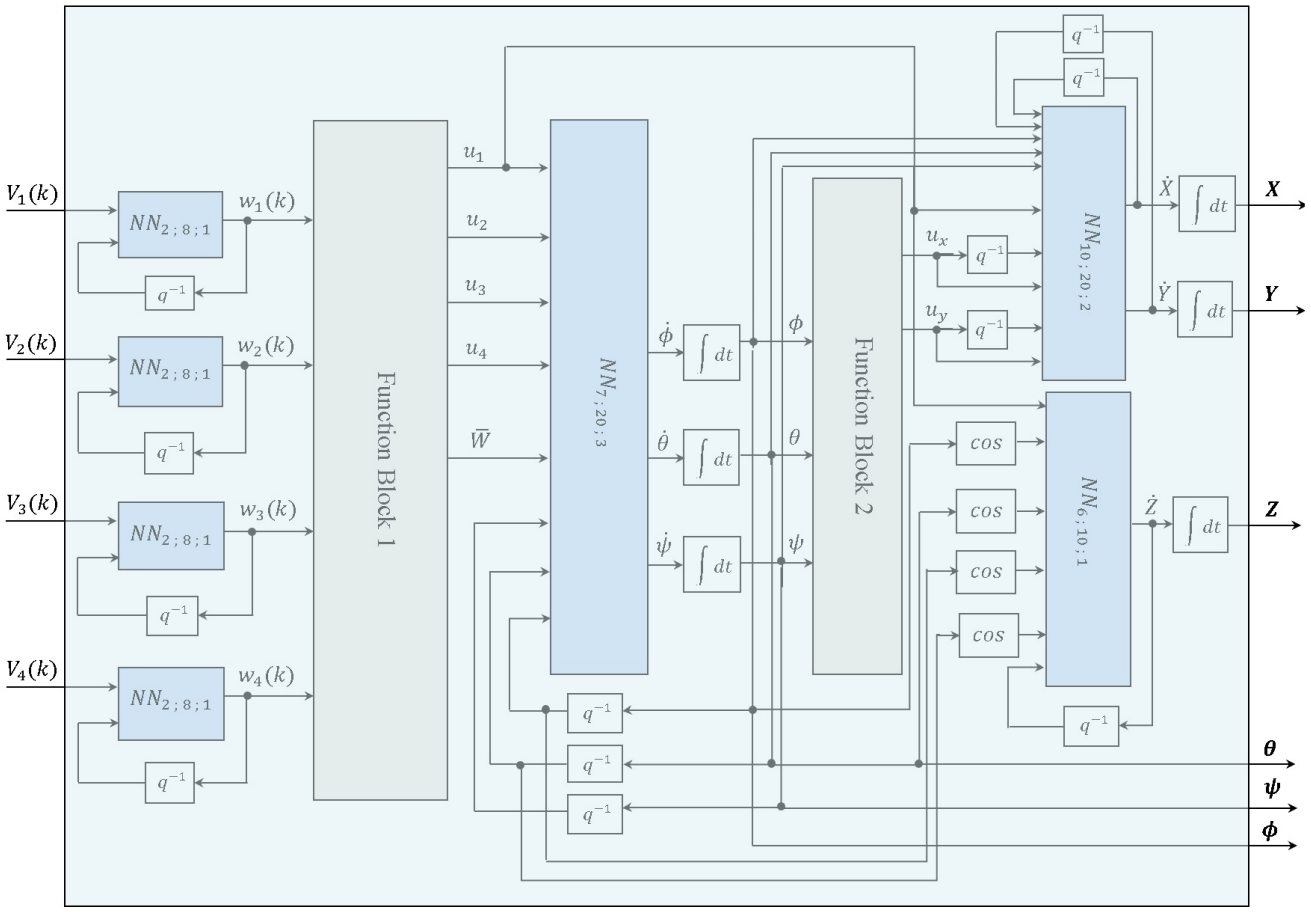

Fig. 3 Block diagram of the proposed model

Table 1 Quadrotor constants values [31]

Constant	Value	Constant	Value
I_x (N·m/rad/s^2)	$3.8278 \cdot 10^{-3}$	k_{fax} (N/rad/s)	$5.567 \cdot 10^{-4}$
I_y (N·m/rad/s^2)	$3.8288 \cdot 10^{-3}$	k_{fay} (N/rad/s)	$5.567 \cdot 10^{-4}$
I_z (N·m/rad/s^2)	$7.6566 \cdot 10^{-3}$	k_{faz} (N/rad/s)	$5.567 \cdot 10^{-4}$
k_{ftx} (N/m/s)	$5.567 \cdot 10^{-4}$	J_r (N·m/rad/s^2)	$2.8385 \cdot 10^{-5}$
k_{fty} (N/m/s)	$5.567 \cdot 10^{-4}$	d (m)	0.25
k_{ftz} (N/m/s)	$5.567 \cdot 10^{-4}$	m (kg)	0.486
g (m/s^2)	9.81		

Table 2 Rotors parameters values [31]

Constant	Value	Constant	Value
k_m (N·m/A)	$3.4629 \cdot 10^{-7}$	J_r (N·m/rad/s^2)	$2.8385 \cdot 10^{-5}$
k_e (V/rad/s)	$2.1632 \cdot 10^{-2}$	R_m (Ω)	$4.3541 \cdot 10^{-5}$
k_r (N·m)	$3.4629 \cdot 10^{-7}$	C_s (N·m·s/rad)	$5.3826 \cdot 10^{-3}$

3.3 Modulization results

To effectively train and test the developed prediction model, two datasets were generated using the dynamic models of the quadrotor and its rotors. Where, at each sampling time (10 ms), four random values between 0 and 12 were generated and applied as input voltages (V_1, V_2, V_3, V_4) to the four rotors. These voltages were then used in the rotor dynamic model to compute the corresponding rotor angular velocities (w_1, w_2, w_3, w_4) through numerical integration of the state model.

Using these velocities and Eq. (4), the corresponding control inputs of the quadrotor (u_1, u_2, u_3, u_4, $\bar{W}$) were calculated. After that, the full quadrotor dynamic model was numerically integrated to compute the resulting state variables ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$, $\dot{X}$, $\dot{Y}$, $\dot{Z}$, ϕ, θ, ψ, X, Y, Z).

For each iteration, the input voltages (V_1, V_2, V_3, V_4), the intermediate computed values (u_1, u_2, u_3, u_4, $\bar{W}$), and the resulting quadrotor states ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$, $\dot{X}$, $\dot{Y}$, $\dot{Z}$, ϕ, θ, ψ, X, Y, Z) were stored to form one data sample. This process was repeated 9000 times to construct the full dataset. This dataset was then used to train the neural networks in the proposed prediction model, where the networks were provided with their corresponding input values and trained to approximate the desired output responses.

The results of training the proposed prediction model are illustrated in Figs. 4 and 5.

According to Figs. 4 and 5, it is clear that the proposed model is well-trained. The modeling error remains below 0.01% for all outputs. However, to further validate the trained model, the second dataset was used to calculate the predicted outputs of the quadrotor, which were then compared to those obtained using the state model. The results are presented in Figs. 6 and 7.

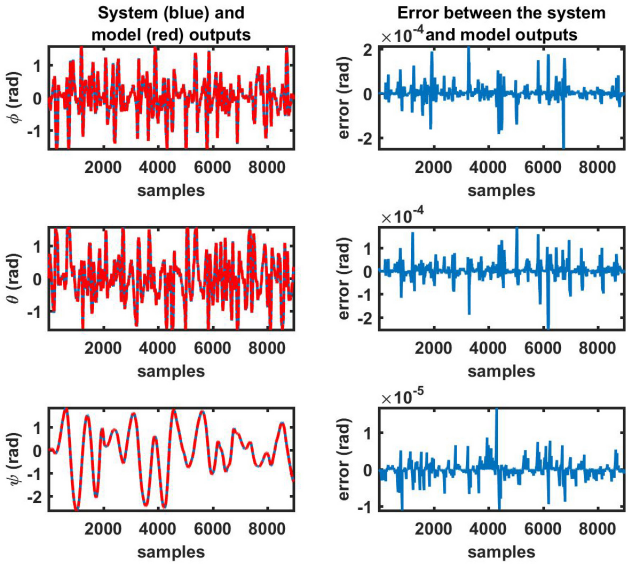

Fig. 4 Training results of the prediction model for the outputs (ϕ, θ, ψ)

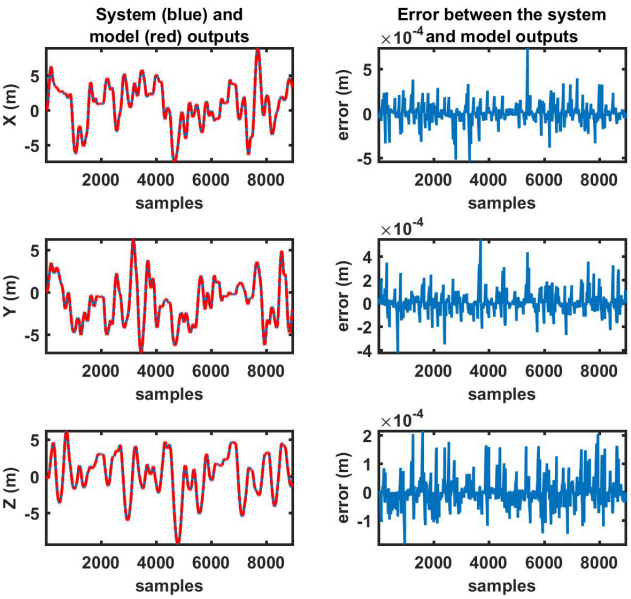

Fig. 5 Training results of the prediction model for the outputs (X, Y, Z)

According to Figs. 6 and 7, it can be seen that the modeling error for all predicted outputs is very small, remaining below 0.02%. This demonstrates that the proposed model is highly accurate and can be validated as a reliable prediction model of the quadrotor's behavior.

4 Constrained neural networks based model predictive control using SCA optimizer

4.1 Principle

Nearly all MPC algorithms operate on the same fundamental principle, which relies on an explicit prediction model combined with an optimization algorithm to generate a control sequence. This process can be summarized in the following steps:

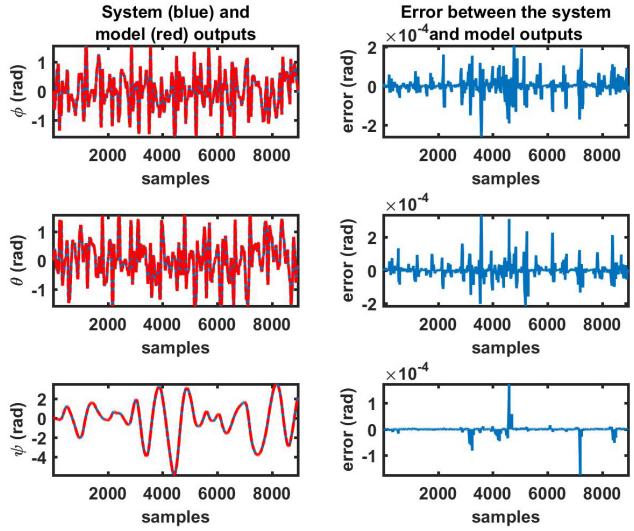

Fig. 6 Testing results of the prediction model for the outputs (ϕ, θ, ψ)

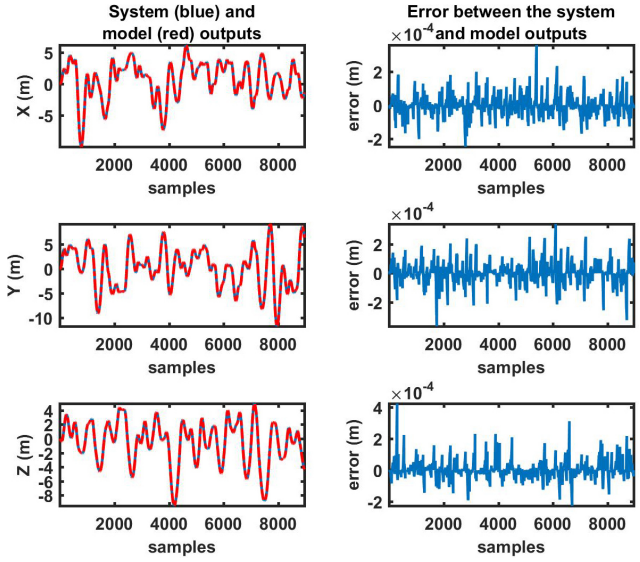

Fig. 7 Testing results of the prediction model for the outputs (X, Y, Z)

1. The future values of the system's outputs are calculated over the prediction horizon N_p using the prediction model.
2. The reference trajectory must be defined over at least the prediction horizon.
3. A control sequence is computed over the control horizon (N_p), but only the first element applied to the input to the controlled system.

4.2 CNNMPC-SCA control algorithm

To compute the control sequence, as outlined in the third step of the MPC principle, an objective function must be defined. This function incorporates all desired control performances. In the proposed CNNMPC-SCA algorithm, the primary objective is to minimize the tracking error between the predicted future outputs of the quadrotor ($\hat{X}$, $\hat{Y}$, $\hat{Z}$, $\hat{\psi}$), calculated using the prediction model over N_p, and their respective reference trajectories (X_r, Y_r, Z_r, ψ_r). The secondary objective is to minimize the variations in the control signals (ΔV_1, ΔV_2, ΔV_3, ΔV_4) to optimize energy consumption. Therefore, the first version of the objective function (J) is given in Eq. (8):

$$\begin{aligned} & J\left(\Delta \boldsymbol{V}(k), \hat{\boldsymbol{Y}}(k), \boldsymbol{C}(k)\right) \\ & = \sum_{i=N_1}^{N_2}\left[\left(\hat{\boldsymbol{Y}}(k+i|k)-\boldsymbol{C}(k+i)\right)^T \cdot \boldsymbol{Q} \cdot\left(\hat{\boldsymbol{Y}}(k+i|k)-\boldsymbol{C}(k+i)\right)\right] \\ & +\sum_{i=1}^{N_u}\left[\Delta \boldsymbol{V}^T(k+i-1) \cdot \boldsymbol{R} \cdot \Delta \boldsymbol{V}(k+i-1)\right] \end{aligned} \tag{8}$$

subject to:

$$\begin{aligned} & \Delta \boldsymbol{V}(k+i-1)=0 \text{ for } i>N_u \\ & \boldsymbol{Y} \min <\hat{\boldsymbol{Y}}(k)<\boldsymbol{Y} \max \\ & \boldsymbol{V} \min <\boldsymbol{V}(k)<\boldsymbol{V} \max \\ & \Delta \boldsymbol{V}_{\min }<\Delta \boldsymbol{V}(k+i-1)<\Delta \boldsymbol{V}_{\max } \text{ for } i \leq N_u . \end{aligned}$$

Where:

- $\hat{\boldsymbol{Y}}(k)=\left[\hat{X}(k), \hat{Y}(k), \hat{Z}(k), \hat{\psi}(k)\right]^T$: are the predicted outputs.
- $\boldsymbol{C}(k)=\left[X_r(k), Y_r(k), Z_r(k), \psi_r(k)\right]^T$: are the future reference trajectories.
- N_1 and N_2 are the minimum and the maximum of the prediction horizon N_p.
- $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive and semi-positive definite weighting matrices, respectively.

In addition to the first two objectives of the proposed controller, a third objective is introduced to manage the input and output constraints of the quadrotor within the CNNMPC-SCA algorithm. Indeed, the rotors of the quadrotor operate within a voltage range of [0, $V_{\max}$] where $V_{\max}$ varies depending on the rotor type. To address this, the proposed controller incorporates input constraints, given in Eq. (9), by penalizing the objective function.

$$0 \leq V \leq V_{\max} \tag{9}$$

Besides the input constraints, and to improve the stability of the quadrotor, the angles (ϕ, θ) are also subject to constraints, as shown in Eq. (10). These constraints are incorporated into the proposed control algorithm using a penalty-based approach. Consequently, the cost function adopted for the proposed controller is defined in Eq. (11):

$$-\frac{\pi}{2} \le \phi,\ \theta \le \frac{\pi}{2} \tag{10}$$

$$\begin{aligned} & J\left(\Delta \boldsymbol{V}(k), \hat{\boldsymbol{Y}}(k), \boldsymbol{C}(k)\right) \\ & = \sum_{i=N_1}^{N_2}\left[\left(\hat{\boldsymbol{Y}}(k+i|k)-\boldsymbol{C}(k+i)\right)^T \cdot \boldsymbol{Q} \cdot \left(\hat{\boldsymbol{Y}}(k+i|k)-\boldsymbol{C}(k+i)\right)\right] \\ & + \sum_{i=1}^{N_u}\left[\Delta \boldsymbol{V}^T(k+i-1) \cdot \boldsymbol{R} \cdot \Delta \boldsymbol{V}(k+i-1)\right] \\ & + \sum_{i=1}^{N_2}\left[\boldsymbol{\delta}(k+i|k) \cdot \boldsymbol{I}_n \cdot \Gamma(\delta_i)\right] \end{aligned} \tag{11}$$

subject to:

$$\Delta \boldsymbol{V}(k+i-1) = 0 \text{ for } i > N_u .$$

Where:

- $\boldsymbol{\delta}(\boldsymbol{k}) = [V_1(k), V_2(k), V_3(k), V_4(k), \phi(k), \theta(k)]$ represents the vector of constrained variables.
- $\boldsymbol{I}_n$ = ones(6, 6) is the identity matrix.
- $\Gamma(\delta_i)$ is the output-dependent weight function, it is defined as follows:

$$\Gamma(\delta_i) = \begin{cases} 0 & \text{if} \quad \delta_{\min_i} \le \delta_i \le \delta_{\max_i} \\ C_{i_\delta} & \text{if} \quad \delta_i < \delta_{\min_i} \text{ or } \delta_i > \delta_{\max_i} \end{cases} .$$

- C_{i_δ} is the penalization factor, where $C_{i_\delta} = 0$ indicates the absence of a constraint, and $C_{i_\delta} = \infty$ represents a hard constraint for i = 1:6.

The proposed CNNMPC-SCA is described in Algorithm 1.

5 Simulation study

To evaluate the efficiency of the proposed CNNMPC-SCA algorithm, a comparative study is carried out in Section 5 using the Adaptive Fuzzy PID (AFLC-PID) controller and the hybrid Fuzzy PID (FPID) controller while considering various operating conditions. In addition to the proposed controller, both the AFLC-PID and FPID controllers were implemented to control the quadrotor system that was defined by the state model in Eqs. (6) and (7). The design process of the AFLC-PID and the FPID controllers is detailed in [9], where their effectiveness was demonstrated. The control block diagram of the proposed CNNMPC-SCA algorithm is given in Fig. 8.

Several simulation scenarios are considered to ensure a comprehensive evaluation. The parameter values used for the proposed CNNMPC-SCA algorithm are provided in Table 3.

In the first simulation, a multistep reference trajectory is selected for all three quadrotor output positions (*X*, *Y*, *Z*), while a stationary reference trajectory equal to zero is set for the roll angle (ψ). The obtained control results are presented in Fig. 9. The control performances, including response time, overshoot, Mean Squared Error (MSE), and Mean Absolute Error (MAE), are summarized in Tables 4–6 for the considered controllers (CNNMPC-SCA, AFLCPID, and FPID).

From Fig. 9, it can be seen that the proposed CNNMPC-SCA controller provides the best control results for the quadrotor positions (*X*, *Y*, *Z*) compared to those obtained using the AFLCPID and FPID controllers. Indeed, the tracking error between the quadrotor positions and their respective reference trajectories, as given by the CNNMPC-SCA algorithm, is much smaller than those of the other controllers. This implies that the quadrotor adapts its positions rapidly and precisely according to the desired trajectories. To further reinforce this result, the control performances summarized in Tables 4–6 indicate that the CNNMPC-SCA controller is very efficient compared to the other considered controllers. Specifically, the MSE and MAE values provided by the proposed controller are much smaller than those obtained with the AFLCPID and FPID controllers, indicating better tracking accuracy of the desired reference trajectories for the quadrotor positions (*X*, *Y*, *Z*), except for the roll angle, where the FPID performs better. Furthermore, the response time of the proposed controller is notably fast compared to the other controllers, and the overshoot value obtained using the CNNMPC-SCA algorithm is very small, indicating superior stability of the quadrotor when using the proposed controller.

In the second simulation, a helical reference trajectory is selected to evaluate the effectiveness of the proposed controller in tracking nonlinear paths. The control results for each quadrotor output are shown in Fig. 10, while a 3D representation is provided in Fig. 11. Additionally, the response time, overshoot, MSE, and MAE for all considered controllers are summarized in Tables 7–9. The response time was measured at the moment when the quadrotor trajectory matched the shape of the desired trajectory, while the overshoot was determined at the peaks of each sinusoidal trajectory.

According to Figs. 10 and 11, the proposed CNNMPC-SCA algorithm demonstrates the best control performance for the quadrotor in tracking the helical reference trajectory. The tracking error provided by the CNNMPC-SCA is significantly smaller compared to those obtained with AFLCPID and FPID when controlling the quadrotor's positions (*X*, *Y*, *Z*). Furthermore, from Tables 7–9, it is evident that the quadrotor reaches the desired trajectory faster when controlled by CNNMPC-SCA, which, unlike the other controllers, does not produce any overshoot.

Algorithm 1 CNNMPC-SCA algorithm

Initialize the MPC parameters $\left(N_u, N_1, N_2, C_{i_5}, \boldsymbol{Q}, \boldsymbol{R}\right)$ for i = 1:6.

Initialize the SCA parameters $\left(D, N_{pop}, k_{\max}, a, X\max_{ij}, X\min_{ij}\right)$ for i = 1:N_{pop}, j = 1:D.

Load the trained proposed model given in Section 3.

for i = 1:N_{pop}
  for j = 1:D
    Generate the initial solutions (X_{ij}) using Eq. (1).
  End for
End for

Specify the reference trajectories $\left(X_r, Y_r, Z_r, \psi_r\right)$ for at least the next prediction horizon.

if the first iteration of the optimization process $\left(j = 1 \middle| j = 1 : k_{\max}\right)$ is being executed, **Do**:
  for i = 1:N_{pop}
    Compute the predicted outputs over N_p using the developed prediction model.
    Evaluate the objective function $J\left(\Delta\hat{\boldsymbol{V}}, \hat{\boldsymbol{Y}}, \boldsymbol{C}\right)$ using Eq. (11) to calculate the fitness (f).
  End for
  for i = 1:N_{pop}
    if $\left(f(i) <= \min(f)\right)$ **Do**:
      if $\left(\min(f) < f_{\text{best}}\right)$ **Do**:
        $f_{\text{best}} = \min(f)$.
        $P_g = X_{i,(1:D)}$.
      End if
      Break.
    End if
  End for
End if

Update the value of r_1 linearly from a to 0 according to Eq. (3).

for i = 1:N_{pop}
  Compute r_2, r_3 and r_4 as indicated in Section 2.
  Calculate the new position of each particle ($X\text{new}_i$) using Eq. (2).
End for

for i = 1:N_{pop}
  Compute the predicted outputs over N_p using the developed prediction model.
  Evaluate the objective function $J\left(\Delta\hat{\boldsymbol{V}}, \hat{\boldsymbol{Y}}, \boldsymbol{C}\right)$ using Eq. (11) to calculate the new fitness (f_{new}).
End for

for i = 1:N_{pop}
  if $\left(f_{\text{new}}(i) < f(i)\right)$ **Do**:
    $f(i) = f_{\text{new}}(i)$.
    $X_i = X\text{new}_i$.
  End if
End for

for i = 1:N_{pop}
  if $\left(f(i) <= \min(f)\right)$ **Do**:
    if $\left(\min(f) < f_{\text{best}}\right)$ **Do**:
      $f_{\text{best}} = \min(f)$.
      $P_g = X_{i,(1:D)}$.
    End if
    Break.
  End if
End for

if none of the stopping criteria of the optimization process are met, **Do**:
  Go back to line 7.
else:
  Apply the obtained control value (the first elements of P_g) on the quadrotor's inputs using the state model defined in Eq. (6).
  Wait for the next sampling time.
  if the predicted outputs of the model differ from the actual outputs of the quadrotor, **Do**:
    Retrain the prediction model to minimize the modeling error.
  End if
  Go back to line 4 (first loop).
End if

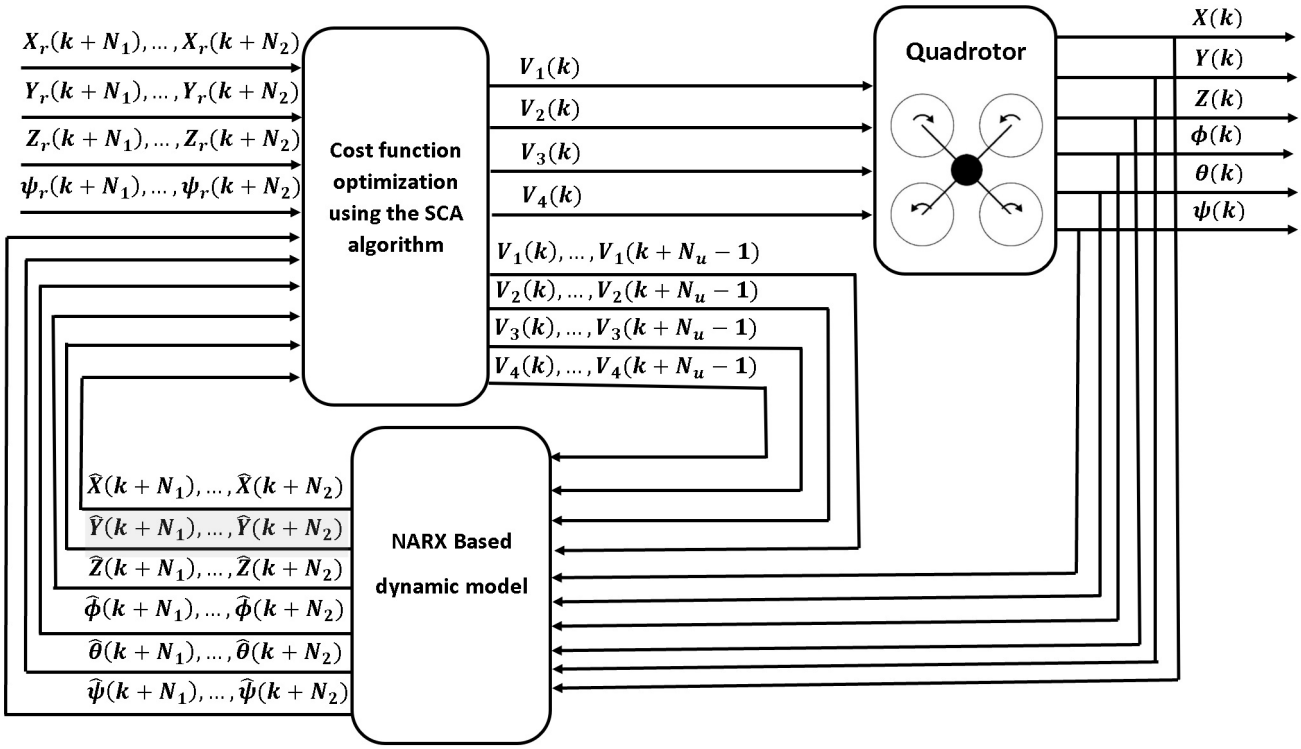

Fig. 8 CNNMPC-SCA control block diagram

Table 3 CNNMPC-SCA parameters values

Parameter	Value	Parameter	Value
N_1	1	D	50
N_2	50	N_{pop}	10
N_u	50	$k_{\max}$	100
C_{i_6}	10^6	a	2
Q	eye(4,4)	Xmax	$\left[12,12,12,12,\frac{\pi}{2},\frac{\pi}{2}\right]$
R	zeros(4,4)	Xmin	$\left[0,0,0,0,-\frac{\pi}{2},-\frac{\pi}{2}\right]$

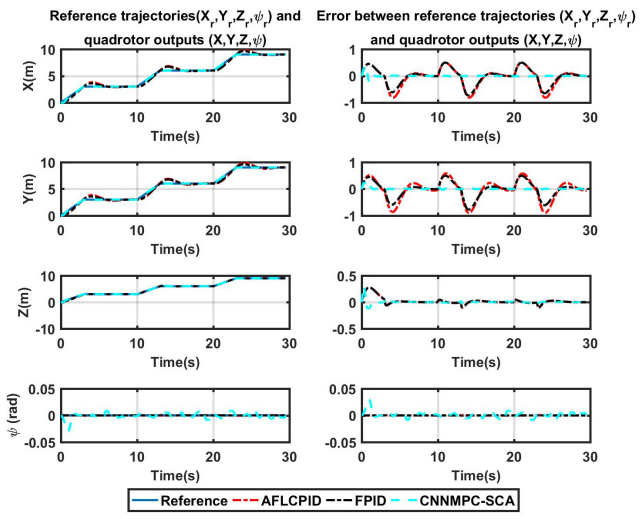

Fig. 9 Control results of the quadrotor using CNNMPC-SCA, AFLCPID, and FPID controllers for the multistep reference trajectory case

Table 4 CNNMPC-SCA's obtained control performance for the multistep reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	1.61 s	1.37 s	1.29 s	/
Overshoot	1.73%	0.8%	1.16%	/
MSE	6.2866	2.4942	3.1067	0.0966
MAE	49.2081	37.6245	33.8916	11.1805

Table 5 AFLCPID's obtained control performance for the multistep reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	7.91 s	5.71 s	3.37 s	/
Overshoot	27.4%	28.8%	2.16%	/
MSE	367.2685	454.2523	12.9054	8.928e-05
MAE	737.6834	867.1814	93.7593	0.1938

Table 6 FPID's obtained control performance for the multistep reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	7.44 s	5.74 s	3.34 s	/
Overshoot	20.6%	19.83%	2.16%	/
MSE	271.1004	267.7256	12.8291	2.794e-05
MAE	635.5665	628.9148	93.1249	0.0541

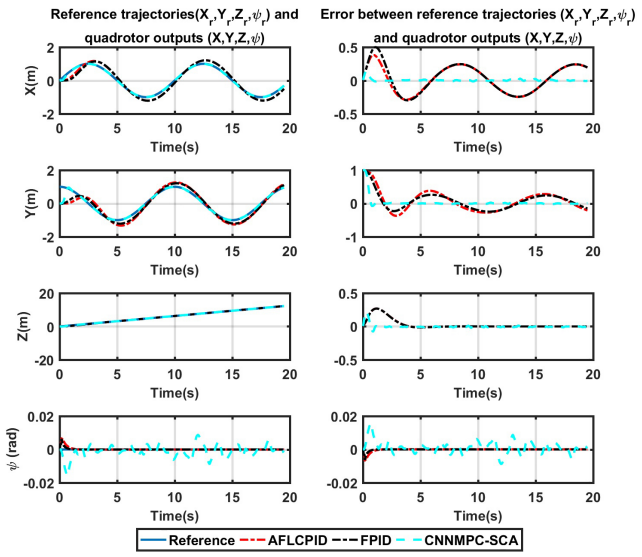

Fig. 10 Control results of the quadrotor using CNNMPC-SCA, AFLCPID, and FPID controllers for the helical reference trajectory case

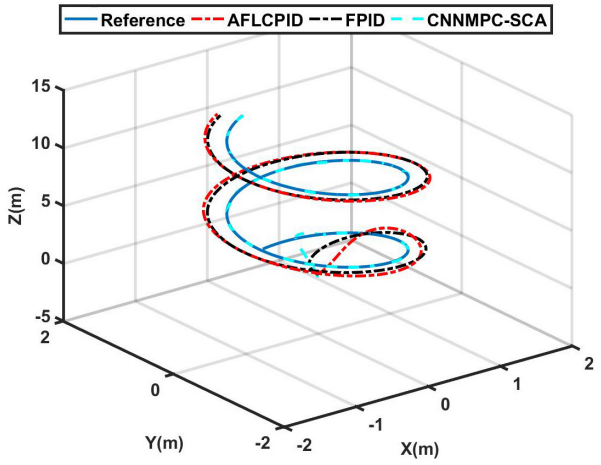

Fig. 11 3D representation of the quadrotor's trajectory for the helical reference case

Table 7 CNNMPC-SCA's obtained control performance for the helical reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	1.38 s	1.43 s	1.03 s	/
Overshoot	0%	0%	/	/
MSE	0.7858	40.0121	1.7046	0.0288
MAE	17.2903	62.5988	22.2561	5.5917

Table 8 AFLCPID's obtained control performance for the helical reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	2.28 s	1.90 s	3.59 s	/
Overshoot	16.8%	25.8%	/	/
MSE	73.5447	187.5627	11.0165	0.0013
MAE	339.3687	469.5831	58.4978	0.4584

Table 9 FPID's obtained control performance for the helical reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	2.55 s	1.78 s	3.59 s	/
Overshoot	14.2%	21.1%	/	/
MSE	87.0459	133.2267	10.9257	5.7664e-04
MAE	360.3316	376.0431	58.2344	0.2285

Additionally, the MSE and RMSE obtained with the proposed controller are considerably lower than those of the AFLCPID and FPID which demonstrates the superiority of CNNMPC-SCA in tracking a helical trajectory.

In the third simulation, a square reference trajectory is chosen for the quadrotor. The obtained control results are presented in Figs. 12 and 13 and Tables 10–12. Fig. 12 illustrates the reference trajectories alongside the quadrotor outputs and the tracking error for each output, while Fig. 13 provides a 3D representation of the desired and actual trajectory of the quadrotor. Tables 10–12 summarize the control performances: response time, overshoot, MSE, and MAE.

According to Figs. 12 and 13, the best control performances are achieved using the proposed controller. The tracking errors produced by CNNMPC-SCA are smaller than those obtained with the AFLCPID and FPID. This is further supported by the response time presented in Tables 10–12, where the proposed controller gives the shortest response time. Additionally, it achieves the smallest overshoot, highlighting its superiority. Furthermore, the MSE and RMSE values in Tables 10–12 demonstrate

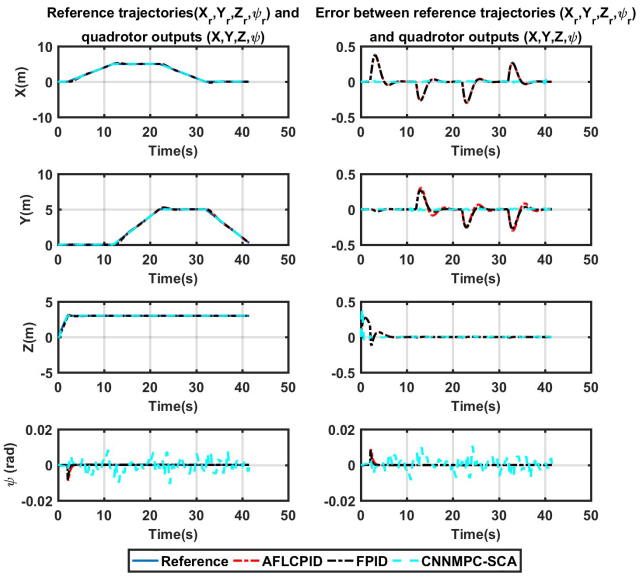

Fig. 12 Control results of the quadrotor using CNNMPC-SCA, AFLCPID, and FPID controllers for the square reference trajectory case

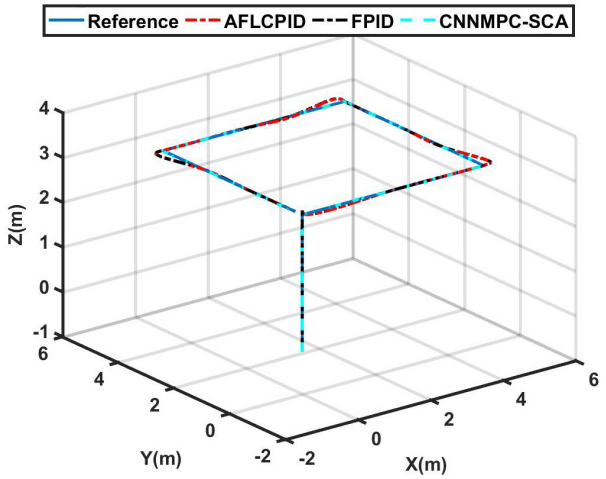

Fig. 13 3D representation of the quadrotor's trajectory for the square reference case

Table 10 CNNMPC-SCA's obtained control performance for the square reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	0 s	0 s	0.57 s	/
Overshoot	0%	0%	0%	/
MSE	0.2922	0.2950	4.7376	0.0439
MAE	26.1765	27.1524	33.3796	10.4919

Table 11 AFLCPID's obtained control performance for the square reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	2.49 s	2.21 s	4.37 s	/
Overshoot	7,54%	6%	4%	/
MSE	48.3302	32.9438	11.2891	0.0028
MAE	230.6753	191.7029	67.2219	0.8131

Table 12 FPID's obtained control performance for the square reference trajectory case

	Position (X)	Position (Y)	Position (Z)	Angle (ψ)
Response time (T_r)	2.49 s	2.18 s	4.37 s	/
Overshoot	7,54%	5.4%	4%	/
MSE	48.4100	27.5962	11.2821	0.0014
MAE	230.7780	156.7155	67.0204	0.4172

the high precision of the CNNMPC-SCA in tracking the square reference trajectory when compared to the other considered controllers.

For all three simulations, the proposed CNNMPC-SCA algorithm was subjected to the six constraints listed in Table 3. To assess its ability to handle these constraints, the minimum and maximum values of the variables $[V_1, V_2, V_3, V_4, \phi, \theta]$ are provided in Table 13 for all three simulations.

Table 13 Minimum and maximum values of the constrained variables obtained with the CNNMPC-SCA

	First simulation		Second simulation		Third simulation	
	Max	Min	Max	Min	Max	Min
V_1	12	0	12	0	12	0
V_2	12	0	12	0	12	0
V_3	12	0	12	0	12	0
V_4	12	0	12	0	12	0
ϕ	0.6577	−0.399	1.0431	−0.676	0.2045	−0.268
θ	0.8911	−0.855	0.291	−0.213	0.2436	−0.241

From Table 13, it can be concluded that all imposed constraints were satisfied, demonstrating the ability of the proposed CNNMPC-SCA algorithm to handle various constraints.

In order to evaluate the performance of the proposed controller under parameter variation, a fourth simulation is conducted. Where, a step reference trajectory is applied simultaneously to all three positional outputs of the quadrotor (X, Y, Z), while the roll angle (ψ) is assigned a fixed reference of zero. During this simulation, the quadrotor mass m is changed at t = 4 s from 0.486 kg to 1 kg, simulating the act of picking up an object. The resulting control behavior is illustrated in Fig. 14.

As shown in Fig. 14, the proposed controller effectively compensates for the parameter change. The impact of the mass variation is quickly attenuated, and the CNNMPC-SCA succeeds in re-stabilizing the system and driving it back to the target position within 1.03 s.

In the final simulation, the robustness of the CNNMPC-SCA controller is evaluated under the influence of external

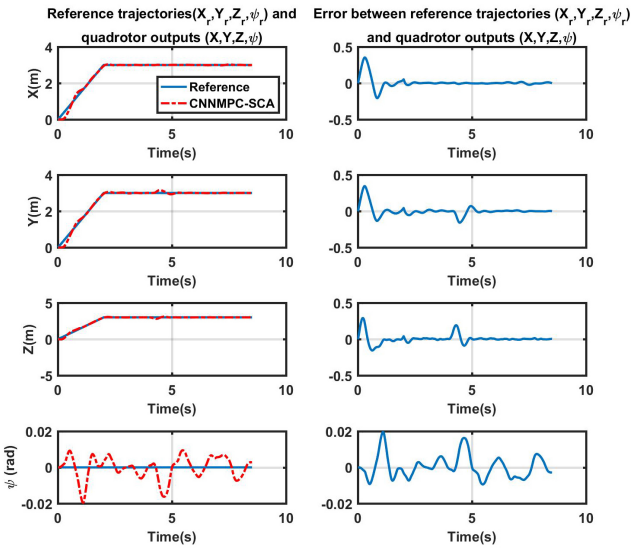

Fig. 14 Control results of the quadrotor using CNNMPC-SCA during a parameter variation scenario

disturbances applied to the quadrotor's outputs. Specifically, a constant positional disturbance of −50 cm is introduced at $t = 4$ s on the output positions (X, Y, Z). The reference trajectory remains unchanged from the previous test. The corresponding results are illustrated in Fig. 15.

From Fig. 15, it is clear that the proposed control strategy handles the imposes outputs disturbances effectively. The quadrotor quickly recovers and returns to the desired position, compensating for the X output disturbance in 750 ms, the Y output disturbance in 810 ms and the Z output disturbance in 630 ms. These results confirm the robustness of the CNNMPC-SCA in the presence of outputs perturbations.

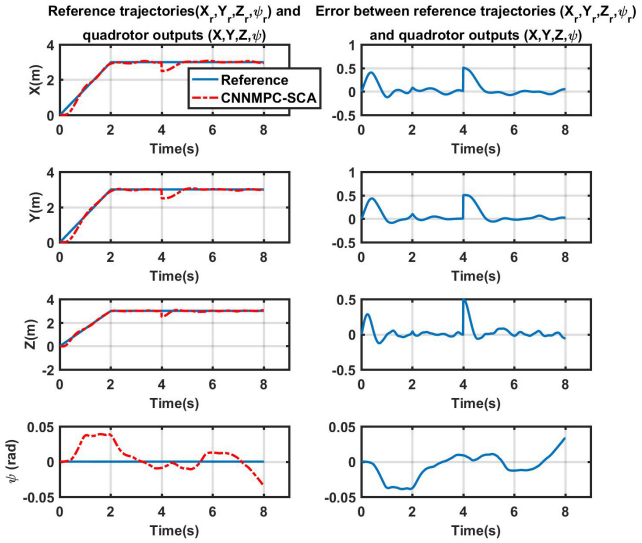

Fig. 15 Control results of the quadrotor using CNNMPC-SCA under output disturbance conditions

6 Conclusion

An efficient Constrained Neural Network-based Model Predictive Control using the Sine Cosine Algorithm has been developed to control the dynamics of quadrotors. This controller leverages a predictive model of quadrotor dynamics and a modified SCA metaheuristic to solve the associated optimization problem. The proposed model relies on seven NARX neural networks, which are known for their accuracy in modeling dynamic systems along with two functional blocks. The simulation results confirmed the high precision of the developed model, with prediction errors for all outputs remaining below 0.02%.

Additionally, the modified SCA incorporates a greedy selection step to enhance the convergence rate of the optimization process. The proposed controller is designed with six hard constraints specific to the quadrotor system while maintaining a simple architecture, making it well-suited for real-time implementation.

To evaluate its effectiveness, a comparative study was conducted against the AFLCPID and FPID controllers. The results demonstrated that the CNNMPC-SCA controller achieves superior performance in terms of response time, overshoot reduction, tracking accuracy, and constraint handling. Furthermore, the controller was evaluated under conditions involving parameter variations and output disturbances. The obtained performances confirm the robustness of the CNNMPC-SCA in such scenarios, highlighting its reliability for quadrotor control applications.

Acknowledgement

The project presented in this article is supported by the DGRSDT (Direction Générale de la Recherche Scientifique et du Développement Technologique).

References

[1] del Cerro, J., Cruz Ulloa, C., Barrientos, A., de León Rivas, J. "Unmanned aerial vehicles in agriculture: A survey", Agronomy, 11(2), 203, 2021. https://doi.org/10.3390/agronomy11020203

[2] Zeng, Y., Zhang, R., Lim, T. J. "Wireless communications with unmanned aerial vehicles: Opportunities and challenges", IEEE Communications Magazine, 54(5), pp. 36–42, 2016. https://doi.org/10.1109/MCOM.2016.7470933

[3] Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., Guizani, M. "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges", IEEE Access, 7, pp. 48572–48634, 2019. https://doi.org/10.1109/ACCESS.2019.2909530

[4] Yao, H., Qin, R., Chen, X. "Unmanned aerial vehicle for remote sensing applications—A review", Remote Sensing, 11(12), 1443, 2019. https://doi.org/10.3390/rs11121443

[5] Mohsan, S. A. H., Othman, N. Q. H., Li, Y., Alsharif, M. H., Khan, M. A. "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends", Intelligent Service Robotics, 16(1), pp. 109–137, 2023. https://doi.org/10.1007/s11370-022-00452-4

[6] Spica, R., Giordano, P. R., Ryll, M., Bülthoff, H. H., Franchi, A. "An open-source hardware/software architecture for quadrotor UAVs", IFAC Proceedings Volumes, 46(30), pp. 198–205, 2013. https://doi.org/10.3182/20131120-3-FR-4045.00006

[7] Lee, T., Leok, M., McClamroch, N. H. "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3)", [preprint] arXiv, arXiv:1003.2005v4, 09 September 2011. https://doi.org/10.48550/arXiv.1003.2005

[8] Stiti, C., Kara, K., Benrabah, M. "Optimized FLPID using TLBO algorithm: Applied to Quadrotor (UAV) system", In: 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS), Blida, Algeria, 2023, pp. 1–6. ISBN 978-1-6654-6311-9
https://doi.org/10.1109/ICAECCS56710.2023.10104893

[9] Amrouche, R., Benrabah, M., Fas, M. L. "A PSO-Based Type-2 Fuzzy Logic Controller Designed for Quadrotor Systems", In: 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1–6. ISBN 979-8-3503-4975-7
https://doi.org/10.1109/ICEEAC61226.2024.10576341

[10] Noordin, A., Mohd Basri, M. A., Mohamed, Z., Zainal Abidin, A. F. "Modelling and PSO fine-tuned PID control of quadrotor UAV", International Journal on Advanced Science, Engineering and Information Technology, 7(4), pp. 1367–1373, 2017.
https://doi.org/10.18517/ijaseit.7.4.3141

[11] Cádenas, J. A., Carrero, U. E., Camacho, E. C., Calderón, J. M. "Optimal PID ø axis Control for UAV Quadrotor based on Multi-Objective PSO", IFAC-PapersOnLine, 54(14), pp. 101–106, 2022.
https://doi.org/10.1016/j.ifacol.2022.07.590

[12] Runcharoon, K., Srichatrapimuk, V. "Sliding mode control of quadrotor", In: 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Konya, Turkey, 2013, pp. 552–557. ISBN 978-1-4673-5612-1
https://doi.org/10.1109/TAEECE.2013.6557334

[13] Shao, X., Sun, G., Yao, W., Liu, J., Wu, L. "Adaptive sliding mode control for quadrotor UAVs with input saturation", IEEE/ASME Transactions on Mechatronics, 27(3), pp. 1498–1509, 2022.
https://doi.org/10.1109/TMECH.2021.3094575

[14] Rinaldi, F., Chiesa, S., Quagliotti, F. "Linear quadratic control for quadrotors UAVs dynamics and formation flight", Journal of Intelligent & Robotic Systems, 70(1–4), pp. 203–220, 2013.
https://doi.org/10.1007/s10846-012-9708-3

[15] Panomrattanarug, B., Higuchi, K., Mora-Camino, F. "Attitude control of a quadrotor aircraft using LQR state feedback controller with full order state observer", In: The SICE Annual Conference 2013, Nagoya, Japan, 2013, pp. 2041–2046. ISBN 978-4-907764-43-2 [online] Available at: https://ieeexplore.ieee.org/document/6736320 [Accessed: 20 June 2025]

[16] Lopez-Sanchez, I., Moreno-Valenzuela, J. "PID control of quadrotor UAVs: A survey", Annual Reviews in Control, 56, 100900, 2023.
https://doi.org/10.1016/j.arcontrol.2023.100900

[17] Li, S., Wang, Y., Tan, J., Zheng, Y. "Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft", Neurocomputing, 216, pp. 126–134, 2016.
https://doi.org/10.1016/j.neucom.2016.07.033

[18] Nan, F., Sun, S., Foehn, P., Scaramuzza, D. "Nonlinear MPC for quadrotor fault-tolerant control", IEEE Robotics and Automation Letters, 7(2), pp. 5047–5054, 2022.
https://doi.org/10.1109/LRA.2022.3154033

[19] Salzmann, T., Kaufmann, E., Arrizabalaga, J., Pavone, M., Scaramuzza, D., Ryll, M. "Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms", IEEE Robotics and Automation Letters, 8(4), pp. 2397–2404, 2023.
https://doi.org/10.1109/LRA.2023.3246839

[20] Sun, S., Romero, A., Foehn, P., Kaufmann, E., Scaramuzza, D. "A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight", IEEE Transactions on Robotics, 38(6), pp. 3357–3373, 2022.
https://doi.org/10.1109/TRO.2022.3177279

[21] Eskandarpour, A., Sharf, I. "A constrained error-based MPC for path following of quadrotor with stability analysis", Nonlinear Dynamics, 99(2), pp. 899–918, 2020.
https://doi.org/10.1007/s11071-019-04859-0

[22] Stiti, C., Benrabah, M., Aouaichia, A., Oubelaid, A., Bajaj, M., Tuka, M. B., Kara, K. "Lyapunov-based neural network model predictive control using metaheuristic optimization approach", Scientific Reports, 14, 18760, 2024.
https://doi.org/10.1038/s41598-024-69365-9

[23] Benrabah, M., Kara, K., AitSahed, O., Hadjili, M. L. "Constrained Nonlinear Predictive Control Using Neural Networks and Teaching–Learning-Based Optimization", Journal of Control, Automation and Electrical Systems, 32(5), pp. 1228–1243, 2021.
https://doi.org/10.1007/s40313-021-00755-4

[24] Normey-Rico, J. E., Camacho, E. F. "Model Predictive Control of Dead-time Processes", In: Control of Dead-time Processes, Springer, 2007, pp. 271–308. ISBN 978-1-84628-829-6
https://doi.org/10.1007/978-1-84628-829-6_9

[25] Maciejowski, J. M., Huzmezan, M. "Predictive control", In: Robust Flight Control: A Design Challenge, Springer, 1997, pp. 125–134. ISBN 978-3-540-40941-0
https://doi.org/10.1007/BFb0113856

[26] Liu, H., Song, X. "Nonlinear system identification based on NARX network", In: 2015 10th Asian Control Conference (ASCC), Kota Kinabalu, Malaysia, 2015, pp. 1–6. ISBN 978-1-4799-7862-5
https://doi.org/10.1109/ASCC.2015.7244449

[27] Diaconescu, E. "The use of NARX neural networks to predict chaotic time series", WSEAS Transactions on Computer Research, 3(3), pp. 182–191, 2008.
https://dl.acm.org/doi/10.5555/1466884.1466892

[28] Mirjalili, S. "SCA: A sine cosine algorithm for solving optimization problems", Knowledge-Based Systems, 96, pp. 120–133, 2016.
https://doi.org/10.1016/j.knosys.2015.12.022

[29] Bansal, J. C., Bajpai, P., Rawat, A., Nagar, A. K. "Sine Cosine Algorithm for Optimization", [e-book] Springer, 2023. ISBN 978-981-19-9722-8
https://doi.org/10.1007/978-981-19-9722-8

[30] Abualigah, L., Diabat, A. "Advances in sine cosine algorithm: A comprehensive survey", Artificial Intelligence Review, 54(4), pp. 2567–2608, 2021.
https://doi.org/10.1007/s10462-020-09909-3

[31] Zeghlache, S. "Commande non linéaire d'un appareil à vol vertical" (Nonlinear control of a vertical flight vehicle), PhD Thesis, University of M'Sila - Mohamed Boudiaf, 2014. (in French)

[32] Derafa, L., Madani, T., Benallegue, A. "Dynamic modelling and experimental identification of four rotors helicopter parameters", In: 2006 IEEE International Conference on Industrial Technology, Mumbai, India, 2006, pp. 1834–1839. ISBN 1-4244-0725-7
https://doi.org/10.1109/ICIT.2006.372515