COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-1374
APPLICATION OF DIFFERENTIAL EVOLUTION FOR ADJUSTING THE PARAMETERS OF THE LQR TO A QUADROTOR

Laura Ribeiro
Aniel Silva de Morais
Gabriela Vieira Lima
Kenji Fabiano Avila Okada
Federal University of Uberlândia, Faculty of Electrical Engineering, Uberlândia – MG, Brazil
laurar@ufu.br; aniel@ufu.br; gabriela.lima@ufu.br; kenji_okada09@hotmail.com;

Abstract. *This work proposes an innovative approach that uses Differential Evolution (DE), an evolutionary algorithm, to adjust the parameters of the Q and R matrices that describe the Linear Quadratic Regulator (LQR) for the controller of a quadrotor, an Unmanned Aerial Vehicle (UAV). This paper presents the modeling, control system, and comparison of the results of eleven different techniques using MATLAB software with simulations. In addition, this study highlights the importance and benefit of an adequate control system for this type of UAV, owing to the high demand for the quality and reliability of these products, aiming to decrease as much as possible the number of accidents and reduce financial and environmental damage. The differential evolution algorithm application proved to be challenging, but the development and results were successful, asserting its applicability and effectiveness.*

Keywords: *differential evolution, quadrotor, LQR controller, Unmanned Aerial Vehicle, genetic algorithms*

1. INTRODUCTION

Society has witnessed the development of the famous Unmanned Aerial Vehicles (UAVs), with different physical and modeling characteristics, and involved in several applications in numerous areas, such as research, industrial, military, or civilian. With varying physical and modeling characteristics concerning the form of support to keep UAV in the air, the quadrotors are the fastest, lowest cost, and implementable type of UAV (Elmokadem and Savkin, 2021), becoming a straightforward choice for simpler applications. Also, the quadrotor UAV has a more stable performance with autonomy (Abdelghany et al., 2022).

However, to numerous adaptations, the quadrotor needs stability and making this control can be complicated. In this manner, it is necessary that these systems have desirable behaviors to generate practical applications, and these are achievable through the use of an appropriate control strategy (Rodríguez-Molina et al., 2020). Studies of controller Linear Quadratic Regulator (LQR) applications for problems in different fields are not new, and the design of control methods based on the optimal control theory is an improvement status (Fan et al., 2023).

The LQR methodology, with its optimal and very effective solution, is the basis of modern control and is widely used in the industry for its margin of gain, besides being applied to complex plants with Multiple-Input Multiple-Output (MIMO) systems. Thus, the problem appears when defining the matrices Q and R: these matrices need adequate weights to minimize the cost function, establishing a balance between the responses and the required forces. There are studies in control theory for the selection of the weights of LQR matrices using evolutionary optimization algorithms and their derivations (Katebi et al., 2020).

Evolutionary computation is inspired by the theory of evolution and natural selection, applying models from these processes to solve problems. In (Slowik and Kwasnicka, 2020), five main algorithms techniques have multiple varieties in the industry: Genetic Algorithms (GA), Evolution Strategies (ES), Genetic Programming (GP), Differential Evolution (DE) and Evolutionary Programming (EP).

Based on (Storn and Price, 1997), the DE algorithm shows a straightforward strategy search engine. It has strategies for minimization with above-expected performances when looking at the required numbers of function evaluations necessary to locate a global minimum of the test functions.

Therefore, the innovation proposed in this work is applying the Differential Evolution algorithm as a solution of choosing the ideal weights of Q and R matrices of the LQR controller in a control system for a quadrotor UAV. Moreover, all the topics covered and required will be explained later in the paper, including the main parameters for an LQR controller, that is, the Q and R weighting matrices. Since reaching a result for the setting time is done by trial and error (Deng et al., 2017), the optimization technique applied to adjust the Q and R matrices is a differential evolution algorithm; aiming to automate and find a better fit for 16 variables by means of multiple operators with varying methodologies.

2. MODELING THE QUADROTOR

The quadrotor model is based on the quadrotor from (Zhong et al., 2018) and was also used in previous work by the authors (Avila Okada et al., 2021). The rigid structure of the quadrotor UAV has four motors that take off and are sustained by the thrust of their propellers, acting simultaneously and independently, as seen as a representation in Figure 1.

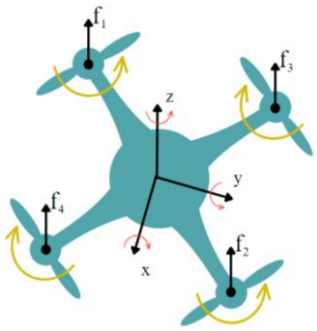

Figure 1. Schematic and Representation of a Quadrotor UAV.

The propeller's forces and momentum vary with the motor's angular velocity, and there are two motor pairs, one of the pairs rotates clockwise, while the other pair rotates counterclockwise. This setup counterbalances the moments, resulting in motion. The UAV under analysis has a quadrotor system with a nonlinear, underactuated model due to the need to control all six degrees of freedom through four control variables. Obtains the roll angle "ϕ" (movement of an aircraft around the x–axis) and pitch "θ" (movement of an aircraft around the y–axis) by varying the speed of the opposed engines. The yaw motion "ψ" (movement around the z–axis) occurs when there is a difference between the clockwise and counterclockwise motor speeds. Concerning linear displacement, the variation of the combined speed of the four motors equally results in the position of the UAV in the inertial coordinate system (x, y, z).

Equation (1) presents the nonlinear dynamics of the quadrotor, considering small-scale variations of the angular velocities *p*, *q*, and *r* and the speeds ϕ, θ, and ψ.

$$\begin{cases} \ddot{x} = (cos\,\phi\, sin\,\theta\, cos\,\psi + sin\,\phi\, sin\,\psi)\dfrac{u_z}{m} \\ \ddot{y} = (cos\,\phi\, sin\,\theta\, sin\,\psi - sin\,\phi\, cos\,\psi)\dfrac{u_z}{m} \\ \ddot{z} = cos\,\phi\, cos\,\theta\dfrac{u_z}{m} - g \\ \ddot{\phi} = \dfrac{I_y - I_z}{I_x}\dot{\phi}\dot{\theta} + \dfrac{u_\phi}{I_x} \\ \ddot{\theta} = \dfrac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \dfrac{u_\theta}{I_x} \\ \ddot{\psi} = \dfrac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \dfrac{u_\psi}{I_z} \end{cases} \quad (1)$$

To design the LQR controller with Integrator, one must consider that the design will base on linear models of the system. This way, the UAV is hovering relative to the vertical, that is, $u_z \approx mg$, and the variations will be small in roll and pitch angles so that approximately $sin\,\phi \approx \phi$ and $sin\,\theta \approx \theta$. Also, there will be no changes or variations in the yaw angle, where the value remains zero during the aircraft's displacement. In this context, the linear model of the quadrotor can be represented by Eq. (2).

$$\begin{cases} \ddot{x} = \theta g \\ \ddot{y} = -\phi g \\ \ddot{z} = \dfrac{u_z}{m} - g \\ \ddot{\phi} = \dfrac{u_\phi}{I_x} \\ \ddot{\theta} = \dfrac{u_\theta}{I_x} \\ \ddot{\psi} = \dfrac{u_\psi}{I_z} \end{cases} \quad (2)$$

In Eq. (3), the linear and angular motions generated by the forces of each of the four motors, f_1, f_2, f_3, and f_4, is a result of a function of the total force, u_z, and torques u_ϕ, u_θ, and u_ψ.

$$\begin{bmatrix} u_z \\ u_\psi \\ u_\theta \\ u_\phi \end{bmatrix} = T_f \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{3}$$

And the Eq. (4) define the torques.

$$\begin{bmatrix} u_z \\ u_\psi \\ u_\theta \\ u_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{C_t}}{\sqrt{2}r} & -\frac{\sqrt{C_t}}{\sqrt{2}r} & \frac{\sqrt{C_t}}{\sqrt{2}r} & -\frac{\sqrt{C_t}}{\sqrt{2}r} \\ -\frac{d\sqrt{2}}{2} & -\frac{d\sqrt{2}}{2} & \frac{d\sqrt{2}}{2} & \frac{d\sqrt{2}}{2} \\ -\frac{d\sqrt{2}}{2} & \frac{d\sqrt{2}}{2} & \frac{d\sqrt{2}}{2} & -\frac{d\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{4}$$

The simplification in the gain is justified by the fact that the engine dynamics are sufficiently faster than those of other air vehicles, not generating the transients of each engine response, significantly influencing the quadrotor behavior. The electrical voltage signal of each motor is associated with the force and momentum generated by the propeller per gain. That is due to the simplification of the motor dynamic, seen in Eq. (3) and Eq. (4). Without affecting and degrading the performance of the engines in the system, Eq. (2) is the linear model of the vehicle.

Table 1 describes the variables and the UAV model parameters settings. Also, the parameters values in Table 1 are related to the UAV for model building and code initialization.

Table 1. Experimental Variables and Settings Quadrotor's Parameters

Specification	Definition	Value
x, y, z	Inertial coordinate system	-
ϕ, θ, ψ	Angles roll, pitch, and yaw	-
m	Mass of the Quadrotor	$6.3\text{x}10^{-2}$ kg
I_x	Moment of inertia in relation to the x axis of the coordinate system fixed to the rigid body	$5.829\text{x}10^{-5}$ kg.m²
I_y	Moment of inertia in relation to the y axis of the coordinate system fixed to the rigid body	$7.169\text{x}10^{-5}$ kg.m²
I_z	Moment of inertia in relation to the z axis of the coordinate system fixed to the rigid body	$1\text{x}10^{-4}$ kg.m²
u_z	Total lift force	-
u_ϕ, u_θ, u_ψ	Moment roll, pitch, and yaw	-
f_1, f_2, f_3, f_4	Forces 1, 2, 3, and 4 generated by propellers	-
C_t	Lift factor	$1.07\text{x}10^{-2}$
d	Quadrotor constant	$6.24\text{x}10^{-2}$ m
r	Propeller Radius	$3.3\text{x}10^{-2}$ m
g	Gravity acceleration	9.81 m/s²

3. LQR CONTROLLER

The LQR controller was chosen for the system mainly because it provides optimal control in a feedback control solution with minimum cost. As mentioned, several applications and control studies employ this methodology (Ur Rehman et al., 2021).

However, the weights adjustments that compose the matrices Q and R, i.e., their parameters, can be a task that demands time and computational cost, especially for more complex systems characterized by the number of states and the coupling between them. The system response and its characteristics, such as the spare signal, the settling time, the response time, and its behavior throughout the simulation, among others, must be considered when adjusting the values of these matrix components. Figure 2 illustrates a generic block diagram of a control system, having the plant with the effect of the controller in its closed-loop.

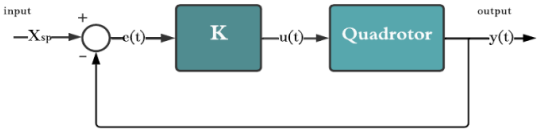

Figure 2. Schematic diagram of the control strategy.

For the controller design, it is necessary to consider which vehicle model will be represented by equations in the form of state space, mainly to facilitate the use and application of the controller. The state space representation of the quadrotor model can be expressed in Eq. (5), considering that all states are measured, where the matrices are $A \in R^{n\,x\,n}$, $B \in R^{n\,x\,m}$, $C \in R^{p\,x\,n}$, $D \in R^{n\,x\,d}$ and $G \in R^{n\,xl}$.

$$\begin{cases} \dot{x} = Ax + Bu + Gg \\ y = Cx \end{cases} \tag{5}$$

Therefore, the state vector is $x = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ \phi\ \theta\ \psi\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T \in R^n$, the control signal vector is $u = [u_z\ \tau_\phi\ \tau_\theta\ \tau_\psi]^T \in R^m$, and the system measurement signal vector is $y = x \in R^p$.

Consequently, by the state space representation of Eq. (5), the controllability matrix $Cx = [B\ AB\ ...\ A^{(n-1)}B]$ can be obtained, which has, for this model, a rank equal to that of matrix A. That is, there is no rank loss, which satisfies the condition of being a controllable system. The software verified the system as controllable, meaning a controller can be implemented. The controller gain, K in Eq. (6), defines the state feedback law for the continuous system. The minimization of the cost function, J, shown in Eq. (7), generates the gain calculation.

$$u = -K(x - x_{sp}) \tag{6}$$

$$J(u) = \int_0^\infty (x^T Q x + u^T R u) dt \tag{7}$$

Where the variable x_{sp} represents the desired value for each state. The Eq. (6) and Eq. (7) submitted to the dynamics of the system represented by Eq. (5), being obtained through the solution of the algebraic Riccati equation, in Eq. (8), finding the K in Eq. (9) with $P \in R^{n\,x\,n} > 0$.

$$A^T P + PA - PBR^{-1}B^T P + Q = -\dot{P} \tag{8}$$

$$K = R^{-1}B^T P \tag{9}$$

The parameters of the controller $Q \in R^{n\,x\,n} \geq 0$ and $R \in R^{m\,x\,m} > 0$ are matrices composed of weights and are related to both the states Q and R.

In the LQR design, choosing the K value aims to achieve optimization of the performance index J. This ensures that the feedback gain K is the most favorable in terms of the index performance (Ur Rehman et al., 2021). Still, these matrices vary in size and dimension according to the system, also determining the complexity of the system. Although the weights values are hard to define, there were empirical observations that the LQR method provides a better performance when compared to different controllers, like the classic PID (Quresh et al., 2019).

4. THE DIFFERENTIAL EVOLUTION ALGORITHM

The evolutionary computing algorithms area has a good effect in solving many complex problems (Chen et al., 2019), being a powerful method, solving many real-world multi-objective optimization problems (Quresh et al., 2019) and is mainly popular in industrial applications (Slowik and Kwasnicka, 2020). The classic problem of search and optimization is recurrent for scientists and engineers (Das and Suganthan, 2011), reading optimization as an act of searching for the most suitable solution to the problem with constraints and flexibilities.

DE has interesting characteristics, like the ability to deal with nonlinear, non-differentiable, and multimodal cost functions, besides presenting accessibility (a few control variables whose values are simplified) and self-adjustment of the adaptation step. It also has good convergence properties, ideal for maximization and minimization problems in consecutive trials (Storn and Price, 1997). Furthermore, it is well known the use of differential evolution algorithm in UAV problem applications.

Over simplifying, DE follows an evolution methodology based on the weighted difference between two population members with respect to another individual. Proposed in 1996 (Gonzalez et al., 2020) it takes inspiration from a genetic algorithm composed of three basic concepts: differential mutation, crossover, and selection. Due to the similarities, the

DE algorithm starts with an initial population, challenging individuals, applying mutations and crossovers until it finds a new population with individuals containing a better fitness result for the given problem (Gonzalez et al., 2020; Majid et al., 2022). The DE algorithm developed here tests with ten different structures, shown in Table 2, based on (Qiang et al., 2016), which are common and conventional variations.

In this work, the DE starts with a random population capable of respecting the restrictions of the limits of the variables under analysis. Sequentially, perform the mutation process. In the mutation process, the user chooses which structure to use, generating a new individual via crossover. Then, compare this newly created individual with the ones of the past generation of the DE, or, in other words, the individual with the best aptitude prevails in the population.

Hence, the structures vary according to three principal parameters: choosing an individual, the number of differences to use during mutation, and the type of crossover strategies.

Table 2. Strategies of Differential Evolution Mutation.

DE Notation	Strategy Mutation
Best/1/bin	$\vec{v}_{i,G+1} = \vec{x}_{best,G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G})$
Best/2/bin	$\vec{v}_{i,G+1} = \vec{x}_{best,G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}) + F(\vec{x}_{r3,G} - \vec{x}_{r4,G})$
Rand/1/bin	$\vec{v}_{i,G+1} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G})$
Rand/2/bin	$\vec{v}_{i,G+1} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) + F(\vec{x}_{r4,G} - \vec{x}_{r5,G})$
Best/1/exp	$\vec{v}_{i,G+1} = \vec{x}_{best,G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G})$
Best/2/exp	$\vec{v}_{i,G+1} = \vec{x}_{best,G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}) + F(\vec{x}_{r3,G} - \vec{x}_{r4,G})$
Rand/1/exp	$\vec{v}_{i,G+1} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G})$
Rand/2/exp	$\vec{v}_{i,G+1} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) + F(\vec{x}_{r4,G} - \vec{x}_{r5,G})$
Rand-to-best/2/bin	$\vec{v}_{i,G+1} = \vec{x}_{old,G} + F(\vec{x}_{best,G} - \vec{x}_{old,G}) + F(\vec{x}_{r2,G} - \vec{x}_{r1,G})$
Rand-to-best/2/exp	$\vec{v}_{i,G+1} = \vec{x}_{old,G} + F(\vec{x}_{best,G} - \vec{x}_{old,G}) + F(\vec{x}_{r2,G} - \vec{x}_{r1,G})$
Rand-to-best/2/exp	$\vec{v}_{i,G+1} = \vec{x}_{best,G} + F(\vec{x}_{r1,G} - \vec{x}_{r2,G})$

Where for each target vector or population member $\vec{x}_{i,G}$, $i = 1, 2, 3 \ldots, NP$. And $r_1, r_1, r_1 \in 1, 2 \ldots, NP$ are mutually distinct indices and different from index i. The constant F is real $\in [0,2]$, which determines the amplification and the step taken in the direction defined by the difference vector. The $\vec{x}_{best,G}$ is the best solution for the generation G. In addition, DE notations introduced before is to facilitate the differentiation of its main variants (Das and Suganthan, 2011). The generic notation in literature is *DE/a/b/c*, where:

- *a:* Specifies the vector to be mutated or perturbed.
- *b:* Determines the number of difference vectors (directions) used in the mutation step.
- *c:* Indicates the crossover adopted. Specifies the vector to be mutated or perturbed.

Note that there are two possibilities of crossover: binomial and exponential. Binomial, or binary, crossover tests the individual vector on each term as determined by the crossover rate. In this case, the previous test result does not interfere in the crossing of the next term. As for the exponential crossover, once the crossover rate test determines that the term will not undergo crossover, all following terms of the vector will not undergo crossover, too.

5. RESULTS AND DISCUSSION

Firstly, the main challenge of this application was the determination of an initial random population that covered the whole possible field for the *Q* and *R* variables since these can vary from a small value (in the order of 10^{-7}) to a high value (in the order of 10^{3}) and apply the necessary restrictions so that the final result found is producing suitable signal that does not introduce instability in the system and that respects the operating limits of the actuators, which in this case, represent the quadrotor motors.

For the optimization algorithm tests, it was noticed, by empirical means, that the use of a population larger than 500 individuals and a number above 300 generations have no effect of significant improvements in the results compared to the computational effort used. In view of this fact, for the final tests, analysis and plots contained in the paper, the number of generations and the number of individuals chosen, among other parameters for the execution of the algorithm, can be seen in Table 3. This work's application, the graphs of the mean and standard deviation of the fitness function, and the parameters *Q* and *R* utilize the software MATLAB.

As a comparison and analysis, a code with all the necessary parameters for a Genetic Algorithm (GA), was developed, too. The GA, created in 1975, is an optimization and stochastic global search technique which takes inspiration from the principles of genetics and natural selection. The basic structure of the DE is similar to the GA: coding, selection, crossover, and mutation.

Therefore, for other parameters, such as the crossover rate and mutation constants, applied values commonly pointed out as ideal for each of the methodologies (DE and GA), and when analyzing the results with these and other values, the

"ideal" values produced better results and with less variation. Table 3 also shows the parameters and values for the GA code used. In addition, the limits for Q = [1e-7 1e3], for R = [1e-7 1e3] and the boundary for stability = [-10 10].

Table 3. Parameters for the Differential Evolution Algorithm and Genetic Algorithm.

Parameters	Differential Evolution	Genetic Algorithm
Number of generations	300	300
Number of individuals	500	500
Number of runs	10	10
Crossover method	-	Wright
Crossover rate	0.9	0.9
Elitism	-	1 individual
Probability of mutation	-	0.05
Real mutation constant	0.5	-
Selection of individuals	-	Tournament (3)

The calculation of the Absolute Error Integral (IAE) applies the fitness function of the algorithms. The IAE, popularly known in Control Systems areas as a benchmark for performance indices, is calculated by adding the difference between the desired linear positions with the values obtained by the system over time.

Table 4 displays the results of the DE structures obtained after the execution of the algorithm code.

Table 4. Final Values of Mean and Standard Deviation Per Structure.

S	***Strategy***	***Fitness***	***Standard Deviation***
1	Best/1/bin	0.9989	0.0104
2	Best/1/exp	0.9935	0.0052
3	Best/2/bin	0.9934	0.0042
4	Best/2/exp	1.0128	0.0161
5	Rand/1/bin	1.0215	0.0073
6	Rand/1/exp	1.0718	0.0146
7	Rand/2/bin	1.0520	0.0055
8	Rand/2/exp	1.0831	0.0108
9	Rand-to-best/2/bin	0.9962	0.0062
10	Rand-to-best/2/exp	1.0011	0.0075
11	GA	1.0679	0.0209

Figure 3 and Figure 4 illustrate the evolution of the mean and standard deviation values of the fitness function, the IAE, over the generations. It can be observed that the high value of the standard deviation of the GA is due to the randomness imposed on its population according to the mutation rate.

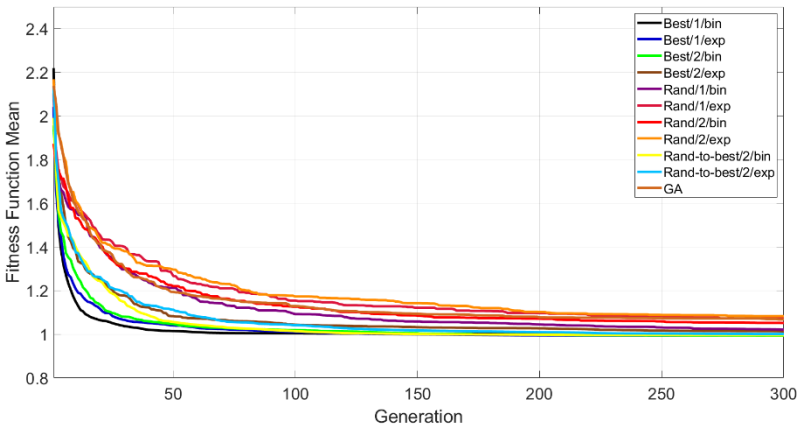

Figure 3. Mean of the Values of the Fitness Function (IAE) of each Structure over the Generations.

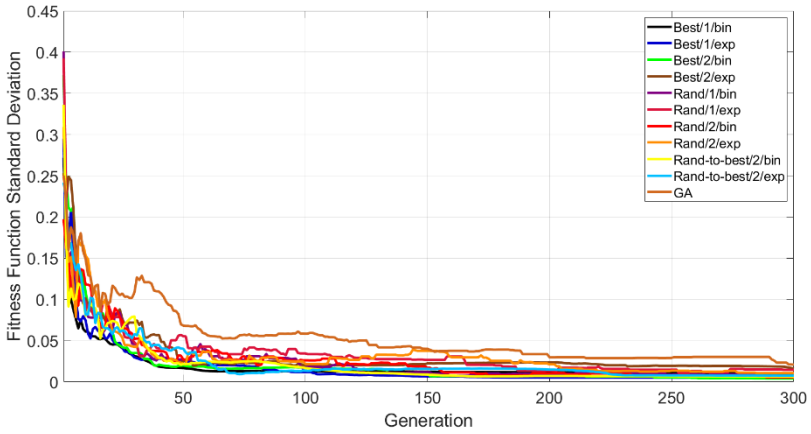

Figure 4. Standard Deviation of the Values of the Fitness Function (IAE) of each Structure over the Generations.

The analyses of the structure performance derive from the graphs of the results of the linear positions (x, y, z). Figure 5 and Figure 6 show each structure tested. Figure 5 illustrates in 2D the behavior of each structure and its linear positions over time in a closed-loop with different references. In this case, the setpoints were 1, 1.5, and -1. On the other hand, Figure 6 represents the evolution of linear positions for each structure in 3D with a unit setpoint.

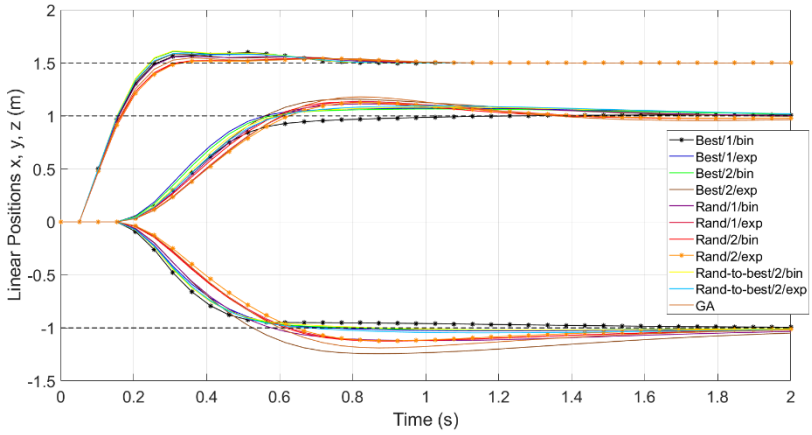

Figure 5. Evolution of linear positions as a function of Q and R for each structure.

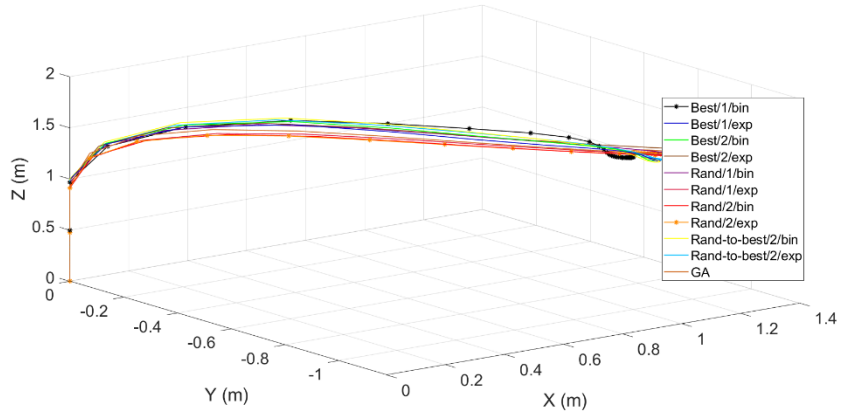

Figure 6. Evolution of linear positions as a function of Q and R for each structure (3D).

Considering the stabilization criterion of the fitness function average over the generations, the structure with better performance is the Best/1/bin, whereas the worst is the Rand/2/exp. The best result presents a fast response with the minimum possible spare signal without harming the system's robustness.

Table 5 and Table 6 show the results of the mean found for the variables Q and R for each structure, the first table is related to Q, and the other relates to R. As well, Table 7 and Table 8 present the results of the standard deviation found for the variables Q and R for each structure, as mentioned before.

Table 5. Mean of the Parameters Q Adjusted Per Structure

Structure	$Q_{1,1}$	$Q_{2,2}$	$Q_{3,3}$	$Q_{4,4}$	$Q_{5,5}$	$Q_{6,6}$	$Q_{7,7}$	$Q_{8,8}$	$Q_{9,9}$	$Q_{10,10}$	$Q_{11,11}$	$Q_{12,12}$
1	76.1	58.1	140.8	10.9	2e-6	0.31	235.4	223.7	284.2	1.30	0.01	1.63
2	67.8	36.5	59.9	25.9	9e-5	0.02	172.0	212.1	79.2	0.25	0.15	0.05
3	236.3	159.3	774.3	76.0	0.02	0.19	720.2	726.7	552.1	1.87	0.37	0.01
4	263.3	224.2	738.5	51.2	2.48	0.67	787.2	691.0	161.3	5.40	1.84	113.2
5	313.3	290.5	845.1	72.0	3.15	1.63	832.9	836.2	489.4	8.65	1.65	468.2
6	534.2	513.2	869.6	103.6	16.0	3.81	824.3	931.0	733.3	19.8	9.58	454.8
7	467.8	416.2	943.9	93.5	11.3	4.07	826.0	890.9	324.6	12.1	7.94	103.5
8	640.3	517.2	920.4	134.3	28.5	4.16	850.1	825.7	574.3	20.2	17.8	351.7
9	33.8	52.6	109.0	10.4	5e-4	3e-3	208.6	102.3	134.1	1.15	0.05	5.68
10	64.2	56.7	368.5	17.8	5e-3	0.31	201.4	188.8	96.0	1.51	0.14	25.0
11	362.9	488.3	738.8	76.6	29.5	1.88	622.2	575.8	250.5	24.5	8.86	73.9

Table 6. Mean of the Parameters R Adjusted Per Structure

Structure	$R_{1,1}$	$R_{2,2}$	$R_{3,3}$	$R_{4,4}$
1	0.56	100.0	0.20	0.75
2	0.28	73.3	18.1	13.5
3	3.59	102.8	176.3	143.4
4	3.30	75.4	123.3	174.2
5	3.57	589.2	382.2	432.0
6	3.52	447.6	550.4	611.6
7	3.31	396.1	148.0	298.3
8	3.49	343.7	163.3	389.3
9	0.51	57.8	29.0	26.6
10	1.63	49.2	33.4	47.0
11	3.27	102.6	75.8	1.95

Table 7. Standard Deviation of the Parameters Q Adjusted Per Structure

Structure	$Q_{1,1}$	$Q_{2,2}$	$Q_{3,3}$	$Q_{4,4}$	$Q_{5,5}$	$Q_{6,6}$	$Q_{7,7}$	$Q_{8,8}$	$Q_{9,9}$	$Q_{10,10}$	$Q_{11,11}$	$Q_{12,12}$
1	109.5	101.2	232.4	15.1	8e-6	0.97	408.9	323.3	401.1	2.30	0.04	5.16
2	92,.6	34.7	67.1	39.2	1e-4	0.04	177.2	292.2	115.5	0.37	0.23	0.15
3	93.7	77.2	205.1	52.0	0.03	0.54	329.9	294.4	378.1	2.49	0.34	0.02
4	106.6	84.3	282.0	43.1	5.38	0.56	266.2	299.3	245.6	4.74	4.63	313.8
5	56.0	113.2	110.0	76.2	3.39	1.02	155.1	126.8	392.3	7.61	2.24	432.5
6	192.4	293.0	132.3	145.8	28.1	2.43	208.3	163.6	382.1	21.5	14.0	484.9
7	161.9	277.9	77.1	74.0	19.7	1.41	216.4	191.9	303.9	20.2	8.24	162.6
8	323.9	306.3	142.5	138.4	48.8	1.70	286.8	259.3	325.8	29.1	16.8	308.2
9	14.1	49.3	125.8	8.22	7e-4	8e-3	164.8	45.1	123.4	1.73	0.05	10.0
10	43.7	57.1	332.3	21.6	0.01	0.50	148.7	132.0	119.9	2.42	0.28	42.3
11	325.7	279.5	358.7	151.6	41.1	2.07	384.1	342.5	338.1	23.2	16.1	203.9

Table 8. Standard Deviation of the Parameters R Adjusted Per Structure

Structure	$R_{1,1}$	$R_{2,2}$	$R_{3,3}$	$R_{4,4}$
1	0.91	316.2	0.63	1.77
2	0.31	204.9	41.1	38.4
3	0.94	130.2	294.3	295.5
4	1.21	121.6	192.0	237.2
5	0.39	412.8	394.0	295.0
6	0.81	313.7	398.5	454.0
7	0.81	371.4	267.9	329.9
8	0.82	371.5	183.4	436.6
9	0.59	66.0	5.0	26.9
10	1.41	78.1	67.8	49.2
11	2.01	275.7	124.5	6.16

The values vary, being in the tens to hundreds, remaining aligned with the objective of the application, that is, to reach such results with the algorithm used. Finally, it is possible to observe a significant improvement in the results throughout the generations, resulting in a suitable individual for the system, presenting an optimal and ideal solution for the problem.

6. CONCLUSIONS

This paper presents the results of the parameterization of an LQR controller applied to a quadrotor using different methodologies, applications, and junctions of major engineering areas. Besides promoting the study and highlighting the complexity of working with intelligent systems, the results also demonstrate the comparison and the advantage of using a Differential Evolution algorithm with an operator and appropriate settings for the best results, diversity in the structures, and less computational work.

The application of the differential evolution algorithm proved to be challenging. The system must produce a proper output signal, not show instability, remain controllable, and respect the operating limits of the actuators that, in this case, represent the quadrotor motors, respecting all the necessary constraints for a satisfactory result and performance.

Finally, it also emphasizes the importance and usefulness of an adequate control system for this type of aircraft due to the great demand for quality and reliability level of the systems, aiming to reduce the number of accidents and financial and environmental expenses surrounding it.

In search of acquiring new knowledge and promoting innovations, there is room for future improvements or modifications. Some of them are the comparison of the DE code with other strategies, the simplification of the code, the reduction in execution time, and finally, the creation of a graphic interface for the work.

7. ACKNOWLEDGEMENTS

The authors would like to thank the Postgraduate Program in Electrical Engineering (PPGEELT) of the Faculty of Electrical Engineering (FEELT) at the Federal University of Uberlândia (UFU) for the financial support provided. In addition, the author would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for supporting the execution of this project.

8. REFERENCES

Abdelghany, M. B., A. M. Moustafa, and M. Moness. 2022. “Benchmarking Tracking Autopilots for Quadrotor Aerial Robotic System Using Heuristic Nonlinear Controllers.” *Drones* 6 (12): 379. https://doi.org/10.3390/drones6120379.

Avila Okada, K. F., A. Silva De Morais, L. C. Oliveira-Lopes, and L. Ribeiro. 2021. “Neuroadaptive Observer-Based Fault-Diagnosis and Fault-Tolerant Control for Quadrotor UAV.” In *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*, 285–92. São Paulo, Brazil: IEEE. https://doi.org/10.1109/INDUSCON51756.2021.9529573.

Chen, H., Y. Yang, and J. Sun. 2019. “Improved Genetic Algorithm Based Optimal Control for A Flying Inverted Pendulum.” In *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, 1428–32. Xiamen, China: IEEE. https://doi.org/10.1109/EITCE47263.2019.9094769.

Das, S., and P. N. Suganthan. 2011. “Differential Evolution: A Survey of the State-of-the-Art.” *IEEE Transactions on Evolutionary Computation* 15 (1): 4–31. https://doi.org/10.1109/TEVC.2010.2059031.

Deng, X., X. Sun, R. Liu, and W. Wei. 2017. "Optimal Analysis of the Weighted Matrices in LQR Based on the Differential Evolution Algorithm." In *2017 29th Chinese Control And Decision Conference (CCDC)*, 832–36. Chongqing, China: IEEE. https://doi.org/10.1109/CCDC.2017.7978635.

Elmokadem, T., and A. V. Savkin. 2021. "Towards Fully Autonomous UAVs: A Survey." *Sensors* 21 (18): 6223. https://doi.org/10.3390/s21186223.

Fan, X., J. Wang, H. Wang, L. Yang, and C. Xia. 2023. "LQR Trajectory Tracking Control of Unmanned Wheeled Tractor Based on Improved Quantum Genetic Algorithm." *Machines* 11 (1): 62. https://doi.org/10.3390/machines11010062.

Gonzalez, V., C. A. Monje, S. Garrido, L. Moreno, and C. Balaguer. 2020. "Coverage Mission for UAVs Using Differential Evolution and Fast Marching Square Methods." *IEEE Aerospace and Electronic Systems Magazine* 35 (2): 18–29. https://doi.org/10.1109/MAES.2020.2966317.

Katebi, J., M. Shoaei-parchin, M. Shariati, N. T. Trung, and M. Khorami. 2020. "Developed Comparative Analysis of Metaheuristic Optimization Algorithms for Optimal Active Control of Structures." *Engineering with Computers* 36 (4): 1539–58. https://doi.org/10.1007/s00366-019-00780-7.

Majid, F., M. Mostafa, A. Hassan, and E. K. Abdeljalil. 2022. "Differential Evolution Approach for Identification and Control of Stable and Unstable Systems." In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, 218–23. Istanbul, Turkey: IEEE. https://doi.org/10.1109/CoDIT55151.2022.9804077.

Qiang, J., C. Mitchell, and A. Qiang. 2016. "Tuning of an Adaptive Unified Differential Evolution Algorithm for Global Optimization." In *2016 IEEE Congress on Evolutionary Computation (CEC)*, 4061–68. Vancouver, BC, Canada: IEEE. https://doi.org/10.1109/CEC.2016.7744305.

Quresh, K., S. Rahnamayan, Y. He, and R. Liscano. 2019. "Enhancing LQR Controller Using Optimized Real-Time System by GDE3 and NSGA-II Algorithms and Comparing with Conventional Method." In *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2074–81. Wellington, New Zealand: IEEE. https://doi.org/10.1109/CEC.2019.8789894.

Rodríguez-Molina, A., E. Mezura-Montes, M. G. Villarreal-Cervantes, and M. Aldape-Pérez. 2020. "Multi-Objective Meta-Heuristic Optimization in Intelligent Control: A Survey on the Controller Tuning Problem." *Applied Soft Computing* 93 (August): 106342. https://doi.org/10.1016/j.asoc.2020.106342.

Slowik, A., and H. Kwasnicka. 2020. "Evolutionary Algorithms and Their Applications to Engineering Problems." *Neural Computing and Applications* 32 (16): 12363–79. https://doi.org/10.1007/s00521-020-04832-8.

Storn, R., and K. Price. 1997. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." *Journal of Global Optimization* 11 (4): 341–59. https://doi.org/10.1023/A:1008202821328.

Ur Rehman, A., M. U. Khan, M. S. Shah, Q. Maqsood, M. F. Ullah, and S. Uba. 2021. "Aircraft Pitch Control Based on Genetic Algorithm Tuning with PID and LQR Controller." In *2021 6th International Multi-Topic ICT Conference (IMTIC)*, 1–6. Jamshoro & Karachi, Pakistan: IEEE. https://doi.org/10.1109/IMTIC53841.2021.9719736.

Zhong, Y., Y. Zhang, W. Zhang, J. Zuo, and H. Zhan. 2018. "Robust Actuator Fault Detection and Diagnosis for a Quadrotor UAV With External Disturbances." *IEEE Access* 6: 48169–80. https://doi.org/10.1109/ACCESS.2018.2867574.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.