



Real Time Optimal Tuning of Quadcopter Attitude Controller Using Particle Swarm Optimization

Musa Abdalla^{1*} & Salam Al-Baradie

The University of Jordan, Mechanical Engineering Department, Amman 11942, Jordan

*E-mail: musa_abdalla@mechatronix.us

Highlights:

- This work highlights a Quadcopter real-time PID controller tuning.
- Real-time application of the PSO search technique is presented.
- Effective blended control of fuzzy logic and conventional PID controllers is used.
- Experimental verification and validation of the method is presented.

Abstract. A real-time novel algorithm for proportional, integral and derivative (PID) controller tuning for quadcopters is introduced. The particle swarm optimization (PSO) method is utilized to search the quadcopter solution space to find the best PID controller parameters. A fuzzy logic (FL) controller is used to provide proper velocity reference signals to serve as tracking set points to be achieved by the PID controller. This nested loop design is proposed for stabilizing the quadcopter, where the fuzzy logic controller (FL) is used in the stable loop (i.e. outer loop) to control the desired angle, while the PID controller is used for the rate loop (i.e. inner loop). Finally, the optimum generated PID parameters were achieved in real time using the PSO search algorithm. The generated parameters were tested successfully using an experimental quadcopter setup at the University of Jordan.

Keywords: *fuzzy control; PID control; PSO tuning; quadcopter; PID tuning.*

1 Introduction

The popular unmanned aerial vehicles (UAV), self-propelled aerial vehicles that operate without the presence of a human pilot, are increasingly catching the attention of researchers and engineers. These vehicles are found in many important applications, including for military and surveillance purposes of great importance [1-4]. The control of the UAV's attitude, altitude and position poses many difficulties and challenges and it has been the subject of numerous studies [3,5-14]. Researchers and engineers are overwhelmed by the number of advanced algorithms that have been proposed to address the quadcopter control problem in particular.

The nonlinearity of the quadrotor behavior poses major obstacles in synthesizing effective controllers and control algorithms. Proportional, integral and derivative (PID) controllers are among the popular controllers that have been studied due to

Received December 20th, 2019, Revised April 23rd, 2020, Accepted for publication September 24th, 2020.

Copyright ©2020 Published by ITB Institute for Research and Community Services, ISSN: 2337-5779,

DOI: 10.5614/j.eng.technol.sci.2020.52.5.10

their simple structure, good performance, reliability and effectiveness when tuned properly [15-20].

A wide range of control systems have been introduced with a classic PID controller. The ubiquitous PID controller was successfully used for achieving attitude stabilization and horizontal position control in several studies [11,19-21]. Recent studies have revealed the proposal of a multi-loop (i.e. inner-loop and outer-loop) control architecture to control quadcopters for specific applications. The outer-loop controllers were designed based on a diversity of controllers while the inner-loop controllers were all implemented using a conventional PID control strategy [22-25].

This work proposes an optimal tuning method for the ubiquitous PID controller focusing on attitude stabilization using particle swarm optimization (PSO). PSO was used previously in the literature for controller tuning [26-29], where several authors proposed to perform tuning using simulation of a dynamical model and then implemented the results to a physical model of a quadcopter. However, the unmodeled quadcopter dynamics and the diversity of uncertainties in dimensions and physical parameters all combined prevented finding well-tuned parameters. In this work, to overcome these problems the tuning of the PID controller is performed online in a real-time environment using the particle swarm optimization (PSO) search algorithm.

This paper follows the following flow of material. Section 2 gives an overview of fuzzy PID controllers, while section 3 introduces the quadcopter controller design and section 4 discusses the theory of the fuzzy logic controllers (FL). Section 5 introduces the particle swarm optimization (PSO) and proposes its use for online FL-PID controller parameter tuning as a research methodology. Section 6 presents the experimental results and their discussion, which is used to verify and validate the online controller's tuning technique as applied to a quadcopter to demonstrate its effectiveness. Finally, section 7 highlights the main conclusions of this work.

2 Fuzzy PID Controller Overview

PID controllers have been used successfully in the industry for a long time owing to their simple structure and robust performance in a wide range of operating conditions. However, lately more demands on performance are placed on these controllers. Unfortunately, the PID controllers are linear in nature and they are ill suited for strongly nonlinear systems. This has motivated many researchers to complement the PID controller with FL based controller because FL is typically parameterized using rules and membership functions, which makes it easy to add nonlinearities, logic, and additional input signals to the controller's control law.

The PID controller currently takes many forms: direct action (DA), gain scheduling (GS) and hybrid types of FL-PID controllers.

For DA control, the FL controller replaces the PD controller part in the conventional PID structure, while the I-control action stays the same. However, this modification in the PID structure requires providing two measured inputs for the FL controller, as depicted in Figure 1. Several researchers have proven that the two PD systems are equivalent with the addition of providing more intelligence due to FL's heuristic nature and ability to handle both linear and nonlinear systems [30-32].

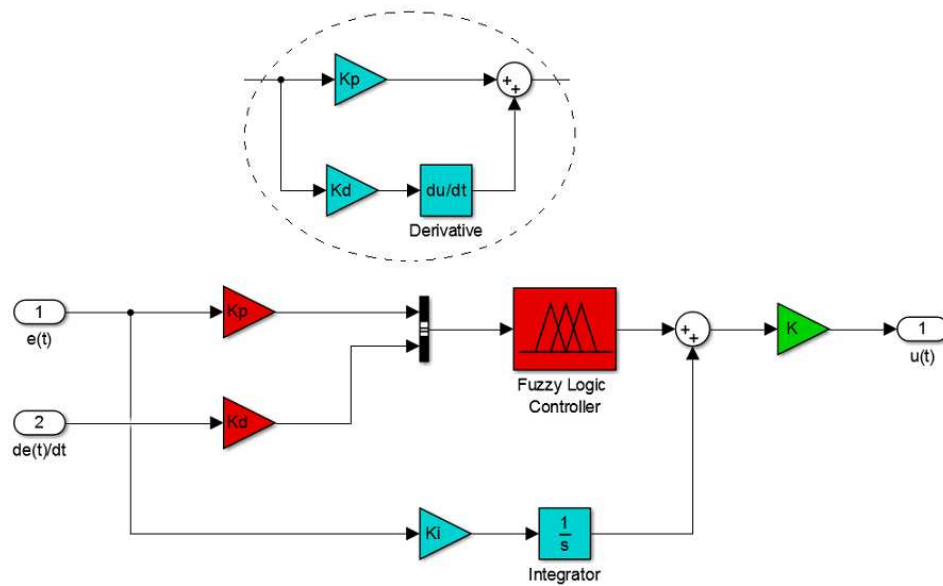


Figure 1 Direct action FL-PID controller implementation.

In GS controllers the FL part provides a dynamical mean to adjust the conventional PID controller's parameters in relation to the current dynamical system behavior. This adaptation turns out to be very useful in handling nonlinear systems with varying properties, characteristics, uncertainties, etc. Figure 2 illustrates the GS fuzzy based PID controller implementation scheme.

In hybrid FL-PID controller designs, the PID controller's effective working space is extended by using an FL based controller. However, both controllers utilize the error signal as input. Figure 3 illustrates the implementation of a hybrid FL-PID controller. An intelligent switching scheme that makes a decision on the priority of the controllers' contributions may be utilized for further enhancement.

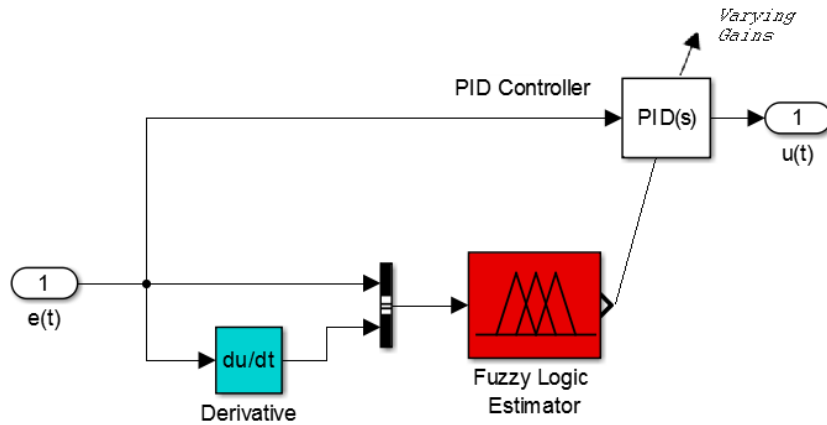


Figure 2 Gain scheduling FL-PID controller implementation.

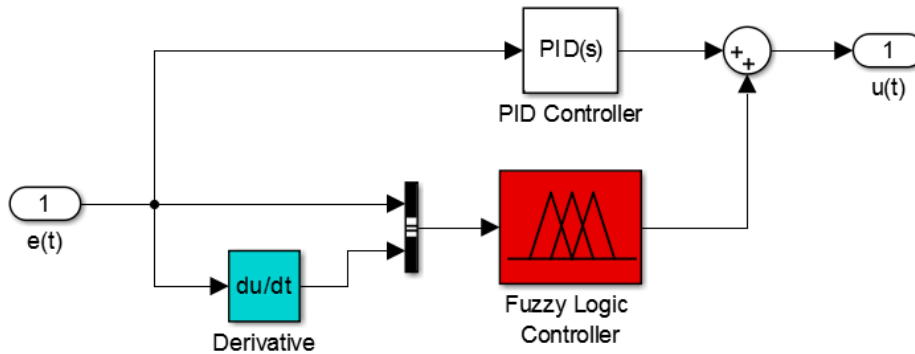


Figure 3 Hybrid FL-PID controller implementation.

3 Quadcopter Controller Design

Typically, the attitude controller of a quadcopter requires the controlling of angular velocities and the corresponding Euler angles (i.e. yaw, roll and pitch). The velocities and angles are measured using an accurate inertial measurement unit (IMU). An FL controller will be used to provide proper velocity reference signals to serve as tracking set points to be achieved by the PID controller. This nested loop design was used in this study for stabilizing the quadcopter, where an FL controller was used for the stable loop (i.e. outer loop) to control the desired angle and the PID controller was used for the rate loop (i.e. inner loop). Figure 4 depicts the nested control loops that are proposed to stabilize the altitude. The top of the diagram shows a box designating the PSO algorithm that is used online to dynamically adjust the PID parameters searching for optimal parameters.

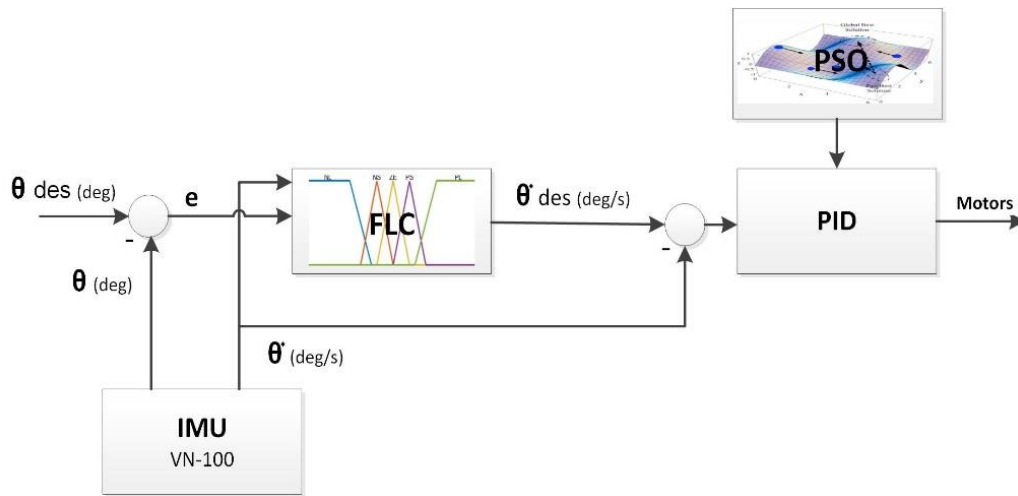


Figure 4 Controller strategy for attitude stabilization.

4 Fuzzy Logic Controller (FLC)

Since Zadeh first introduced fuzzy set theory and Mamdani applied it to replace operators in control many newly developed FL applications have emerged. All these applications may be categorized into two distinct fields: control and expert systems. Both fields apply inference and approximate reasoning using rule bases with the aid of membership functions over fuzzy logic sets $[0, 1]$ in contrast to classical logic sets $\{0, 1\}$. In control applications, the two most widely accepted fuzzy linguistic inference tools are the Mamdani and the Takagi-Sugeno methods. This work considered a Mamdani-based FL controller design methodology.

The FL controller provides its crisp output (defuzzified) based on crisp input (fuzzified internally) by utilizing the inference process via a rule base. Hence, the FL controller design involves designing three stages: fuzzification, rule-base, and defuzzification. Normally the FL controller design is sequential in the fuzzification stage; the membership functions are first selected and partitioned and then the rule base is constructed; finally, the defuzzification stage is implemented. Figure 5 summarizes the conventional top/bottom design approach and the main components of the FL controller.

The difficulty in designing a FL controller comes from having to capture and cover all the aspects of the system's dynamics, which is heavily dependent on the rule base and the selection of the membership functions (MF). The rule base should be constructed in such a way that the rule set spans the solution space.

On the other hand, the designer must choose the type of fuzzification (singleton or non-singleton), the number of membership functions, the functional forms of the membership functions (piecewise linear, Gaussian, sigmoidal), the parameters of the membership functions (fixed or tuned during a training procedure), the conjunction operator (t-norm, t-conorm), the implication or inference operator [33], the aggregation operator (t-norm, t-conorm) and the type of defuzzification (centroid, maxima, height). This demonstrates the richness and flexibility of fuzzy controllers, but also it reveals the need for some guidelines for their practical design.

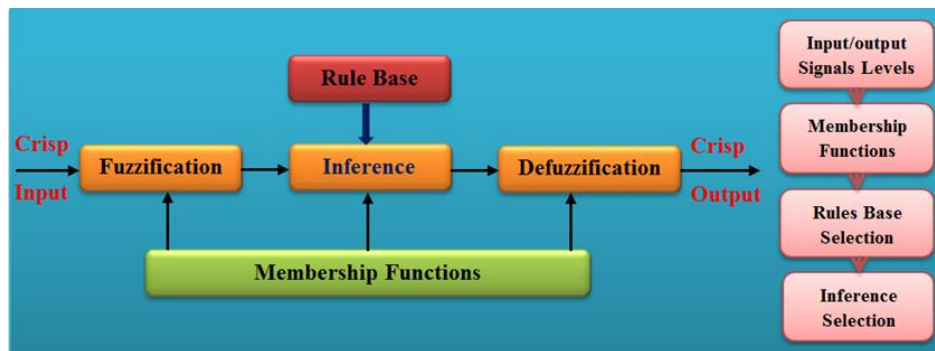


Figure 5 FL controller structure and conventional design methodology.

The FL controller regulates the dynamical system according to a collection set of rules (i.e. rule base) in the form of linguistic IF-THEN rules. For example: IF error (e_i) is big and the rate of change of error (de_i/dt) is small, THEN the controller output (u_i) is medium, where: e_i and de_i/dt are the inputs and u_i is the FL controller's rule fired output. Also, the big, medium and small are linguistic terms in the membership function (fuzzy subsets) of universe U of discourse. Traditionally, in the literature these sets of rules are presented in table form instead of as a cascaded IF-THEN statement.

Since the quadcopter model being used is symmetric in the x and y directions, the roll and pitch movements present the same dynamic. Therefore, the same controller is capable of controlling both angles. The error in the roll and pitch angle input is divided into five regions in the membership function: negative large (NL), negative small (NS), zero (Z), positive small (PS) and positive large (PL). Also, the rate of change is divided into five regions in the membership function: negative large (NL), negative small (NS), zero (Z), positive small (PS) and positive large (PL). Figure 6 shows the finalized input membership functions.

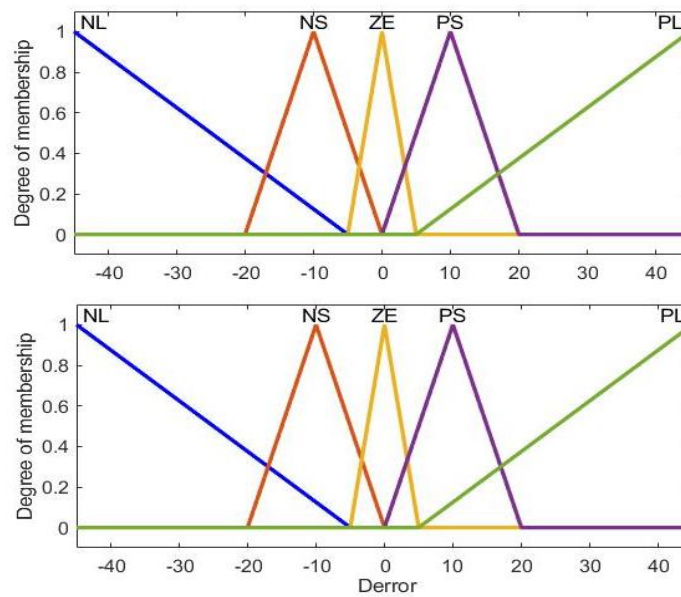


Figure 6 Input membership functions.

The controller output for the angle reference is divided into five membership functions: negative large (NL), negative small (NS), zero (Z), positive small (PS) and positive large (PL). Figure 7 shows the output membership function.

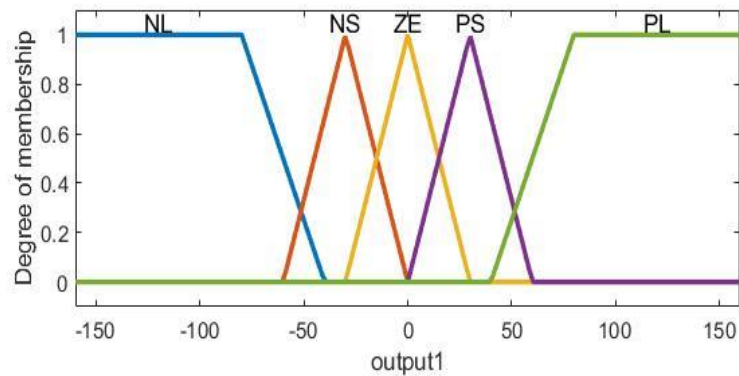
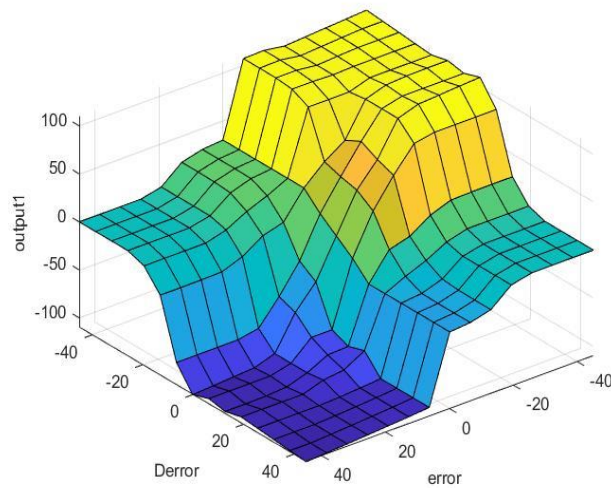


Figure 7 Output membership function.

Table 1 lists the used set of rules of knowledge base for the fuzzy inference system for the roll and pitch controllers, while Figure 8 depicts the FL generated control surface.

Table 1 Fuzzy Rules for Attitude Control

$\theta \backslash \dot{\theta}$	NL	NS	Z	PS	PL
NL	PL	PL	PS	NS	Z
NS	PL	PS	PS	Z	NS
Z	PL	PS	Z	NS	NL
PS	PS	Z	NS	NS	NL
PL	Z	NS	NS	NL	NL

**Figure 8** FL control surface.

5 PID Controller Tuning Using Particle Swarm Optimization

The literature shows that PID controllers have been applied successfully to quadrotors with some limitations [7,9,10]. However, the tuning of the PID controller may present challenges as it has to provide effective quadcopter dynamical behavior around the balance point. Several researchers have successfully used particle swarm optimization (PSO) based techniques to tune the PID controller parameters during the design stage or as an offline experimental technique; interested readers can consult the following recent references [25-27]. To achieve the desired performance in practice, an automated search technique that is based on PSO needs to be implemented for tuning the PID controller's

gains [34-39]. The PSO search technique was first introduced by Kennedy and Eberhart in 1995, which was based on fish school and bird flock behaviors.

Particle swarm optimization aims to find the minimum cost function in the search space according to a few simple dynamical formulas. A good metaphor for the PSO algorithm is to imagine a swarm of bees in a field searching for flowers. The objective of the swarm is to find the location with the most flowers.

Basically, each bee conducts a random search and memorizes the densest location of flowers it encountered. Also, by assuming that all the individual bees may share the information about their best findings of locations, each bee will be guided by its own personal discovery and by the best location reported by the others. Consequently, by altering the direction of their trajectory to fly somewhere between the two locations, the bees will explore the fields by overflying locations with the highest concentrations and eventually will be drawn to the densest flower location [40].

The following definitions and notes are needed to realize the PSO algorithm:

1. A number $n = 3$ of particles is used in PSO.
2. Every particle is identified by: current position P_i , velocity v_i and P_{i_Best} in the searched solution space.
3. All particles share their information about their best findings of locations.
4. The individual particle best position P_i corresponds to its minimum evaluated cost function $J(p_i)$ throughout its search journey.
5. The global best position is the best of all the best particle positions, P_{Global} .
6. The solution space is explored by driving the particles to move according to their best findings and also influenced by the best locations reported by the other particles. The mathematical representation of the swarm iterative updates for all particles, n , is given by the following equations:

$$P_{(i)Best}(t+1) = \begin{cases} P_{(i)Best}(t) & \text{if } J(P_{(i)Best}(t)) \leq J(P_{(i)Best}(t+1)) \\ P_{(i)Best}(t+1) & \text{if } J(P_{(i)Best}(t)) > J(P_{(i)Best}(t+1)) \end{cases} \quad (1)$$

$$P_{Global(t)} = \min \left(J(P_{(i)Best(t)}), J(P_{Global(t)}) \right) \quad (2)$$

Figure 9 illustrates how the particles are moving and sweeping the solution space intelligently and it also summarizes the used pseudo-algorithm. Each particle in the swarm tracks its position by means of two vectors one accelerates the particle in the direction of its own best and another towards the global best for the whole swarm (best of the best). The advancement of the particles in the solution space is controlled by a simple kinematic equation:

$$P_i(t+1) = P_i(t) + \Delta t \cdot v_i(t) \quad (3)$$

where Δt is the advancement in time (increment) and P_i is the current location of the particle (i.e. the solution). On the other hand, the particles' velocities are updated during the search with the following formula:

$$v_i(t+1) = \omega v_i(t) + c_1 \text{rand}() (P_{(i)Best}(t) - P_i(t)) + c_2 \text{rand}() (P_{(i)Global}(t) - P_i(t)) \quad (4)$$

where w is a scalar that generates some form of momentum for the particle taken from the previous iteration. The value that was used in this study was changed dynamically as well (in an adaptive fashion).

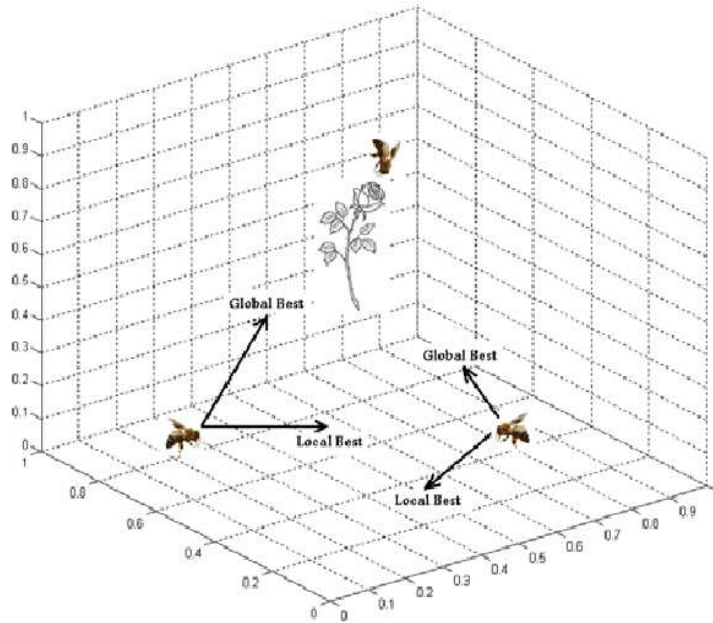


Figure 9 Particle movement in solution space.

The constants c_1 and c_2 represent the emphasis toward the particle's best or toward the global swarm's best, weighted by random terms. Typical values from the literature for these weights are within the interval $[0, 4]$. The local best is the best value of the solution attained by the particles and the global best is the best value of the solution attained by the whole swarm. The stochastic behavior of the PSO algorithm is realized by giving two random sequences.

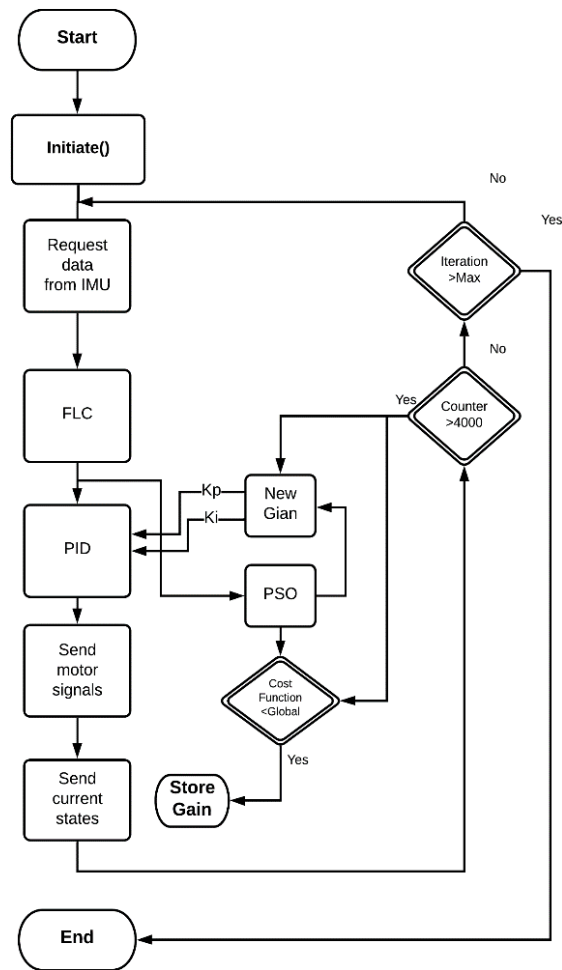


Figure 10 PID tuning process.

In this implementation, the objective is to find the minimum error in the fitness function, where the fitness of the particle is evaluated based on the system's step responses. It is estimated using the sum square error (SSE). This will also act as a stopping criterion for the algorithm in conjunction with the maximum number of iterations allowed, i.e. once the iteration reaches the set maximum value the quadcopter stops the calibration process and the best result obtained is stored for further processing. The complete controller's tuning process is summarized and depicted in Figure 10 as a flowchart. This algorithm will be executed in real time as mentioned above. This poses some challenges but the produced results are worth the effort of getting them.

6 Experimental Verification and Validation

An in-house built quadcopter and testing platform was used to verify and validate the effectiveness of the proposed real-time PSO algorithm for tuning the PID controller parameters. Figures 11 and 12 depict an image of the quadcopter and the corresponding block diagram of the main units. The control algorithms were implemented on an STM32F4 - ARM Cortex-M4 board, where the board used different methods and protocols to communicate with the remaining hardware units. Each quadcopter motor was powered and controlled by its own electronic speed controller (ESC) that provided a 3-phase alternating current. The ESCs keep each motor rotating at an rpm value determined by a pulse width they receive from the MCU board. An inertial magneto unit (IMU) equipped with magnetometers, gyroscopes and accelerometers was used to provide all the needed measurements (i.e. filter algorithms, bias estimation and magnetometer calibration) to operate the quadcopter. The IMU ran at a frequency of 400 Hz and the control loop at a rate of 250 Hz, which is the frequency of main loop.

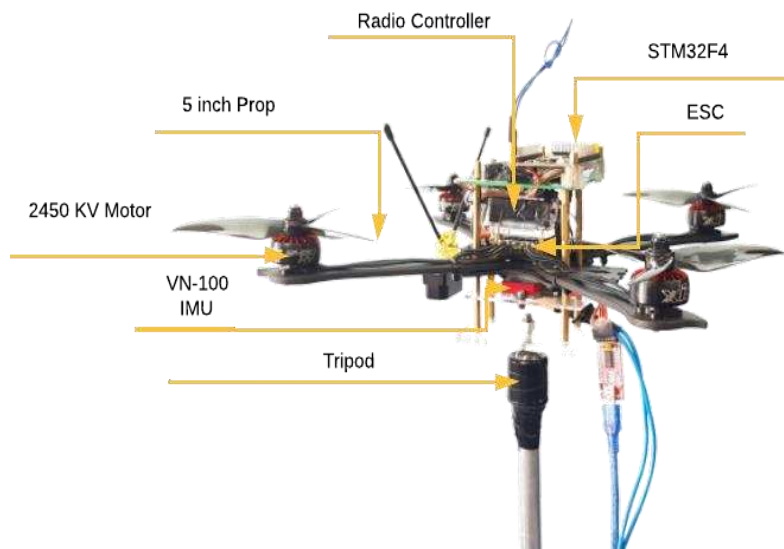


Figure 11 Quadcopter testing platform.

Initially, the parameters ϕ (roll), θ (pitch), and ψ (yaw) were all set to zero degrees. Since the model used is symmetric in the x and y directions, the same controller is capable of controlling both angles. The quadcopter is started with a

stock PID value and when the PSO is activated by the user, the initialization of the PSO will start to set the maximum iteration fixed to 25 with three swarm particles. After that the PSO starts updating the particle position and velocity every 16 seconds for pitch angle with a fixed desired angle (zero degrees). After the iteration reaches the maximum number of iterations, the stop criterion is satisfied and the best gain is stored. Performance is satisfied around a constant set point (zero degrees), however, the user can select another state to optimize it. After achieving the optimized gain, the user can select the optimization under a variable desired angle, where the user's desired value is changed every 1.5 seconds to find the optimal gains, which should improve the stabilization of the quadcopter.

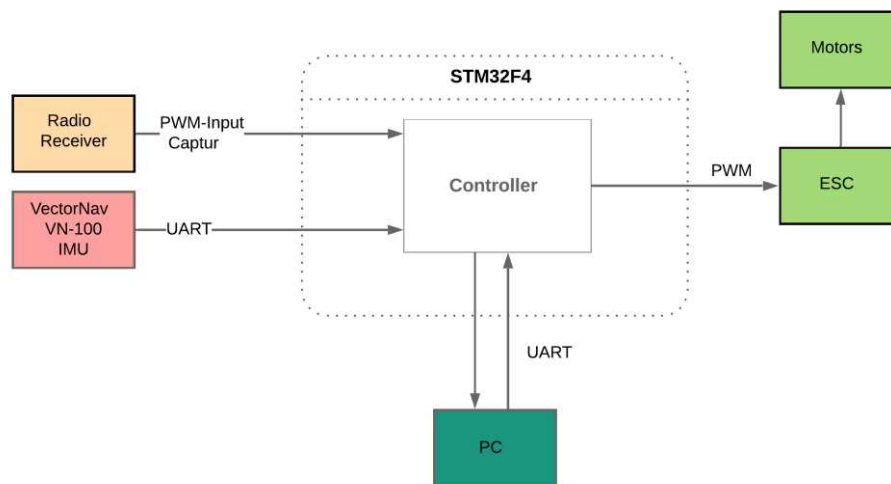


Figure 12 Quadcopter testing platform.

Figure 13 depicts the tuned PID controller parameters and the corresponding cost function. The quadcopter roll controller performance was evaluated for a period of 16 seconds. The initial controller gains were taken as ($k_p = 0.85$, $k_i = 0.04$, with fix $k_d = 40$). After releasing 25 swarms searching the solution space, the algorithm reached the optimal gain, marked with white points in the top area, while the initial controller gain in the bottom area was taken from the previous global best swarm and the integration gain was fixed. The (k_p and k_d) was optimized after reaching the 25th iteration; the optimized gain is marked with white points in the bottom area. The quadcopter yaw controller performance was also evaluated for a period of 16 seconds with the initial controller gain taken as ($k_p = 1.1$, $k_i = 0.07$). After 25 swarms the algorithm reached the optimal gain, marked by white points in the surface, as shown in Figure 14.

Finally, the quadrotor controller performance was evaluated for a period of 2.5 seconds while varying the desired angle. The initial controller (shown in blue in Figure 15) performed poorly but was stable. In contrast, the optimized controller (shown in red) showed an optimized, smooth, and fast response. The other controller parameters that were evaluated are shown in gray.

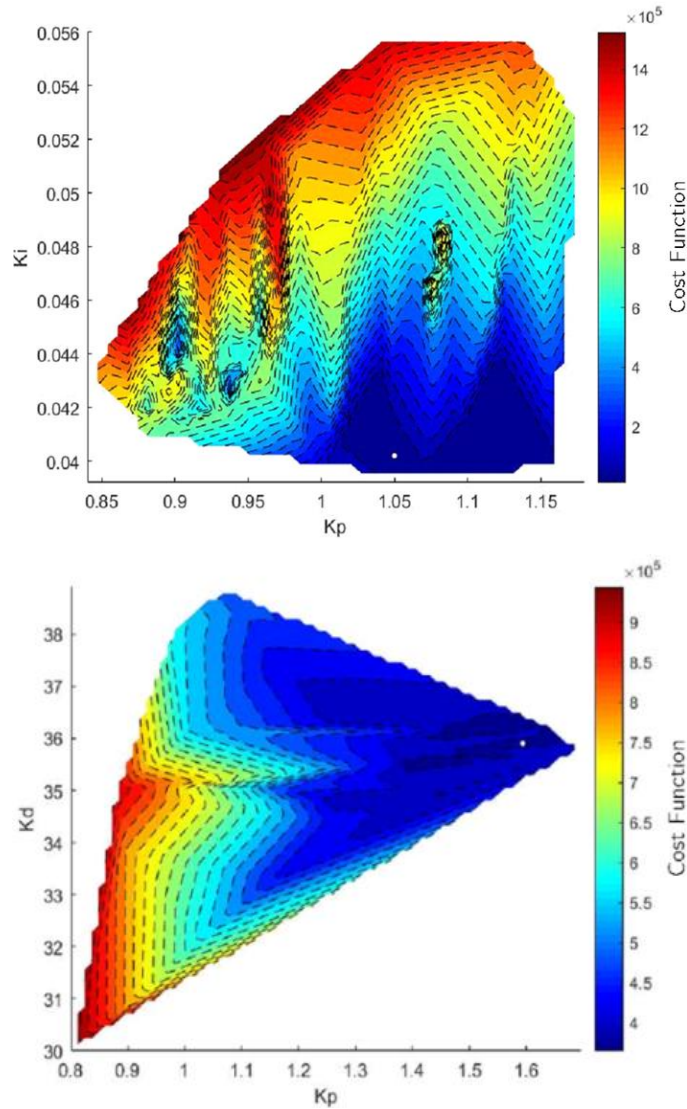


Figure 13 Quadcopter iterating tuning using real-time PSO.

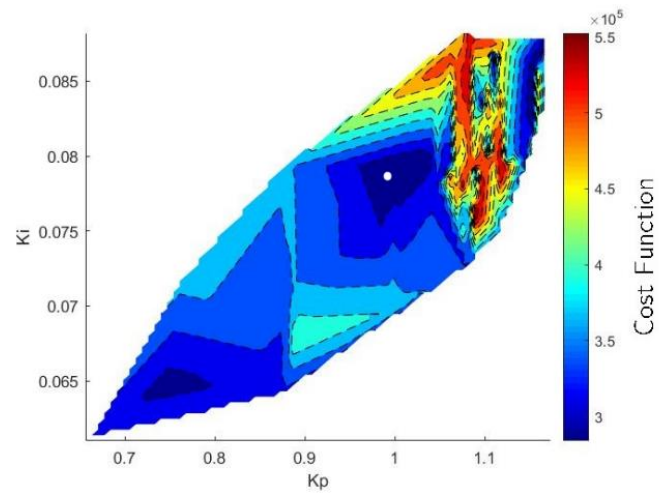


Figure 14 Quadcopter roll iterating tuning using real-time PSO.

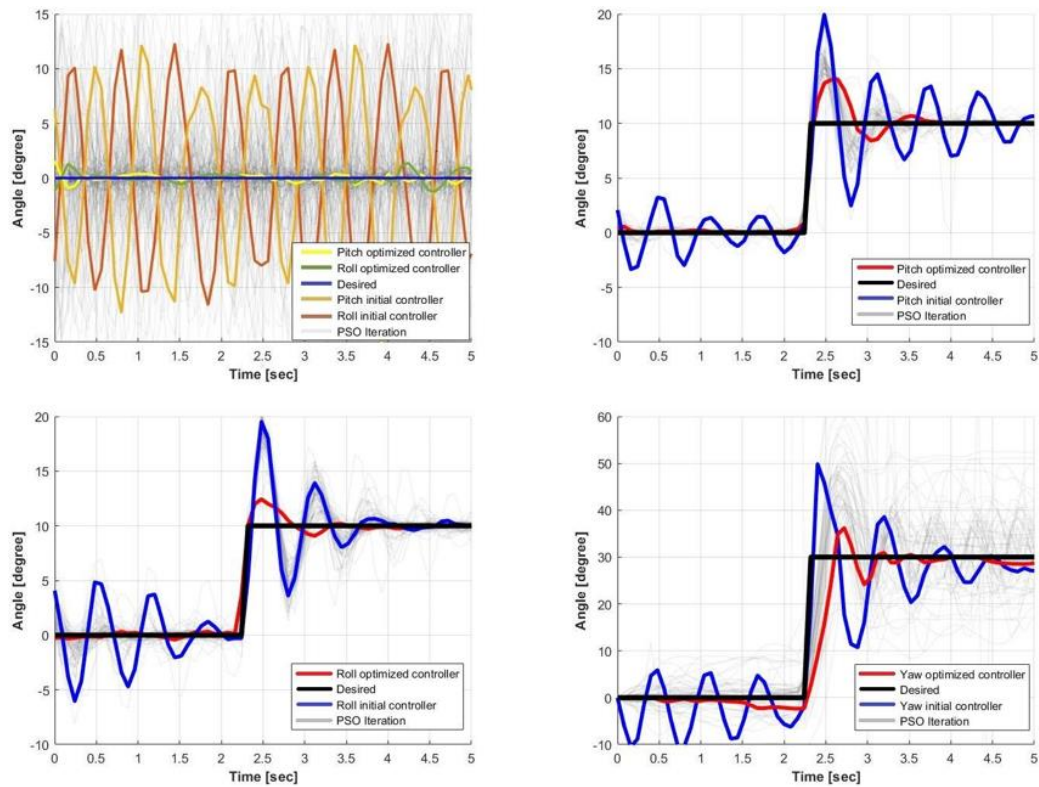


Figure 15 Quadcopter step responses for all swarms for pitch, yaw and roll.

Figure 16 depicts a series of user-defined steps and the corresponding quadcopter angle responses that resulted from using the optimized real-time tuning using PSO. The figure shows the roll angle command response of the quadcopter in carrying out the user inputs. Such good performance would not be possible without real-time tuning.

The figure also depicts the quadcopter roll angle response during a ten-second disturbance period generated by artificial wind produced by a fan. The curve illustrates how the quadcopter oscillated and represents the effectiveness of the tuned PID controller. The controller was able to compensate for the disturbance effect and preserved quadcopter stability.

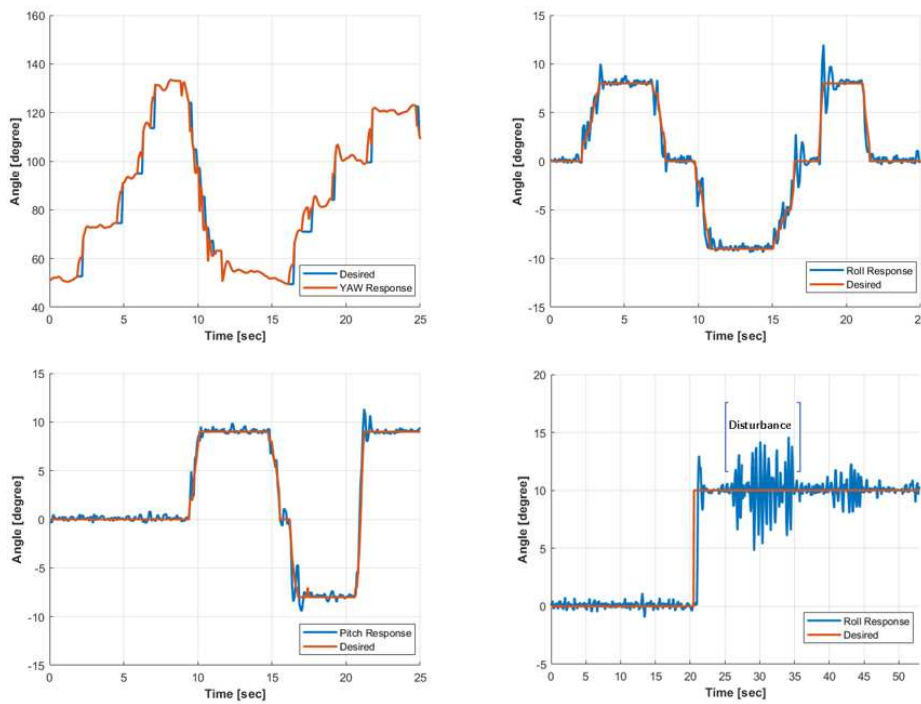


Figure 16 Roll, pitch and yaw angle step responses.

7 Conclusions

In this paper, a novel proportional, integral and derivative (PID) controller real-time tuning algorithm based on particle swarm optimization (PSO) search strategy was presented. The tuning strategy was tested on a laboratory experimental quadcopter platform with a fuzzy logic controller on board to

provide proper velocity reference signals to serve as tracking set points to be achieved by the PID controller.

Optimum PID working-parameters were successfully produced in real time by the PSO algorithm. The quadcopter response enhancement due to utilizing the new tuned parameters was clear. This technique seems to be insensitive to errors in the mathematics or modeling errors of the system in contrast to offline tuning strategies but is a little riskier.

References

- [1] Ahirwar, S., Swarnkar, R., Bhukya, S. & Namwade, G., *Application of Drone in Agriculture*, Int. J. Curr. Microbiol. App. Sci., **8**(1), pp. 2500-2505, 2019. DOI: 10.20546/ijcmas.2019.801.264.
- [2] Olov, A., Mariusz, W. & Patrick, D., *Deep Learning Quadcopter Control via Risk-aware Active Learning*, AAAI Conference on Artificial Intelligence, Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [3] Chen, M.Y., *Designing a Spatially Aware and Autonomous Quadcopter*, in 2013 IEEE Systems and Information Engineering Design Symposium, 26-26 April, pp. 213-218, 2013. DOI: 10.1109/SIEDS.2013.6549521.
- [4] Vergouw, B., Nagel, H., Bondt, G. & Custers, B., *Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments*, in *the Future of Drone Use: Opportunities and Threats from Ethical and Legal Perspectives*, B. Custers Ed. The Hague: T.M.C. Asser Press, pp. 21-45, 2016.
- [5] Zhang, B., Yoon, G. & Lim, H., *Attitude Control and Altitude Control of a Four-rotor Flying Robot*, in 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 28-30 July, pp. 1323-1328, 2018, DOI: 10.1109/FSKD.2018.8687282.
- [6] Wang, J., *Attitude Free Position Control of a Quadcopter using Dynamic Inversion*, in Infotech@Aerospace, Proceedings of the American Institute of Aeronautics and Astronautics, 2011.
- [7] Njinwoua, B.J. & Wouwer, A.V., *Cascade Attitude Control of a Quadcopter in Presence of Motor Asymmetry at University of Mons*, Mons, Belgium, IFAC-PapersOnLine, **51**(4), pp. 113-118, 2018/01/01/ 2018, DOI: 10.1016/j.ifacol.2018.06.055.
- [8] Nemati, A. & Kumar, M., *Modeling And Control of a Single Axis Tilting Quadcopter*, in 2014 American Control Conference, 4-6 June, pp. 3077-3082, 2014, DOI: 10.1109/ACC.2014.6859328.

- [9] Praveen V. & Pillai A., *Modeling and Simulation of Quadcopter Using PID Controller*, International Journal of Control Theory and Applications, **9**, pp. 7151-7158, 2016.
- [10] Cedro, L. & Wieczorkowski, K., *Optimizing PID Controller Gains to Model the Performance of a Quadcopter*, Transportation Research Procedia, **40**, pp. 156-169, 2019/01/01/ 2019, DOI: 10.1016/j.trpro.2019.07.026.
- [11] Ferdaus, M.M., Pratama, M., Anavatti, S.G., Garratt, M.A. & Lughofer, E., *PAC: A Novel Self-adaptive Neuro-fuzzy Controller for Micro Aerial Vehicles*, Information Sciences, **512**, pp. 481-505, 2020/02/01/ 2020. DOI: 10.1016/j.ins.2019.10.001.
- [12] Argentim, L.M., Rezende, W.C., Santos, P.E. & Aguiar, R.A., *PID, LQR and LQR-PID on a Quadcopter Platform*, in 2013 International Conference on Informatics, Electronics and Vision (ICIEV), 17-18 May, pp. 1-6, 2013. DOI: 10.1109/ICIEV.2013.6572698.
- [13] Olivares-Mendez, M.A., Mejias, L., Campoy, P. & Mellado-Bataller, I., *Quadcopter See and Avoid Using a Fuzzy Controller*, in *Uncertainty Modeling in Knowledge Engineering and Decision Making*, World Scientific Proceedings Series on Computer Engineering and Information Science, **7**, pp. 1239-1244, 2012.
- [14] Jiinec, T., *Stabilization and Control of Unmanned Quadcopter*, Master Thesis, Luleå University of Technology Department of Computer Science, Electrical and Space Engineering, 2011.
- [15] Pyrkin, A.A., Bobtsov, A.A., Kolyubin, S.A., Borisov, O.I. & Gromov, V.S., *Output Controller for Quadcopters Based on Mathematical Model Decomposition*, in 22nd Mediterranean Conference on Control and Automation, 16-19, pp. 1281-1286, 2014. DOI: 10.1109/MED.2014.6961552.
- [16] Ferdaus, M.M., Anavatti, S.G., Garratt, M.A. & Pratama, M., *Fuzzy Clustering Based Nonlinear System Identification and Controller Development of Pixhawk Based Quadcopter*, in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 4-6 Feb, pp. 223-230, 2017. DOI: 10.1109/ICACI.2017.7974513.
- [17] Andaluz, V.H., *Nonlinear Controller of Quadcopters for Agricultural Monitoring*, in *Advances in Visual Computing*, Bebis, G., Boyle, R., Parvin, B., Koracin, D., Ushizima, D., Chai, S., Sueda, S., Lin, X., Lu, A., Thalmann, D., Wang, C., Xu, P., Eds., 2015// 2015: Springer International Publishing, pp. 476-487, 2015.
- [18] Fatan, M., Sefidgari, B.L. & Barenji, A.V., *An Adaptive Neuro PID for Controlling the Altitude of Quadcopter Robot*, in 2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR), 26-29 Aug., pp. 662-665, 2013. DOI: 10.1109/MMAR.2013.6669989.

- [19] Gonzalez-Vazquez, S. & Moreno-Valenzuela, J., *A New Nonlinear PI/PID Controller for Quadrotor Posture Regulation*, in 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, IEEE, pp. 642-647, 2010.
- [20] Moreno-Valenzuela, J., Pérez-Alcocer, R., Guerrero-Medina, M. & Dzul, A., *Nonlinear PID-Type Controller for Quadrotor Trajectory Tracking*, IEEE/ASME Transactions on Mechatronics, **23**(5), pp. 2436-2447, 2018.
- [21] Sahul, M.P.V., Chander, V.N. & Kurian, T., *A Novel Method on Disturbance Rejection PID Controller for Quadcopter Based on Optimization Algorithm*, IFAC Proceedings Volumes, **47**(1), pp. 192-199, 2014/01/01/ 2014. DOI: 10.3182/20140313-3-IN-3024.00016.
- [22] Lukmana, M.A. & Nurhadi, H., *Preliminary Study on Unmanned Aerial Vehicle (UAV) Quadcopter Using PID Controller*, in 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), 15-17 Oct., pp. 34-37, 2015. DOI: 10.1109/ICAMIMIA.2015.7507997.
- [23] Paiva, E., Soto, J., Salinas, J. & Ipanaque, W., *Modeling, Simulation and Implementation of A Modified PID Controller for Stabilizing A Quadcopter*, in 2016 IEEE International Conference on Automatica (ICA-ACCA), 19-21 Oct., pp. 1-6, 2016. DOI: 10.1109/ICA-ACCA.2016.7778507.
- [24] Tesch, D.A., Eckhard, D. & Guarienti, W.C., *Pitch and Roll control of a Quadcopter using Cascade Iterative Feedback Tuning*, IFAC-PapersOnLine, **49**(30), pp. 30-35, 2016/01/01/ 2016. DOI: 10.1016/j.ifacol.2016.11.118.
- [25] Astudillo, A., Muñoz, P., Álvarez, F. & Rosero, E., *Altitude and Attitude Cascade Controller for A Smartphone-Based Quadcopter*, in 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 13-16 June 2017, pp. 1447-1454, 2017. DOI: 10.1109/ICUAS.2017.7991400.
- [26] Santos, M.F., *Simulation and Comparison between a Linear and Nonlinear Technique Applied to Altitude Control in Quadcopters*, in 2017 18th International Carpathian Control Conference (ICCC), 28-31 May, pp. 234-239, 2017. DOI: 10.1109/CarpathianCC.2017.7970403.
- [27] Paiva, E.A., Soto, J.C., Salinas, J.A. & Ipanaque, W., *Modeling And PID Cascade Control of A Quadcopter for Trajectory Tracking*, in 2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 28-30 Oct., pp. 809-815, 2015. DOI: 10.1109/Chilecon.2015.7404665.
- [28] Khodja, M., Tadjine, M., Boucherit, M. & Benzaoui, M., *Tuning PID Attitude Stabilization of A Quadrotor Using Particle Swarm Optimization (Experimental)*, Int. J. Simul. Multisci. Des. Optim., **8**, A8, 2017.
- [29] Abbas, N. & Sami, A., *Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm – Particle*

- Swarm Optimization*, International Journal of Computer Applications **172**(4), pp. 9-18, August 2017.
- [30] Astrom, K.J., The Future of PID Control, Control Engineering Practice, **9**(11), pp. 1163-1175, 2000.
 - [31] Mann, G.K.I., Hu, B. G. & Gosine, R.G., *Two-level Tuning of FL-PID Controllers*, IEEE Trans. on System, Man, and Cybernetics, **31**(2), pp. 263-269, 2001.
 - [32] Carvajal J., Chen G. & Ogmen H., *FL-PID Design, Performance, and Stability*, Info Science, **123**(3), pp. 249-270, 2000.
 - [33] Fodor, J, *Left-continuous T-norms in Fuzzy Logic: An Overview*, Acta Polytechnica Hungarica, **1**(2), pp. 35-47, 2004.
 - [34] El Gmili, N., Mjahed, M., El Kari, A. & Ayad, H., *Particle Swarm Optimization Based Proportional-Derivative Parameters for Unmanned Tilt-Rotor Flight Control and Trajectory Tracking*, Automatika, **61**(2), pp. 189-206, 2020.
 - [35] Noordin, A., Basri, 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. DOI: 10.18517/ijaseit.7.4.3141.
 - [36] Tran, H.K. & Chiou, J.S., *PSO-Based Algorithm Applied to Quadcopter Micro Air Vehicle Controller Design*, (in eng), Micromachines (Basel), **7**(9), pp. 168, 2016. DOI: 10.3390/mi7090168.
 - [37] Mac, T.T., Copot, C., Duc, T.T. & Keyser, R.D., *AR.Drone UAV Control Parameters Tuning Based on Particle Swarm Optimization Algorithm*, in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 19-21 May, pp. 1-6, 2016. DOI: 10.1109/AQTR.2016.7501380.
 - [38] Robinson, J. & Rahmat-Samii, Y., *Particle Swarm Optimization in Electromagnetics*, IEEE Transactions on Antennas and Propagation, **52**(2), pp. 397-407, 2004. DOI: 10.1109/tap.2004.823969.
 - [39] Lim, S.Y., Montakhab, M. & Nouri, H., *A Constriction Factor Based Particle Swarm Optimization for Economic Dispatch*, paper presented at The European Simulation and Modelling Conference, October 2009.
 - [40] Abdalla, M., Al-Jarah, T., *Optimal Fuzzy Controller: Rule Base Optimized Generation*, Control Engineering and Applied Informatics **20**(1), pp. 76-86, 2018.