Received: October 18, 2020. Revised: December 20, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

The Use of an Optimal Fuzzy Controller Algorithm for a Low-cost Microcontroller

Mohammed Abdallah Khodja[1,2]* **Sami Mahfoudhi**[3]

[1]Department of Electronics, College of Engineering, Mustaqbal University, Buraidah, 51452, Saudi Arabia
[2]Department of Electrical Engineering, Faculty of Technology, M'sila University, Saudi Arabia
[3]Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University P.O.B. 6677, Buraidah, 51452, Saudi Arabia
* Corresponding author's Email: medkhodja28@gmail.com

Abstract: Fuzzy gain-scheduled PID (FuGSPID) controllers have attracted significant interest in contemporary research. This paper provides mathematical description help to reduce the code program for a Fuzzy gain-scheduled controller FuGSPID that is subsequently used to stabilize the altitude of a home-constructed Quadcopter. The subject particle swarm optimization approach was employed to optimize the fuzzy output sets of the controller. MATLAB was used to generate the dynamic model, the FuGSPID, and the optimization. A simulation exercise revealed that the new parameters employed in the FuGSPID that were generated as a result of the particle swarm optimization produced fewer trajectory tracking errors. Fuzzy systems are required to process large volumes of tasks and processes. This subsequently impedes the performance of the microcontroller. In light of this, this paper outlines a mathematical description that can potentially reduce the algorithm code employed in the FuGSPID controller and, thereby, making it more intuitive and reducing the processing speed of the microcontroller and reducing the sampling time of the controller and the whole flight controller. The findings of this study revealed that the mathematical description it be useful to reduce instruction of fuzzy controller program to implement it in low cost microcontroller and tested effectively for a quadcopter altitude stabilization.

Keywords: Quadcopter dynamic, Fuzzy control, Particle swarm optimization, Microcontroller.

1. Introduction

In more recent times, scientists have focused on the development of an effective control algorithm that can be used with unmanned aerial vehicles (UAVs) as a means of improving their performance in civil and industrial applications. UAVs engage in two types of flying: horizontal and vertical. For the purposes of the current study, vertical flying was selected on the basis that it can hold a steady position at an altitude and offers hovering stabilization.

PID (proportional integral derivative) controllers play a fundamental role in control applications and theories. They have attracted the attention of many researchers because they have real-world applications in a range of fields spanning both civil and industrial applications. The primary issue with PIDs is that they are not particularly suitable for nonlinear systems. Furthermore, their performance cannot be guaranteed in an evolving environment. For the purposes of the study outlined in this paper, a subject was selected and transformed into a hybrid in combination with a PID controller. This makes it possible to track a required reference under a variety of environmental conditions. The fuzzy logic artificial intelligence form was employed in this study to assist us to develop a FuGSPID controller.

The configuration for the proposed controller was developed by applying Particle Swarm Optimization of FuGSPID with the objective of controlling altitude stabilization. The self-constructed quadcopter that was used in previous papers was used in the current study. The steps outlining the experimental setup are outlined in [1, 2]. The plan of work was divided into two modes: simulation mode and experiment mode.

DOI: 10.22266/ijies2021.1031.26

The simulation mode was tested in the MATLAB environment, and executed through a dynamic model, which is described in [1, 2]. The use of FuGSPID to maintain the altitude of the quadcopter was discussed in previous work, and the configurations are outlined in [3-5]. These configurations were programmed in Simulink after the simulation revealed positive results. The main part of this paper describes the Particle Swarm Optimization algorithm that was added to the simulation project to optimize the subject and subsequently identify new parameter values for FuGSPID. The index that will be used for the purposes of the comparative criteria will be formed of an integral of the square error between the achieved altitude and the desired altitude of a quadcopter. The second stage will involve the real-time implementation of the parameters that were obtained during the first phase to validate the performance of these parameters. This stage of the research will assist the generation of a mathematical model that will subsequently be used to implement the FuGSPID code such that it is efficient. The code will be implemented on Teensy 3.2 to stabilize the quadcopter, which was validated in prior studies [1, 2, 6].

Particle Swarm Optimization (PSO), which was first introduced by Kennedy and Eberhart in 1995[8], is a population-based approach that is used to identify an optimal solution for an objective function [7]. The swarm aspect of the model is formed of “n” particles, with the position of each particle representing a feasible solution in space. The conditions of the particles vary according to three control objectives: inertia, the position of the particles in terms of the optimal position, and control of the particle velocity. The optimal position influences the position of each particle in the swarm. This optimal position is derived from the particle’s individual experience and its global best position in terms of its neighborhood. The PSO is underpinned by multiple algorithms [9, 10]. In the current study, the integral of the square error (ISE) represents the fitness function (objective function). This, in addition to the constriction coefficient [7, 11] method has been used in prior papers [12, 1] and has generated positive results.

The study will aim to enhance the performance of FuGSPID controllers further to achieve advanced control within the system of interest. In [13], moving target-tracking will be performed for a Quadcopter using two different approaches, the performance of each of which will subsequently be compared. The first approach will involve linearly adjusting the proportional gain of the PID controller, while the second approach will involve a hybrid of the fuzzy logic to adapt the gain of the proportional and integral controller, referred to as Fuzzy-PI, control approaches. These approaches will be implemented on Arduino Duo (84Mhz clock) and wired to a Pixy camera, which will transmit the generated trajectory for each 20ms to the Pixhawk autopilot that positions the Quadcopter. In [14], researchers performed trajectory tracking control of a quadcopter using Genetic Algorithm, Particle Swarm Optimisation, and Artificial Bee Colony to self-tune the Fuzzy Logic Controller that was implemented on a drone. [15] proposed a combination of a nonlinear controller and a PID controller to enhance the altitude stabilization of a Quadcopter. Again, the controller was installed on the Pixhawk autopilot that drove the Quadcopter. [16] proposed a safe landing system for a Pixhawk autopilot that was based on fuzzy logic that was implemented on Arduino Mega(16Mhz). This was connected to a laser rangefinder sensor to improve the efficiency of altitude sensing. The control loop to support safe landing was updated every 200ms. [17] proposed the use of a Fuzzy-PID controller for the purpose of tracking the trajectory of a Quadcopter. The Fuzzy-PID controller was compared with a traditional PID controller and Fuzzy controller. The performance of all the controllers was simulated using MATLAB software. [18] proposed the development of a controller for attitude stabilization that offered a hybrid of particle swarm optimization and fuzzy logic to achieve optimized PID gains within a brief period. The controller was subsequently tested in MATLAB. [19] presented a dynamical model of the quadcopter that was based on Newton-Euler equations. This study involved a fuzzy system that automatically adjusted the PD to achieve controller gains, a classical PID controller, and a fuzzy controller. The system also incorporated a fuzzy system that made automatic adjustments to the controller gains of the PD to stabilize the quadcopter and control the altitude at which it flew. The Takagi-Sugeno inference mechanism fuzzy inference type was employed in this study and was validated in a MATLAB\Simulink environment. [20] proposed additional membership functions to distribute the input and output of the Fuzzy Gain Scheduling of the PID Controlled Quadrotor for Payload Drop Missions. This was then compared to the performance of linear gain scheduling of the PID and a classical PID controller. The various control strategies involved were installed in an Arduino MEGA2560 autopilot that controlled the quadcopter. [21] proposed a hybrid of different variations of the PSO and fuzzy logic to achieve gain scheduling of a PID that controlled a quad-tilt wind. The optimized algorithm fuzzy gain-scheduled GS-PID controllers were simulated within a processor-in-the-loop framework. The PSO in this work tuned the inputs and outputs of the fuzzy logic system that is responsible for adjusting the PID gains. [22] presented a comparative study that compared the performance of a fuzzy gains scheduled PID controller with that of a classic PID controller that is in common use within the process control industry. [3] described the development of a PID and Fuzzy-PID Control Model that was designed to stabilize a Quadcopter. MATLAB was used to simulate the performance of the models. [4] proposed a FuGSPID to stabilize the altitude of a quadrotor in the Presence of Actuator Faults.

According to the existing literature, FuGSPID controllers have been specifically designed for altitude stabilization; however, the methods developed were not able to achieve the right optimization. For this reason, Particle Swarm Optimization was incorporated. Fuzzy logic typically incorporates a variety of parameters and conditioning; as such, the coding algorithms involved are relatively complex. The size of the algorithms undermines performance and reduces the speed at which a low-cost microcontroller responds through the Teensy development board. To overcome this challenge, we suggested an algorithm that reduces the C++ coding programs for the FuGSPID controller to achieve the fastest processing time. The code for the controller is presented in the Appendix.

The remainder of this paper is organized as follows. The dynamical model of the quadrotor is described in Section 2. The mathematical description of the FuGSPID controller is described in Section 3, while Section 4 describes the particle swarm optimization strategy for the controller. Finally, the results of the simulation together with the conclusion, will be presented in Sections 5 and 6 respectively.

2. The quadrotor model

In Based on the previous work [1, 2, 6] We used the simplified altitude dynamical model of a quadrotor same in [6]. For that just the equation of Z axis was selected and is given by:

$$\begin{cases} \ddot{z} = \frac{T}{m}\cos\theta\cos\varphi - g \\ T = f_1 + f_2 + f_3 + f_4 \\ \qquad \theta \approx \varphi \\ f_i = C_T\omega_i^2, i = 1,2,3,4 \end{cases} \tag{1}$$

Where *fi (i=1,2,3,4)* are the t forces produced by motors, *(g)* gravity constant; a quadrotor mass m. In hovering condition the roll and pitch angles *(θ,φ)* approximately are zero degree. Some of parameters taken from [23] Table 1.

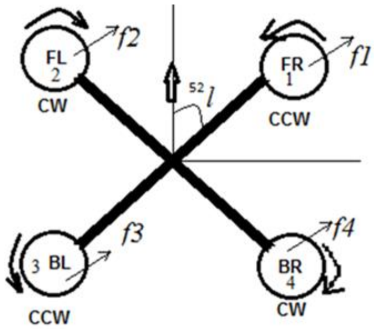


Figure. 1 A quadrotor configuration

Table 1. Parameters of the quadcopter [1, 2, 23]

Calculated Trust Coefficient	8.5 10^{-7}
Trust Coefficient C_T	8.8 10^{-7}
Mass	490^{g}
l	0.125 meter

The trust coefficient calculated in experiment is approximate to trust coefficient in [23] see Table 1. The frequency response of the propeller (GemFan 5030) speed with respect to the duty ratio of the motor speed controller (see Table 2) showed that dynamic thrust and duty ratio could be modelled as a first-order transfer function [25, 26].

The frame is a 250 mm (Q250) fiberglass frame similar to experiment in this work [27]. The flight controller based on Teensy development board [28], FreeIMU library [29] and Mahony library [30, 31]. The real photo of a quadrotor in Fig. 2 and picture during flying in Fig. 3. The Ultrasonic and Barometer sensors are attached to the Teensy development board for altitude calculation and estimation. Since the software was mostly custom-developed, it could easily interface with any additional sensors and modify the control laws strategy [4, 27].

3. Fuzzy gain-scheduled pid

In this Figure 4 presents a general scheme of a

Table 2. Some experiment results for trust calculation

	Min	Max
Duty (ms)	1410	1900
RPM(Round Per Minute)	11225	20730
Motor speed (rad/s)	1175.48	2170.84
Weight (kg)	0.125	0.39
Trust (N)= Weight*gravity	1.23	3.83
Trust coefficient= Trust/(Motor speed2)	8.12 10^{-7}	8.87 10^{-7}

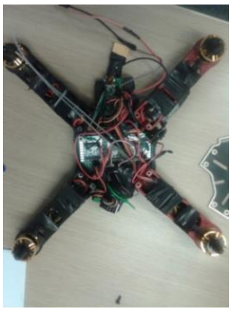


Figure. 2 The realized quadrotor

Figure. 3 Altitude hold

FuGSPID. The PID controller transfer function as follows:

$$g(s) = K_p + \frac{K_i}{s} + K_d s \tag{2}$$

Many studies have demonstrated the efficiency of PID controllers, and they have found widespread application in industrial fields. However, when the parameters that underpin a nonlinear system are violated, they may not perform effectively. It is for this reason that it is more effective to choose a hybrid that is developed according to an advanced theory and, thereby, modify its behavior from linear to nonlinear. To achieve this, we propose the use of hybrid fuzzy logic within the PID controller. The fuzzy logic will adjust the PID gains in response to error, and this will optimize the performance of the system. The proposed strategy will be referred to as Fuzzy Gain-Scheduled PID (FuGSPID).

The online adjustment is based on a simple linear formulation [4]:

$$\begin{cases} K_p = (K_{pmax} - K_{pmin})F_{kp} + K_{pmin} \\ K_i = (K_{imax} - K_{imin})F_{ki} + K_{imin} \\ K_d = (K_{dmax} - K_{dmin})F_{kd} + K_{dmin} \end{cases} \tag{3}$$

We defined ranges for the subsequent PID gains, which were achieved using the Z-N method, as presented in Table 3.

To calculate the three outputs of the fuzzy system F_{kp}, F_{ki}, F_{kd} , we consider the following rules:

$$IF\ e\ is\ X\ and\ \Delta e\ is\ Y\ then\ output\ c(j,i) \tag{4}$$

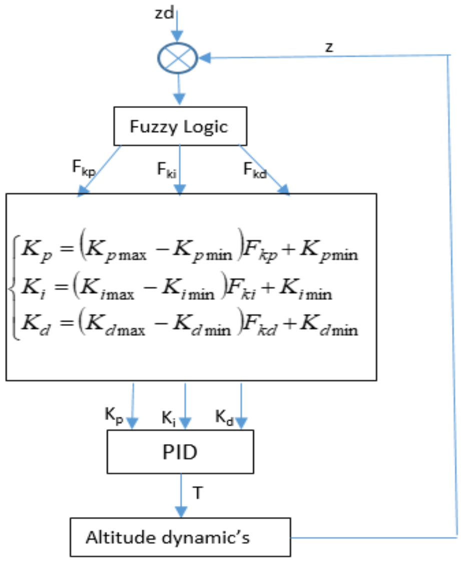

zd
z
Fuzzy Logic
F_{kp}
F_{ki}
F_{kd}
$K_p = (K_{p\max} - K_{p\min})F_{kp} + K_{p\min}$
$K_i = (K_{i\max} - K_{i\min})F_{ki} + K_{i\min}$
$K_d = (K_{d\max} - K_{d\min})F_{kd} + K_{d\min}$
K_p
K_i
K_d
PID
T
Altitude dynamic's

Figure. 4 A general scheme of a Fuzzy Gain-Scheduled PID (FuGSPID)

Table 3. PSO searching interval

	Z-N produced gains	Chosen interval for PSO
P	24	[K_{pmin}, K_{pmax}]=[13 30]
I	0.1	[K_{imin}, K_{imax}] =[0.01 1.5]
D	0.025	[K_{dmin}, K_{dmax}]=[0.01 1.5]

Habitually programming fuzzy logic can be immensely time-consuming in comparison to programming alternative controllers. This is particularly true in the case of the current study, which was implemented on the Teensy Development Board 3.2. The clock involved is 98Mhz, which is relatively slow; as such, we will attempt to identify the smallest mathematical description for FuGSPID, which has never previously been published. For this purpose, we will employ the Mamdani type inference engine (and, implication) for multiplication, the aggregation will be the summation, and the defuzzification will be performed by the centroid formulate. The degree of membership for the input is denoted by triangular shapes, and the outputs are singleton (as in Fig. 6). The input of the fuzzy logic is distributed according to five sets of linguistic variables: negative large (*NL*), negative medium (*NM*), negative small (*NS*), zero (*Ze*), positive small (*PS*), positive medium (PM), and positive large (*PL*). These linguistic variables represent both the degree of error and any error in the change in altitude while moving along a vertical trajectory. The details of these variables can be seen in Fig. 5. Two inputs (*e*,

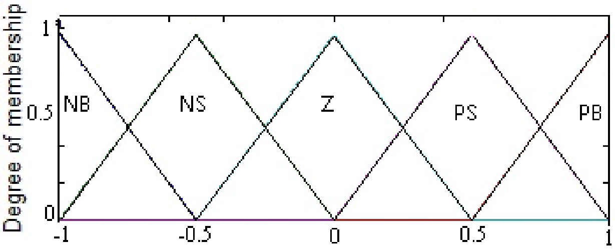


Figure. 5 Fuzzy sets fuzzification of error and its change (*e*, *ce*)

Table. 4 Fuzzy baserules for Fuzzy Gains-Scheduled PID

fK_p	Δe					
		NB	NS	Z	PS	PB
e	NB	B	B	B	B	B
	NS	S	B	B	B	S
	Z	S	S	S	S	S
	PS	S	B	B	B	S
	PB	B	B	B	B	B

Note: S=0.1; B=0.9

fK_i	Δe					
		NB	NS	Z	PS	PB
e	NB	Z	Z	Z	Z	Z
	NS	M	Si	Z	Si	M
	Z	B	M	Si	M	B
	PS	M	Si	Z	Si	M
	PB	Z	Z	Z	Z	Z

Note: Z=0;Si=0.3; M=0.7; B=1

fK_d	Δe					
		NB	NS	Z	PS	PB
e	NB	S	S	S	S	S
	NS	B	S	S	S	B
	Z	B	B	B	B	B
	PS	B	S	S	S	B
	PB	S	S	S	S	S

$$outfuzzy = c(j,i) = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} & c_{04} \\ c_{10} & c_{11} & c_{12} & c_{13} & c_{14} \\ c_{20} & c_{21} & c_{22} & c_{23} & c_{24} \\ c_{30} & c_{31} & c_{32} & c_{33} & c_{34} \\ c_{40} & c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix}$$

N: Negative; B: Big; S: Small; Z: Zero; P: Positive; M: Medium

de) and three outputs (F_{kp}, F_{ki}, F_{kd}) are normalized in the same interval [-1, 1]. Table 4 presents an overview of the linguist rules used in the Fuzzy Logic Controller "*FLC*." Three sets of 25 rules were used to determine the three outputs (F_{kp}, F_{ki}, F_{kd}) employed to calculate the PID controller gains.

The optimized parameters of the FuGSPID, which were used in [32], were developed according to a standard nonlinear system. However, the present work will optimize these parameters with the intention of stabilizing the altitude of a quadrotor.

The FuSGPID will be implemented in Teensy 3.2, which is a microcontroller 32 bit ARM Cortex-M4 72 MHz. This is a low-cost microcontroller. As such, the incorporated clock is relatively slow. Furthermore, fuzzy logic incorporates a significant number of parameters and details that can have a negative impact on the time it takes for the microcontroller functions to be executed. Furthermore, in this case, repetitive tasks that undermine the performance of the microcontroller will need to be avoided.

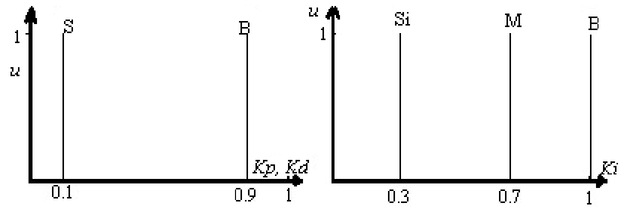


Figure. 6 Output fuzzy set and membership functions for *Kp*, *Kd* and *Ki*.

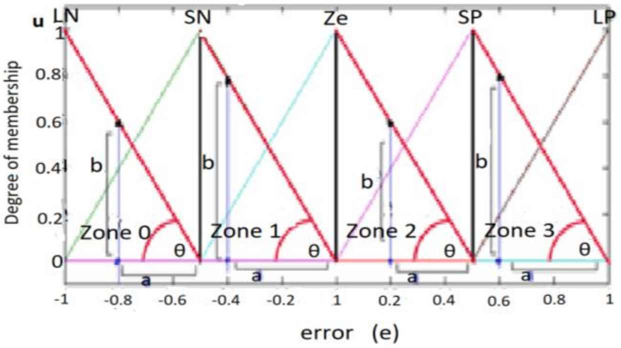


Figure. 7 Fuzzy sets with some triangle geometry.

Table 5. Fuzziffication of error and its change cases

Error(*e*)	Error in zone *i*	*a*
-0.8	0	-0.5-(*e*)=-0.5-(-0.8)
-0.4	1	0-(*e*)=0-(-0.4)
0.2	2	0.5-*e*=0.5-0.2
0.6	3	1-*e*=1-0.6

The presented mathematical fuzzy system description will help to reduce the coding program of the controller's next section and explains all the steps required to achieve the lowest algorithm, which will be implemented on Teensy.

The membership function of the error and its change in Fig. 7 is divided into four zones: Zone 0, 1, 2, 3). The red line indicates where their tangent is negative. Here, we can observe that they create a theta angle, which can be used to facilitate a calculation of the degree of membership, u, for the error and its associated change. Since the membership triangle is symmetrical, all red lines are equal.

Fig. 7 depicts fuzzy sets with some triangular geometry. This approach will help to derive the most basic mathematical description for the fuzzy system without requiring significant C++ coding for a Teensy development board or Arduino platform.

By using Fig. 7 we can notice that:

$$\tan\theta = \frac{1}{0.5} = \frac{a}{b} \Longrightarrow b = 2a \tag{5}$$

Let study all the cases of the variation of the error between these zones as in the next table.

$$a = \frac{i-1}{2} - e \tag{6}$$

Where *i* is zone number and e is value of error.

Implies that degree of the membership will be as:

$$\begin{cases} u = b = 2a = 2\left(\frac{i-1}{2} - e\right) \\ = i - 1 - 2e = i - (1 + 2e) \end{cases} \tag{7}$$

In same zone there is tow fuzzy sets the grade of membership for the first calculated as in Eq. (7), the second one will the compliment of the first as in Eq. (8) because the shapes of the fuzzy sets are triangle and symmetric.

$$compliment\ of\ u = u_c = 1 - u \tag{8}$$

Let take example if the error and its derivative where *e*=0.4 and *Δe*=-0.2.

$$e = 0.4 \rightarrow i = 2 \rightarrow active\ sets\ (Z, SP) \tag{9}$$

Calculate their memberships grads using Eq. (7) and replace *i* by 2 and *e*=0.4:

$$for\ set\ Z: u = i - (1 + 2e) = 0.2 \tag{10}$$

$$for\ set\ SP: u_c = 1 - u = 0.8 \tag{11}$$

Same thing for *Δe*, their membership grads are: $\Delta e = -0.2 \rightarrow active\ sets(Z, NS)$

$$for\ set\ Z: u = j - (1 + 2e) = 0.4 \tag{12}$$

$$for\ set\ NS: u_c = 1 - u = 1 - 0.4 = 0.6 \tag{13}$$

For example, to calculate output of *Fkp*, the four rules actives, here will add link between zone number *(i,j)* and *(line,column)* number of the rule base for *Fkp* which is :

For error *i=2* the active fuzzy sets are *Z* and *PS* which are correspond to line 3 and 4 in *Fkp* rule Table 6 which fuzzy sets active, can use zone number by the next formulate:

Table 6 Fuzzy rulebase for *Fkp*

	Δe					
		NB	**NS**	**Z**	**PS**	**PB**
e	**NB**	B	B	B	B	B
	NS	S	B	B	B	S
	Z	S	S	B	S	S
	PS	S	B	B	B	S
	PB	B	B	B	B	B

Note: S=0.1; B=0.9

For error each zone has two fuzzy sets

For the first fuzzy set: Line number in rule table of *Fkp=zone number+1 =i+1*

For the second fuzzy set: Line number in rule table of *Fkp=zone number+2=i+2*

For error change (*Δe*) same formulate in error case.

The zone number in the example are: *i=2, j=1*

Rule 1: i=2 and j=1 : if e is Z and Δe is NS then output is c(2,1)

Rule 2: i+1=3 and j=1 : if e is SP and Δe is NS then output is c(3,1)

Rule 3: i=2 and j+1=2: if e is Z and Δe is Z then output is c(2,2)

Rule 4: i+1=3 and j+1=2: if e is SP and Δe is Z then output is c(3,2)

There many ways to calculate fuzzy inference [33-35], the output of the fuzzy inference or fuzzy rules such as:

Rule 1: $R_1 = \mu_i(e) * \mu_j(\Delta e) = \mu_2(e) * \mu_1(\Delta e)$

Rule 2: $R_2 = \mu_i(e) * \mu_{j+1}(\Delta e) = \mu_i(e) * (1-\mu_j(\Delta e))$

Rule 3: $R_3 = \mu_{i+1}(e) * \mu_j(\Delta e) = (1-\mu_i(e)) * \mu_j(\Delta e)$

Rule 4: $R_4 = \mu_{i+1}(e) * \mu_{j+1}(\Delta e) = (1-\mu_i(e)) * (1-\mu_j(\Delta e))$

The are many defuzzification method [33-35], the Centroid of area method selected in our case:

$$output_{FS} = \frac{\sum_{k=1}^{4} R_k c(j,i)}{\sum_{k=1}^{4} R_k} \tag{14}$$

Using output of inference in Eq. (14), we find:

$$\sum_{k=1}^{4} R_k c(j,i) = 1 \tag{15}$$

By using the numeric example which calculated before for fuzzy inference and defuzzification, replace each variable by their values:

$R_1 = \mu_i(e) * \mu_j(\Delta e) = \mu_2(e) * \mu_1(\Delta e) = 0.2*0.4 = 0.005$

$R_2 = \mu_i(e) * \mu_{j+1}(\Delta e) = \mu_2(e) * \mu_2(\Delta e) = 0.2*0.6 = 0.045$

$R_3 = \mu_{i+1}(e) * \mu_j(\Delta e) = \mu_3(e) * \mu_1(\Delta e = 0.8*0.4 = 0.095$

$R_4 = \mu_{i+1}(e) * \mu_{j+1}(\Delta e) = \mu_3(e) * \mu_2(\Delta e) = 0.8*0.6 = 0.855$

$\sum_{k=1}^{4} R_k c(j,i) = R_1 c(1,2) + R_2 c(2,2) + R_3 c(1,3) + R_4 c(2,3)$

By using Table 6: *c(2,1)=S=0.1*, *c(2,2)=B=0.9*, *c(3,1)= B=0.9*, *c(3,2)=B=0.9.*

4. Fuzzy gain-scheduled PID tuning by particle swarm optimization

PSO, introduced by Kennedy and Eberhart in 1995 [36]. It is based on Swarm Intelligence methods, for optimal values searching.

There are several algorithms used for PSO [36-38]. In this work, the integral of square error (ISE) were fitness functions, and the constriction coefficient [36, 39] method was selected. The converge of the algorithm is guaranteed and the convergence in our case is clear in figures 9 [36] and PSO formulate according the following:

$$v_{ij}(t+1) = X\left[v_{ij}(t) + \phi_1\left(y_{ij}(t) - x_{ij}(t)\right) + \phi_2\left(\hat{y}_j - x_{ij}(t)\right)\right] \tag{16}$$

$$X = \frac{2k}{2-\phi-\sqrt{\phi(\phi-4)}} \tag{17}$$

$$\phi = \phi_1 + \phi_2, \phi_1 = c_1 r_1, \phi_2 = c_2 r_2, \phi \geq 4, k \in [0\ 1]$$

c1 and *c2* are the acceleration coefficients, and r_1 and r_2 are random values [0 1]

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{18}$$

To verify the performance, we used the integral of square error (ISE) as comparison criteria, defined as follows:

$$ISE = \int_0^\infty e^2(t)dt \tag{19}$$

e= The desired angle –The actulle angle

Therefore, for optimization tuning of controller we used criteria Eq. (19) as the objective function for seek a set of parameters (*S, B, Si, M*) such that the control system met the minimum performance criteria.

PSO used to tune four parameters in fuzzy rules Table 4 (*S, B* in *FKp, Fkd* and *Si,M* in *FKi,*)[4]. Each particle represented a solution for controller. A good set of four parameters yielded a good system response, and resulted in minimization of the performance criteria is listed in Eq. (19). The general strategy of the optimization problems is represented in Fig. 8.

The PSO needed to determine number of particles and iterative number in order to calculate the PID gains. In our case the number, of particles was set to 15 and the iterative number was set to 20. Fig. 8 shows the movements of the particles during the searching for optimized fuzzy output parameters of Fuzzy Gain-Scheduled PID according to the objective function. The blue points represent the swarms, the black points represent the local best

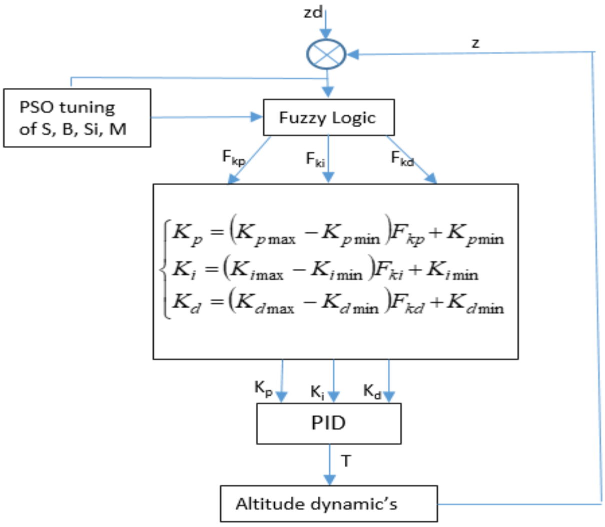
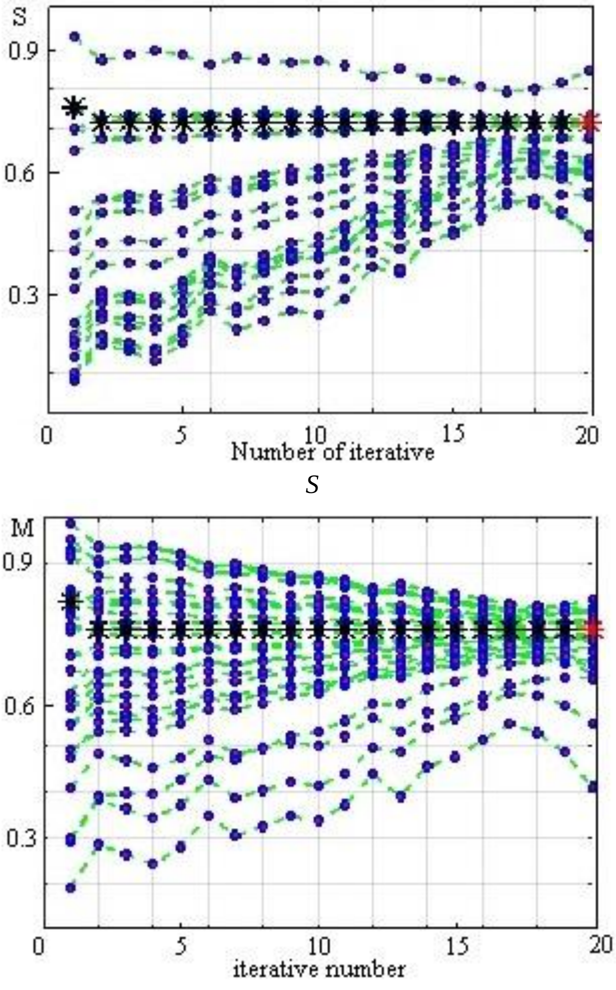

zd
z
PSO tuning of S, B, Si, M
Fuzzy Logic
F_kp
F_ki
F_kd
$K_p = (K_{p\max} - K_{p\min})F_{kp} + K_{p\min}$
$K_i = (K_{i\max} - K_{i\min})F_{ki} + K_{i\min}$
$K_d = (K_{d\max} - K_{d\min})F_{kd} + K_{d\min}$
K_p
K_i
K_d
PID
T
Altitude dynamic's

Figure. 8 The general strategy of optimization



S

M

Figure. 9 Movement of particles during the search for the optimal parameters (*S, B, Si, M*)

Table 10. Summary of PID gains obtained by Z-N tuning and PSO

	Controller parameters	ISE
PID	PID (P=5, I=0.15, D=0.01)	5.51 10^{-6}
Before optimization	S=0.1, B=0.9, Si=0.3; M=0.7	4.43 10^{-6}
After optimization	S=0.7153, B=0.9456, Si=0.5425; M=0.7554	3.507 10^{-6}

Table 11. Output fuzzy sets after optimization

FK_p	fK_d	fK_i
No optimization: S=0.1; B=0.9	No optimization: S=0.1; B=0.9	No optimization: Z=0;Si=0.3; M=0.7; B=1
After optimization S=0.7153; B=0.9456	*After optimization* S=0.7111; B=0.9456	*After optimization* S=0.5425; M=0.7554

particles, and the red points represent the global best particles.

5. Results and discussion

The PID gains and output fuzzy sets parameters of FuGSPID both before and after optimization with Integral Squire Error (ISE) for each case with 40 second simulation time, are summarized in Table 7.

The new parameters (*S*, *B*, *Si*, *M*) for Fuzzy Gain-Scheduled PID controller after optimization in Table 7, with the new parameters the objective function which is Integral Square Error is lower error.

The parameters according to the output fuzzy gains (F_{kp}, F_{ki}, F_{kd}) are in Table 8:

Table 11 Output fuzzy sets after optimization

Wherever Times is specified, Times Roman of Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred.

The simulation related to trajectory tracking problem, the three controllers response are plotted in same plot Fig. 10 to visualize the response difference between the three controllers (PID, no optimized and optimazed FuGSPID).

Fig. 11 show the output of controller response which is the global thrust produced by four motors to keep the quadcopter in desired reference and to perform trajectory tracking.

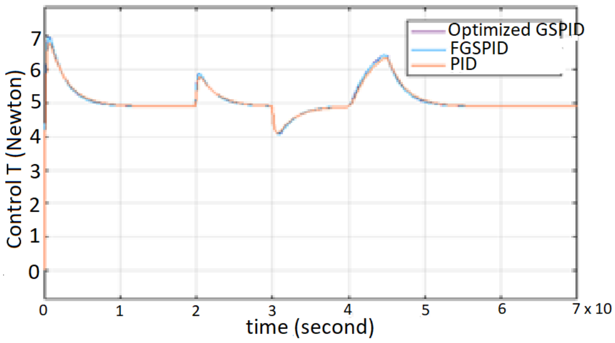


Figure. 10 Altitude tracking.

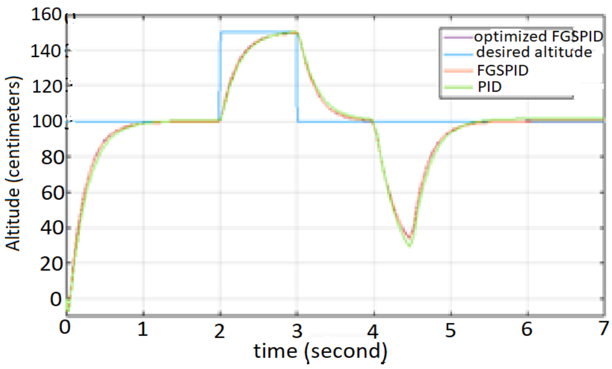


Figure. 11 thrust control variation.

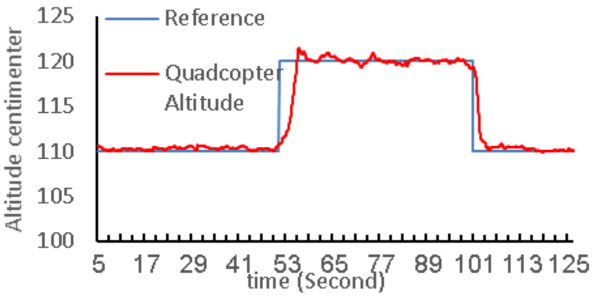


Figure. 12 experimental response.

Fig. 12 show simple trajectory tracking which is altitude variation. The quadcopter was kept hold at (110cm) and at (53secnd) the desired altitude changed to (120cm) and after retuned again to the old desired altitude. Can notice in real picture Figure 3 during flying that a quadcopter is attached with superfine wire because no (*x*,*y*) axis controller which need specific configuration dealing as adding camera and that is not goals of the paper. The goals of paper related with just altitude controller.

6. Conclusion

The presented work describes the use of a FuGSPID to stabilize the altitude of a homebuilt Quadcopter. Many researchers have highlighted the effectiveness of FuGSPID controllers. To improve its performance or verify the optimal configuration of the controller, Particle Swarm Optimization (PSO) was selected. PSO identifies new configurations that will generate a low trajectory tracking error, as proven in MATLAB simulations. Both simulations and experiments were employed to validate the effectiveness of the controller. A fuzzy logic system incorporates multiple parameters and conditioning instructions. The complexity of the system has a negative impact on the processing speed. As such, a new approach is required by which it is possible to identify the lowest algorithm code for FuGSPID. The mathematical discerption it be useful to reduce number of instructions and loops and that will help microcontrollers to operate at optimal speed when performing tasks that involve a minimal amount of repetition.

7. Notations

fi	(i=1,2,3,4) are the t forces produced by motors
g	gravity constant
m	a quadrotor mass
θ, φ	the roll and pitch angles of a quadcopter attitude
C_T	Trust Coefficient
l	A quadcopter length
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimization
FuGSPID	Fuzzy gain-scheduled PID
UAV	Unmanned Aerial Vehicle
ISE	Integral of the Square Error
FL, FR	Front Left, Front Right
BL, BR	Back Left, Back Right
IMU	Inertial Measurement Unit

Appendix

The implemented code of Fuzzy Gain-Scheduled PID

```
float z=0;
float si=0.3;
float m=0.7;
float bi=1;
float b=0.9;
float s=0.1; //0.3
float ukp=0;
float uki=0;
float ukd=0;
float e=0;
float de=0;
float ckp[5][5]={{b,b,b,b,b},
          {s,b,b,b,s},
          {s,s,b,s,s},
          {s,b,b,b,s},
          {b,b,b,b,b}};

float cki[5][5]={{bi,bi,bi,bi,bi},
          {m,m,bi,m,m},
          {z,si,bi,si,z},
          {m,m,bi,m,m},
          {bi,bi,bi,bi,bi}};

float ckd[5][5]={{s,s,s,s,s},
          {b,s,s,s,b},
          {b,b,b,b,b},
          {b,s,s,s,b},
          {s,s,s,s,s}};

void fuzzy ()
{
 float ua,ub,d1,d2,d3,d4;
 ua=0;ub=0;d1=0;d2=0;d3=0;d4=0;
 int i,j;

  e =err_hzold/5;
  de = err_vzold/5;
 i=0;j=0;
 if (e<=-1) {e=1;}
 if (e >=1) {e=1;}
 if (e<=-1/2) {i=0; }
 else if (e>-1/2 && e<=0) {i=1;}
 else if (e>0 && e<= 1/2) {i=2;}
 else if (e>1/2 ) { i=3;}

 if (de<=-1) {de=1;}
 if (de >=1) {de=1;}
 if (de<=-1/2) {j=0; }
 else if (de>-1/2 && de<=0) {j=1;}
 else if (de>0 && de<= 1/2) {j=2;}
 else if (de>1/2 ) { j=3;}

 ua= ((-1+(i/2))-e)/2;
 ub= ((-1+(j/2))-de)/2;
 d1=ua*ub;
 d2=ua*(1-ub);
 d3=(1-ua)*ub;
 d4=(1-ua)*(1-ub);

ukp=ckp[j][i]*d1+ckp[j+1][i]*d2+ckp[j][i+1]*d3+ckp[j+1][i+1]*d4;

ukd=ckd[j][i]*d1+ckd[j+1][i]*d2+ckd[j][i+1]*d3+ckd[j+1][i+1]*d4;

uki=cki[j][i]*d1+cki[j+1][i]*d2+cki[j][i+1]*d3+cki[j+1][i+1]*d4;
 }
```

Conflicts of interest

“The authors declare no conflict of interest.”.

Author contributions

“Conceptualization, methodology, validation, Dr Khodja; software, formal analysis Dr Khodja and Dr Mahfoudhi; writing—original draft preparation, Dr Khodja; writing—review and editing, Dr Mahfoudhi; project administration, Dr Khodja; funding acquisition, Dr Mahfoudhi”.

References

[1] M. A. Khodja, M. Tadjine, M. S. Boucherit, and M. Benzaoui, “Tuning PID attitude stabilization of a quadrotor using particle swarm optimization (experimental)”, *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 8, No. A8, 2017.

[2] M. A. Khodja, M. Tadjine, M. S. Boucherit, and M. Benzaoui, “Experimental dynamics identification and control of a quadcopter”, In: *Proc. of 2017 6th International Conference on Systems and Control, Batna, Algeria*, pp. 498-502, 2017.

[3] E. Kuantama, T. Vesselenyi, S. Dzitac, and R. Tarca, “PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter”, *International Journal of Computers Communications & Control*, Vol. 12, No. 4, pp. 519-532, 2017.

[4] M. H. Mohammad, C. Abbas, and Z. Youmin, “Fault-Tolerant Fuzzy Gain-Scheduled PID for a Quadrotor Helicopter Testbed in the Presence of Actuator Faults”, In: *Proc. of IFAC Conference on Advances in PID Control*, Vol. 45, No. 3, pp. 28-30, 2012.

[5] M. F. Rahmat, “Application of selftuning fuzzy PID controller on industrial hydraulic actuator using system identification approach”, *International Journal on Smart Sensing and Intelligent Systems*, Vol. 2, No. 2, pp. 246-261, 2009.

[6] M. A. Khodja, C. Larbes, N. Ramzan, and A. Ibrahim, “Implementation of Heuristical PID Tuning for Nonlinear System Control”, *International Review of Automatic Control (IREACO)*, Vol. 12, No. 2, pp. 108-114, 2019.

[7] A. P. Engelbrecht, *Computational intelligence: an introduction*, John Wiley & Sons, 2007.

[8] R. Eberhart and J. Kennedy, “Particle swarm optimization”, In: *Proc. of the IEEE international conference on neural networks*, Vol. 4, pp. 1942-1948, 1995.

[9] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial”, *Chemometrics and Intelligent Laboratory Systems*, Vol. 149, pp. 153-65, 2015.

[10] P. R. Dian, M. S. Siti, and S. S. Yuhaniz, “Particle swarm optimization: Technique, system and challenges”, *International Journal of Computer Applications*, Vol. 14, No. 1, pp. 19-26, 2011.

[11] G. Štimac, S. Braut, and R. Žigulić, “Comparative analysis of PSO algorithms for PID controller tuning”, *Chinese Journal of Mechanical Engineering*, Vol. 27, No.5, pp. 928-936, 2014.

[12] M. A. Khodjal, M. Tadjine, M. S. Boucherit, and K. Busawon, “Optimization of a proportional derivative (PD) fuzzy controller using the particle swarm optimization (PSO) technique for a 3DOF robot manipulator”, *The Mediterranean Journal of Measurement and Control*, Vol. 12, No. 4, pp. 670-677, 2016.

[13] M. Rabah, A. Rohan, S. A. Mohamed, and S. H. Kim, “Autonomous Moving Target-Tracking for a UAV Quadcopter Based on Fuzzy-PI”, *IEEE Access*, Vol. 7, pp. 38407-38419, 2019.

[14] E. Yazid, M. Garratt, and F. Santoso, “Position control of a quadcopter drone using evolutionary algorithms-based self-tuning for first-order Takagi–Sugeno–Kang fuzzy logic autopilots”, *Applied Soft Computing*, Vol. 78, No. 1, pp.373-392, 2019.

[15] N. Xuan-Mung and S. K. Hong, “Improved Altitude Control Algorithm for Quadcopter Unmanned Aerial Vehicles”, *Applied Sciences*, Vol. 9, No. 10, p. 2122, 2019.

[16] M. Talha, F. Asghar, A. Rohan, M. Rabah, and S. H. Kim, “Fuzzy logic-based robust and autonomous safe landing for UAV quadcopter”, *Arabian Journal for Science and Engineering*, Vol. 44, No. 3, pp. 2627-2639, 2019.

[17] M. Rabah, A. Rohan, Y. J. Han, and S. H. Kim, “Design of fuzzy-PID controller for quadcopter trajectory-tracking”, *International Journal of Fuzzy Logic and Intelligent Systems*, Vol. 18, No. 3, pp. 204-213, 2018.

[18] H. K. Tran and J. S. Chiou, “PSO-based algorithm applied to quadcopter micro air vehicle controller design”, *Micromachines*, Vol. 7, No. 9, 2016.

[19] Y. J. Ryoo, “An autonomous control of fuzzy-PD controller for quadcopter”, *International Journal of Fuzzy Logic and Intelligent Systems*, Vol. 17, No. 2, pp. 107-113, 2017.

[20] I. H. Gue and A. Y. Chua, “Development of a Fuzzy GS-PID Controlled Quadrotor for

Payload Drop Missions", *Journal of Telecommunication Electronic and Computer Engineering (JTEC)*, Vol. 10, No. 1-5, pp. 55-58, 2018.

[21] K. B. Khoud, S. Bouallègue and M. Ayadi, "Design and co-simulation of a fuzzy gain-scheduled PID controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle", *Transactions of the Institute of Measurement and Control*, Vol. 40, No. 14, pp. 3933-3952, 2018.

[22] S. Ahmad, S. Ali, and R. Tabasha, "The design and implementation of a fuzzy gain-scheduled PID controller for the Festo MPS PA compact workstation liquid level control", *Engineering Science and Technology, an International Journal*, Vol. 23, No. 2, pp. 307-315, 2019.

[23] http://ecalc.ch/ (date access: 15/11/2019)

[24] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies", *IET Control Theory & Applications*, Vol. 6, No. 12, pp. 1812-1827, 2012.

[25] Y. Myunggon, "A transfer function model of thrust dynamics for multi-rotor helicopters", *International Journal of Engineering Research & Technology*, Vol. 5, No. 1, pp. 15-18, 2016.

[26] M. H. Tanveer, S. F. Ahmed, D. Hazry, F. A. Warsi, and M. K. Joyo, "Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions", *American Journal of Applied Sciences*, Vol. 10, No. 8, pp. 819-831, 2013.

[27] C. Liu and S. D. Prior, "Design and implementation of a mini quadrotor control system in GPS denied environments", *International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, USA, pp. 462-469, 2015.

[28] https://www.pjrc.com/store/teensy32.html (last access 15-11-2019).

[29] http://www.varesano.net/projects/hardware/FreeIMU (last access 15-11-2019)

[30] https://github.com/PaulStoffregen/MahonyAHRS (last access 15-11-2019).

[31] R. Mahony, T. Hamel, and M. J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group", *IEEE Transactions on automatic control*, Vol. 53, No. 5, pp. 1203-1218, 2008.

[32] K. W. Yu, and J. H. Hsu, "Fuzzy gain scheduling PID control design based on particle swarm optimization method", In: *Proc. of Second International Conference on Innovative Computing, Information and Control*, Japan, pp. 337-337, 2007.

[33] J. Mendel, H. Hagras, W. W. Tan, W. W. Melek, and H. Ying, *Introduction to type-2 fuzzy logic control: theory and applications*, John Wiley & Sons, 2014.

[34] M. M. Adnan, A. Sarkheyli, A. M. Zain, and H. Haron, "Fuzzy logic for modeling machining process", *Artificial Intelligence Review*, Vol. 43, No. 3, pp. 345-379, 2015.

[35] E. Dadios, *Fuzzy Logic - Controls, Concepts, Theories and Applications*, INTECH, Rijeka, Croatia, 2012.

[36] M. I. Menhas, L. Wang, M. Fei, and H. Pan, "Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design", *Expert Systems with Applications*, Vol. 39, No. 4, pp. 4390-4401, 2012.

[37] W. W. Tan and T. W. Chua, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions (Mendel, J.M.; 2001) [book review]", *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, pp. 72-73, 2007.

[38] A. Hadeel N, "An Improvement in LQR Controller Design based on Modified Chaotic Particle Swarm Optimization and Model Order Reduction", *International Journal of Intelligent Engineering & Systems*, Vol. 14, No. 1, pp. 157-168, 2021

[39] F. A. Hasan, L. J. Rashad, and A. T. Humod, "Integrating Particle Swarm Optimization and Routh-Hurwitz's Theory for Controlling Grid-Connected LCL-Filter Converter", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 102-113, 2020.