Back-stepping control of a quadcopter: A comparison between two different methods of gain tuning

Mahdi. Mirhadi
Electrical and Electronic Engineering Department
Kharazmi University
Tehran, Iran
mahdi.mirhadi96@gmail.com

Zahra. Goorkani Zarandi
Department of Mechanical Engineering
Kharazmi University
Tehran, Iran
goorkani1373@gmail.com

Iman. Zamani
Electrical and Electronic Engineering Department
Shahed University
Tehran, Iran
zamaniiman@shahed.ac.ir

Ehsan. Sadeghi Ghasemabad
Electrical and Electronic Engineering Department
Shahed University
Tehran, Iran
sadeghi1365@gmail.com

***Abstract*— Today quadcopters are used for a wide range of applications due to their excellent movement and high-quality flights. In this paper, the nonlinear dynamic model of the quadcopter is derived using the Newton-Euler method. The back-stepping control method is applied to the quadcopter's dynamic system along with the genetic algorithm to optimize and tune the design parameters of the controller. The adaptive control method is also used for online tuning the design parameters of the controller in order to compare the performance of the two gain tuning methods.**

Keywords— **Genetic algorithm, Adaptive back-stepping controller, Optimization**

I. Introduction

Early ideas for quadcopters appeared in the 1920s and 1930s, but those early designs had a high level of instability. There are four main forces acting on a quadcopter: (i) gravity: the force produced by a quadcopter due to its mass; (ii) lift: this is the upward force that the blades create for the quadcopter; (iii) drift: this is the horizontal motion force in the quadcopter that the propellers are involved in producing; and (iv) pulling: this is the quadcopter's deterrent force that enters from the air. Quadcopters, like helicopters, use the pressure difference in the atmosphere around them to take off. Using four engines to rotate blades. The blade rotation direction is reverse two by two to counteract the torque force and create the pressure difference necessary to lift off the ground.

In [1], a method is given, which uses an integral terminal sliding mode surface to ensure that the tracking error will converge to zero in finite time. The mention article also focuses on tracking trajectory with the disturbance consisting of 3 parts: 1-Height control, 2-Horizontal position, and 3-An attitude subsystem. The authors of the paper [2] have been designed a sliding mode controller, and it has been used in simulation in quadcopter. In this paper is demonstrated complete control for the quadcopter using the Proportional Derivative (PD) sliding mode controller, where a sliding mode controller tests the sliding surface of the PD. That results show that the proposed controller is Simpler on the PD controller, which is used as a sliding surface.

The paper [3] has been used a sliding mode disturbance observer that the dynamic model used is based on the first model proposed by Lozano, et al. That controller in this paper is designed to use a new sliding mode approach that uses a sliding mode observer to allow the calculation of continuous control for perturbations and model uncertainty. In paper [4] has been proposed method for sliding mode control base on independent quadcopter. The controller is based on Lyapunov theory, so the benefits are robust behavior to the uncertains and other disturbances. This article proofs that SMC can be reached quadcopter to the desire position, with the operating appropriate switching even with the injection of disturbance.

In paper [5], two methods of nonlinear control, sliding mode control and back-stepping control have been used. The performance and traceability of the controllers is also ensured by simulating the MATLAB program using Lyapunov's global stability theorem and comparing it with a traditional linear regulator controller. In paper [6] a second-order slip mode control method is proposed to design controllers for a small quadcopter drone. The result of this paper is that all state variables converge to their reference values, respectively, even if their reference values suddenly change at different times, different quadcopter paths are obtained by changing reference positions and different attitudes by changing reference angles. Other results include position tracking errors and the speed of all system mode variables are reduced to zero.

In paper [7], quadcopter problems are stability and tracking route with changing dynamics in the middle of gravity quadcopter that provides a systematic approach to derive the quadcopter model. This article designs an adaptive controller base on input-output linear feedback that recovers dynamic changes in the middle of the gravity quad. In paper [8], a simple signal-parametric adaptive control for quadcopter stabilization based on the passage method is proposed for both scalar modes (vertical plate stabilization and yawning angle stabilization) and vector control (roll and step angle stabilization). In paper [9], an adaptive robust tracking control system is designed for an unmanned rotor in which the dynamic quadrotor model is derived from the Newton-Euler method. The controller proposed in this paper includes two internal and external control loops that control the inner loop of rotational motion and Euler quadcopter angles. The outer loop controls the position and transfer motion of the quadcopter and calculates the desired angles for tracking the reference path. In paper [10], a nonlinear composite adaptive control algorithm for a six-degree quadcopter system is performed. The system under the operator is divided into two subsystems using dynamic inversion. Compound adaptive control is driven using information from both tracking error and parameter error. In paper [11], two types of adaptive mode space controllers are performed to stabilize the attitude and automatically adjust a four-rotor air robot, a quadcopter. In this work, numerical simulations based on a nonlinear model without singularity in quaternion representation show their superiority over the integral state controller with a fixed parameter. In [12-14], the genetic algorithm (GA) optimization method was used for the gain tuning of the PID controller implemented on a quadcopter. In the paper [19], the quadcopter has been controlled using a PID controller based on the Extended Kalman Filter (EKF) algorithm. One of the PID control limitations is that proportional gain, integration gain, and derivative gain must be manually adjusted, one solution being self-tuning [20]. Also, in the article [21], they presented a new method based on fuzzy logic to control a quadrotor.

In this paper, the nonlinear dynamic model of the quadcopter is derived using the Newton-Euler method. The back-stepping control method is applied to control the quadcopter. While it is popular to tune the design parameters of the controller by trial and error, this paper suggests two procedures for gain tuning, including genetic algorithm and fuzzy adaptive technique, and studies their performances. In fact, the novelty of the paper is in suggesting and comparing two different methods of gain tuning.

In this paper, we first model a quadcopter and then use a powerful nonlinear method to control it. Next, we ensure the stability of the system by using a suitable Lyapunov function. But to optimize the tracking error, we use genetic and fuzzy algorithms, which are validated by the simulation result.

We will optimize the control gains according to a cost function in the neural network optimization method and obtain fixed gains. In the fuzzy method, the obtained gains are variable with time, and the simulation results show that the fuzzy response is better than the neural one.

The results of the obtained method are also compared with the feedback linearization control method. The response to this method is much better than the feedback linearization method.

This paper is organized as follows: In Section 2, the mathematical model of the quadcopter system is presented. Design of the back-stepping controller for related system is explained in Section 3. Afterwards, in Section 4, we use a Genetic algorithm to find the best gain for the main controller based on a cost function. Section 4 is to design an adaptive fuzzy system for the back-stepping gains to have a better operation. Results and verification of the proposed methods are conducted in Section 6 by the aid of some simulation scenarios conducted in MATLAB. And at last in Section 7, conclusion of the article and comparing of the methods are proposed.

II. QUADCOPTER MODEL

The nonlinear dynamic model of the quadcopter is derived using the Newton-Euler method from [12]. The quadcopter has six main parameters; three of them are about the quadcopter position, including x, y, and z, and the other are for the angle of the quadcopter, including θ, φ, and ψ as shown in fig (1).

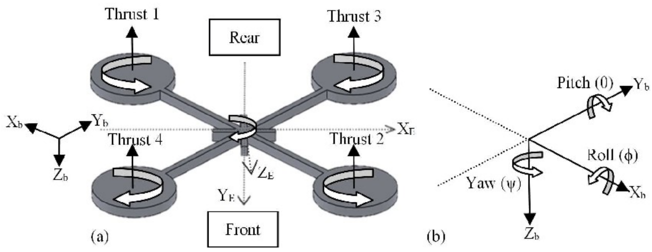

Fig (1).Quadcopter dynamic

A. Kinematics Model

Control torques and forces are created by changing the speed of the rotors (Ω1, Ω2, Ω3, Ω4) [12], which are shown in Table 1. The + sign in Table 1 means that the increase in rotor speed is related to the positive force or moment. A negative sign indicates that the rotor's deceleration is associated with the force or moment [15].

Table 1: generate positive control forces and moments varying rotor speeds

Force/moment	Ω1	Ω2	Ω3	Ω4
Roll moment		-		+
Pitch moment	+		-	
Yaw moment	+	-	+	-
Vertical thrust	+	+	+	+

Using the rotation matrix in paper [16]:

$$R = \begin{bmatrix} c\psi c\theta & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ s\psi c\theta & s\varphi s\theta s\psi + c\varphi c\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (1)$$

Where c is the cosine function and s is the sine function.

In paper [17], the Euler rates and angular body Rates are denoted by $\dot{\eta} = [\dot{\varphi}\dot{\theta}\dot{\psi}]^T$ and $\omega = [p\ q\ r]^T$, respectively. Eventually $\omega = R_r\dot{\eta}$ where

$$R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \quad (2)$$

B. Dynamic Model

The dynamic model consists of two parts, the rotational subsystem is the same Roll, Pitch, and Yaw, and the translational subsystem is the same altitude, x, y and z. The rotational equations of motion based on Newton-Euler are as follows[17]:

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \tag{3}$$

where J is quadcopter's diagonal inertia matrix that $J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$, ω is angular body rates, M_G is gyroscopic moments due to rotors' inertia and M_B moments acting on the quadcopter in the body frame.

Equation for translational of quadcopter is expressed according to Newton-Euler's law:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \tag{4}$$

where r is $[x\ y\ z]^T$ quadcopter's distance from the inertial frame, m is quadcopter's mass, g is the gravitational acceleration ($g = 9.81\, m/s^2$), and F_B is non gravitational forces acting on the quadcopter in the body frame that:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{5}$$

C. *States Space Model*

Using the state-space model, one can formulate the state variables of the rotational and translational dynamics as

$$\boldsymbol{X} = \left[\, \varphi\ \dot{\varphi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ z\ \dot{z}\ x\ \dot{x}\ \ y\ \dot{y} \right]^T \tag{6}$$

Now the state redefined the state vector as:

$$\boldsymbol{X} = [\, x_1\ x_2\ x_3\ \ x_4\ x_5\ x_6\ \ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ \ x_{12}]^T \tag{7}$$

Showing the inputs vector U, the vector of control inputs is expressed as a function of rotation speed as follows:

$$U = [\, U_1\ U_2\ U_3\ \ U_4\ ] \tag{8}$$

$$\begin{cases} U1 = K\mathrm{p}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U2 = K\mathrm{p}(-\omega_2^2 + \omega_4^2) \\ U3 = K\mathrm{p}(\omega_1^2 - \omega_3^2) \\ U4 = K\mathrm{d}(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \tag{9}$$

Introducing the following constants and state vector, it can express the six second-order differential equations as:

$$\begin{cases} a_1 = \frac{I_{yy}-I_{zz}}{I_{xx}}, a_2 = \frac{J_r}{I_{xx}} \\ a_3 = \frac{I_{zz}-I_{xx}}{I_{yy}}, a_4 = \frac{J_r}{I_{xx}} \\ a_5 = \frac{I_{xx}-I_{yy}}{I_{zz}} \\ b_1 = \frac{I}{I_{xx}}, b_2 = \frac{I}{I_{yy}}, b_3 = \frac{I}{I_{zz}} \end{cases} \tag{10}$$

$$\begin{cases} \ddot{x} = \frac{-U_1}{m}(sinx_1\ sinx_5 + cosx_1\ cosx_5\ sinx_3) \\ \ddot{y} = \frac{U_1}{m}(cosx_1\ sinx_5\ sinx_3 - cosx_5\ sinx_1) \\ \ddot{z} = g - \frac{U_1}{m}(cosx_1\ cosx_3) \end{cases} \tag{11}$$

Where the states are given as:

$$\begin{cases} \dot{x}_1 = \dot{\varphi} = x_2 \\ \dot{x}_2 = \ddot{\varphi} = a_1x_4x_6 - a_2x_4\omega_r + b_1U_2 \\ \dot{x}_3 = \dot{\theta} = x_4 \\ \dot{x}_4 = \ddot{\theta} = a_3x_2x_6 + a_4x_2\omega_r + b_2U_3 \\ \dot{x}_5 = \dot{\psi} = x_6 \\ \dot{x}_6 = \ddot{\psi} = a_5x_2x_4 + b_3U_4 \\ \dot{x}_7 = \dot{z} = x_8 \\ \dot{x}_8 = \ddot{z} = g - \frac{U_1}{m}(cosx_1\ cosx_3) \\ \dot{x}_9 = \dot{x} = x_{10} \\ \dot{x}_{10} = \ddot{x} = \frac{-U_1}{m}(sinx_1\ sinx_5 + cosx_1\ cosx_5\ sinx_3) \\ \dot{x}_{11} = \dot{y} = x_{12} \\ \dot{x}_{12} = \ddot{y} = \frac{U_1}{m}(cosx_5\ sinx_1 - cosx_1\ sinx_3 sinx_5) \end{cases} \tag{12}$$

$$F(\boldsymbol{X}, U) = \begin{bmatrix} x_2 \\ a_1x_4x_6 - a_2x_4\omega_r + b_1U_2 \\ x_4 \\ a_3x_2x_6 + a_4x_2\omega_r + b_2U_3 \\ x_6 \\ a_5x_2x_4 + b_3U_4 \\ x_8 \\ g - \frac{U_1}{m}(cosx_1\ cosx_3) \\ x_{10} \\ \frac{-U_1}{m}(sinx_1\ sinx_5 + cosx_1\ cosx_5\ sinx_3) \\ x_{12} \\ \frac{U_1}{m}(cosx_5\ sinx_1 - cosx_1\ sinx_3 sinx_5) \end{bmatrix} \tag{13}$$

III. CONTROL DESIGN

This section has two parts. part A explains how to design back-stepping controller and part B explains how to design feedback Linearization.

A. *Back-stepping controller*

This part proposes a back-stepping controller for parameter θ, φ and Z to track the desire function. For this purpose, the roll of the quadcopter (φ), the dynamics are defined in below:

$$\begin{aligned} &\dot{x}_1 = x_2 \\ &\dot{x}_2 = a_1x_4x_6 - a_2x_4\omega_r + b_1U_2 \end{aligned} \tag{14}$$

To analyze the stability, a Lyapunov function is defined as below:

$$V = \frac{1}{2}z_1^2 \tag{15}$$

Where the z_1 is an error variable:

$$z_1 = x_{1d} - x_1 \quad (16)$$

The time derivative of Lyapunov function is derived to be:

$$\dot{V}_1 = z_1\dot{z}_1 = z_1(\dot{x}_{1d} - \dot{x}_1) = z_1(\dot{x}_{1d} - x_2) \quad (17)$$

Based on the Krasovskii -LaSalle principle, the system's stability is guaranteed if the time derivate of a positive definite Lyapunov function is negative semi-definite. To achieve so, in this paper, an arbitrary negative function is considered. So, the control signal should be designed to achieve Lyapunov function to satisfy the below equation.

$$\dot{V}_1 = z_1(\dot{x}_{1d} - x_2) \leq -k_1 z_1^2 \quad (18)$$

where $k_1$is a positive constant. To satisfy this inequality, the virtual control input can be chose as

$$x_{2d} = \dot{x}_{1d} + k_1 z_1 \quad (19)$$

Then, new error variable z_2 is defined to be the deviation of the state x_2 from its desired.

$$z_2 = x_2 - \dot{x}_{1d} - k_1 z_1 \quad (20)$$

Rewriting the Lyapunov function based on z_1 and z_2 variables is given:

$$\dot{V}_1 = z_1\dot{z}_1 = -z_1 z_2 - k_1 z_1^2 \quad (21)$$

Note that the presence of term $-z_1 z_2$ cannot lead $\dot{V}_1$to be a negative semi definite function, so to solve this problem in the next step of the back-stepping algorithm. For the next step, V_2 is defined the new Lyapunov function as below:

$$V_2 = V_1 + \frac{1}{2} z_2^2 \quad (22)$$

Time derivate of this function is:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + z_1 z_2 \rightarrow \\ &= -z_1 z_2 - k_1 z_1^2 + z_2(\dot{x}_2 - \dot{x}_{1d} - k_1 \dot{z}_1) \end{aligned} \quad (23)$$

Again, to guarantee the stability of the system, below function is given:

$$W_{2(z)} = -k_1 z_1^2 - k_2 z_2^2 \quad (24)$$

and the inequality is defined:

$$\begin{aligned} \dot{V}_2 &= -z_1 z_2 - k_1 z_1^2 \\ &+ z_2(a_1 x_4 x_6 - a_2 x_4 \omega_r + b_1 U_2 - \dot{x}_{1d} - k_1 \dot{z}_1) \\ &\leq -k_1 z_1^2 - k_2 z_2^2 \end{aligned} \quad (25)$$

Now by solving the inequality, the input control signal is derived.

$$\begin{aligned} U_1 = \frac{1}{b_1}(&-k_2 z_2 + z_1 - a_1 x_4 x_6 + a_2 x_4 \omega_r \\ &+ \ddot{x}_{1d} + k_1 \dot{x}_{1d} - k_1 x_2) \end{aligned} \quad (26)$$

For controlling pitch (θ) and the position Z is the way of calculations is as same as controlling the roll (φ). So the control signal in sequence for the pitch (θ) and the position Z is derived as below:

$$\begin{aligned} U_2 = \frac{1}{b_2}(&-k_4 z_4 + z_3 - a_3 x_2 x_6 \\ &+ a_4 x_2 \omega_r + \ddot{x}_{3d} + k_3 \dot{x}_{3d} - k_3 x_4) \end{aligned} \quad (27)$$

$$\begin{aligned} U_3 = \frac{m}{cosx_1 cosx_3}(&-z_7 + g - \ddot{x}_{7d} - c_7 \dot{x}_{7d} \\ &+ k_7 x_8 + k_8 z_8) \end{aligned} \quad (28)$$

B. Feedback Linearization controller

In this part controller will be designed by using the Input Output Feedback Linearization (IOFL) method. In this method, output will be derivative till the input signal appearance, then by input we linearize the equation and design a controller for linear system, so consider the phi position as an output:

$$y = \varphi \rightarrow \dot{y} = \dot{\varphi} = x_4 \quad (29)$$
$$\ddot{y} = \dot{x}_4 = (a_3 x_2 x_6 + a_4 x_2 \omega_r + b_2 U_3) \quad (30)$$

As it is shown in the above equation, input signal is appeared in the second derivative of output. Now control signal can design to remove nonlinear parameters and control the output.
In order to achieve, control signal is define as below:

$$U_3 = \frac{1}{b_2}(-a_3 x_2 x_6 + a_4 x_2 \omega_r) + V \quad (31)$$

That V is:

$$V = -k_1 \dot{y} - k_0 y \quad (32)$$

By substituting the control signal and after some calculations the output equation is derived as below:

$$\ddot{y} + k\dot{y} + k_0 y = 0 \quad (33)$$

The laplacian of the above equation is:

$$s^2 + k_1 s + k_0 = 0 \quad (34)$$

To guarantee the stability, coefficient k_0 and k_1 should be positive constants that we use fuzzy tuning for the choosing the coefficients.
There is the same method for other two parameters to control. Below figures are shown in order to compare IOFL with the back-stepping control method, reference is the constant number 1.

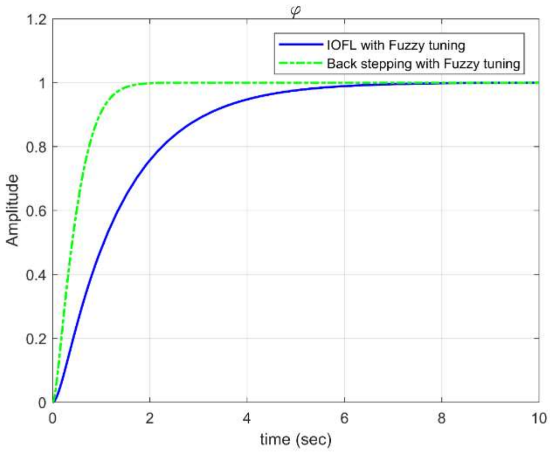

Fig. 2- φ position taraking reference IOFL & back-sepping control

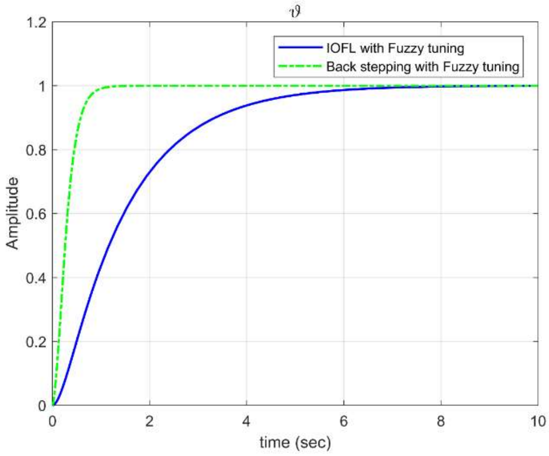

Fig. 3- θ position taraking reference IOFL & back-sepping control

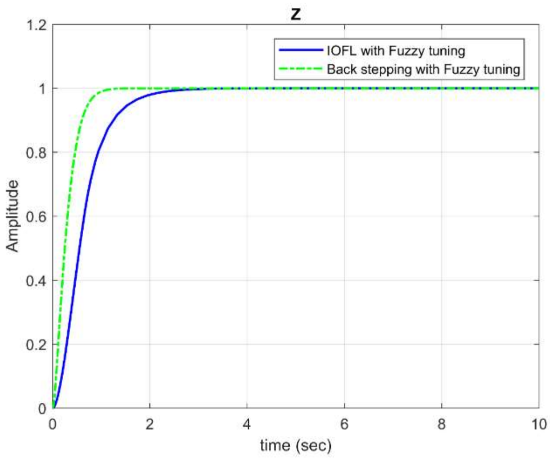

Fig. 4- Z position taraking reference IOFL & back-sepping control

IV.GENETIC ALGORITHM

Here, the numerical method of genetic algorithm (GA) is employed to optimize the designed back-stepping controller's gains. This algorithm is one of the strongest and most popular optimization methods. Here an objective function should be defined and a set of genes is defined according to these parameters. The values of this gene's parameters change internationally, and the objective function is updated in each iteration. The changes of the gene parameters mutate like the human gene and some biologically inspired algorithms will be employed to find the optimum values of the parameters so the objective function would set on its extremum value. This algorithm's implementation in each computational iteration is performed by three main operands, including two genetic operations and one randomly selected one. Since the control target is a biological parameter and follows the biological behaviors, selecting this algorithm increases cancer's optimization process. Here eight parameters are tuned in the GA process, k_1 through k_8. Our main goal is to tune the parameters to minimize the following fitness function which is the Integral of the Squared Error (ISE):

$$J = \int_0^{\tau} \left[e(t) \right]^2 dt \tag{35}$$

Fig. 5 shows their block diagram of the controller using GA for tuning the gains

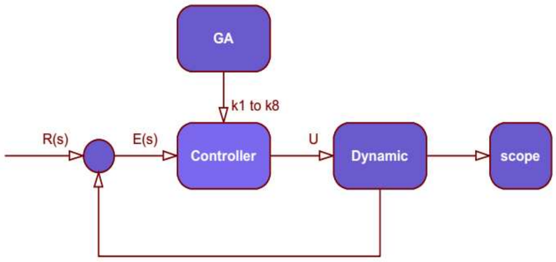

Fig. 5 – Block diagram of the controller using GA for the gains

V. FUZZY TUNNING

This section uses a fuzzy system to achieve dynamic coefficient gain for the controller to have better operating in tracking the desire reference [18]. Fig. 6 shows the block diagram of the controller using adaptive fuzzy control for the gains

For the state φ, first, the input membership functions is described that are the states of the roll and derivate of the roll. The parameters k_1 through k_8 has a similar membership functions. For example, input membership function and the output membership function are the coefficient k_5, k_6 are shown in Fig. 7, and Fig. 8, respectively.

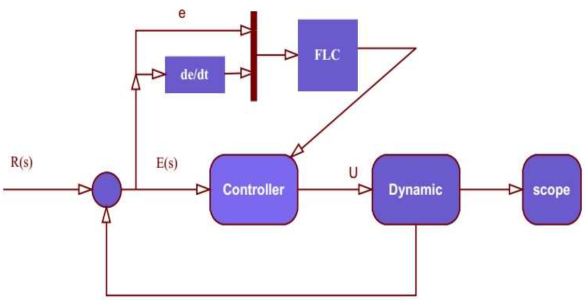

Fig. 6 – Block diagram of the controller using adaptive fuzzy control for the gains

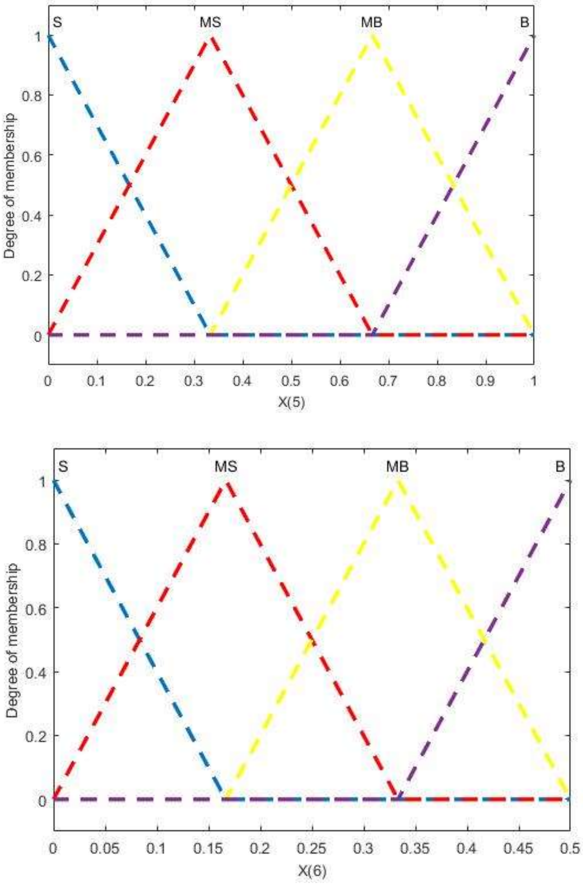

Fig. 7 - Input membership function

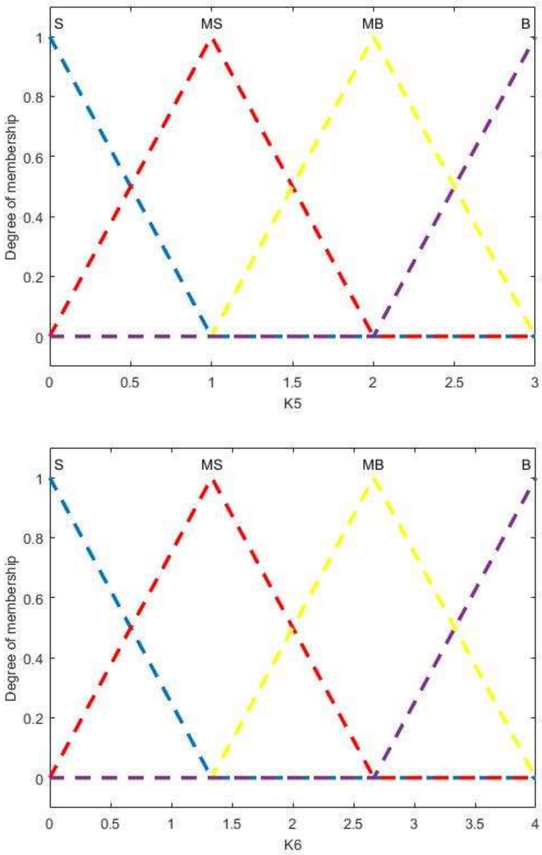

Fig. 8 - Output membership function

The rules of the fuzzy system are set so that if the roll state and its derivate are small (far from the desire), the output of the fuzzy system is big, and whatever input goes bigger, the output gains go smaller, the surface of fuzzy rules are shown in Fig. 9. Table 2 shows the rules of the membership function for fuzzy system

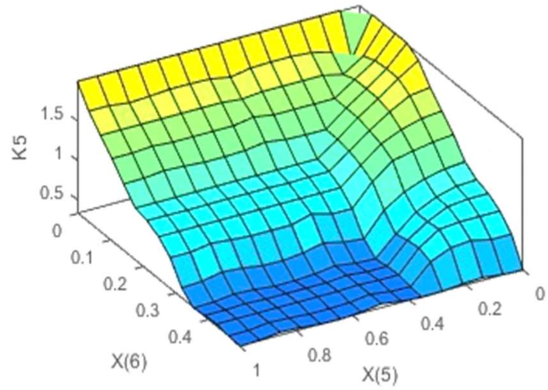

Fig. 9 - Surface of the membership function rules

Table 2 rules of membership function for fuzzy system

x(5) \ x(6)	S	MS	MB	B
S	B	MB	MS	MS
MS	MB	MS	S	S
MB	MS	S	S	S
B	MS	S	S	S

The membership functions and the rules for the gains of fuzzy controller for the pitch (θ) and the position Z are as the same as fuzzy system for the roll (φ).

The benefit of using this method is that the system can adapt gains based on the current position that causes better performance in tracking the desired reference.

VI. RESULT SIMULAION

In this section, the proposed controller and its related process are simulated by conducting some MATLAB simulation scenarios.

Fig. 10 shows the roll state tracking reference with GA Control and adaptive control, and Fig. 11 shows pitch state tracking reference with GA Control and adaptive control. Finally, Fig. 12 shows Z position tracking reference with GA Control and adaptive control. According to the figures, it can be concluded that the adaptive fuzzy controller can operate better to track the reference and because this method receives feedback from states and, Base on this signal, produces a net gain to have better efficiency.

The parameters used in section 2, for the modeling the quadcopter system are described in below:

Table 3 parameters of the quadcopter mathematical model

Parameter	**Name**	**Value**	**Unit**
Ixx	Inertia on x axis	7.5e-3	**$Kg.m^2$**
Iyy	Inertia on y axis	7.5e-3	**$Kg.m^2$**
Izz	Inertia on z axis	1.3e-2	**$Kg.m^2$**

Jr	Rotor inertia	6e-5	**$Kg.m^2$**
l	Arm length	0.23	**m**
g	Gravity constant	9.81	**$m.s^{-2}$**
m	Mass	0.650	**Kg**
kp	Thrust coefficient	3.13e-5	**$N.s^2$**
kd	Drag coefficient	7.5e-7	**$N.m.s^2$**

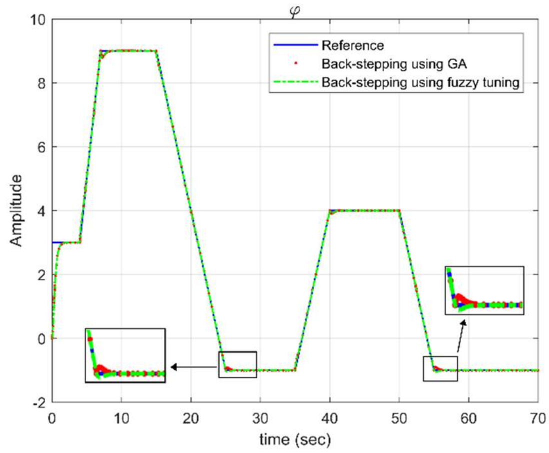

Fig. 10 – roll state tracking reference with GA Control and adaptive control

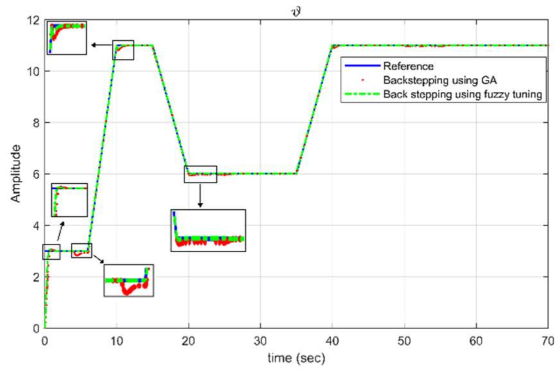

Fig. 11 – pitch state tracking reference with GA Control and adaptive control

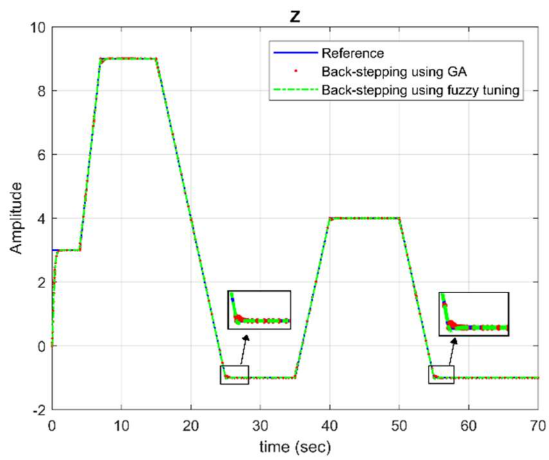

Fig. 12 – Z position tracking reference with GA Control and adaptive control

VII. Conclusion

This paper studied the dynamics of quadcopter and its mathematical model. Then the back-stepping controller was designed to control the states of the quadcopter. While the controller design parameters or controller gains are usually chosen by trial and error; in this paper, two ways were proposed to calculate the optimal controller gains to achieve better efficiency. The first one was using GA that gives constant gains, and the second way was using an adaptive fuzzy method that gives different gains based on the states' positions. We will optimize the control gains according to a cost function in the neural network optimization method and obtain fixed gains. In the fuzzy method, the obtained gains are variable with time, and the simulation results show that the fuzzy response is better than the neural one. Simulation results showed the efficiency of the proposed methods.

References

[1] M. Labbadiand, and M. Cherkaoui, “Robust integral terminal sliding mode control for quadrotor UAV with external disturbances,” International Journal of Aerospace Engineering, Vol. 5, August 2019.

[2] M. Herrera, A. P. Gómez, W. Chamorro, and O. Camacho, “Sliding mode control: An approach to control a quadrotor,” Pacific Conference on Computer Aided System Engineering, pp. 314-319, July 2015.

[3] L. Besnard, Y. B. Shtessel, S. Member, and B. Landrum, “Control of a quadrotor vehicle using sliding mode disturbance observer,” in 2007 American Control Conference, pp. 5230-5235, July 2007.

[4] K. Runcharoon, and V. Srichatrapimuk, "Sliding mode control of quadrotor," In 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), pp. 552-557, May 2013.

[5] V. K. Tripathi, L. Behera, and N. Verma, “Design of sliding mode and backstepping controllers for a quadcopter,” 2015 39th National Systems Conference (NSC), pp. 1-6, December 2015.

[6] E. H. Zheng, J. J. Xiong, and J. L. Luo, “Second order sliding mode control for a quadrotor UAV,” ISA transactions, vol. 53, pp.1350-1356, July 2014.

[7] I. Palunko, and R. Fierro, “Adaptive control of a quadrotor with dynamic changes in the center of gravity,” International Federation of Automatic Control, vol. 44.1, pp.2626-2631, September 2011.

[8] S. Tomashevich, and A. Belyavskyi, “Passification based simple adaptive control of quadrotor,” IFAC,Papers OnLine, vol.49, pp. 281-286, Jan 2016.

[9] A. Mottahedi, and A. Akbarzadeh, “Adaptive robust sliding mode control of quadrotor in the presence of wind/disturbance,” Modares Mechanical Engineering, vol. 16, pp.95-102, October 2016.

[10] B. J. Imran, and A. Yesildirek, “Nonlinear Composite Adaptive Control for Quadrotor,” 2013.

[11] M. Schreier, “Modeling and adaptive control of a quadrotor,” 2012 IEEE international conference on mechatronics and automation, pp. 383-390, August 2012.

[12] S. E. I. Hasseni, and L. Abdou,” Decentralized PID control by using GA optimization applied to a quadrotor,” Journal of Automation Mobile Robotics and Intelligent Systems, vol. 12, pp. 33-44, 2018.

[13] J. Cuellar, Y. Arturo, R. d. J. R. Troncoso, L. M. Velazquez, and R. A. O. Rios, “PID-controller tuning optimization with genetic algorithms in servo systems,” International Journal of Advanced Robotic Systems, vol. 10, September 2013.

[14] Khuwaja, Komal, I. C. Tarca, and R. C. Tarca. “PID Controller Tuning Optimization with Genetic Algorithms for a Quadcopter,” Recent Innovations in Mechatronics, vol. 5, pp.1-7, 2018.

[15] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, “Control and navigation framework for quadrotor helicopters,” Journal of Intelligent & Robotic Systems, vol. 70, pp.1-12, April 2013.

[16] D. Lee, H. J. Kim, and S. Shankar, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," International Journal of control, Automation and systems, vol. 7, pp. 419-428, June 2009.

[17] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, H. Li, “Control and navigation framework for quadrotor helicopters,” Journal of intelligent and robotic systems, vol. 70, pp.1-12, April 2013.

[18] N.Kanagasabai, N.Jaya, "Fuzzy gain scheduling of PID controller for a MIMO process," International Journal of Computer Applications, vol. 91, April 2014.
[19] D. Gautam, and C. Ha, "Control of a quadrotor using a smart self-tuning fuzzy PID controller," International Journal of Advanced Robotic Systems, vol. 10, November 2013.
[20] P. J. Gawthrop, and P. E. Nomikos, "Automatic tuning of commercial PID controllers for single-loop and multiloop applications," IEEEControl Syst. Mag, vol. 10, pp. 34-42, June 1990.
[21] M. Santos, V. Lopez, and F. Morata, "Intelligent fuzzy controller of a quadrotor," International Conference on Intelligent Systems and Knowledge Engineering (ISKE). pp 141-146, November 2010