Performance Evaluation of a PID and a Fuzzy PID Controllers Designed for Controlling a Simulated Quadcopter Rotational Dynamics Model

Laith Jasim Saud
Assistant Professor
Control & Systems Eng. Dept.
University of Technology
Laithjasim15@yahoo.com

Rasha Shehab Mohammed
M.Sc. Student
Control & Systems Eng. Dept.
University of Technology
rasha_shehab@yahoo.com

ABSTRACT

This work is concerned with designing two types of controllers, a PID and a Fuzzy PID, to be used for flying and stabilizing a quadcopter. The designed controllers have been tuned, tested, and compared using two performance indices which are the Integral Square Error (ISE) and the Integral Absolute Error (IAE), and also some response characteristics like the rise time, overshoot, settling time, and the steady state error. To try and test the controllers, a quadcopter mathematical model has been developed. The model concentrated on the rotational dynamics of the quadcopter, i.e. the roll, pitch, and yaw variables. The work has been simulated with "MATLAB". To make testing the simulated model and the controllers more realistic, the testing signals have been applied by a user through a joystick interfaced to the computer. The results obtained indicated a general superiority in performance for the Fuzzy PID controller over the PID controller used in this work. This conclusion is based by the following figures: $70\%, 70\%, \text{and } 52\%$ lesser ISA for the roll, pitch, and yaw consequently, $70.5\%, 70.5\%, 56.4\%$ lesser IAE for the roll, pitch, and yaw consequently, $53\%, \text{and } 80.6\%$ lesser rise time and settling time for the roll and pitch consequently, and 77% lesser settling time for the yaw. Moreover, the FPID gave zero overshoot versus $18\%, 18\%$, and 25% in the PID case for the roll, pitch, and yaw consequently. Both controllers gave zero steady state error with close rise times for the yaw. This superiority of the FPID controller is gained as the fuzzy part of it continuously and online adapts the parameters of the PID part.

Key words: unmanned aerial vehicle quadcopters, quadcopter modeling, PID controller, fuzzy PID controller, performance indices.

تقييم اداء مسيطر تناسبي تكاملي تفاضلي و مسيطر تناسبي تكاملي تفاضلي ضبابي مصممان للتحكم بنموذج للديناميكا الدورانية لمروحية رباعية محاكى رقميا

الاستاذ المساعد الدكتور ليث جاسم سعود
قسم هندسة السيطره والنظم
الجامعة التكنولوجي

رشا شهاب محمد
قسم هندسة السيطره والنظم
الجامعة التكنولوجية

الخلاصة

يتضمن هذا البحث تصميم نوعين من المسيطرات، اولهما المسيطر التناسبي التكاملي التفاضلي، وثانيهما والمسيطر التناسبي التكاملي التفاضلي الضبابي، مستخدمان لتحقيق طيران مستقر لمروحية رباعية. لقد تم تنغيم المسيطرات المصممة واختبارها و مقارنة اداؤها باستخدام معياري اداء احدهما تكامل مطلق الخطأ وثانيهما تكامل مربع الخطأ، إضافة الى استخدام بعض خصائص الاستجابة مثل زمن الصعود و تجاوز الحد وزمن الاستقرار وقيمة الخطأ عند الاستقرار. لتجربة واختبار المسيطرات، تم بلورة

نموذج رياضي للمروحية الرباعية و الذي يركز تحديدا على الديناميكا الدورانية للمروحية. وقد تم محاكاة المنظومة رقميا باستخدام الكيان البرمجي (MATLAB). ولجعل عملية اختبار النموذج المحاكى والمسيطرات المصممة اكثر واقعية فقد تم ادخال اشارات التحكم عبر عصا تحكم تربط مع الحاسوب. لقد عكست النتائج المستحصلة من المحاكاة تفوق واضح في الاداء للمسيطر التناسبي التكاملي التفاضلي الضبابي على المسيطر التناسبي التكاملي التفاضلي. إن هذا الاستنتاج مبني على المعطيات التالية: تكامل مربع الخطأ أقل بنسبة 70%، 70%، 52% لكل من زوايا العطوف والخطران والانعراج على التوالي، وتكامل مطلق الخطأ أقل بنسبة 70.5%، 70.5%، 56.4% لكل من زوايا العطوف والخطران والانعراج على التوالي، وزمني صعود و استقرار أقل بنسبة 53% و 80.6% لكل من زوايا العطوف والخطران على التوالي، وزمن استقرار أقل بنسبة 77% لزاوية الانعراج. وإضافة لهذا فان نسبة الطفرة مع المسيطر التناسبي التكاملي التفاضلي الضبابي كانت صفرا مقابل 18%، 18%، و 25% مع المسيطر التناسبي التكاملي التفاضلي ولكل من زوايا العطوف والخطران والانعراج على التوالي. أما بالنسبة لخطأ حالة الاستقرار فكانت صفراً لكلا المسيطرين ولكل الحالات. وأما بالنسبة لزمن الصعود لزاوية الانعراج فان المسيطرين أعطيا نتائج متقاربة. إن الافضلية في الاداء التي ابداها المسيطر التناسبي التكاملي التفاضلي الضبابي على المسيطر التناسبي التكاملي التفاضلي متأتية من قيام الجزء الضبابي فيه بالتعديل المستمر وخلال عمل المنظومة لقيم معاملات الجزء التناسبي التكاملي التفاضلي.

الكلمات الرئيسية: المروحيات الرباعية من دون طيار، نمذجة المروحية الرباعية، المسيطر(التناسبي+التكاملي+المشتقة)، المسيطر(التناسبي+التكاملي+المشتقة) الضبابي، مؤشرات الاداء.

1. INTRODUCTION

In control, modeling represents a very important issue as it can serve the process of system testing, analysis, as well as the design process. It reduces effort, cost, problems, and provides the possibility of getting fair solutions in considerably shorter times. Mathematical modeling of systems provides the possibility of numerical simulation with computers with its well-known capability and software versatility which eases and speeds up the test, analysis, and design of well controlled systems that obey the targeted performance objectives. That is why a lot of efforts have been done through simulation work. Some of the researches focused on modeling the quadcopter **Benic, et al., 2016**. Some concentrated on developing and evaluating different controllers **Anjum, et al., 2016, Ribas** and **Engel**, **2014. Abbasi,** and **Mahjoob, 2013. Bouadi, et al., 2007.** and **Harrag, et al., 2012.** And some dealt with the two issues, i.e. the model and the controllers, **Mahdi, 2006**. **Bouabdallah, 2007. Brito, 2009.** and **Basta, 2012.** In addition, practical implementation issues got part of the research interest, **Hystad, 2015.** as well as the trajectory tracking problems, **Luis, 2016.**

In this work, a mathematical model will be derived specifically for the quadcopter rotational dynamics. The derivation will be carried out using Newton Euler equations. The quadcopter model will be then, through simulation and software, used in the process of testing the quadcopter action and designing and trying different controllers that can help in achieving the goal of flying the quadcopter accurately and stably. For the control part, two controllers are used, the Proportional+ Integral+Derivative (PID) controller, and the fuzzy PID controller. The controllers are to be designed to achieve the goal of flying the quadcopter accurately and stably.

For evaluating the controllers' performance and to compare among them, four response characteristics will be used which are the rise time, the overshoot, the settling time, and the steady state error. That is besides two performance indices which are the ISE and the IAE.

2. THE QUADCOPTER MODEL

Before derivation, some important relevant terms and concepts will be defined or clarified.

2.1 Definitions

i-The coordinates system: There are two coordinate systems; Earth frame and Body frame, **Mahdi, 2006**. These two frames are shown in **Fig.1**

ii- Kinematics description model: In this description, any point in the body frame can be defined with the following relation:

$$S = R * v$$

where; S represents the displacement vector in the Earth frame, R represents the rotation matrix, and v represents the displacement vector in the body frame. The velocity equation is found through derivation of the displacement equation.

iii - Dynamics description model: the dynamics of a quadcopter can be derived by Lagrange Euler equations and Newton Euler equations. The dynamics of the quadcopter take into consideration the mass and the moments of inertia about the axes.

2.2 Notes about modeling the quadcopter

The motion of the quadrotor can be divided into two subsystems; rotational subsystem (roll, pitch and yaw) and translational subsystem (z, x, and y positions), and as shown in **Fig.2**, It is worth it to mention that the rotational subsystem is fully actuated while the translational subsystem is under-actuated **Nagaty, et al., 2013**.

The aim here will be the model derivation of the rotational subsystem of the quadcopter dynamics, that is, developing the equations of the roll (ϕ), pitch (θ), and yaw (ψ) for the quadcopter. In the derivation, the quadcopter body and earth fixed frames considered in the derivation are shown in **Fig.1**. In this figure, B represents the body fixed frame, and E represents the earth fixed frame. The quadcopter orientation in space is given by a rotation from B to E, and as clarified in **Fig.1**.

2.3 Modeling with Newton-Euler equation

The mathematical model for the quadcopter dynamics and motion could be developed either using either Euler Lagrange equations or Newton-Euler equations. The results obtained in both cases will be the same, but the procedure with the Newton-Euler equations is shorter, and that is why it will be used in this section.

In order to derive a model for the quadcopter motion, the various moments and forces on the quadcopter must be taken into consideration. Both forces and moments on the quadcopter can be derived using Newton-Euler equations and as given in Eq. (1).

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} mI_{3*3} & 0 \\ 0 & I_{3*3} \end{bmatrix} * \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} w \times mv \\ w \times Iw \end{bmatrix} \tag{1}$$

where; m represents the mass of the quadcopter, $\dot{v}$ represents the linear acceleration vector, $\dot{w}$ represents the angular acceleration vector, F represents the force vector acting on the quadcopter, Ʈ represents the torque vector acting on the quadcopter, all with the respect to the body frame, and I_{3*3} represents the moments of inertia matrix and is given as:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2)$$

The equations for the quadcopter motion can be developed using Eq. (1), with the individual moments and forces described for all degrees of freedom **Schmidt, 2011.**
From Eq. (1), the equivalent moments' equation is given by:

$$Ʈ = I\dot{w} + (w \times Iw) \quad (3)$$

$$\begin{bmatrix} Ʈ_\phi \\ Ʈ_\theta \\ Ʈ_\Psi \end{bmatrix} = \begin{bmatrix} Ixx & 0 & 0 \\ 0 & Iyy & 0 \\ 0 & 0 & Izz \end{bmatrix} * \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\Psi} \end{bmatrix} + [(\dot{\phi}\imath + \dot{\theta}j + \dot{\Psi}k) \times (Ixx\,\dot{\phi}\imath + Iyy\dot{\theta}j + Izz\dot{\Psi}k) \quad (4)$$

where $i \times i$ =0, $j \times j$ =0 and $k \times k$ =0, the moments for the roll, pitch, and yaw are obtained:

$$\begin{bmatrix} Ʈ_\phi \\ Ʈ_\theta \\ Ʈ_\Psi \end{bmatrix} = \begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\Psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta}\dot{\Psi}(I_{zz} - I_{yy}) \\ \dot{\phi}\dot{\Psi}(I_{xx} - I_{zz}) \\ \dot{\phi}\dot{\theta}(I_{yy} - I_{xx}) \end{bmatrix} \quad (5)$$

the moments acting round x, y, and z axis are given by:

$$Ʈ_x = bl\,(-\Omega_1{}^2 + \Omega_2{}^2 + \Omega_3{}^2 - \Omega_4{}^2) \quad (6)$$
$$Ʈ_y = bl\,(-\Omega_1{}^2 - \Omega_2{}^2 + \Omega_3{}^2 + \Omega_4{}^2) \quad (7)$$
$$Ʈ_z = d\,(-\Omega_1{}^2 + \Omega_2{}^2 - \Omega_3{}^2 + \Omega_4{}^2) \quad (8)$$

where; b represents the thrust factor of the quadcopter, d represents the drag factor of the quadcopter, l represents the distance from the center of gravity to the center of a motor, and ($\Omega_1{}^2, \Omega_2{}^2, \Omega_3{}^2$ and $\Omega_4{}^2$) represents the squared angular velocities of the propellers. The given moment equations are for "X" configuration adopted in this work. Some researches, like **Gopalakrishnan, 2016**, adopts the "+" configuration.

The configuration used for the quadcopter is the (X) configuration and as depicted in **Fig.3**. Considering this configuration, the speed equations are given by the following equations:

$$\Omega_1{}^2 = Throttle - Ur - Up - Uy \quad (9)$$
$$\Omega_2{}^2 = Throttle + Ur - Up + Uy \quad (10)$$
$$\Omega_3{}^2 = Throttle + Ur + Up - Uy \quad (11)$$

$$\Omega_4{}^2 = Throttle - Ur + Up + Uy \tag{12}$$

where; $Ur, Up,$ and Uy represents the control signals for each of the roll, pitch, and yaw motion. These equations take the correction command for the roll, pitch, yaw and throttle and combine them in a way to allow all corrections be sent to the motors. An important point to state here is that the signals of all parts in all of the equations depend on each axis reference as well as where each motor is located **Ribas,** and **Engel**, **2014.**

The gyroscopic effect resulting from the propellers rotation for roll is defined as:

The gyroscopic effect for roll $= -Jr * wy(-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)$

and, the gyroscopic effect resulting from the propellers rotation for pitch is defined as:

The gyroscopic effect for pitch $= Jr * wx\ (-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)$

where; wx, wy represents the body axis angular rate, and Jr represents the rotor inertia of the quadcopter. And considering that:

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\Psi \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{13}$$

The following equation can be obtained:

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\Psi} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} \dot{\theta}\dot{\Psi}(I_{zz} - I_{yy}) \\ \dot{\phi}\dot{\Psi}(I_{xx} - I_{zz}) \\ \dot{\phi}\dot{\theta}(I_{yy} - I_{xx}) \end{bmatrix} \tag{14}$$

Then, the dynamics of the subsystem can be written as follows:

$$\ddot{\phi} = [\frac{\dot{\theta}\ \dot{\Psi}(Iyy - Izz)}{I_{xx}}] + \frac{\tau_x}{I_{xx}} - \frac{Jr * wy * \ (-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)}{I_{xx}} \tag{15}$$

$$\ddot{\theta} = [\frac{\dot{\Psi}\dot{\phi}(Izz - Ixx)}{I_{yy}}] + \frac{\tau_y}{I_{yy}} + \frac{Jr * wx * (-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)}{I_{yy}} \tag{16}$$

$$\ddot{\Psi} = [\frac{\dot{\theta}\ \dot{\phi}(Ixx - Iyy)}{I_{zz}}] + \frac{\tau_z}{I_{zz}} \tag{17}$$

Although this work is based on the dynamics given by Eqs. (15) to (17), it is important to state that many researchers, like **Kotarski, et al. 2016.** and **Bresciani, 2008.** simplify the dynamics by keeping the terms with the moments and ignoring the others. Such simplifications are based on justifications which in turn are based on assumed conditions. For example, ignoring coupling terms is justified by assuming the motion of the quadrotor to be close to the hovering condition, which means small angular changes occur (especially for roll and pitch). And so, these terms can be simplified because they are smaller than the main ones. In reality, to work close to the hovering condition can help a lot in analysis and design. One other simplification example based on the "close to hovering" condition, is that the angular accelerations which are referred to the angles of the quadrotor measured in its fixed frame, can be referred directly to the Euler angle accelerations that are referenced to the earth frame.

3. THE QUADCOPTER MODEL WITH THE CONTROLLERS

To test the quadcopter rotational dynamics and control its rotational variables $\phi, \theta,$ and ψ, the closed loop control system shown in **Fig.4** has been suggested.

Before going into the controllers details, some important points will be clarified about the system:

- The system outputs or controlled variables are $\phi, \theta,$ and ψ.
- There are three input signals to the system which are ϕ_d, θ_d and ψ_d. These signals represent the desired values for the outputs
- The throttle signal is intended to play the role of the control signal responsible for altitude control. And as the simulation is intended for the quadcopter rotational dynamics, the throttle value is kept constant throughout the simulation.
- During testing the quadcopter model and the controllers used, the input signals ($\phi_d, \theta_d$and ψ_d) will be input to the computer by a user through a joystick connected to the computer. This arrangement is used to resemble the actual case when a person is controlling a real quadcopter through a joystick.
- The (X) configuration will be used for the quadcopter and as depicted in **Fig.3**.

3.1 The PID and the Fuzzy PID Controllers

In this part, two controllers will be suggested for controlling the quadcopter. A justification for this suggestion will be given, and an analysis for its performance will be covered, and also their feasibility will be discussed.

The PID control is one of the early used control methods. Its first version was an analog pneumatic one, and its latest version is the digital PID that is implemented as software. The PID with the proportional, derivative, and integral manipulation of the error signal, represents a mixture that aims at fast response with minimum overshoot and minimum if not null steady state error **Ribas,** and **Engel**, **2014.**The reason for the wide and surviving use of PID is because it offered an easy structure that was simple to understand and operate with, and the more important is that it does well with a lot of systems **Basta, 2012.** Not for just this reason the PID has been adopted in this research for quadcopter fly control, but for many other reasons, like:

- It is still an efficient controller that can meet, when tuned, the design objectives (not for just the quadcopter, but for a lot of other systems).
- It has some sort of robustness.
- It can be tuned manually or with many procedural methods.
- It is easy to implement by software in real time applications.
- Its execution time is small which makes it suitable for real time applications with strict timing conditions.
- The huge number of researches and researchers, it is considered as a reference to compare the performance of other controllers with it.

To do well, the PID controller must be tuned. Different methods exist for tuning like the manual, Tyreus-Luyben, close loop Ziegler-Nichols, open loop Ziegler-Nichols, damped oscillation, Cohen-

Coon, and C-H-R **Shahrokhi,** and **Zomorrodi, 2010**. Besides these methods, there are newer ways that depend non classical optimization methods, and the GA algorithm is just one of many examples.

The approach of the tuning methods mentioned above is to tune the controller parameters and then use them in the real time work of the system. A different magnificent way of tuning is introduced with the fuzzy PID controller. With this controller, the fuzzy logic part continuously tunes and adjusts the PID parameters while the controlled system is in its actual long run real time work. In other words, the Fuzzy part adapts the PID parameters moment by moment at each sampling instant. Although some concentrates when talking about tuning methods on the capability of minimizing cost and training time, like **Dicesare, et al.,2009.** who suggests the use of Ziegler-Nichols method or the manual method, in the opinion of researchers in this paper, it is more important to get a method with adaptive online tuning that provides the controller with capability of self-adapting to go on with the nonlinear nature of the quadcopter dynamics and other external effects, for example. The Fuzzy logic is nonlinear and can adapt tuning to suffice the quadcopter control needs, as will be seen.

A lot could be found in the literature about the PID controller and the fuzzy logic **Seidabad, et al., 2014. Abbasi,** and **Mahjoob, 2013. Tanaka, 1996. Ross, 2010.** and **Harris, 2000.** and so, the focus here will be on issues relating to use them.

3.2 The Quadcopter System with the PID Controller

The overall control system for the quad copter with the PID controllers for the three basic quadcopter motions, roll, pitch, and yaw is shown in **Fig.5** Three controllers' outputs are used to control these variables; $Ur, Up,$ and Uy respectively. These three control signals together with the "throttle" signal will be used to calculate the actual signals needed to control the quadcopter four motors, and as given by Eqs. (9) to (12).

3.2.1 The roll controller

The PID control equation for the roll of the quadcopter is defined by:

$$Ur = K\phi_p * e(t) + K\phi_i * \int_0^t e(\tau)\, dt + K\phi_d * \frac{de(t)}{dt} \quad (18)$$

The error equation is given by:

$$e(t) = (\phi_d - \phi) \quad (19)$$

where; $K\phi_p$, $K\phi_i$, and $K\phi_d$, are three roll PID controller parameters, ϕ_d represent the roll desired value, and ϕ represent the roll actual value.

3.2.2 The pitch controller

The PID control equation for pitch of the quadcopter is defined by:

$$Up = K\theta_p * e(t) + K\theta_i * \int_0^t e(\tau)\, dt + K\theta_d * \frac{de(t)}{dt} \quad (20)$$

The error equation is given by:

$$e(t) = (\theta_d - \theta) \tag{21}$$

where; $K\theta_p$, $K\theta_i$, and $K\theta_d$, are the three pitch PID controller parameters, θ_d represents the pitch desired value, and θ represents the pitch actual value.

3.2.3 The yaw controller

The PID control equation for yaw of the quadcopter is defined by:

$$Uy = K\psi_p * e(t) + K\psi_i * \int_0^t e(\tau)\, dt + K\psi_d * \frac{de(t)}{dt} \tag{22}$$

The error equation is given by:

$$e(t) = (\psi_d - \psi) \tag{23}$$

where; $K\psi_p$, $K\psi_i$, and $K\psi_d$ are the three yaw PID controller parameters, ψ_d represent the yaw desired value, and ψ represent the yaw actual value.

3.2.4 PID manual tuning procedure

In this method, which is much like Ziegler-Nichols method, the tuning takes place while the system is running. Initially, K_i and K_d values are set to zero, and K_p value is changed until the output of the loop oscillates. The value of K_p should be set to half of the value that caused the oscillation. Then K_i value is to be changed to minimize settling time without causing instability. Finally, the K_d value is to be changed until the overshoot is as small as possible without damping the system. The advantage of this method lies in that the PID parameters can be changed online; and no calculations are needed. The main disadvantage is that it is not as precise as other methods **Dicesare, et al., 2009.**

3.3 The Quadcopter System with the Fuzzy PID Controller

The overall control system for the quad copter with the Fuzzy PID controllers for the three basic quadcopter motions, roll, pitch, and yaw is shown in **Fig.6**. The FPID controller contains two parts, the fuzzy logic tuner, and the PID controller. The fuzzy part is used to tune the PID controllers' parameters. Each fuzzy tuner system contains two inputs; error and error derivative, and three outputs; K_p, K_i and K_d. The triangular membership function has been used for each of the inputs and the outputs and for all of the controllers. The inputs, the error and the error derivative, are defined by five membership functions: NB (negative big), NS (negative small), Z (zero), PS (positive small), and PB (positive big). On the other hand, each of the outputs is defined by three membership function: S (small), M (medium), and B (big). The range of the fuzzy set for the error input has been chosen as [-2, 2], and for the error derivative as [-4, 4]. **Fig.7** shows the membership functions for the inputs.

The range for the outputs fuzzy set has been chosen as follows:

- [1, 75] for K_p, and is the same for roll, pitch, and yaw, and as indicated in **Fig.8**.
- [0, 0.02] for K_i, and is the same for roll, pitch, and yaw, and as indicated in **Fig.9**.
- [2, 18] for K_d, and is the same for roll, and pitch, and as indicated in figure **Fig.10**.
- [36, 72] for K_d for yaw case and as is indicated in **Fig.11**.

After defining the inputs and the output for each fuzzy tuner, the next step is to set the rules base. In this case, 25 rules are needed. This is because there are two inputs each of which contains five membership functions. The number of rules equal five multiplied by five. **Tables 1** and **2** show the rules for K_p, K_i, and K_d for each controller.

The outputs of each fuzzy tuner system provide gains values for the PID controllers used to control the roll, pitch, and yaw of the quadcopter. After using the fuzzy tuner to calculate the optimal gains, these gains are used by the PID controllers to find the Ur, Up, Uy control signals via the three closed control loop as shown in **Fig.6** These three control signals, together with the "throttle" signal, will be used to calculate the actual signals needed to control the quadcopter four motors, and as given by Eqs. (9) to (12).

4. THE SIMULATION RESULTS

The quadcopter model and the controllers developed in sections 2 and 3 will be simulated and results will obtained using "MATLAB" software. The outputs or the controlled variables will be the roll, pitch, and yaw. The desired values for these variables will be entered to the "MATLAB" environment through a joystick interfaced to the computer running the "MATLAB" to resemble of actual quadcopter control, **Fig.12** shows the joystick and the computer used.

The quadcopter parameters used in the simulation are given in **Table 3**. These parameters are based on a quadcopter designed and built by the researchers of this work.

To get close to a realistic situation, the quadcopter model parameters used in the simulation are those of an actual quadcopter being designed and implemented by the researchers of this work.

4.1 The Result Obtained with the PID Controller

In this case, each parameter of the PID controller has been tuned to get a stable system and improved performance. **Tables 4** and **5** show the parameters values for each of the PID controllers, and **Figs. 13** to **15** show the response for each PID controller.

4.2 The Fuzzy PID Controller

In this part, the fuzzy logic system has been used to tune the PID controller's parameters in such a way to get a stable response with good characteristics. **Figs. 16** to **19** show the parameters for each PID controller during the tuning/response time. **Figs. 20** to **22** show the quadcopter responses for the roll, pitch, and yaw signals.

5. PERFORMANCE EVALUATION

Two performance indices have been used to judge the performance of the two controllers used. The two indices are the Integral Square Error (ISE) and the Integral Absolute Error (IAE). The time over which each of these two indices has been measured is (20 seconds) for the roll and pitch cases, and (40 seconds) for the yaw case. **Table 6** shows the values for the ISE and the IAE for each controller, with attention to that the results obtained for the PID controller regards the PID after being manually tuned.

Moreover, the system performance has been evaluated using four response characteristics which are the rise time, the overshoot, the steady state error, and the settling time. The results obtained are summarized in **Table 7**. It must be pointed out here that the results obtained for the PID controller regards the PID after being manually tuned. The results given in **Tables 6** and **7** indicates a clear superiority of the FPID.

6. RESULTS DISCUSSION

The results obtained for controlling the rotational dynamics variables, the pitch, the roll, and the yaw, with the PID controller have been compared to those obtained with the FPID controller to judge the performance of each controller. Doing the comparison required calculating some performance indices and response characteristics and as given in **Tables 6** and **7**. The comparison indicates that the FPID is more capable than the PID in reducing the ISA and the IAE. And this is true whether for the roll or the pitch or the yaw. Also, the FPID showed the capability of reducing rise and settling times and at the same time reducing the overshoot in the response. This was the case for the pitch and the roll. For the yaw case, the FPID managed to reduce the settling time and the overshoot, but with no improvement regarding the rise time. Considering the steady state error, both of the controllers managed to eliminate it.

The percentage of improvement in the responses, with the FPID controller, for the different evaluation cases considered ranged (approximately) between 50% and 80%.

The main feature of the FPID which gave it this efficiency is its capability in adapting the PID parameters from sample to sample, and so, giving it the capability to cope with changes in the system working conditions keeping in mind that it is a nonlinear one. It is important to point out here that the fuzzy part of the FPID, could itself be tuned to be capable of achieving better tuning of the PID part.

7. CONCLUSIONS

In this work adopted the software simulation approach and aimed at testing, evaluating, and comparing two controllers, namely the PID and the FPID, when used to control the pitch, roll, and yaw of a quadcopter. Two performance indices, the Integral Square Error (ISE) and the Integral Absolute Error (IAE), and four response characteristics, the rise time, the overshoot, the settling time, and the steady state error, have been used for evaluation and compare. The obtained results indicated a superiority in performance for the FPID over the PID, and this is because the FPID involves continuous adjusting of the PID parameters over the entire response time to cope with the changing situation in a way to achieve best possible response. This conclusion about performance superiority is based by the following figures that are calculated using the results obtained: $70\%, 70\%,$ and 52% lesser ISA for the roll, pitch, and yaw consequently, $70.5\%, 70.5\%, 56.4\%$ lesser IAE for the roll, pitch, and yaw consequently, $53\%,$ and 80.6% lesser rise time and settling time for the roll and pitch consequently, and 77% lesser settling time for the yaw. In addition, the FPID gave zero overshoot versus $18\%, 18\%,$ and 25% in the PID case for the roll, pitch, and yaw consequently. Apart from this, both controllers gave zero steady state error with close rise times for the yaw.

The FPID fits quite great for simulation cases. For control of quadcopter in real time, the processing unit must be capable and fast enough to compute the control signals using the FPID controller. And this is because the computations needed for the FPID controller is much greater than that for the PID.

One more important thing to point out is that the fuzzy part itself could be tuned with the aim of a compromise between an improved efficiency and a reduced computation time.

8. REFERENCES

- Abbasi, E. and Mahjoob, M., 2013, *Controlling of Quadrotor UAV Using a Fuzzy System for Tuning the PID Gains in Hovering Mode*, Mechanical Engineering, University of Tehran, College of Engineering Journal, Vol.12, No.5.
- Anjum, A., Sufian, R., Abbas, Z., and Qureshi, I., 2016, *Attitude Control of Quadcopter Using Adaptive Neuro Fuzzy Control*, International Journal of Hybrid Information Technology, Vol.9, No.4.
- Basta, P. O., 2012, *Quadcpoter Flight*, M. Sc. Thesis, Electrical Engineering, University of California, USA.
- Benic, Z., Piljek, P., and Kotarski, D., 2016, *Mathematical Modeling of Unmanned Aerial Vehicles with Four Rotors*, Interdisciplinary Description of Complex Systems, Vol.14, No.2 pp.88-100.
- Bouabdallah, S., 2007, *Design and Control of Quadrotors with Application to Autonomous Flying*, M.Sc. Thesis, Faculty of Science, Engineering and Technology, University of Aboubekr Belkaid.
- Bouadi, H., Bouchoucha, M., and Tadjine, M., 2007, *Modeling and Stabilizing Control Laws Design Based on Sliding Mode for an UAV Type-Quadrotor*, Engineering Letters, Vol.15, No.2, pp. 42-47.
- Brito, J. M., 2009, *Quadcopter Prototype*, M.Sc. Thesis, Department of Mechanical Engineering, University of tecniac de Lisboa, Portugal.
- Bresciani, T., 2008, *Modelling, identification and control of a quadrotor helicopter*, Thesis, Lund university, Sweeden.
- Dicesare, A., Gustafson, K., and Lindenfelzer, P., 2009, *Design Optimization of a Quad-Rotor Capable of Autonomous Flight*, B.Sc. Thesis, Department of Mechanical Engineering, Institute Worcester Polytechnic, USA.
- Gopalakrishnan, E., 2016, *Quadcopter flight mechanics model and control algorithms*, Thesis, Czech Technical University, Czech.
- Harrag, A., Zeghlache1, S., Saigaa, D., Kara, K., and Bouguerra, A., 2012, *Fuzzy Sliding Mode Control with Chattering Elimination for a Quadrotor Helicopter in Vertical Flight*, in Corchado et al. (Eds), *Hybrid Artificial Intelligent System*, Berlin Heidelberg: Springer, Vol.10, No.4, pp.125-136.
- Hystad, A., 2015, *Model, design and control of a quadcopter*, Thesis, Norwegian University of Science and Technology, Norway.
- Harris, J., 2000, *An Introduction to Fuzzy Logic Applications*, Springer.

- Kotarski, D., Benic, Z., and Krznar, M., 2016, *Control design for unmanned aerial vehicles with four rotors*, journal of Interdisciplinary Description of Complex Systems, Vol.14, No.2, Croatia.
- Luis, C., 2016, *Design of a trajectory tracking controller for a nano quadcopter*, Technical report, Polytechnique Montreal, Canada.
- Mahdi, O., 2006, *Modeling and Control of Unmanned Aerial Vehicle of Technology*, M.Sc. Thesis, Department of Automation and Robotics, Poznan University of Technology, Poland.
- Nagaty, A., Saeedi, S., Thibault, C., Seto, M., and Li, H., 2013, *Control and Navigation Framework for Quadrotor Helicopters*, Journal of Intelligent and Robotic Systems, Vol.70, No.1.
- Ribas, F., and Engel, P. M., 2014, *Complete System for Quadcopter*, B.Sc. Thesis, Computer Engineering, Federal University of Rio Grande do Sul, Brazil.
- Ross, T. J., 2010, *Fuzzy Logic with Engineering Applications*, 3rd. ed., John Wiley & Sons.
- Schmidt, M. D., 2011, *Simulation and Control of Quadcopter Unmanned Aerial Vehicle*, M.Sc. Thesis, University of Kentucky, USA.
- Seidabad, E. A., Vandaki, I., Kamyad, I. V., 2014, *Designing Fuzzy PID Controller for Quadrotor*, International Journal of Advanced Research in Computer Science and Technology, Vol.2, No.4.
- Shahrokhi, M. and Zomorrodi, A., 2010, *Comparison of PID Controller Tuning Methods*, Project Report, Chemical and Petroleum Engineering, Sharif University of Technology, Iran.
- Tanaka, K., 1996, *An Introduction to Fuzzy Logic for Practical Applications*, Newyork: Springer.

Table 1. The rules for K_p and K_i for each of the three PID controllers.

e / de	NB	NS	Z	PS	PB
NB	M	S	S	S	M
NS	B	M	S	M	B
Z	B	B	M	B	B
PS	B	M	S	M	B
PB	M	S	S	S	M

Table 2. The rules for K_d for each of the three PID controllers.

e / de	NB	NS	Z	PS	PB
NB	M	B	B	B	M
NS	S	M	B	M	S
Z	S	S	M	S	S
PS	S	M	B	M	S
PB	M	B	B	B	M

Table 3. The parameters of the quadcopter.

Parameter	Description	Value	Units
m	mass	0.79	kg
L	Distance from the center of gravity to the center of a motor	0.27	m
I_{xx}	Roll inertia moment	$7.035 * 10^{-3}$	$kg * m^2$
I_{yy}	Pitch inertia moment	$7.035 * 10^{-3}$	$kg * m^2$
I_{zz}	Yaw inertia moment	$1.4 * 10^{-2}$	$kg * m^2$
J_r	Rotor inertia	$6.5 * 10^{-5}$	$kg * m^2$
b	Thrust factor	$3.13 * 10^{-5}$	-
d	Drag factor	$7.5 * 10^{-7}$	-

Table 4. The tuned simulation parameters of the PID controllers for roll and pitch cases.

Parameter	Value
k_P	3
k_i	0.01
k_d	2.5

Table 5. The tuned simulation parameters of the PID controller for the yaw case.

Parameter	Value
k_P	10
k_i	0.01
k_d	18

Table 6. The ISE and IAE values for the responses with the PID and the FPID controllers.

Controller	ISE	IAE
Roll PID	0.5702	0.9852
Pitch PID	0.5702	0.9852
Yaw PID	1.834	3.554
Roll fuzzy PID	0.1747	0.2998
Pitch fuzzy PID	0.1747	0.2998
Yaw fuzzy PID	0.8836	1.55

Table 7. The response characteristics values with the PID and the FPID controllers.

Roll and Pitch cases	PID controller	FPID controller	Yaw case	PID controller	FPID controller
Rise time (sec)	1.7	0.8	Rise time (sec)	4.5	4.7
Overshoot (%)	0.18	0	Overshoot (%)	0.25	0
Steady state error	0	0	Steady state error	0	0
Settling time (sec)	4.9	0.95	Settling time (sec)	25	5.73

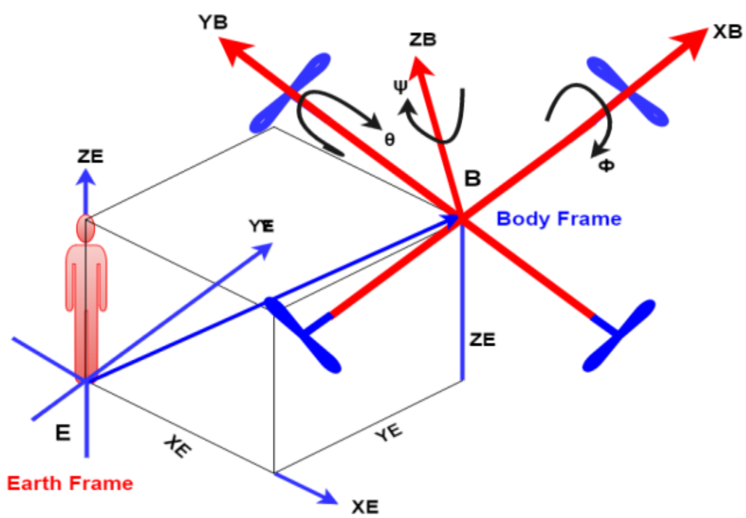

Figure 1. Body and earth fixed frame.

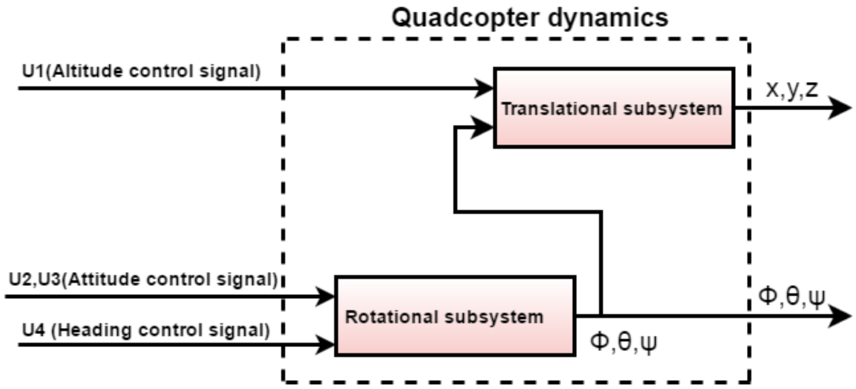

Figure 2. The quadcopter dynamics subsystems being given within a general block diagram for a quadcopter control system.

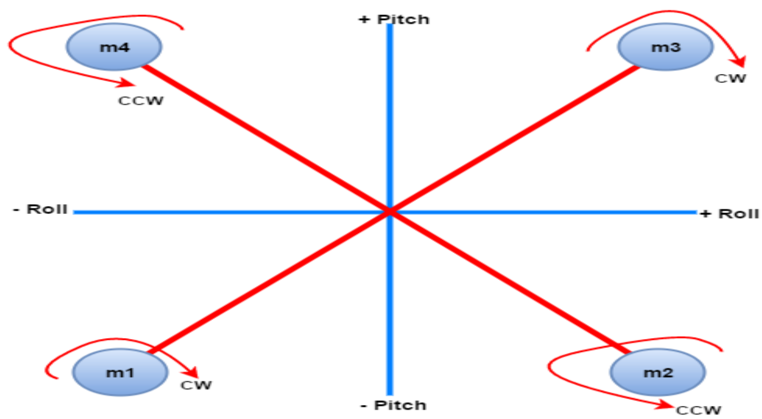

Figure 3. Quadcopter (X) configuration.

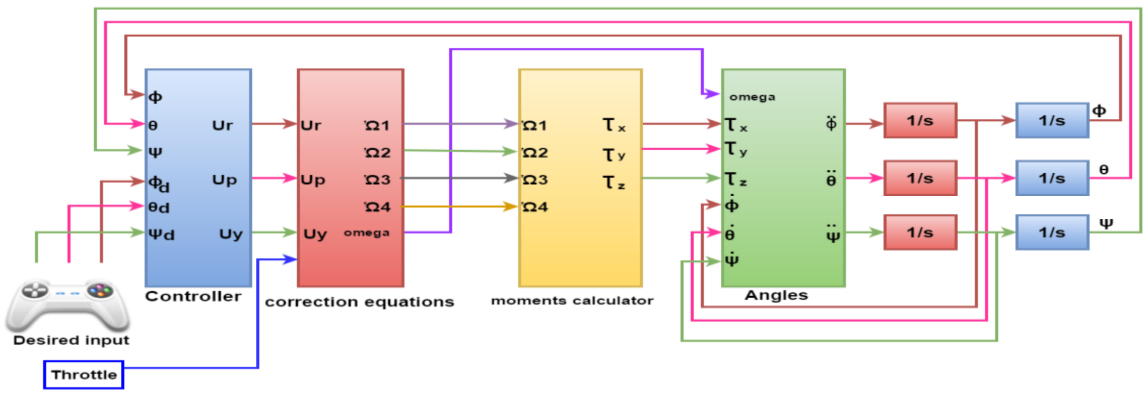

Figure 4. A picture giving an overall depiction of the closed loop quadcopter control system.

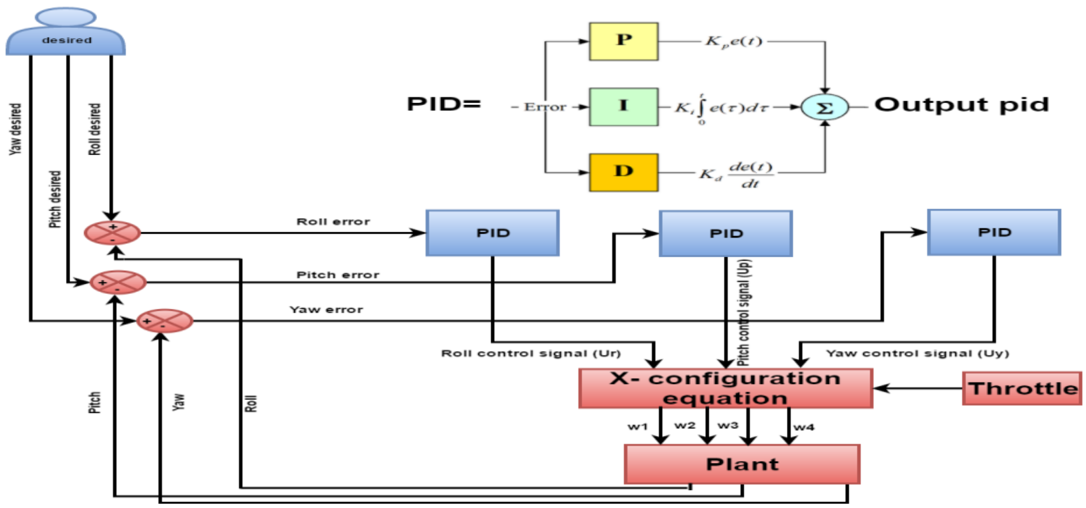

Figure 5. The quadcopter system with three PID controllers.

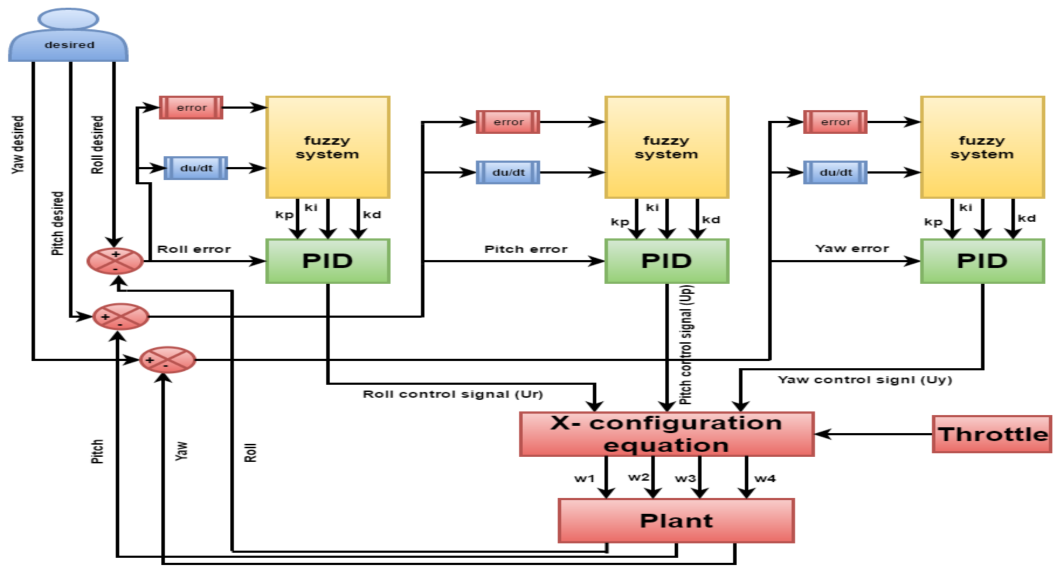

Figure 6. The quadcopter system with three fuzzy PID controllers.

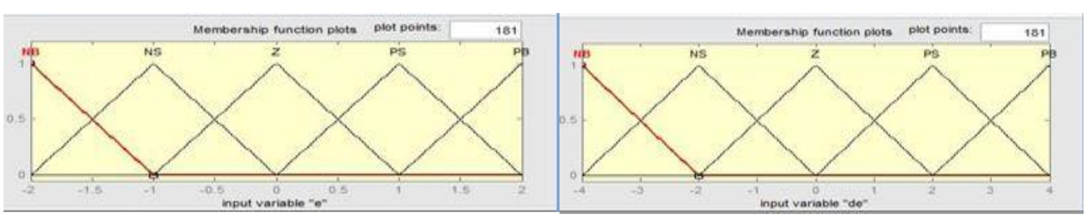

Figure 7. The membership functions for the inputs.

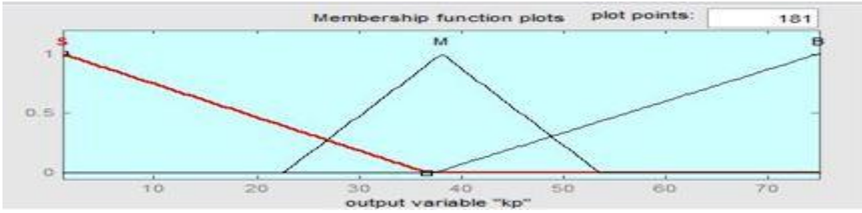

Figure 8. The memberships for the parameter K_p for roll, pitch, and yaw cases.

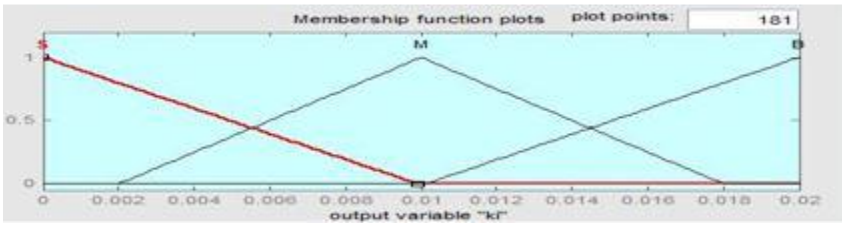

Figure 9. The memberships for the parameter K_i for roll, pitch, and yaw cases.

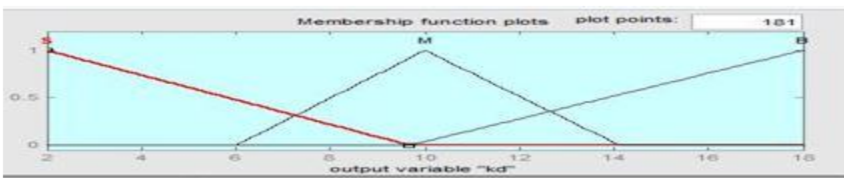

Figure 10. The memberships for the parameter K_d for roll, and pitch.

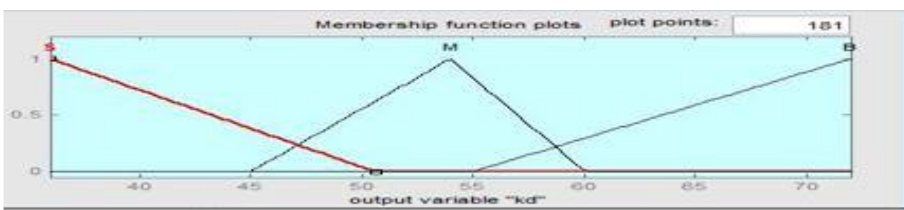

Figure 11. The memberships for the parameter K_d for yaw case.

Figure 12. The joystick interfaced to the computer for inputting the desired roll, pitch, and yaw values.

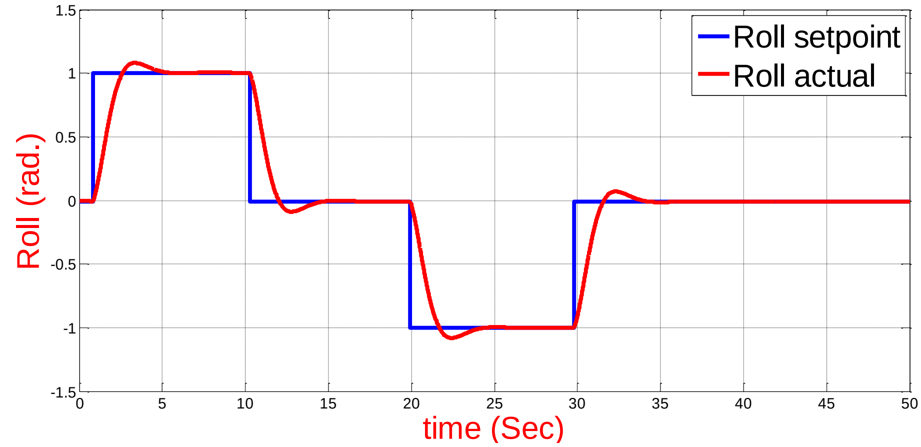

Figure 13. The roll signal Response.

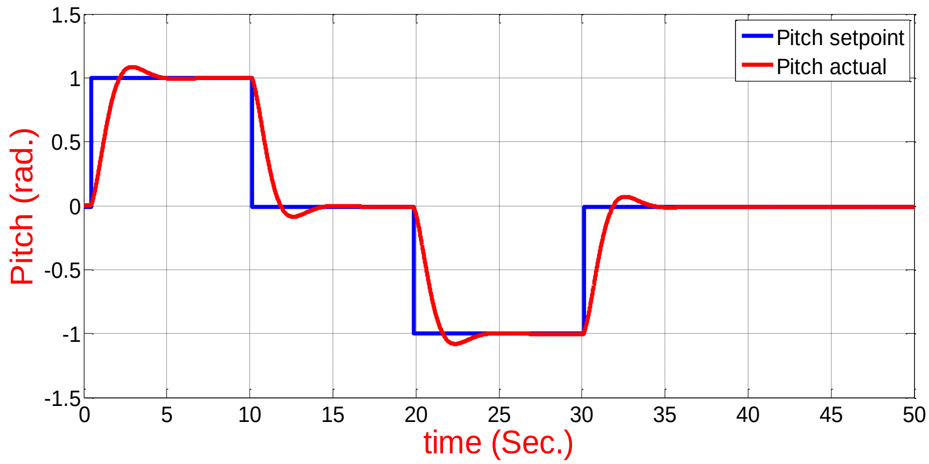

Figure 14. The pitch signal Response.

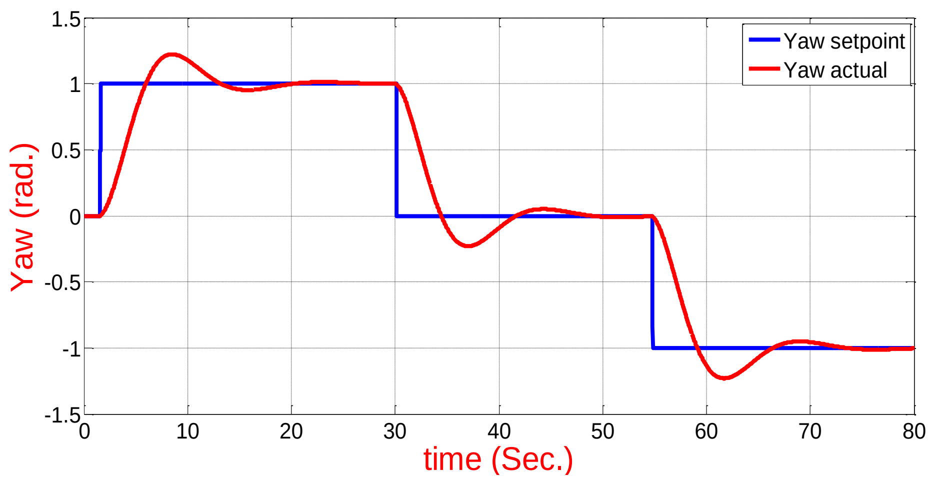

Figure 15. The yaw signal Response.

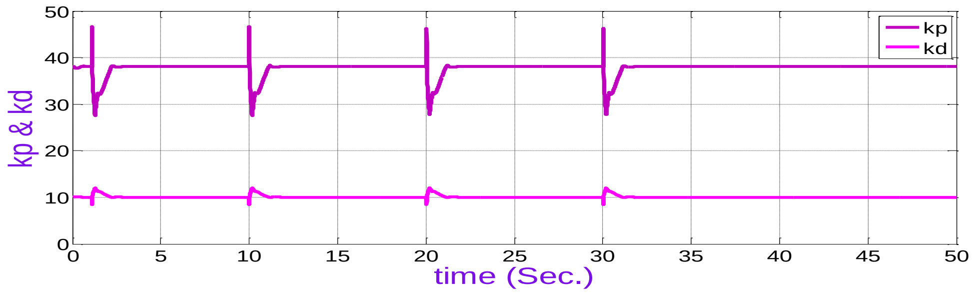

Figure 16. The k_p and k_d values for the roll and the pitch signals for the fuzzy PID controller case.

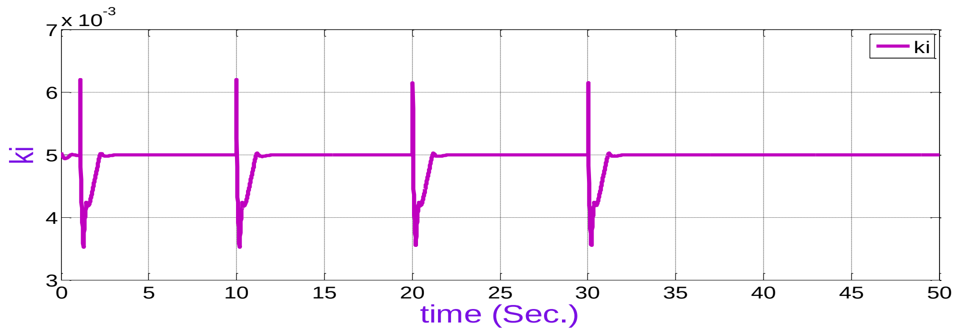

Figure 17. The k_i value for the roll and the pitch signals for the fuzzy PID controller case.

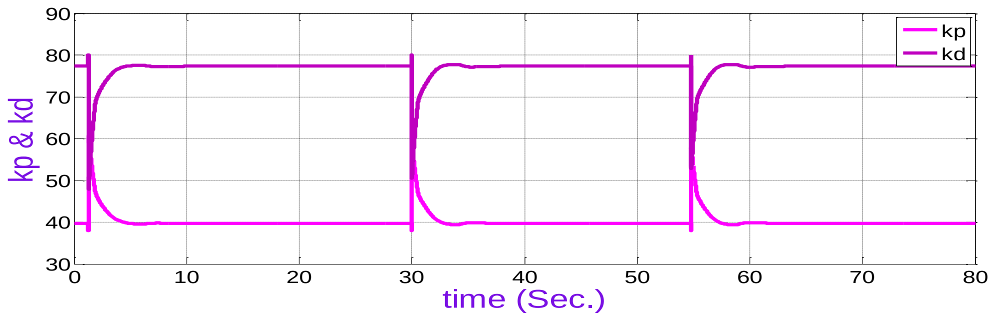

Figure 18. The k_p and k_d values for the yaw signal for the fuzzy PID controller case.

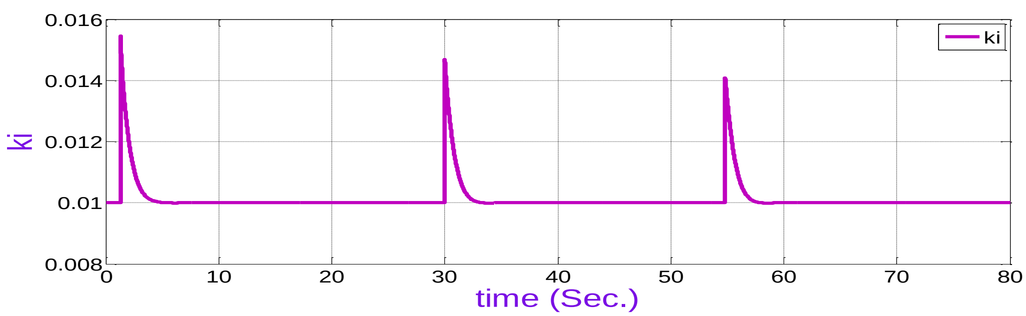

Figure 19. The k_i value for the yaw signal for the fuzzy PID controller case.

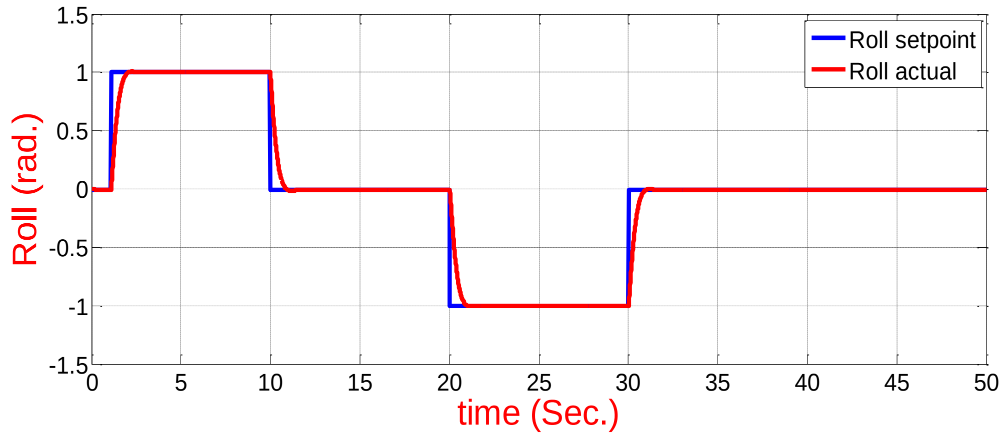

Figure 20. The roll response for the Fuzzy PID controller case.

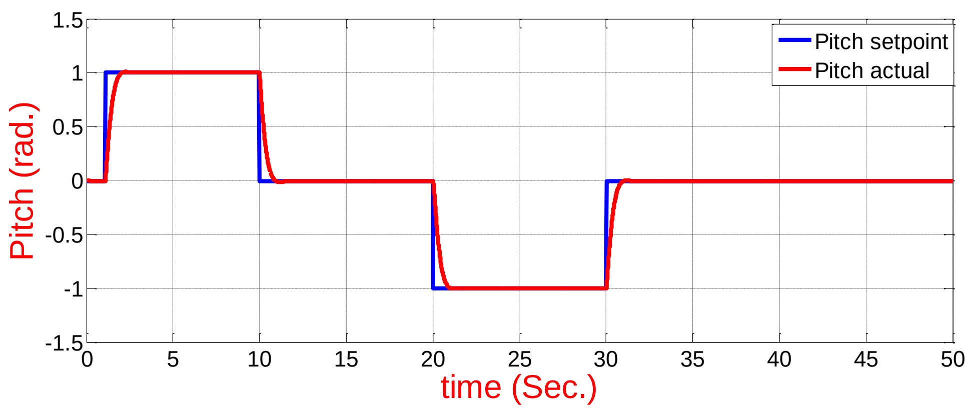

Figure 21. The pitch response for the Fuzzy PID controller case.

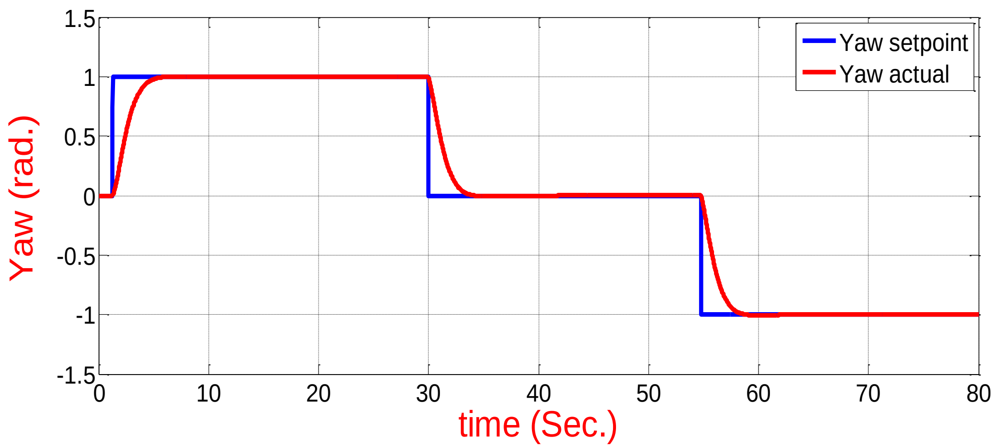

Figure 22. The yaw response for the Fuzzy PID controller case.