Multi-Objective TEO and GOA Optimization of PID Controllers Applied to an Altitude and Yaw Mission of Quadrotor UAV Swarm

Bilel HERMOUCHE [(1)*], Youcef ZENNIR [(1)]
[(1)] Automatic Laboratory of Skikda, University of 20 August 1955, Skikda, Algeria
[*]b.hermouche@univ-skikda.dz; y.zennir@ univ-skikda.dz

Abstract: This paper produces a distributed control system to be used for collaborative UAV quadrotor simultaneous hovering and yawing missions in a leader-follower architecture, PID control is used in the altitude controller and the yaw controller for each single quadrotor with well-tuned parameters provided by two multi-objective optimization algorithms MOTEO and MOGOA, in the simulation and discussion section we show that due to the well system performances the produced parameters by MOTEO-PID combination are chosen to be the gains of the altitude and yaw controllers of the distributed control system for the hovering and yawing mission using a leader-follower architecture.

Keywords: Quadrotor, UAV swarm, Multi-objective Grasshopper optimization algorithm, Multi-Objective Thermal Exchange Optimization, Leader-follower architecture, PID controller.

1. INTRODUCTION

As the interest about quadrotors particularly and Unmanned Airial Vehicles UAVs in general is raising up day by day, and even the complexity of stabilizing a quadrotor due to its unstable nature, the under-actuated dynamics and the non-linearities that it have, many industrials have focused their interest of implementing these UAVs to benefit from the advantages they can provide starting from the vertical take-off and landing which is known as VTOL, the high maneuverability, precision to its hovering feature [1]. To overcome the difficulty of stabilizing quadrotors many works have tried different control methods like MPC and PID controller like in [1], LQR is being used in [8] where the parameters are optimized using grasshopper optimization algorithm and also it is used [2] in purpose of trajectory planning which is an interesting field where the work in [1] focused on tuning the control parameters by particle swarm optimization for trajectory planning as a single objective optimization problem, but due to the requirements of dynamic systems of many applications and the need of satisfying different and conflicting necessities we find in the literature work that tackle the subject of multi-objective optimization for trajectory planning like in [3], and in [4] and [5] for controller tuning, after solving the tuning problem there is many works that developed a distributed control systems for quadrotor swarms in a leader-follower like in [6, 7, 13].

The outline of this work is organized as follows: the proposed distributed control system for controlling a UAV quadrotor altitude and yaw is presented in the second section besides to the UAV swarm leader-follower architecture, then in section 3 an overview of the multi-objective optimization and a presentation of the meta-heuristic algorithms MOTEO and MOGOA where they are used in this work for optimization purpose are being given, the simulation results of the behavior of the altitude and yaw after the optimization using the multi-objective meta-heuristic algorithms for a single UAV quadrotor is shown in section 4, in addition to that and to choose who is the best algorithm to associate with the swarm of UAVs the section 5 is considered as a discussion section and it contains also the simulation results of a leader-follower swarm of two UAVs as a proof of concept, finally a conclusion part was there to sum up the obtained results of the paper.

2. QUADROTOR SWARM CONROL SYSTEM AND ARCHITECTURE

based on the mathematical modeling of the quadrotor dynamics of the X4-Flyer developed in [11, 12, 14, 15], and building on our previous work [16] that assumes the z-direction is upward which that mean it is positive.
In addition to our previous work [16], we try in this paper to control the altitude and yaw angle of the UAV quadrotor model, after that we try to put successfully pass a hovering and yawing mission of a swarm of two UAV quadrotor model by

developing a distributed control system for altitude and yaw.

Differently from Peter Corke's work in [11] and [12] we chose a parallel PID controller architecture for both altitude and yaw controllers without the use of a feedforward term like in [11,12] and the altitude controller of our previous work [16].

As we are dealing with a swarm mission of quadrotors, we define a distributed control system for both altitude and yaw.

The distributed control outputs of the ith quadrotor which are based on the altitude and yaw errors that are calculated as follows:

For the altitude error of the ith quadrotor, we have:

$$e_{z,i} = \int_0^T z_i^* - z_i \tag{1}$$

so, the distributed altitude control law of the of the ith quadrotor is as follows:

$$U_{z,i}(t) = -kp_z e_{z,i}(\mathrm{t}) - ki_z \int_0^T e_{z,i}(t)dt - kd_z\left(\frac{de_{z,i}}{dt}\right) \tag{2}$$

The yaw error of the ith quadrotor is:

$$e_{yaw,i} = \int_0^T \mathrm{yaw}_i^* - yaw_i \tag{3}$$

The yaw control law of the ith quadrotor is as follows:

$$U_{yaw,i}(t) = -kp_{yaw} e_{yaw,i}(\mathrm{t}) - ki_{yaw} \int_0^T e_{yaw,i}(t)dt - kd_{yaw}\left(\frac{de_{yaw,i}}{dt}\right) \tag{4}$$

Due to the identicality of the quadrotor agents of the swarm we kept the same PID gains of the altitude controller for all the ith quadrotor agents that are involved in the swarm, we did the same thing for the PID gains of the yaw controller.

Here is in figure 1 the control diagram of the ith quadrotor:

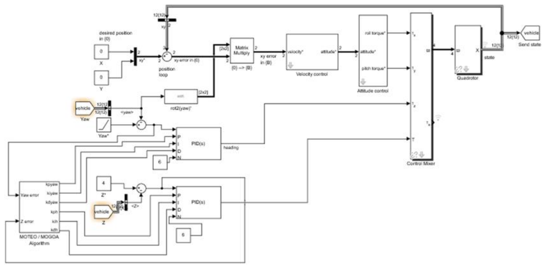

Fig. 1 Quadrotor control diagram

As our work is focusing only on altitude and yaw conrol of UAV quadrotors, we kept the same PD controllers for velocity and attitude and their gains as they are in the works of peter corke [11,12], the values are the following in table 1.

Table 1 Velocity and attitude PD gains

PID GAINS	Values
$Kp_{velocity}$	0.1
$Kd_{velocity}$	2
$Kp_{attitude}$	100
$Kd_{attitude}$	1

As a proof of concept, we chose a mission of two quadrotor UAVs to be the agents of our hovering and yawing mission with a leader and follower architecture where i = {1,2}, and here is the swarm configuration diagram shown in figure 2.

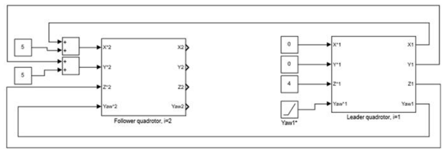

Fig. 2 Swarm of two UAVs in a leader-follower architecture.

As we see in figure 1, the leader UAV denoted as i=1 feeds the setpoint for the follower UAV denoted as I = 2 from its current states, we clearly remark that the x and y position input for the follower agent is shifted by 5 meters than the leader staring position.

3. OPTIMIZATION OF ALTITUDE AND YAW CONTROLLERS

In order to avoid the manual tuning methods, we suppose that using meta-heuristic algorithms can solve the problem of how well are the gains of the controllers? and produce fine-tuned PID coefficients.

In this work we have two controllers to search for their optimal gains, so the use of multi-objective optimization is highly recommended.

3.1 MULTI-OBJECTIVE OPTIMIZATION ALGORITHM FOR TUNING CONTROLLERS

Multi-objective optimization is a mathematical tool that is widely used by academics and engineers to solve computational life problems, unlike the single objective optimization, the multi-objective optimization method tries to give solutions for problems that are involving two or more objective functions which are in conflict with each other [18] [19], this optimization method does not produce a single solution that can satisfy all the requirements but it provides a set of solution that is called the pareto-optimal solutions [18] that follows the concept of pareto optimality where the improvement of one objective function can harm and worsen the others [19].

In literature [19], we find the usual steps to follow for a multi-objective controller tuning problem, where the first step is to study the plant features, in our case we have a quadrotor UAV system that we have to tune altitude and yaw controllers.

The second step is to select the architecture of the controllers where we have chosen a parallel PID controllers for both altitude and yaw.

So, in the third step we have to select the tunable parameters, the parallel PID gains are selected which are (Kp_{z}, Ki_{z}, Kd_{z}) for altitude controller and (Kp_{yaw}, Ki_{yaw}, Kd_{yaw}) for yaw controller.

Then in the fourth step the required objectives must be chosen, for our UAV quadrotor system we try to minimize two objective functions, so we formulate the multi-objective optimization problem as:

$$\min \mathbf{J} = (J_z, J_{yaw}) \tag{5}$$

The integral of the squared error of the altitude is the first objective function which is described as follows:

$$J_z = \int (e_z)^2 (t)dt \tag{6}$$

The second objective function is the integral of the squared error of the yaw angle and it is given:

$$J_{yaw} = \int (e_{yaw})^2 (t)dt \tag{7}$$

After the end of the optimization process, sets of the accepted solution is generated and the fifth step is to test all these provided sets of PID gains on the system.

The sixth and last step is to select the set thar satisfy more your requirements.

3.2 MULTI-OBJECTIVE THERMAL EXCHAGE OPTIMIZATION

Many engineering solutions over the time were inspired from nature and physical phenomena, one of these solutions is the Thermal exchange optimization algorithm (TEO) a meta-heuristic optimization algorithm which is first introduced by Kaveh and Dadras in 2017, this algorithm is inspired from the Newtonian cooling law where the environment and cooling objects are two groups of search agents, each cooling object has its own temperature and by associating it with an environment agent a heating transfer is happening and we consider the new temperature of the object as its new temperature [17].

In 2022, MOTEO the multi-objective version of the TEO algorithm is presented in [9] to handle multiple conflicting objectives.

3.3 MULTI-OBJECTIVE GRASSHOPER ALGORITHM

Inspired from nature the Grasshopper optimization algorithm (GOA) mimicked the swarming behavior of grasshoppers, this population based meta-heuristic algorithm was first introduced in 2017 [10], in the same paper, MOGOA the multi-objective version of GOA is presented and extended to deal with objective functions that conflict with each other [10].

4. SIMULATION

As a proof of concept, we use simulation to confirm our assumption that Multi-Objective Thermal Exchange Optimization and Multi-Objective Grasshopper Optimization Algorithm can easily satisfy the tuning problem of PID controller gains of an altitude and yaw controllers and that a single UAV can perform a stable hovering and yawing mission, then we use simulation later in the discussion section to prove the success of the best algorithm while applied on two UAVs with a leader and a follower architecture to perform a hovering and yawing mission.
First, we start by presenting the MOTEO obtained results, then we present the MOGOA obtained results, we mention that the simulation time for each the UAV mission is 10 seconds contrarily to the figures where we will show the altitude and yaw responses over 100 seconds of simulation flight time.

4.1 MOTEO-PID

To start the process of optimizing PID controllers gains for altitude and yaw we set the number of iterations to 20 iterations for the hole process, each iteration has a number of populations of 20 which gives us the opportunity of 400 trial to look for a better result. Table 2 presents the algorithm parameters used during the optimization process.

Table 2 MOTEO parameters

MOTEO	Values
Iterations	20
Population	20
dimension	6
objectives	2
Lower bond	0.5
Upper bond	100
c1	1.1
c2	c1×2
pCrossover	0.7
nCrossover	2*round(pCrossover×Population/2)
pMutation	0.4
nMutation	round (pMutation × Population)
mu	0.02
sigma	0.1× (**Upper** bond - Lower bond)

Here in table 3 are the 6 PID gains that were produced using MOTEO to minimize our two objective functions.

Table 3 PID gains obtained by MOTEO

PID GAINS	Values
Kp_z	85.01
Ki_z	4.58
Kd_z	58.62
kp_{yaw}	43.06
ki_{yaw}	43.40
kd_{yaw}	14.98

As we see in figure 3 and 4 the solution of altitude PID controller gains provided by MOTEO algorithm has allowed the quadrotor to achieve the altitude desired value with a stable way and an error converging to zero.

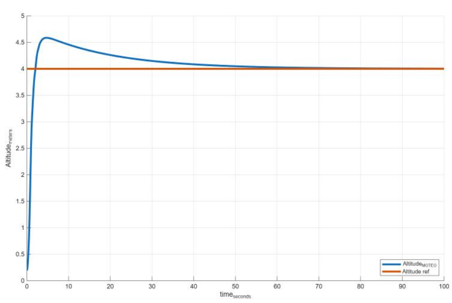

Fig. 3 UAV altitude response while using PID gains obtained by MOTEO.

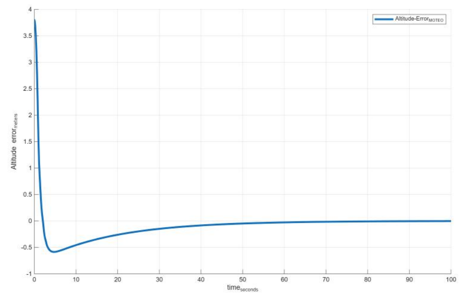

Fig. 4 UAV altitude error while using PID gains obtained by MOTEO.

Same as the altitude behavior of the quadrotor, the yawing movement were achieved using the yaw MOTEO-PID gains with a decreasing error over time, results are shown in figure 5 and 6.

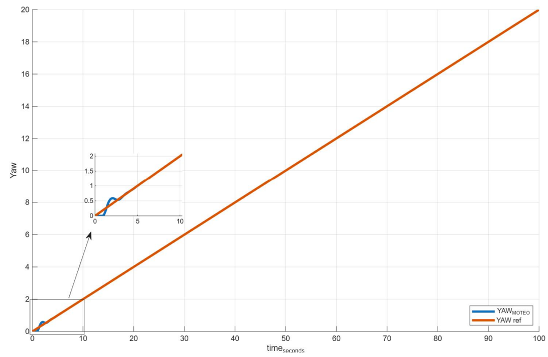

Fig. 5 UAV yaw response while using PID gains obtained by MOTEO.

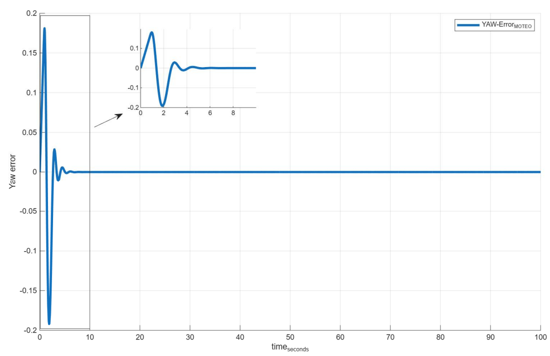

Fig. 6 UAV yaw error while using PID gains obtained by MOTEO.

Table 4 explains more the quadrotor altitude behavior where we can see a fast rise time but a slightly long settling time with an acceptable peak and overshoot percentage.

Table 4 System performances by MOTEO-PID altitude controller

System performance	Values
Settling time	40.46 s
Rise time	1.30 s
Overshoot	14.56 %
Peak	4.59 m
Peak time	4.73 s

4.2 MOGOA-PID

To ensure the equality of comparison between MOTEO and MOGOA table 5 show that we kept the same number of iterations, populations and the other common parameters between the two algorithms.

Table 5 MOGOA parameters

MOGOA	Values
Iterations	20
Population	20
Archive size	100
dimension	6
objectives	2
Lower bond	0.5
Upper bond	100
cMax	1
cMin	0.00004

After the end of the minimization process of the two objective functions, table 6 presents the 6 PID gains provided by MOGOA.

Table 6 PID gains obtained by MOGOA

PID GAINS	Values
Kp_z	7.36
Ki_z	4.38
Kd_z	23.38
kp_{yaw}	100
ki_{yaw}	2.09
kd_{yaw}	22.58

The obtained PID gains by MOGOA has permitted to the UAV to successfully complete the altitude and yaw mission, figures from 7 to fig. 10 show that the setpoints of the altitude and yaw were reached and the errors were converging to zero.

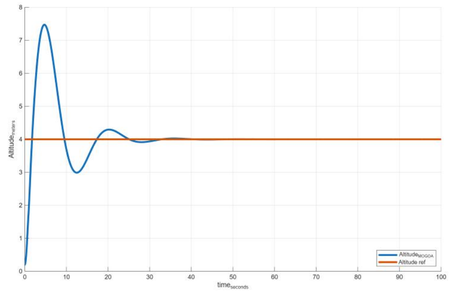

Fig. 7 UAV altitude response while using PID gains obtained by MOGOA.

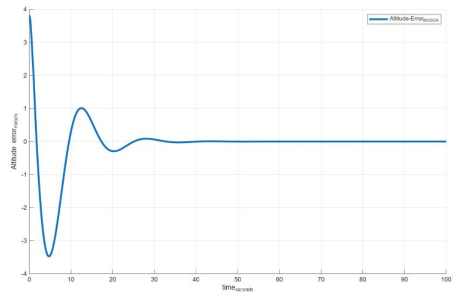

Fig. 8 UAV altitude error while using PID gains obtained by MOGOA.

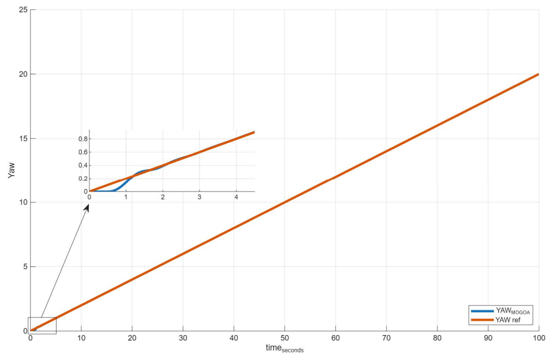

Fig. 9 UAV yaw response while using PID gains obtained by MOGOA.

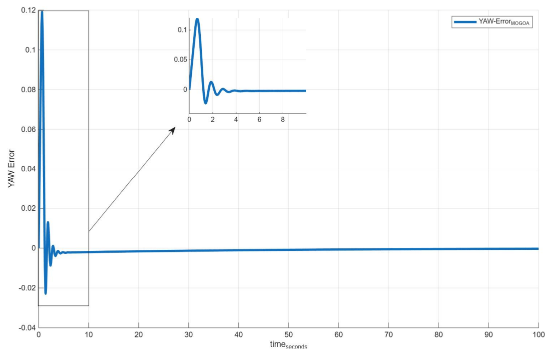

Fig. 10 UAV yaw error while using PID gains obtained by MOGOA.

Table 7 presents the system performances for 4 meters of altitude step response where the peak value and the overshoot percentage were remarkably bad for such a system that requires precision and safety even if the rise time was too good and the settling time was acceptable.

Table 7 System performances by MOGOA-PID altitude controller

System performance	Values
Settling time	28.96 s
Rise time	1.30 s
Overshoot	86.84 %
Peak	7.47 m
Peak time	4.71 s

5. DISCUSSION

The findings in the previous section of this paper showed and proved that both MOTEO and MOGOA are capable of handling the problem of minimization of two objective functions for a UAV system which are the integral of squared error of the altitude and the integral of squared error of yaw angle and successfully the UAV can perform a stable hovering mission on a specific height besides to a ramping yaw value.

Both MOTEO and MOGOA are not far from each other in term of the optimization process time, table 8 can show that MOGOA is a bit faster than MOTEO in producing an acceptable PÏD gains.

Table 8 The optimization time for each algorithm

Algorithm and controller	Optimization time
MOTEO-PID	10432.52 s
MOGOA-PID	9112.89 s

As UAV systems are very critical and prefer safety in the first place to not damage the materials and environment where it could fly, and the optimization times for the two algorithms are very close, that lead us to compare the system performances to choose which algorithm could be better to be used for the UAV swarm.

Regarding to table 4 and table 7 where system performances are shown for MOTEO-PID and MOGOA-PID respectively we can clearly see that the two altitude responses are slightly equal in terms of rise time and peak time, but MOGOA-PID had a much better settling time than MOTEO-PID, in another hand the peak value where the overshoot percentage is calculated from, is showing that MOTEO-PID is providing a better system performance than MOGOA-PID even if it is a little bit slower in the settling time, MOGOA-PID has an overshoot value 5 times bigger than the overshoot value of MOTEO-PID which is contradictory with safety first concept.

Starting from the last remark, we can confidently choose the PID gains provided by MOTEO to be fed in our distributed control especially with the leader-follower swarm navigation architecture where the follower UAV get its set point from the instantaneous position of the leader UAV, so with an 86.84 % overshoot value the follower UAV can diverge more from its desired way particularly with presence of disturbances.

Figures 11 and 12 show the altitude responses and errors of the swarm navigation on a leader-follower architecture where the overshoot value is smaller than the MOGOA-PID one.

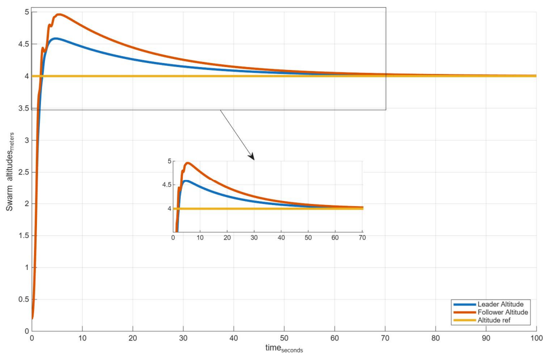

Fig. 11 UAV swarm altitude responses while using PID gains obtained by MOTEO.

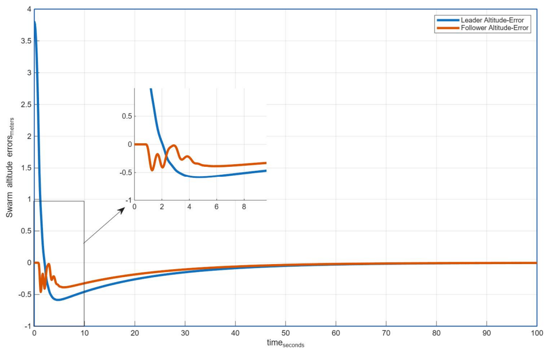

Fig. 12 UAV swarm altitude errors while using PID gains obtained by MOTEO.

We can see clearly in figures 13 and 14 that the swarm of UAV is responding to the request of a ramping yaw angle whether for the leader or the follower agent.

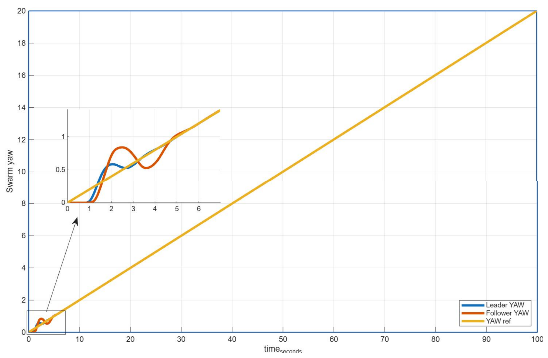

Fig. 13 UAV swarm yaw responses while using PID gains obtained by MOTEO.

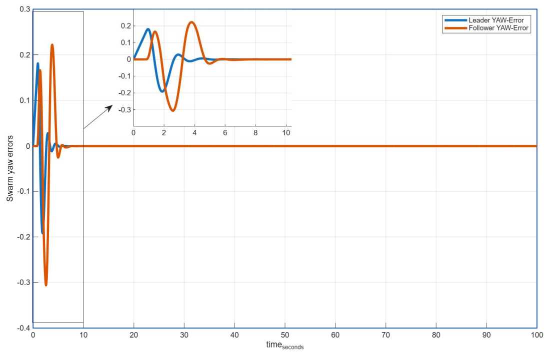

Fig. 14 UAV swarm yaw errors while using PID gains obtained by MOTEO.

6. CONCLUSION AND FUTUR WORK

To conclude, this work was taking in consideration the feasibility of applying the MOTEO-PID and MOGOA-PID controllers to control UAV quadrotors in a collaborative hovering and yawing mission with leader-follower architecture and results showed that:

- Both MOTEO and MOGOA have approved that they are capable of handling the simultaneous tunning process of the altitude and yaw PID controllers.
- MOTEO provided better system performance than MOGOA excluding settling time.
- MOGOA is faster than MOTEO in producing well optimized PID gains.
- The MOTEO and MOGOA can fit easily in the proposed distributed control system.
- The proposed PID controllers tuned by MOTEO have successfully arrived to stabilize the quadrotor swarm in a collaborative leader-follower hovering and yawing mission.

Finally, the obtained results lead us in future to develop our proposed distributed control system to tune all the quadrotor UAV controllers including position controllers (x, y) and attitude controllers (roll, pitch), besides to a validation application on other scenarios.

References

[1] Kapnopoulos, A., & Alexandridis, A. (2022). A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme. *Aerospace Science and Technology*, *127*, 107725.

[2] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., & Cooke, A. K. (2007, July). A prototype of an autonomous controller for a quadrotor UAV. In *2007 European Control Conference (ECC)* (pp. 4001-4008). IEEE.

[3] Ergezer, H. (2021). Multi-objective trajectory planning for slung-load quadrotor system. *IEEE Access*, *9*, 155003-155017.

[4] Gomez, V., Gomez, N., Rodas, J., Paiva, E., Saad, M., & Gregor, R. (2020). Pareto optimal pid tuning for px4-based unmanned aerial vehicles by using a multi-objective particle swarm optimization algorithm. *Aerospace*, *7*(6), 71.

[5] Vera, O., Gomez, V., Rodas, J., Kali, Y., Saad, M., & Comparatore, L. (2024, May). Multi-objective particle swarm optimisation for pareto optimal tuning of super-twisting controller in quadcopter. In *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)* (pp. 1-6). IEEE.

[6] Wang, Z., Zou, Y., Liu, Y., & Meng, Z. (2020). Distributed control algorithm for leader–follower formation tracking of multiple quadrotors: Theory and experiment. *Ieee/aSMe Transactions on Mechatronics*, *26*(2), 1095-1105.

[7] Ali, Q. (2017). *Distributed Control of Cooperating Mini UAVs* (Doctoral dissertation, Universität Würzburg).

[8] Dehnavi, V. S., & Shafiee, M. (2022, May). The design of fractional i-lqr controller for constrained quadrotor using grasshopper optimization algorithm. In *2022 30th International Conference on Electrical Engineering (ICEE)* (pp. 994-999). IEEE.

[9] Khodadadi, N., Talatahari, S., & Dadras Eslamlou, A. (2022). MOTEO: a novel multi-objective thermal exchange optimization algorithm for engineering problems. *Soft Computing*, *26*(14), 6659-6684.

[10] Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, *48*(4), 805-820.

[11] Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised* (Vol. 118). Springer.

[12] Corke PI, Robotics (2020) Vision and Control, viewed on 20 February 2020, retrieved from: https://petercorke.com/download/27/rtb/1045/rtb10-4-mltbx.mltbx

[13] Ali, Q., Gageik, N., & Montenegro, S. (2014). A review on distributed control of cooperating mini UAVs. *International Journal of Artificial Intelligence & Applications*, *5*(4), 1.

[14] Pounds, P. E. I. (2007). Design, construction and control of a large quadrotor micro air vehicle.

[15] Pounds, P., Mahony, R., & Corke, P. I. (2006). Modelling and control of a quad-rotor robot.

[16] Hermouche, B., Zennir, Y., & Kamsu Foguem, B. (2023). Influence of meta-heuristic algorithms on the optimization of quadrotor altitude PID

controller. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *45*(10), 504.

[17] Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in engineering software*, *110*, 69-84.

[18] Youssef, A. (2022). Artificial Corona Algorithm to Solve Multi-objective Programming Problems. *Authorea Preprints*.

[19] Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., & Aldape-Pérez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem. *Applied Soft Computing*, *93*, 106342.