



# Design and Implementation of Autopilot System for Quadcopter

Firas Abdullah Thweny Al-Saedi<sup>1</sup>, Rawad A. Sabar<sup>2</sup>

<sup>1,2</sup>Computer Engineering Department,  
Al-Nahrain University,  
Baghdad, Iraq

**Abstract** - In this paper, a full approach of modeling and control of a four rotor unmanned air vehicle (UAV) known as quadcopter aircraft is presented. The main features of the quadcopter's structure are across-shaped aerial vehicle, cross intersection columns leading to composition of two diagonals of a square. Four motors have been attached to the end of the cross intersection. The body angular rate ( $\omega$ ) of all motors is the key to the control of the quadcopter. The dynamics system modeling is examined to understand the force growth in play. The control algorithm is represented by the smart Fuzzy controller address the stability and robustness. The genetic algorithm (GA) is involved in constructing the flight controller due to its effectiveness and efficiency in optimization of the control systems. The control strategies represented by the smart Fuzzy controller was linked with the GA to obtain further improvements in the system response in both the transient and steady states. The GA is used as an optimization technique because it is becoming a very common tool for resolving optimization problems and it is employed to effectively secure optimal solutions. The Matlab Simulink is a great tool which was used to test the efficiency and the accuracy of the developed system.

**Keywords** - Quadrotor, Fuzzy Controller, Unmanned Aerial Vehicle

## I. INTRODUCTION

UAVs are an important part of scientific study in both the military and space studies. As a substitute for human piloted vehicles, they are advantageous to protect human life in multiple dangerous environments. Their reliabilities in tough circumstances are much higher than their counter parts [1]. UAV can fly autonomously or it can be remotely controlled based on program downloaded into on-board controller. In general, UAVs can be classified basically into two categories, fixed wing and rotatory wing. Rotatory wing UAVs have some advantages over fixed wing UAVs in that they can take off and land vertically, and that they also can keep on their position at a fixed point in 3D space. Maintaining a position at a fixed coordinate axes is called hovering [2].

Modeling is the scientific interpretation of the derived equations, constraints, and logic rules, while simulation is the verification and practical emulation of the model. Modeling and Simulation are considered as primary fundamentals in control engineering which are used in course of designing a new system, improving an already existing system or a model for which a controller has to be designed [3].

Control of a quadcopter is not a trivial task for the following reasons: high nonlinearity response, high maneuverability, intensely coupled multivariable and under actuated condition with six degrees of freedom while there are only four motors. Researchers have designed and implemented numerous quadcopter controllers such as PID controllers, Fuzzy controllers, sliding mode controllers, neuro-Fuzzy controllers [4].

The implementation of the Fuzzy logic controller has become very popular and attractive field for the systems with non-linear response in recent couple of decades, and it has been presented on numerous dynamical systems. The Fuzzy logic has also been used in the quadcopter controller implementation. In the more recent projects conducted by N. I. Vitzilaios, and N. C. Tsourveloudis, they used a Fuzzy controller for achieving the altitude response and hovering control of quadcopter [5]. The traditional control methods use linear practice that is fit with demands of the linear systems behaviors only. Hence, the Fuzzy control which is nonlinear controller is suitable for the nonlinear system control [6].

Fuzzy controller optimization is a vital step in the implementation of a Fuzzy control model. One can see that the Fuzzy model with simple two inputs will have more than thousands of possible arrangements of Fuzzy rules. Even though applying common methods discussed previously along with human knowledge would normally be close to the desired response, they still face the fact that the picked Fuzzy rules are normally not the best tuned and meet with project demands [7].

Genetic algorithm (GA) is becoming a very common tool for resolving optimization problems. It has been employed to effectively secure optimal solutions for a set of problems (e.g. operations research, mathematical problems, hybrid techniques, medical problems, image processing, and even political science problems) [8]. Genetic algorithms are built on the mechanics of natural selection and natural genetics. With GA's abilities, it has been developed to be a novel optimization algorithm [9]. Genetic algorithms have been used for producing and/or tuning membership functions of Fuzzy rules. Kane [10] has tuned membership functions of the Fuzzy rules and Nomura et al. [11] computed Fuzzy partition of input spaces by genetic algorithms.

## II. QUADROTOR'S DYNAMIC MODELLING

The quadcopter is hovering in three dimensional space, two different coordinate systems will appear. One is the body coordinate system, indexed 'b', which is influenced by the motors. The other is the navigation frame, indexed 'n', where forces like gravitation have affected as shown in Figure 1. The body coordinate system will move along with the quadcopter, while the navigation coordinate system is the reference point for the quadcopter. The reference coordinate system can be located anywhere, but has to be fixed once the quadcopter starts flying. In avionics the Z-axis is normally pointing towards the earth. The quadcopter rotation is carried out around its own axes with an angular velocity [12] [13].

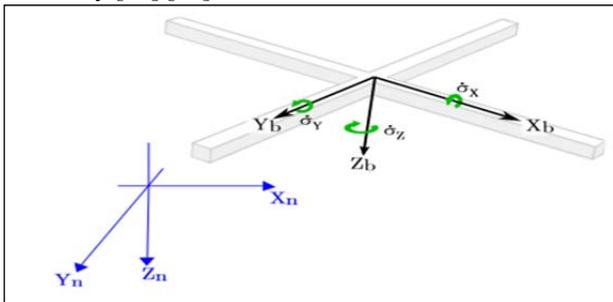


Figure 1: Coordinate Systems

The quadcopter's coordinate system has to be linked to the navigation coordinates. The way achieved is by the implementation of Euler angles. There are three Euler angles, known as roll, pitch and yaw as shown in Figure 2. This symbolization is often used in avionics [13] [14].

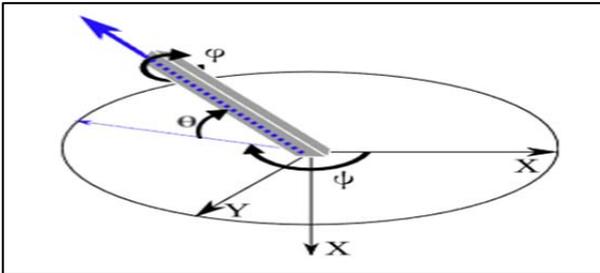


Figure 2: Euler angles

The quadcopter's orientation should be rotated to determine how much force the motors have to put out to keep it hovering. The rotations can be done one at a time. Equation 1 indicates the matrices that can be used to rotate about a single axis [3].

$$\begin{aligned}
 R_{\sigma_x}(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \\
 R_{\sigma_y}(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \\
 R_{\sigma_z}(\varphi) &= \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{1}$$

For the calculation of the total conversion from the body frame to the navigation frame, we can multiply together all the three matrixes. The complete rotation matrix is clearly annotated in Equation 2.

$$C_b^n = R_{\sigma_x}(\phi) R_{\sigma_y}(\theta) R_{\sigma_z}(\varphi) \tag{2}$$

The conversion from body to navigation frame is obtained by the calculated matrix. This can be done by transposing the result matrix  $C_b^n$  to be  $C_n^b$  by performing  $C_n^b = C_b^n^T$ .

Usually UAVs, including quadcopter, are characterized by under actuated and coupled dynamics. Under actuated, because they have six degrees of freedom (3 rotational and 3 translations) but have four actuated DOF and that the translational and rotational dynamics are coupled. The equations of motion are ruled by Newtonian mechanics and their calculation are done through the appropriate choice of modeling technique i.e. Newton-Euler formulation [15].

The quadcopter is very well modeled and represented with four rotors in a cross symmetric configuration. As shown in Figure 3, the front and the rear motors rotate counter-clockwise, while the left and the right ones turn clockwise. This structure of opposite pair's directions removes the tail rotor needs (needed instead in the standard helicopter structure) [3].

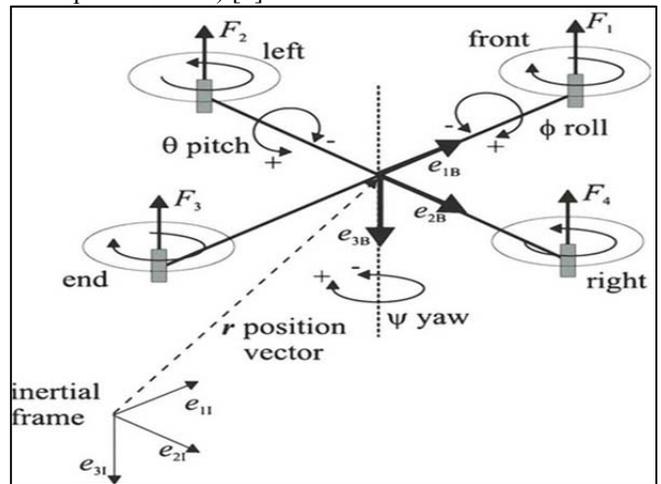


Figure 3: Conceptual Diagram of a Quadcopter

Four basic movements achieved by quadcopter. Altitude is achieved by increasing (or decreasing) the entire propeller speeds by the same amount. It yields to a vertical force body-fixed frame which ascends or descends the quadrotor quadcopter. Figure 4 shows the throttle command on a quadcopter sketch [12] [15].

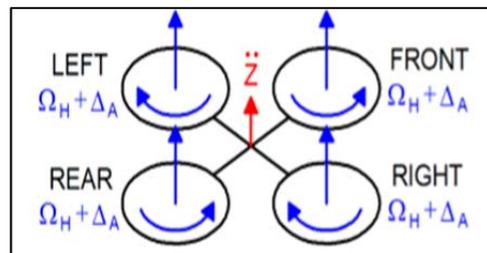


Figure 4: Throttle movement

Throttle is achieved by increasing (or decreasing) the left propeller speed and by decreasing (or increasing) the right one. It results in a torque with respect to the  $X_b$ -axis which makes the quadcopter turn. The overall vertical thrust remains the same as in hovering; hence this action leads only to roll angle acceleration (in first approximation).

Figure 5 shows the roll command on a quadcopter sketch [12] [16].

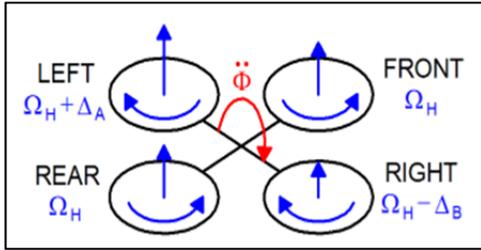


Figure 5: Roll movement

The pitching is very similar to the roll and is achieved by increasing (or decreasing) the rear propeller speed and by decreasing (or increasing) the front one. It yields torque with respect to the  $Y_b$  axis which makes the quadcopter turn. The overall vertical thrust remains the same as in hovering; hence this action leads only to pitch angle acceleration (in first approximation). Figure 6 shows the pitch command on a quadcopter sketch [12] [16].

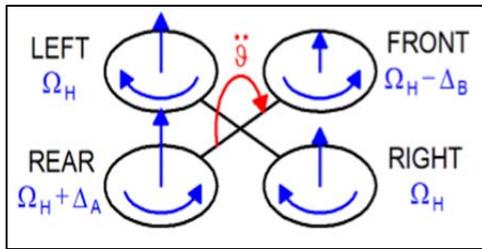


Figure 6: Pitch movement

Yaw is achieved by increasing (or decreasing) the front-rear propellers' speed and by decreasing (or increasing) that of the left-right couple. It leads to a torque with respect to the  $Z_b$  axis which makes the quadcopter turn. The yaw movement is generated as a result of the fact that the left-right propellers rotate clockwise while the front-rear ones rotate counter-clockwise. Hence, when the overall torque is unbalanced, the quadcopter turns on itself around  $Z_b$ . The total vertical thrust is the same as in hovering; hence this command leads only to Yaw angle acceleration. Figure 7 shows the yaw command in a quadcopter sketch [12] [16].

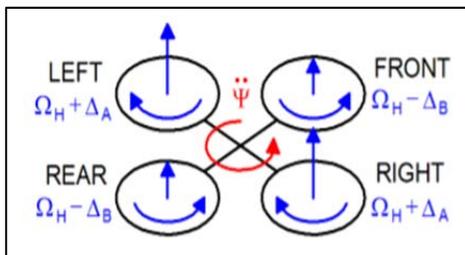


Figure 7: Yaw movement

The motion equations of physical dynamic systems (quadcopter) are described by the Newton-Euler formalism. It has been realized that the quadcopter system is a six DOF system defined with twelve states.

Six out of twelve states represent the attitude of the system (Figure 3). These comprise the angles  $(\phi, \theta, \psi)$  and angular rates  $(p, q, r)$  around the three orthogonal body

axes. The remaining six states are the three positions and three linear velocities of the center of mass of the quadcopter with respect to a fixed reference frame (earth fixed frame) [17]. Rotation matrix is used to guarantee a correct application for Newton's laws of motion since almost all measures are in with respect to the body-fixed frame except gravity, thus, this matrix is used to relate the motions expressed in body-fixed frame with respect to the inertial frame [18].

Linear velocities along body axes can be transformed into inertial frame using transformation matrix as in Equation 2 [3] [19].

$$\dot{X} = \frac{d}{dt} [x_E y_E z_E]^T = C_b^n [u \ v \ w]^T \quad (3)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = C_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4)$$

The state equations are summarized as follow:

$$X = a + b + c \quad (5)$$

where:

$$\begin{aligned} a &= \cos\theta \cos\phi \cdot u \\ b &= (\sin\theta \sin\phi \cos\phi - \cos\theta \sin\phi) \cdot v \\ c &= (\sin\theta \sin\phi + \cos\theta \sin\theta \cos\phi) \cdot w \end{aligned}$$

$$\dot{y} = d + e + f \quad (6)$$

where:

$$\begin{aligned} d &= \sin\theta \cos\theta \cdot u \\ e &= (\cos\theta \cos\phi + \sin\theta \sin\theta \sin\phi) \cdot v \\ f &= (\cos\theta \sin\theta \sin\phi - \sin\theta \cos\phi) \cdot w \end{aligned}$$

$$\dot{z} = -\sin\theta \cdot u + \sin\theta \cos\theta \cdot v + \cos\theta \cos\theta \cdot w \quad (7)$$

Angular rates along body axes can be transformed into Euler rates using the transformation matrix T given below:

$$\frac{d}{dt} [\phi \ \theta \ \psi]^T = T [p \ q \ r]^T \quad (8)$$

$$T = \begin{bmatrix} 1 & \tan\theta \cdot \sin\phi & \tan\theta \cdot \cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sec\theta \cdot \sin\phi & \sec\theta \cdot \cos\phi \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (11)$$

$$\dot{\phi} = p + q \cdot \tan\theta \cdot \sin\phi + r \cdot \tan\theta \cdot \cos\phi \quad (12)$$

$$\dot{\theta} = q \cdot \cos\phi - r \cdot \sin\phi \quad (13)$$

$$\dot{\psi} = q \cdot \sec\theta \cdot \sin\phi + r \cdot \sec\theta \cdot \cos\phi \quad (14)$$

### III. CONTROL MODELING

The researches pertaining to the modeling and controlling of the quadcopter (UAV) have increased rapidly in the recent decades. A number of these investigations might be summarized in following lines; Altuğ et al modeled the quadcopter by using the Euler-Newton method and worked at the stabilization based on vision and feedback tracking control [20]. Suter et al. examined the quadcopter control by using the image based visual servo practice [21]. F. Shepherd et al. constructed robust neuro-control for a micro Quadcopter [22]. Earl et al estimated the quadcopter attitude by using Kalman filter [23]. Lee et al implemented the quadcopter control in term of the feedback linearization versus adaptive sliding mode [24]. Hazry et al used a hybrid FLC as a controller for a quadcopter [25]. Astha et al. implemented a Fuzzy logic controller for a quadcopter and compared result responses with those of conventional PID controller [6].

The intent of this research is to come up with a self-tuning Fuzzy controller so that it will achieve further improvements in the system response performance in both the transient and/or steady states in comparison with the system response gained when either the conventional PID or the normal Fuzzy controller has been implemented. GA-based Fuzzy controller is proposed for controlling both the quadcopter's attitude and position during hovering. The inputs scale factors of Fuzzy controller are achieved by using the GA. A smart optimization technique and special tuning of the Fuzzy's controller parameters were suggested to reduce the computational time that designer might face during the construction. Simulations are performed on Matlab Simulink to demonstrate the efficiency of the suggested method [2].

The most important concepts in quadcopter dynamical model can be identified through Equation 15 and Equation 16. The first one shows how the quadcopter accelerates according to the basic movement commands given [16] [19].

$$\begin{cases} \ddot{X} = (\sin\phi\sin\theta + \cos\phi\sin\theta\cos\phi) \frac{U_1}{m} \\ \ddot{Y} = (-\cos\phi\sin\theta + \sin\phi\sin\theta\cos\phi) \frac{U_1}{m} \\ \ddot{Z} = -g + (\cos\theta\cos\phi) \frac{U_1}{m} \\ \dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} q r - \frac{J_{TP}}{I_{xx}} q \Omega + \frac{U_2}{I_{xx}} \\ \dot{q} = \frac{I_{xx} - I_{zz}}{I_{yy}} q r + \frac{J_{TP}}{I_{yy}} q \Omega + \frac{U_3}{I_{yy}} \\ \dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} q q + \frac{U_4}{I_{zz}} \end{cases} \quad (15)$$

Equation 16 shows the basic movements are linked to the propellers' squared speed [16] [19].

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 = lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (16)$$

Equation 17 shows the actual quadcopter dynamics has been used in the proposed control [18].

$$\begin{cases} \ddot{Z} = -g + (\cos\theta\cos\phi) \frac{U_1}{m} \\ \ddot{\phi} = \frac{U_2}{I_{xx}} \\ \ddot{\theta} = \frac{U_3}{I_{yy}} \\ \ddot{\psi} = \frac{U_4}{I_{zz}} \end{cases} \quad (17)$$

It is valuable to mention and to notice in the latter system that the angles and their relative time derivatives do not rely on translation movements as it is indicated through Equation 17. While the translations movements depend on the angles components, it can ideally be imagined that the overall system composed of two subsystems, the angular rates and the linear movements as shown in Figure 8 [19].

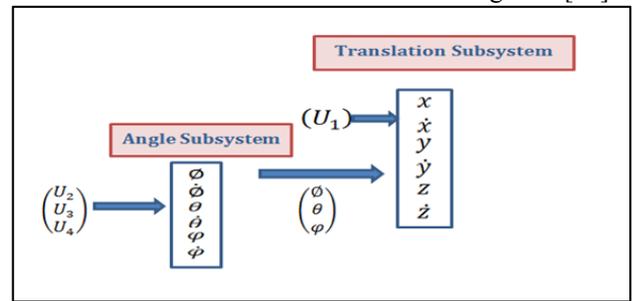


Figure 8: Connection of rotations and translations subsystems

The suggested controller can be divided in four basic components as shown in Figure 9. The two inputs fed into the flight controller are the actual response and the controller's outputs are used to determine the actuated signal of the four rotors.

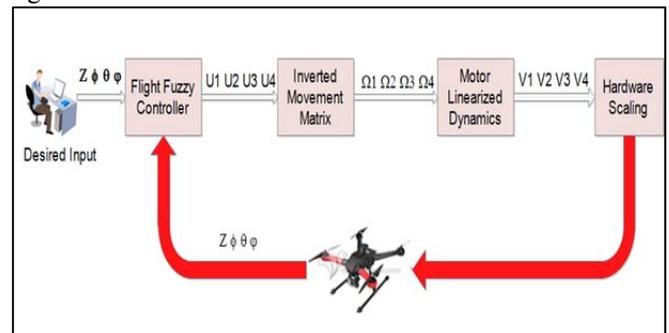


Figure 9: Control block diagram

Fuzzy controller system is considered a type of nonlinear function interpolator which is presented for delivering the required control of variable structure systems. The strongest points in using this type on controller are to obtain stability and the guarantee robustness against parameters, lines and load uncertainties. There are two general procedures in generating Fuzzy controller model. The first one requires expert information in the model's implementation, while in the second the Fuzzy mechanisms are readjusted automatically by the system based on the information retrieved from representative numerical samples [3].

Referring to Equation 18 below, let  $(U_i)$  denote the motor value for the front, right, back, and left motors, respectively where  $(i = 1, 2, 3, 4)$ . Then, the thrust and torque applied to the quadcopter by motor can be stated as [2]:

$$\begin{cases} \Omega_1^2 = \frac{1}{4b} U_1 - \frac{1}{2bl} U_3 - \frac{1}{4d} U_4 \\ \Omega_2^2 = \frac{1}{4b} U_1 - \frac{1}{2bl} U_2 + \frac{1}{4d} U_4 \\ \Omega_3^2 = \frac{1}{4b} U_1 + \frac{1}{2bl} U_3 - \frac{1}{4d} U_4 \\ \Omega_4^2 = \frac{1}{4b} U_1 + \frac{1}{2bl} U_2 + \frac{1}{4d} U_4 \end{cases} \quad (18)$$

$$T_{mot} = b \times U_i \quad (19)$$

$$\tau_{mot} = d \times U_i \quad (20)$$

where  $b$  and  $d$  are predefined motor parameters. This yield:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = G \times \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (21)$$

With

$$G = \begin{bmatrix} b & b & b & b \\ 0 & -l \times b & 0 & l \times b \\ l \times b & 0 & -l \times b & 0 \\ -d & d & -d & d \end{bmatrix} \quad (22)$$

This knowledge will lead us an effective guideline during the implementation of the main blocks in direct Fuzzy logic controller as shown in Figure 10.

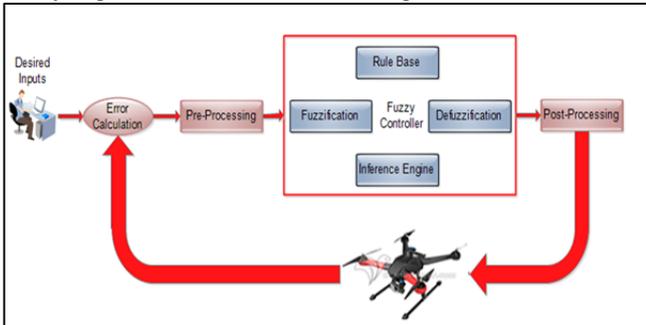


Figure 10: Control scheme

Three Fuzzy controllers are constructed to control the quadcopter’s roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) angles, denoted by  $FLC_\phi$ ,  $FLC_\theta$ , and  $FLC_\psi$ , respectively. The last controller  $FLC_z$ , is designed to control the quadcopter altitude. The internal structure of the flight controller is clearly depicted in Figure 11.

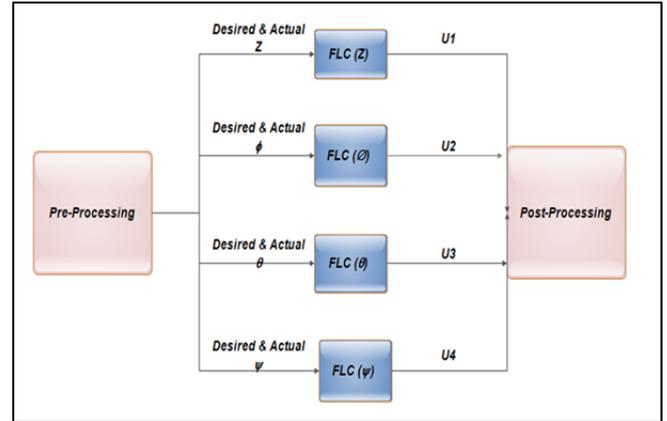


Figure 11: Fuzzy controllers for four basic motions

In this research, the four Fuzzy controllers are identical and each has two inputs which are used first by the fuzzification interface and one output is produced after defuzzification. Inputs into the system are as follows: (i) the error  $e = (\cdot) = (\cdot)_d - (\cdot)$ , which is the difference between the desired signal  $(\cdot)_d$  and its actual value  $(\cdot)$  received as a feedback from the quadcopter, and (ii) the error rate  $\dot{e}$ . The two inputs are normalized to the interval  $[-2, +2]$ . The output ( $U$ ) is normalized to the interval  $[-15, 15]$ .

It is important to refer to the scaling factor insertion into the developed design as a mid-step between inputs value and fuzzification. The main reason for using this mechanism is to improve the Fuzzy logic controller response and select values of these scales will be covered in further detail in genetic algorithm section. Figure 12 is a snapshot of the Simulink model used in the Fuzzy controller system.

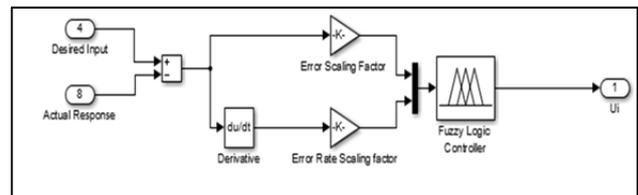


Figure 12: Simulink model of inserting the scaling factors for each FLC

The membership functions were implemented by using a combination of triangular, trapezoidal, and Gaussian shapes. The linguistic variables used for error (Figure 13a) are described by five membership functions: negative/positive big, negative/positive medium, and zero. The linguistic variables used for error rate (Figure 13b) are described by three membership functions: negative/positive and zero. The linguistic variables used for error rate (Figure 13c) are described by three membership functions: negative/positive big, negative/positive medium, and zero.

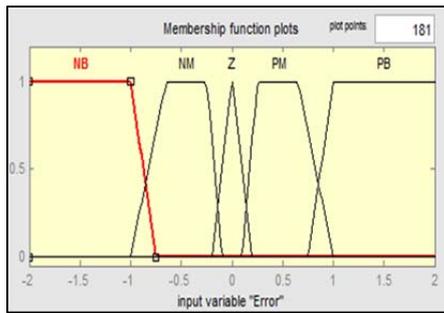


Figure 13a: Input variable error e

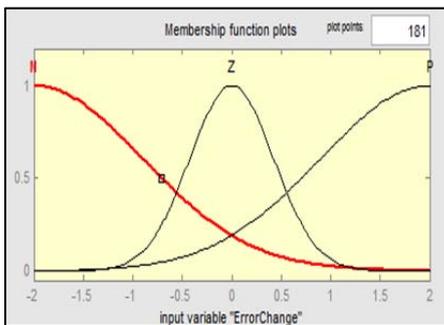


Figure 13b: Input variable error rate  $\dot{e}$

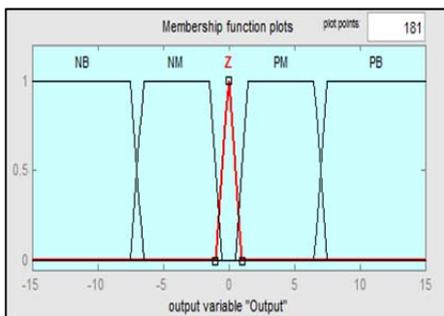


Figure 13c: Output variable  $U$

The input and output rules for the Fuzzy logic controller are put up in the format of if-then statements and are based on experimental and human expert with navigating through an environment (similar to driving a car). In this project, the rules used are 15 rules in total for each FLC. The complete rules-base is listed in Table 1 using the linguistic variables previously outlined as listed below:

Table 1: Rules Base

Rules Base					
Error \ Error Rate	NB	NM	Z	PM	PB
N	NB	NM	NM	Z	PM
Z	NB	NM	Z	PM	PB
P	NM	Z	PM	PM	PB

As noticed, Fuzzy controller optimization is a vital step in the implementation of a Fuzzy control model. One can see that the Fuzzy model with simple two inputs will have more than thousands of possible arrangements of Fuzzy rules. It is a known fact that developing of the Fuzzy model isn't an easy task especially when it comes with complex real world problems. The rules of Fuzzy system were derived from intuition, human experts and knowledge, and a trial and error method in the most implemented Fuzzy systems. Each one of Fuzzy logic controller will have more than  $2 \times 1029$  possible groups of Fuzzy rules. [7].

Recently, several methods were proposed for creating the best Fuzzy rules from the experimental data automatically. General method was used and adopted by Wang and Mendel to produce Fuzzy rules from the experimental data [26]. The self-learning method was proposed by Jang [27] and Berenji and Khedkar [28] have proposed self-learning methods for tuning membership functions of Fuzzy rules. The automatic producing of the Fuzzy rules will lead to lose one of the most important characteristics in Fuzzy logic which is the ability of Fuzzy logic controller of describing the system in a linguistic term. This will enable the design Fuzzy controller with more human-like reasoning, especially with the Fuzzy's statement rules.

#### IV. PARTICLE SWARM OPTIMIZATION

Genetic algorithm (GA) is becoming a very common tool for resolving optimization problems. It has been employed to effectively secure optimal solutions for a set of problems (e.g. operations research, mathematical problems, hybrid techniques, medical problems, image processing, and even political science problems) [8]. Genetic algorithms are built on the mechanics of natural selection and natural genetics. With GA's abilities, it has been developed to be a novel optimization algorithm [9]. Genetic algorithms have been used for producing and/or tuning membership functions of Fuzzy rules. Kane [10] has tuned membership functions of the Fuzzy rules and Nomura et al. [11] computed Fuzzy partition of input spaces by genetic algorithms

The genetic optimization is set in place of the tedious process of trial and error for better combination of Fuzzy controller response. Characteristic data are used from real world or recommended resources for genetic optimization. Studies are performed based on Fuzzy system models. Both newly suggested algorithms, chromosome-length independent mutation operation and fitness mapping mechanism, are described and discussed in coming section [11].

Genetic algorithm consists of two main classes. The first one is referred as "Chrom", covering all the needed information about a chromosome which is belonging to an individual member of the population. The second class is referred as "Population", which consists of all individuals as its members [7].

The typical GA optimization flowchart is shown in Figure 14. The process starts with initialization of the static variable. All the static variables and dynamic variables are

triggered in the GA constructor. Fitness function of initial chromosomes is computed with the overridden fitness computations, include the penalty factor computation [11] [29].

Also, it is most important to refer to the crucial part that fitness function play in GA's implementation. The purpose of using fitness function in The GA is to assign a score (fitness) for each chromosome in the current population. The chromosome's fitness depends on how good that individual (chromosome) resolves the problem at hand. The real dimensional function is maximized, as a quick example for giving better understanding of fitness function application, as shown in the expression below:

$$(y) = y + \sin(32y). \quad 0 \leq y < \pi \quad (23)$$

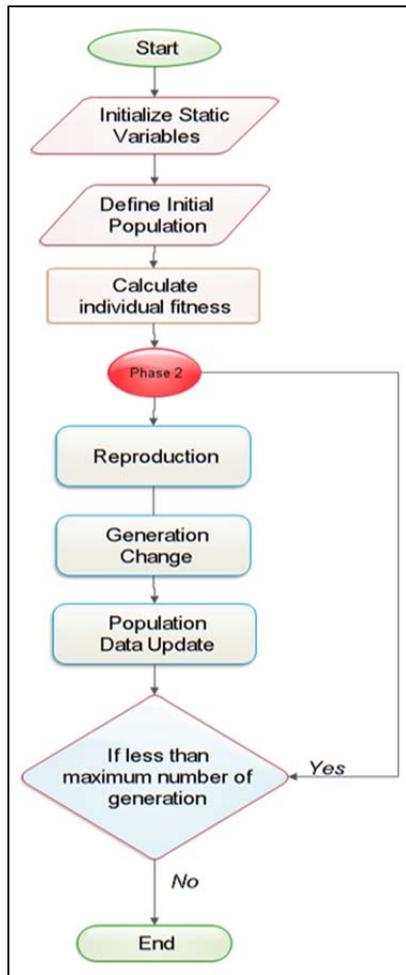


Figure 14: Genetic optimization flowchart.

Here the nominated solutions are values of y that can be coded into bit strings representing the real numbers. The basic explanation of the fitness calculation is to translate a set of bit string x into a real number y and then assess the function at that value. The string's fitness is a function value at that point. In conclusion, the fitness value of an individual is the value of the fitness function for that individual. The best fitness value for a population is the smallest fitness value for any individual in the population [7] [29] [30].

**V. GA TUNED FLC IMPLEMENTATION**

The GA is used to optimize the scaling factors of each FLC that is used in our controller design as shown in Figure 12, while the values and shapes of the member functions and if-then statement (rules base) are predetermined in advance and are kept on fixed through whole the process. The Fuzzy controller used is Mamdani type which contains the same blocks as shown in Figure 10 [31]. Block diagram of the GA based Fuzzy controller is shown in Figure 15.

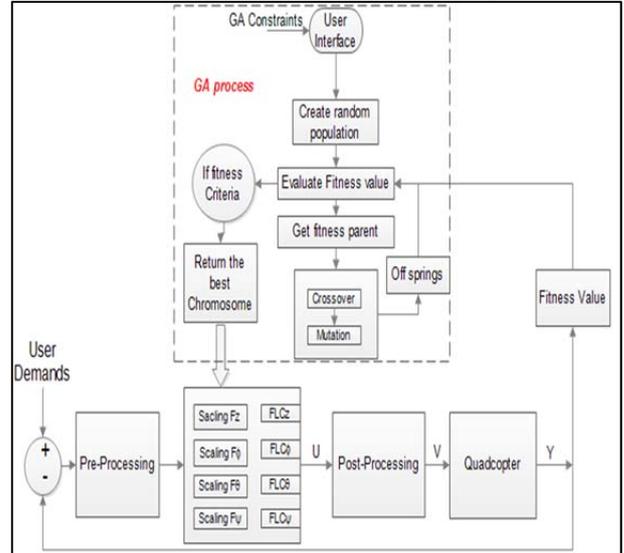


Figure 15: Block diagram of the GA based Fuzzy controller

In the designed model, it can easily notice where exactly the GA interacts with the designed controller. It is solely focused on identifying the optimum values of the scaling factors from the input side. The developed code used to perform this process is written and used in Matlab Simulink.

In the proposed algorithm, the scaling parameters  $K_{z,\phi,\theta,\varphi}$  and  $(K_{z,\phi,\theta,\varphi})$  of the Fuzzy controller have been optimized. In GA, there are two approaches that have been identified in the determination of the scaling factors in the optimal design of Fuzzy controller. In the first method, the scaling factors of the optimal Fuzzy controller have been computed sequentially. This process is in match with the self-tuning adaptive system used in adaptive control systems. The second one is based on the simultaneous optimization of the scaling factor. The Fuzzy controller parameters are not fully independent of each other. In conclusion, the second approach for tuning the scaling factors is the optimum scenario to be used. However, it is necessary to keep in mind that the multi-parameters of the Fuzzy controller system are hard to be achieved by using second method [7].

**VI. SIMULATION AND RESULTS**

The presented GA-based Fuzzy controller system and its implementation on the quadcopter are performed by using Matlab Simulink which is considered as a starting point for ongoing research and development of Fuzzy control of UAVs. It enables the researcher to carry on low cost validation of the proposed control algorithms in real flight

experiments without the need for significant knowledge of hardware design or programming language, because Simulink provides the well-known environment for graphical programming and control system design. Together with MATLAB, it is a very powerful tool for mathematic development and control system design.

The four identical Fuzzy logic controllers are implemented using Fuzzy Logic Toolbox as shown in Figure 16. The input and output values and membership functions are set accordingly. The inputs and outputs of the Fuzzy controller are pre- and post-processed, respectively. The Fuzzy controllers used in the configuration are shown in Figure 11.

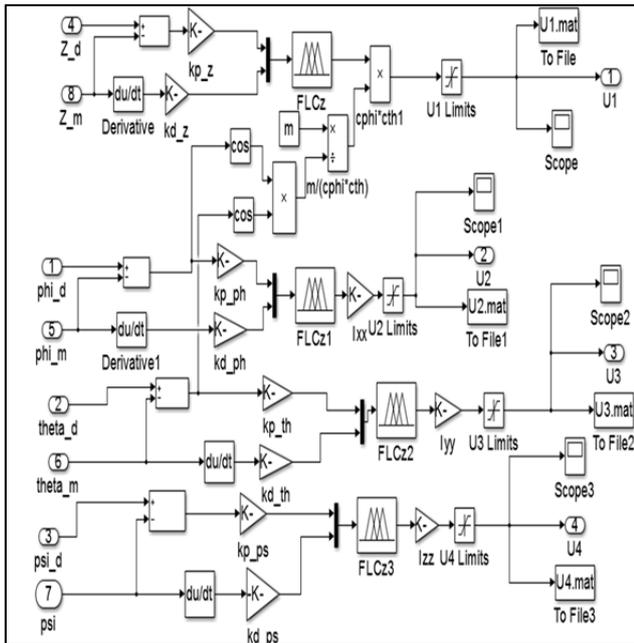


Figure 16: Fuzzy Logic Controllers

The system's initial states are set to zero and a number of simulations were conducted to identify the suitable range of the upper and lower limits of GA that meet the required performance of the Fuzzy logic flight controllers along with inference engines

The user-desired inputs are the desired altitude coordinate ( $z$ ) with respect to the inertial frame, and the desired angular movement represented by roll, pitch, and yaw. The yaw stabilization was easy to identify because the errors calculated in the yaw angle do not make any linear acceleration in hovering condition. Evermore, the dynamic range is much wider than the roll or pitches one. The yaw can range between  $-180$  and  $+180$  degrees while the roll and pitch shows lower flexibility less than 10 degrees. Therefore, good performance in both dynamic and static prospect is required.

Scaling factors found using GA are then applied using a test system based on the step response of the quadcopter. The user-desired inputs chosen for the angular movement represented by roll, pitch, and yaw were (0.3, 0.2, and 0.01) rad and started at (0.3, 0.2, and 0.1) second respectively, while the altitude ( $Z$ ) is chosen to be 10 meter and it is started at the origin. Figure 17, Figure 18, Figure 19, and

Figure 20 reflect the quadcopter response as it is controlled with scaling factor found by GA. It can be noted that the performance of the roll, pitch, yaw and altitude under two different inputs of the quadrotor is excellent and satisfactory. The user-desired input reached an acceptable rising time with small overshoot. In conclusion, one can notice fast response which is a crucial aspect in control of flying objects.

Figure 21 shows the control vectors applied to the quadcopter dynamics system.  $U1$  control representing the total thrust is applied to the quadcopter and is physically limited to the value of 15.7N as according to the actual platform specifications. The minimum value of the thrust is zero. Controls  $U2$ ,  $U3$  and  $U4$  represent the roll, pitch and yaw angles all are restricted between  $-1$  and  $1$  apart from Yaw control where is bounded between  $-0.2$  and  $0.2$ .

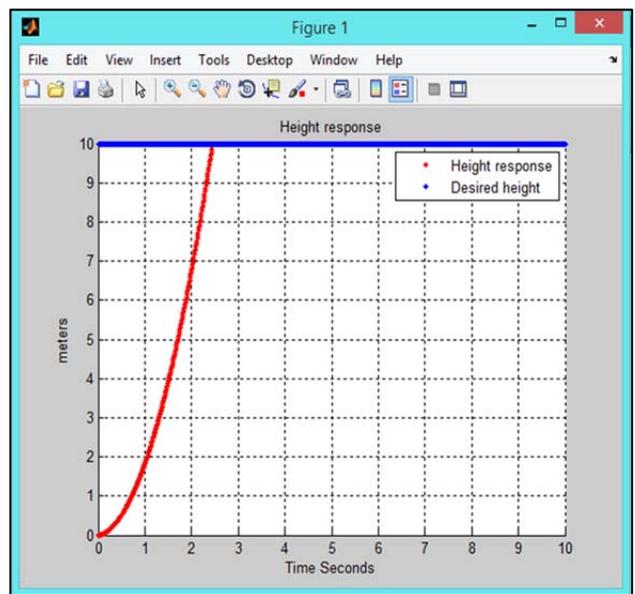


Figure 17: Quadcopter altitude when  $z = 10$  m

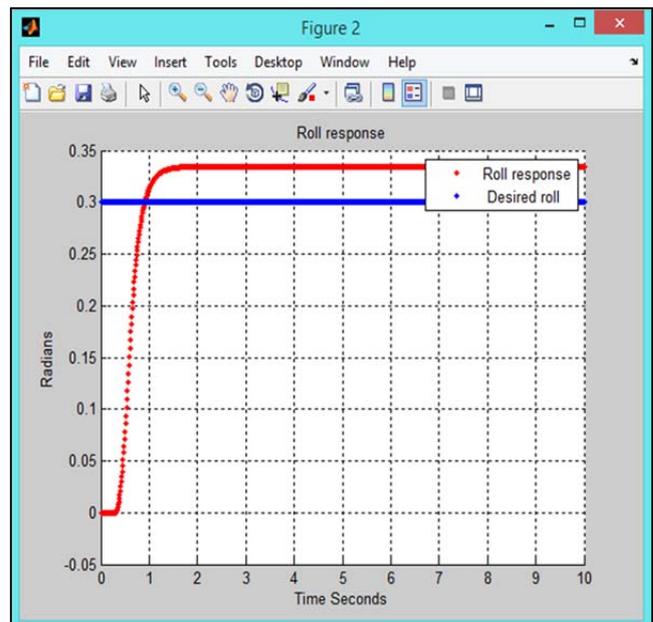


Figure 18: Roll response when  $\phi = 0.3$

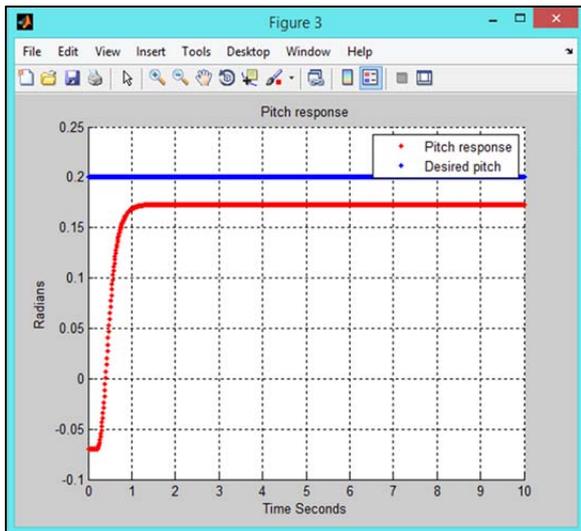


Figure 19: Pitch response when  $\theta = 0.2$

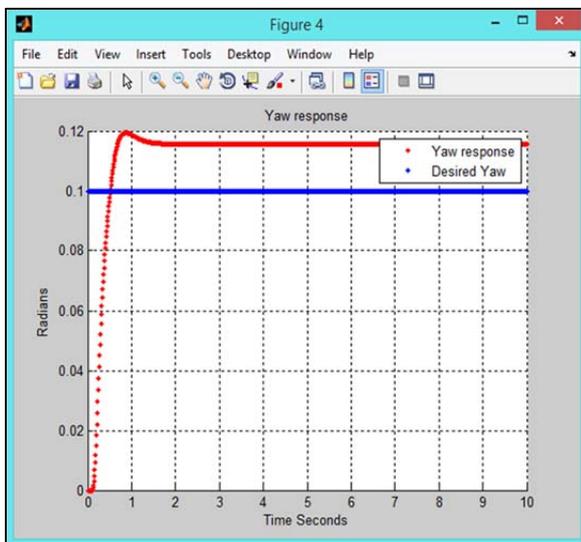


Figure 20: Yaw response when  $\varphi = 0.1$

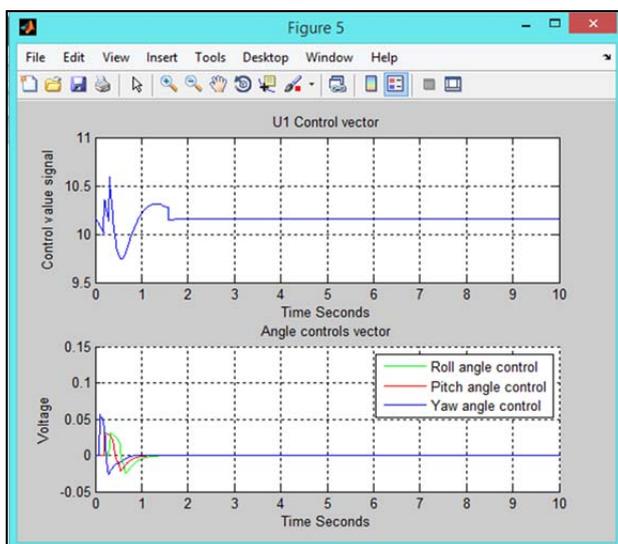


Figure 21: Control vectors signals

## CONCLUSIONS

The intelligent Fuzzy controller based on self-tuning parameters has been verified to be an effective and suitable tool for controlling complex nonlinear-MIMO systems (coupling inputs and decoupling outputs) such as quadcopter. The experiments were conducted in Matlab Simulink. The results prove a successful control response especially with using the optimization technique (GA). When compared to other conventional techniques applied for a similar purpose Syed Ali Raza. (2009), Bouabdallah et al. (2004), show the developed methodology has demonstrated a higher robustness.

The use of GA in the optimization process of the Fuzzy controller indicates significant improvements in the system performance in both the transient and steady state responses. It is also noticed that the used sequential optimization scenario leads to satisfactory results, consumes less time, and it can be considered as a desired preference in such applications. The use of GA with employing dynamic crossover and mutation rates gives more power and great opportunity to further improve the Fuzzy controller parameters in terms of scaling factors. Also, the chromosome size is reduced. Moreover, the suggested method reduces the process time to minutes.

## REFERENCES

- [1] **Lee, Keun.** *Design of Overall Stabilized Controller for Quad-rotor.* Kangar, Perlis, Malaysia: Centre of Excellence for Unmanned Aerial Systems (COEUAS), University Malaysia Perlis, 2013.
- [2] **Syed Ali Raza, Wail Gueaieb.** *Intelligent Flight Control of an Autonomous Quadrotor.* Canada: University of Ottawa, 2009.
- [3] **Ruth Tesfaye.** *Modeling and Control of a Quad-rotor Unmanned Aerial Vehicle at Hovering Position.* Addis Ababa, Ethiopia: University of Addis Ababa, 2012.
- [4] **Deepak Gautam, Cheolkeua Ha.** *Control of Quadcopter Using a Smart Fuzzy-PID Controller,* University of Ulsan, 2013.
- [5] **N. I. Vitzilaios, N. C. Tsourveloudis.** *Test Bed for Unmanned Helicopters.* Journal of Intelligent and Robotic Systems, vol. 54, pp. 769-794, May 2009.
- [6] **A. Sharma, A. Barve.** *Controlling of Quadrotor UAV Using PID Controller and Fuzzy Logic Controller.* International Journal of Electrical, Electronic and computer Engineering, vol. 1, no. 2, pp. 38-41, 2012.
- [7] **Wong SV, Hamouda AMS, Ismail N.** *Optimization of Fuzzy Rules Design using Genetic Algorithm.* Malaysia: University Putra Malaysia, 2000.
- [8] **Goldberg DE.** *Genetic Algorithms in Search, Optimization, and Machine Learning,* Reading, MA: Addison-Wesley, 1989.
- [9] **Abdel-Razzak.** *Novel Bioinspired Stochastic Tuning of a Quadrotor PD Controller.* Australian: Control Conference (AUCC), 2012.
- [10] **Karr CL.** *Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm.* Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, 13-16 July 1991:450-7.
- [11] **A Self-tuning Method of Fuzzy Reasoning by Genetic Algorithm.** [Online] Available at <http://books.google.iq/>.
- [12] **Syed Ali Raza.** *Design and Control of a Quadrotor Unmanned Aerial Vehicle.* Canada: University of Ottawa, 2010.
- [13] **Schmidt, Michael David.** *Simulation and Control of a Quadrotor Unmanned Aerial vehicle.* s.l. : University of Kentucky, 2011.
- [14] **Bak, Thomas.** *Modeling of Mechanical Systems.* Aalborg, Denmark: Department of Control Engineering, Aalborg, 2002.
- [15] **David Allerton.** *Principles of Flight Simulation.* Wiley, 2009.

- [16] **T. Bresciani.** *Modeling, Identification and Control of a Quadrotor Helicopter.* Sweden: Lund University, Department of Automatic Control, 2008.
- [17] **Venditelli, Marilena.** *Quadrotor Modeling.* Rome: Sapienza, University di Roma, 2011.
- [18] **S. Bouabdallah.** *Design and Control of Quadrotors with Application to Autonomous Flying.* University Aboubekr Belkaid, 2007.
- [19] **Qyvind Magnussen, Kjell Eivind.** *Modeling, Design and Experimental Study for a Quadcopter System Construction.* University of Agder, 2011.
- [20] **E. Altuğ, James P, Camillo J. Taylor.** *Vision Based Control of Model Helicopters.* USA: University of Pennsylvania, 2006.
- [21] **D. Suter, T. Hamel, R. Mahony.** *Visual Servo Control Using Homography Estimation for the Stabilization of an X4-Flyer.* Proc. Of the 41th IEEE conf. on Decision and control, PP. 2872-2877, 2002.
- [22] **Jack F. Shepherd, Kagan Tumer.** *Robust Neuro-Control for a Micro Quadrotor.* USA: Oregon State University, 2010.
- [23] **M. G. Earl, R. D'Andra.** *Real-time Attitude Estimation Techniques Applied to a Four Rotor Helicopter.* Proc. of IEEE conf. on Decision and Control, PP. 3956-3961, 2004.
- [24] **D. Lee, H. J. Kim, S. Sastry.** *Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter.* International Journal of Control, Automation, and Systems, vol. 7, no. 3, pp. 419-428, 2009.
- [25] **H. Desa, S. Faiz Ahmed.** *Design and Implementation of Adaptive Hybrid Control Algorithm Design for Attitude Stabilization of Quadrotor (UAV).* Malaysia: University Malaysia Perlis, 2013.
- [26] **Jang JRS.** *Self-learning Fuzzy Controllers Based on Temporal Back Propagation.* IEEE Trans Neural Networks 1992; 3(5):714–23.
- [27] **Jang JRS.** *Self-learning Fuzzy Controllers Based on Temporal Back Propagation.* IEEE Trans Neural Networks 1992; 3(5):714–23.
- [28] **Berenji HR, Khedkar P.** *Learning and Tuning Fuzzy Logic Controllers through Reinforcements.* IEEE Trans on Neural Networks 1992; 3(5):724–40.
- [28] **Azar Zafari.** *Developing a Fuzzy Inference System by Using Genetic Algorithm and Expert Knowledge.* Iran: University of Twente, 2014.
- [29] **Gerardo Acosta, Eli 'as Todorovich.** *Genetic Algorithms and Fuzzy Control: a Practical Synergism for Industrial Applications.* Argentina, 2003.
- [30] **O. Aydogdu, R. Akkaya.** *Design of a Real Coded GA Based Fuzzy Controller for Speed Control of a Brushless DC Motor.* Turkey: Faculty of Engineering and Architecture, 2003.