2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE) | 978-1-6654-6280-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/LARS/SBR/WRE56824.2022.9995998

Optimization of Force and Torque Bounds for the Flight Control System of a Quadcopter using PSO

1st Luiz Giacomossi Jr.
Autonomous Computational Systems Lab (LAB-SCA)
Aeronautics Institute of Technology
São José dos Campos, Brazil
luiz.giacomossi@ga.ita.br

2nd Angelo Caregnato-Neto
Electronic Engineering Division
Aeronautics Institute of Technology
São José dos Campos, Brazil
angelo.neto@ga.ita.br

3rd Marcos R. O. A. Maximo
Autonomous Computational Systems Lab (LAB-SCA)
Aeronautics Institute of Technology
São José dos Campos, Brazil
mmaximo@ita.br

***Abstract*—The development of Unmanned Aerial Vehicles (UAVs) has been a recurrent subject in the literature due to their potential to service many diverse applications and operate in different environments. The design of flight control systems (FCSs) for these vehicles is an essential and challenging aspect of their development. In particular, the appropriate tuning of FCS parameters, such as force and torque bounds, is difficult to determine analytically due to the generally complex dynamics of UAVs. In this work, we propose the use of a meta-heuristic approach for the optimization of the control effort bounds of a quadcopter's FCS, focusing on the improvement of its performance considering a trajectory with sharp maneuvers. Additionally, we study the design and optimization of control laws with the saturation of the control signals for both the position and attitude controllers. Simulation results show that the FCS designed using the proposed approach outperforms manual tuning in terms of trajectory tracking performance.**

***Index Terms*—Meta-heuristic; PSO; Optimization; UAV; Flight Control System.**

I. Introduction

Recently, there has been an upward trend in popularity of Unmanned Aerial Vehicles (UAVs) in a wide range of civil or military applications. In terms of vehicle type, multirotor UAVs are strong candidates in the research community due to their low cost, size, and flying maneuverability in constrained conditions [1]. Despite significant advances, flight control system is still an important research topic for civilian applications, such as logistics, mapping, search and rescue scenarios [2], [3], and for military ones, such as defense and surveillance scenarios [4], [5]. The design of these systems requires the right type of UAV, navigation and control techniques. One special multirotor vehicle with advantage of plain construction and rotor mechanics is the quadrotor, which is the model that we will investigate in this study. Furthermore, the quadrotor is a non-linear and unstable system that must be stabilized by a flight control system, composed of position and attitude controllers for its translational and rotational movements, that vary the speeds and torques created by its four rotors [6].

The flight control system for UAVs often employ the Proportional-Integral-Derivative (PID) controller, which is a linear controller and has shown exceptional performance in many conditions, including in the context of drone racing [7], where accuracy and agility are key factors. The PID controller has three parameters, providing a simplified adjustment space. In this study, we are going to use a variation of this controller, by removing the integrative element, keeping the proportional element and substituting the derivative element by the vehicle's velocity, creating a controller often called P+V (proportional with velocity feedback) controller in the literature [8]. Fast response speed and high controllability can be achieved by implementing proper controller parameters making the method of tuning PID parameters one of the hot research topics in control engineering [1]. In low uncertainty environments, a PID controller performs satisfactorily, but when exposed to unknown dynamics or incorrect controller parameters, its performance may suffer and the controlled process may become unstable [9]. Note that the saturation of the control effort prevents plant instability, due to the rotors actuation limits.

This paper focuses on scenarios where it is essential for a UAV to precisely and rapidly follow a given path. In this context, we analyze how the tuning of proper control saturation limits can affect the controller performance when using analytically calculated gains. As there are no standard linear design methods for saturation limits, we analyze how we can use meta-heuristic optimization algorithms to perform this tuning, thus automating this process that is commonly tuned empirically. In this work, we analyze the position controller force limits and the attitude controller torque limits for a quadcopter model with 6 degrees of freedom (DoF). Note that the PID variation applied can only deal with linear effects, and the saturation limits do not generate non-linear effects.

Our main contribution is the proposal and evaluation of a method to improve the performance of a quadcopter while tracking specific trajectories by automating the tuning of the force and torque saturation using an optimization algorithm.

This paper is organized as follows. In Section II we review the modeling of a quadcopter. In Section III we present the design of the FCS. In the sequel, we define the problem, in Section IV. Later, we describe the experiments in Section V, its results in Section VI, and the discussion in Section VII. Lastly, Section VIII concludes and shares future works.

978-1-6654-6280-8/22/$31.00 ©2022 IEEE

II. Quadcopter Model

Consider a quadcopter (4-rotor UAV) with the coordinate system defined in Fig. 1, with mass m and inertia matrix $\mathbf{J}$, given by

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}. \quad (1)$$

where J_{ii} is the inertia moment along the axis i. Also consider that its propellers are equally spaced with a distance of l around its center of mass (CoM). Furthermore, $d = l\sqrt{2}/2$ is defined as the distance from the CoM to each propeller along the x or y axis.

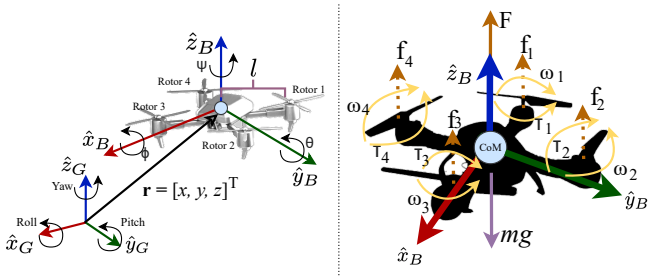

Fig. 1. On the left, the quadcopter coordinate system diagram, the global coordinate system denoted as $\hat{x}_G, \hat{y}_G, \hat{z}_G$, and the quadcopter-frame coordinate system represented by $\hat{x}_B, \hat{y}_B, \hat{z}_B$, with $\mathbf{r}$ being the position vector. On the right, the free body diagram of a quadcopter.

As seen in Fig. 1, we have the weight force mg, the forces f_1, f_2, f_3 and f_4, and the torques τ_1, τ_2, τ_3 and τ_4, generated by the rotors acting on the quadcopter frame. The resulting force $\mathbf{F}$ is the vector sum of all propeller's forces and F is its norm. The force and torque generated by each propeller are modeled respectively according to

$$f_i = k_f \omega_i^2, i \in 1, 2, 3, 4, \quad (2)$$

$$\tau_i = k_\tau \omega_i^2, i \in 1, 2, 3, 4, \quad (3)$$

where k_f and k_τ are force and torque constants, respectively. The conversion from the forces generated by the propellers to the forces and moments on the vehicle body is given by

$$\begin{bmatrix} F \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ d & -d & -d & d \\ -d & -d & d & d \\ k & -k & k & -k \end{bmatrix}}_{\Gamma} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \quad (4)$$

where $k = k_\tau / k_f$. For design, the following linearized dynamic equations at the equilibrium point $[\phi, \theta, \psi]^T = [0, 0, 0]^T$ and $[p, q, r]^T = [0, 0, 0]^T$ are:

$$m\ddot{x} = f_x, \quad (5)$$

$$m\ddot{y} = f_y, \quad (6)$$

$$m\ddot{z} = f_z - mg, \quad (7)$$

$$J_{xx}\ddot{\phi} = \tau_x, \quad (8)$$

$$J_{yy}\ddot{\theta} = \tau_y, \quad (9)$$

$$J_{zz}\ddot{\psi} = \tau_z, \quad (10)$$

where $[\ddot{x}, \ddot{y}, \ddot{z}]^T$ denote the acceleration vector w.r.t. the global coordinate system (GCS), and ϕ, θ and ψ are the roll, pitch, and yaw angles, respectively. Furthermore, $\mathbf{f}_G = [f_x, f_y, f_z]^T$ and $\boldsymbol{\tau}_G = [\tau_x, \tau_y, \tau_z]^T$ are vectors with forces and torques in the GCS.

III. Flight Control System

The following flight control system is implemented for the quadcopter, as shown in Fig. 2.

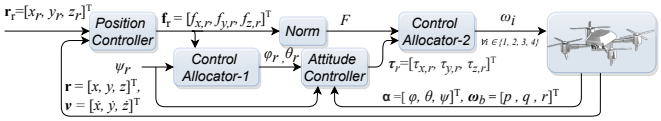

Fig. 2. Flight control system for the quadcopter.

The *position controller* receives as input a position reference $\mathbf{r}_r$, the current position $\mathbf{r}$ and the linear velocity $\mathbf{v}$ and outputs a force command $\mathbf{f}_r$. Through the inputs of $\mathbf{f}_r$ and ψ_r, *control allocator-1* computes the corresponding roll and pitch commands, which are then employed by the *attitude controller* to determine the torque commands, $\boldsymbol{\tau}_r = [\tau_{x,r}, \tau_{y,r}, \tau_{z,r}]^T$. The *norm* block outputs a scalar force F. Afterwards, the *control allocator-2* receives the commanded resulting force and torque commands, and outputs angular velocity commands for each rotor using (4) and (2). Finally, The *UAV block* implements the quadcopter dynamics and is used for simulation purposes, it includes the Euler angle representation of 6 DoF equations of motion using the MATLAB' 6 DoF Simulink block, for more information see [10]. This block outputs the position, linear velocities, attitude and angular velocities of the UAV.

A. Position Controller

The position controller (PC) is designed using the following P+V control laws:

$$f_x = K_{px}(x_r - x) - K_{vx}\dot{x}, f_x \in [f_{x,min}, f_{x,max}], \quad (11)$$

$$f_y = K_{py}(y_r - y) - K_{vy}\dot{y}, f_y \in [f_{y,min}, f_{y,max}], \quad (12)$$

$$f_z = K_{pz}(z_r - z) - K_{vz}\dot{z} + mg, f_z \in [f_{z,min}, f_{z,max}], \quad (13)$$

where K_{px}, K_{py}, and K_{pz} are the proportional gains, and K_{vx}, K_{vy}, and K_{vz} are the velocity gains.

B. Attitude Controller and Control Allocator-1

The attitude controller (AC) is also designed using P+V control laws:

$$\tau_{x,r} = K_{p\phi}(\phi_r - \phi) - K_{v\phi}p, \tau_{x,r} \in [\tau_{x,min}, \tau_{x,max}], \quad (14)$$

$$\tau_{y,r} = K_{p\theta}(\theta_r - \theta) - K_{v\theta}q, \tau_{y,r} \in [\tau_{y,min}, \tau_{y,max}], \quad (15)$$

where $K_{p\phi}$ and $K_{p\theta}$ are the proportional gains and $K_{v\phi}$ and $K_{v\theta}$ are velocity gains. With ϕ_r and θ_r calculated by

$$\phi_r = \text{asin}((F/f_{x,r})sin(\psi) - (F/f_{y,r})\cos(\psi)), \quad \phi_r \in [\phi_{r,min}, \phi_{r,max}], \quad (16)$$

$$\theta_r = \text{atan2}((F/f_{x,r})\cos(\psi + F/f_{y,r})\sin(\psi), F/f_{z,r}), \quad \theta_r \in [\theta_{r,min}, \theta_{r,max}]. \quad (17)$$

TABLE I
QUADCOPTER FLIGHT CONTROL SYSTEM REQUIREMENTS AND GAINS.

Axis	*Natural Frequency* ω_n	*Damping ratio* ξ	*Gains*	
x, y, z	$0.5Hz$	0.7	$Kp_{x,y,z}$	$m\omega_n^2$
			$Kv_{x,y,z}$	$2m\xi\omega_n$
ϕ, θ, ψ	$3Hz$	0.7	$Kp_{\phi,\theta,\psi}$	$J_{i,i}\omega_n^2$, $i \in x, y, z$
			$Kv_{\phi,\theta,\psi}$	$2J_{i,i}\xi\omega_n$, $i \in x, y, z$

The torque in z-axis is found using

$$\tau_{z,r} = K_{p\psi}\tilde{e_\psi} - K_{v\psi}r, \tau_{z,r} \in [\tau_{z,min}, \tau_{z,max}], \quad (18)$$

where $K_{p\psi}$ and $K_{v\psi}$ are proportional and velocity gains. The value of $\tilde{e_\psi}$ is calculated by

$$\tilde{e}_\psi = \text{atan2}(\sin(e_\psi), \cos(e_\psi)), \quad (19)$$

where $e_\psi = \psi_r - \psi$, and $\tilde{e}_\psi$ is in the range $(-\pi, \pi]$.

C. Control Allocator 2

Finally, we can find the forces to be applied to each rotor by

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \mathbf{\Gamma}^{-1} \begin{bmatrix} F \\ \tau_{x,r} \\ \tau_{y,r} \\ \tau_{z,r} \end{bmatrix}, f_1, f_2, f_3, f_4 > 0, \quad (20)$$

then, the angular velocity of each propeller is found as

$$\omega_i = \sqrt{f_i/k_f}, \omega_i \in [\omega_{min}, \omega_{max}], i \in 1, 2, 3, 4. \quad (21)$$

IV. CONTROL EFFORT BOUND OPTIMIZATION

The problem is the optimization of the saturation limits of the position and attitude controllers of a simulated quadcopter with 6 DoF. In this context, we determine by means of meta-heuristics using the PSO optimization algorithm, the force saturation limits of the position controller in (11)-(13), and the torque saturation limits of attitude controller in (14), (15), and (18). A proper tuning of these bounds must be determined to prevent aggressive control commands that may overcome physical limitations of the actuators, which can lead to an unstable plant. On the other hand, a conservative tuning may lead to inferior performance in trajectory tracking. The position and attitude controllers used are the P+V controllers with gains calculated analytically by traditional design techniques based on the control system requirements described in Tab. I.

A. Optimization Method

An efficient and widely used optimization algorithm in the literature for UAVs is the particle swarm optimization (PSO) algorithm, with studies to tune flight control PID gains for quadrotors [11], [12], and in UAV applications, such as parameter optimization for UAV swarms [4], [13]. PSO optimizes a problem by iteratively improving a population of candidate solutions, called particles, and moving these particles in the search-space with regard to a given measure of quality [14]. This method solves the optimization of continuous nonlinear functions and can search very large spaces of candidate solutions. Additionally, PSO is regarded as a meta-heuristic as it makes few or no assumptions about the problem. Also, PSO does not use the gradient of the problem, which means PSO does not require that the problem is differentiable as is required by classic optimization methods such as gradient descent. However, meta-heuristics does not guarantee a global optimal solution [14]. The implementation that we opted for is the one present in MATLAB's optimization toolbox [15].

B. Optimization Problem

In statistics, the mean squared error (MSE) of an estimator that measures average squared difference between estimated values and the actual values. The MSE formula is given by

$$MSE_Y = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_{i,r})^2, \quad (22)$$

where n is the number of points of a vector, and Y is the vector of observed values of the variable, with Y_r being the reference values. We use MSE to calculate the error to a desired reference signal, so a respective MSE will be calculated for the set of axes $\mathcal{S} \triangleq \{x, y, z, \psi\}$, as they are the ones with reference signals given. Additionally, a cost function J evaluates each iteration of the optimization process as

$$J = \sum_{\forall a \in \mathcal{S}} MSE_a, \quad (23)$$

where each MSE is calculated for the set of axes $\mathcal{S}$. Therefore, the optimization problem is the minimization of the cost function J.

V. EXPERIMENTS

In this section, we describe the performed experiments. The following values seen in Tab. II are used for the UAV model.

TABLE II
6 DOF DRONE DYNAMICS PARAMETERS.

Parameters	*Value*
m	0.063kg
J_{xx}	$0.5829 \times 10^{-4} kg.m^2$
J_{yy}	$0.7169 \times 10^{-4} kg.m^2$
J_{zz}	$1.0 \times 10^{-4} kg.m^2$
l	$0.624m$
d	$l\sqrt{2}/2$

Parameters	*Value*
g	$9.81 m/s^2$
k_f	$4.718 \times 10^{-8} Ns^2 rad^{-2}$
k_{tau}	$1.1389 \times 10^{-10} Nms^2 rad^{-2}$
ω_{min}	$0\, rad/s$
ω_{max}	$5.118 \times 10^3 rad/s$

The pre-optimization values of force and torque saturation limits are present in Tab. III, and are used as a baseline to compare the results of the experiments. These values were found empirically based on the authors' design experience.

TABLE III
SATURATION LIMITS PRE-OPTIMIZATION.

Limit	*Value*
$f_{x,y,min}$	$-0.5mg$
$f_{x,y,max}$	$0.5mg$
$f_{z,min}$	$0.1mg$
$f_{z,max}$	$2mg$
$\phi_{r,max}$	$30°$

Limit	*Value*
$\theta_{r,max}$	$30°$
$\tau_{x,y,min}$	$-4dmg$
$\tau_{x,y,max}$	$4dmg$
$\tau_{z,min}$	$-4kmg$
$\tau_{z,max}$	$4kmg$

The PSO algorithm is configured to optimize 10 variables using the parameters set in Tab.IV, further details about the parameters are found in the MATLAB documentation [15]. The variables to be optimized are the force and torque lower and upper bounds for each axis. Note that the limits for τ_x and τ_y are set equally, in order to simplify the search space.

TABLE IV
PSO PARAMETERS SET-UP AND VARIABLES BOUNDS.

PSO Parameter	*Value*
Swarm Size	20
Error tolerance	$1,00e^{-6}$
InertiaRange	[0.1,1.1]
InitialSwarmSpan	2000
MaxIterations	4000
MaxStallIterations	20
MaxTime	Inf
MinNeighborsFraction	0.25
SelfAdjustmentWeight	1.49
SocialAdjustmentWeight	1.49

	Bounds	
Limits	*Lower*	*Upper*
$f_{x,y,z,min}$	$-2mg$	0.00001
$f_{x,y,z,max}$	0.00001	$2mg$
$\tau_{x,min}$	$-2lmg$	0
$\tau_{z,min}$	$-2lmg$	0
$\tau_{x,max}$	0	$2lmg$
$\tau_{z,max}$	0	$2lmg$

A. Experiments description

In order to evaluate and to optimize the implemented controller, three experiments with increasing levels of difficulty were designed. The interested reader can find the implementation on Github. In all experiments, the quadcopter starts at $\mathbf{r} = [0,0,0]^T\ m$ and $[\phi,\theta,\psi]^T = [0,0,0]^T\ rad$. The experiments are as follows:

1) First Experiment: The UAV follows the Lissajous curve, presented in (24) in x–y while keeping constant altitude (z) for a duration of $t_f = 40\ s$. Initially, there is a $t_p = 2\ s$ setup time with $\mathbf{r}_r = [0,0,2]^T\ m$ and ψ_r interpolated linearly from 0 rad to the angle of the start of the Lissajous curve. Then, the quadcopter is commanded to follow the Lissajous curve with the expression

$$\begin{cases} x_r(t) &= 5\ cos\ \left(\frac{3}{t_f-2t_p}\ 2\pi(t-2)\ +\ \frac{\pi}{2}\right)\ m, \\ y_r(t) &= 5\ sin\ \left(\frac{2}{t_f-2t_p}\ 2\pi(t-2)\ \right) m, \end{cases} \tag{24}$$

up to $36s$. Finally, the UAV is commanded to $\mathbf{r}_r = [0,0,2]^T m$ and ψ_r is interpolated from the final angle of the Lissajous curve to 0 rad.

2) Second Experiment: The UAV continues to follow the Lissajous curve in x–y, but now varying the altitude z linearly and with a reduced duration time of $t_f = 25s$. Initially, there is a setup time of $t_p = 4s$ for the x–y axis and ψ_r is interpolated linearly from 0 rad to the angle of the start of the Lissajous curve. Then, the UAV is commanded to follow the Lissajous curve (24) up to 21 s. Meanwhile, the z–axis is commanded to ascend linearly to $5m$ for $t_f/2$. Finally, in the last $4s$ the UAV will descend linearly to $z_r = 1\ m$ with ψ_r being interpolated from the final angle of the curve to 0 rad.

3) Third Experiment: And lastly, this experiment is similar to the second experiment but with $t_f = 20\ s$ and $t_p = 3\ s$. Then, the UAV will follow the Lissajous curve (24) up to $17\,s$ while varying its altitude in a ramp up to $z_r = 4\ m$ to $t_f/2$. Finally, the UAV descends linearly to $z_r = 0\ m$.

https://github.com/luizgiacomossi/optimizationController6dof

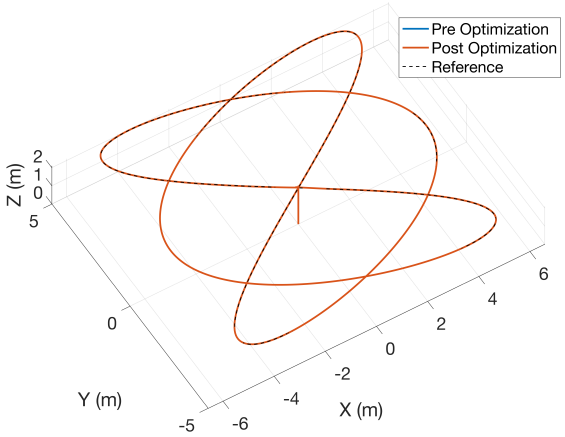

Fig. 3. Experiment 1 – Tracking in x-y-z space with both pre-optimization and post-optimization results. Reference as the black dashed line.

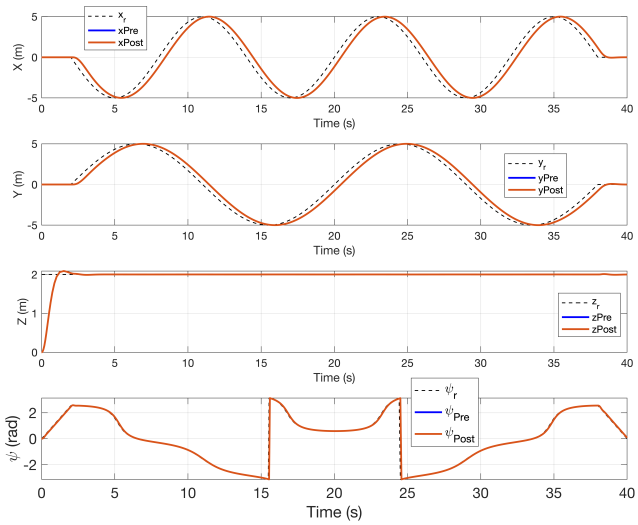

Fig. 4. Experiment 1 – Tracking of x, y, z and ψ in time with pre-optimization and post-optimization results. Reference as the black dashed line.

VI. RESULTS

In this section, we present the graphical and quantitative results of each experiment.

A. Graphical Results

The graphical results of the trajectory in the x-y-z space, as well as the reference command signals in time, of each axis, and its trajectory, pre-and-post optimization. The results of the Experiment 1 are seen in Figs. 3 and 4. The results of the Experiment 2 in Figs. 5 and 6. And finally, the Experiment 3 in Figs. 7 and 8.

B. Quantitative Results

The results of saturation limits for each experiment, after the convergence of the optimization can be seen in Tabs. V and VI. In both tables the pre-optimization values are used as baseline for the evaluation, calculated as shown in Tab. III. The tracking error results based on the MSE calculation for each axis, and the pre-optimization and post-optimization simulation costs are seen in Tab. VII.

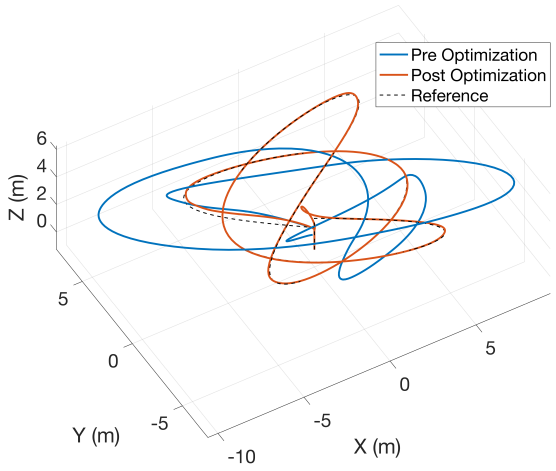

Fig. 5. Experiment 2 – Tracking in x-y-z space with both pre-optimization and post-optimization results. Reference as the black dashed line..

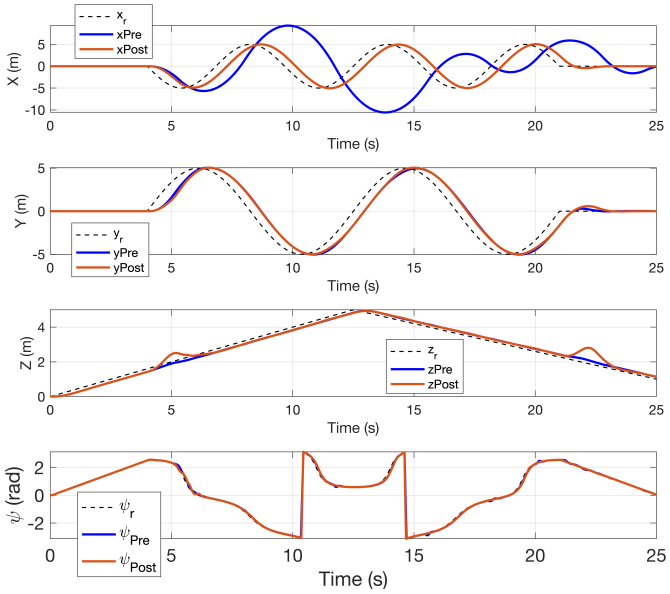

Fig. 6. Experiment 2 – Tracking of x, y, z and ψ in time with pre-optimization and post-optimization results. Reference as the black dashed line.

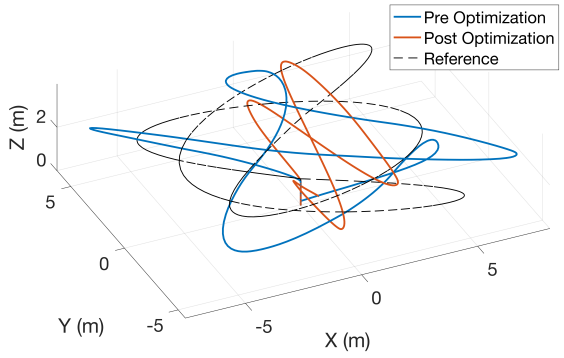

Fig. 7. Experiment 3 – Tracking in x-y-z space with both pre-optimization and post-optimization results. Reference as the black dashed line.

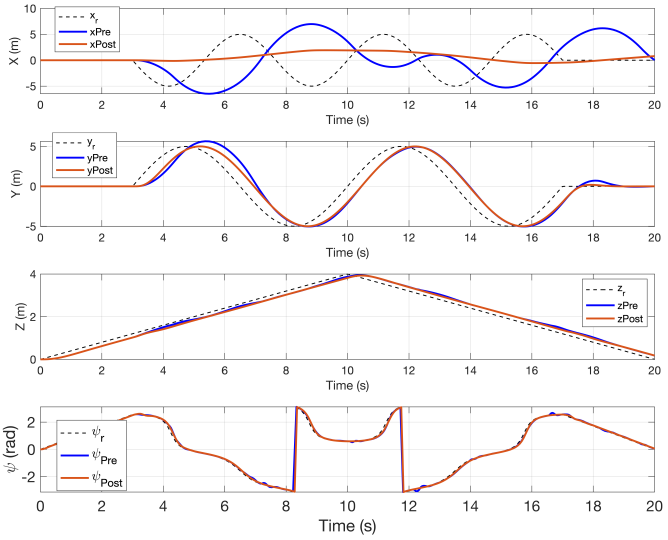

Fig. 8. Experiment 3 – Tracking of x, y, z and ψ in time with pre-optimization and post-optimization results. Reference as the black dashed line.

TABLE V
EXPERIMENTAL RESULTS - FORCE $[N]$ LIMITS AFTER OPTIMIZATION

Exp.	***Force Limit Results***					
Base	$f_{x,min}$	-0,3090	$f_{y,min}$	-0,3090	$f_{z,min}$	0.0618
	$f_{x,max}$	0,3090	$f_{y,max}$	0,3090	$f_{z,max}$	1.2361
1	$f_{x,min}$	-0,2441	$f_{y,min}$	-1,0565	$f_{z,min}$	-0,8279
	$f_{x,max}$	0,8140	$f_{y,max}$	0,8001	$f_{z,max}$	1,1752
2	$f_{x,min}$	-0,9095	$f_{y,min}$	-0,9575	$f_{z,min}$	-0,6849
	$f_{x,max}$	0,9910	$f_{y,max}$	0,3963	$f_{z,max}$	0,7082
3	$f_{x,min}$	0,0000	$f_{y,min}$	-0,3910	$f_{z,min}$	-1,2072
	$f_{x,max}$	0,0032	$f_{y,max}$	1,0908	$f_{z,max}$	1,2338

VII. DISCUSSION

In this section, the results of each experiment will be discussed. The Experiment 1 is the simplest and the optimization algorithm found efficient limits for both force and torque. Note from Fig. 3 that both pre-and-post optimization trajectories are almost coincident, this is an important result, as we observe that the optimization managed to solve the problem as well as the values used for comparison. Fig. 4 shows us that the reference signals were efficiently followed on all axes. Analyzing the quantitative results in Tab. VII, this experiment showed a 10.8 % total improvement in cost. This result is due to a considerable improvement in the MSE_ψ, with a small improvement in the tracking in x and y, and a small deterioration in the tracking in z.

Now, in Experiment 2 the difficulty of the trajectory is increased due to its shorter period of time, and the reference signal in z is no longer constant. Note by Fig. 5 that in the pre-optimization case, the UAV did not follow the reference trajectory, also seen in the x graph in Fig. 6. Then, after the optimization the UAV successfully performs the desired maneuvers. This result shows us the importance of adjusting the controller saturation limits. We can also see in Fig. 5 that after optimization, there was a slight worsening in z, with the presence of two peaks in the signal. The results in Tab. VII shows a considerable improvement of 90.3% by the reduction

TABLE VI
EXPERIMENTAL RESULTS - TORQUE $[Nm]$ LIMITS.

Exp.	***Torque Limit Results***			
Base	$\tau_{x,y,min}$:	-0.1091	$\tau_{z,min}$:	-0.0060
	$\tau_{x,y,max}$:	0.1091	$\tau_{z,max}$:	0.0060
1.	$\tau_{x,y,min}$:	-0,0271	$\tau_{z,min}$:	-0,0012
	$\tau_{x,y,max}$:	0,0273	$\tau_{z,max}$:	0,0601
2.	$\tau_{x,y,min}$:	-0,0362	$\tau_{z,min}$:	-0,0563
	$\tau_{x,y,max}$:	0,0042	$\tau_{z,max}$:	0,0442
3.	$\tau_{x,y,min}$:	-0,0316	$\tau_{z,min}$:	0,0011
	$\tau_{x,y,max}$:	0,0572	$\tau_{z,max}$:	0,0769

TABLE VII
EXPERIMENTAL RESULTS - MSE AND COST

		Evaluation Results - MSE and Cost				
Exp.		MSE_x	MSE_y	MSE_z	MSE_ψ	J
1.	*pre*	0.5897	0.2703	0.0873	0.1213	1,0685
	post	0.5850	0.2684	0.0972	0.0021	0.9526
2.	*pre*	30.7252	1,0452	0.0243	0.0075	31,8022
	post	2.0098	1.0190	0.0616	0.0097	3,1001
3.	*pre*	24.8761	2.2455	0.0318	0.0138	27,1672
	post	8.6567	1.3967	0.0299	0.2403	10.3237

of cost, due mainly to x tracking, previously not successfully achieved by the position controller, and for this improvement the meta-heuristics prioritized x, worsening mostly MSE_z.

Finally, the Experiment 3 is the most difficult with abrupt maneuvers at high speed. The results in both cases were not successful. The table VII show improvements in tracking for all axes with an overall improvement of approximately 62%. However, the ψ axis had a considerable performance deterioration. Notice the x, in which we see both in Tab. V and in Fig. 8, that the its actuation is almost nullified, this is the strategy adopted by the optimizer to achieve an improvement. This result shows the feasible limits of the meta-heuristic method, since it does not improve performance if the desired trajectory is too aggressive and probably unfeasible. In this case, the physical limits of the UAV must be analyzed and if the controller is adequate for aggressive maneuvers.

VIII. CONCLUSIONS AND FUTURE WORKS

In this work we applied meta-heuristics to optimize force and torque saturation limits for the FCS of a quadrotor, these limits are the position and attitude controllers outputs. We carried out three experiments using the PSO algorithm, varying the level of difficulty of the maneuvers. The results showed us that we can optimize the performance of flight controllers with meta-heuristics applied only to the saturations limits, which is difficult to design, in contrast to the controller gains that are often designed using traditional techniques. The reference tracking results showed a final cost improvement in all the cases, indicating an improvement in the controllers' performance. The second experiment stands out, with a considerable improvement of 90.3%. However, note that for tracking improvements, it was necessary to prioritize axes at the expense of others, as observed in the third experiment which demonstrated the limits of the meta-heuristics, that optimized the cost but did not solve the problem, showing that it would be necessary to design new controllers for aggressive maneuvers, or even change the physical components of the UAV. For future works, we suggest enhancing the dynamical model used for the UAV, adding sensor and actuator dynamics and also considering computational and discretization effects. We also suggest to evaluate experiments with different trajectories.

ACKNOWLEDGMENT

The authors acknowledge the Intel Center of Excellence in Artificial Intelligence for the funding support to complete this research. Luiz Giacomossi acknowledges Embraer S.A for his scholarship. Ângelo Caregnato-Neto was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] A. Sheta, M. Braik, D. R. Maddi, A. Mahdy, S. Aljahdali, and H. Turabieh, "Optimization of pid controller to stabilize quadcopter movements using meta-heuristic search algorithms," *Applied Sciences*, vol. 11, no. 14, 2021.

[2] L. Giacomossi, F. Souza, R. G. Cortes, H. M.M. Cortez, C. Ferreira, C. A. C. Marcondes, D. S. Loubach, E. F. Sbruzzi, F. A. N. Verri, J. C. Marques, L. A. Pereira, M. R. O. A. Maximo, and V. V. Curtis, "Autonomous and collective intelligence for uav swarm in target search scenario," in *2021 Latin American Robotics Symposium (LARS)*, 2021, pp. 72–77.

[3] H. D. Yoo and S. M. Chankov, "Drone-delivery using autonomous mobility: An innovative approach to future last-mile delivery problems," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2018, pp. 1216–1220.

[4] L. Giacomossi, S. Schwanz Dias, J. F. Brancalion, and M. R. O. A. Maximo, "Cooperative and decentralized decision-making for loyal wingman uavs," in *2021 Latin American Robotics Symposium (LARS)*, 2021, pp. 78–83.

[5] C. Humphreys, R. Cobb, D. Jacques, and J. Reeger, "Optimal mission path for the uninhabited loyal wingman," in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, p. 2792.

[6] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 153–158.

[7] H. Moon, J. Martinez-Carranza, T. Cieslewski, M. Faessler, D. Falanga, A. Simovic, D. Scaramuzza, S. Li, M. Ozo, C. Wagter, G. Croon, S. Hwang, S. Jung, H. Shim, H. Kim, M. Park, T.-C. Au, and s.-j. Kim, "Challenges and implemented technologies used in autonomous drone racing," *Intelligent Service Robotics*, vol. 12, 2019.

[8] K. Ogata, *Modern control engineering*. Prentice Hall, 2010, vol. 5.

[9] K. N. Maleki, K. Ashenayi, L. R. Hook, J. G. Fuller, and N. Hutchins, "A reliable system design for nondeterministic adaptive controllers in small uav autopilots," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–5.

[10] I. MathWorks, "Aerospace blockset user's guide," pp. 5. 80–88, 2022.

[11] Y. Wang, Y. Chenxie, J. Tan, C. Wang, Y. Wang, and Y. Zhang, "Fuzzy radial basis function neural network pid control system for a quadrotor uav based on particle swarm optimization," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 2580–2585.

[12] T. T. Mac, C. Copot, T. T. Duc, and R. De Keyser, "Ar.drone uav control parameters tuning based on particle swarm optimization algorithm," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2016, pp. 1–6.

[13] J. Cruz, G. Chen, D. Li, and X. Wang, "Particle swarm optimization for resource allocation in uav cooperative control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2012.

[14] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, 1997, pp. 303–308.

[15] I. MathWorks, "Global optimization toolbox user's guide," pp. 18.122–130, 2022.