Tuning of PID Controllers for Quadcopter System using Cultural Exchange Imperialist Competitive Algorithm

Dr. Nizar Hadi Abbas
Assistant Professor
College of Engineering, University of Baghdad.
Email: drnizaralmsaodi@gmail.com

Ahmed Ramz Sami
Research Scholar
College of Engineering, University of Baghdad.
Email: ahmedramz@gmail.com

ABSTRACT

Quadrotors are coming up as an attractive platform for unmanned aerial vehicle (UAV) research, due to the simplicity of their structure and maintenance, their ability to hover, and their vertical take-off and landing (VTOL) capability. With the vast advancements in small-size sensors, actuators, and processors, researchers are now focusing on developing mini UAV's to be used in both research and commercial applications. This work presents a detailed mathematical nonlinear dynamic model of the quadrotor which is formulated using the Newton-Euler method. Although the quadrotor is a 6 DOF under-actuated system, the derived rotational subsystem is fully actuated, while the translational subsystem is under-actuated. The derivation of the mathematical model was followed by the development of the controller to control the altitude, attitude, heading and position of the quadrotor in space, which is, based on the linear Proportional-Derivative- Integral (PID) controller; thus, a simplified version of the model is obtained. The gains of the controllers will be tuned using optimization techniques to improve the system's dynamic response. The standard Imperialist Competitive Algorithm (ICA) was applied to tune the PID parameters and then it was compared to Cultural Exchange Imperialist Competitive algorithm (CEICA) tuning, and the results show improvement in the proposed algorithm. The objective function results were enhanced by (23.91%) in the CEICA compared with ICA.

Key words: Quadcopter, PID, Flying robot, ICA, CEICA.

ضبط قيم المتحكم التناسبي-التفاضلي-التكاملي للطائرات الرباعية باستخدام خوارزمية التبادل المعرفي في تنافس الامبراطوريات

د. نزار هادي عباس
استاذ مساعد
كلية الهندسة، جامعة بغداد

أحمد رمز سامي
باحث
كلية الهندسة، جامعة بغداد

الخلاصة

الطائرات الرباعية بدأت تصبح شائعة الاستخدام في ابحاث الطائرات المسيرة عن بعد وذلك لبساطة بنائها وصيانتها وكذلك قابليتها على الثبات في الهواء وقابلية الاقلاع والهبوط العامودي . نظراً للتقدم الكبير الحاصل في مجال تصنيع الحساسات الدقيقة ، المحركات الكهربائية و المتحكمات ، بدا الباحثون بالعمل في هذا المجال وذلك لتطوير طائرات مصغرة مسيرة عن بعد والتي تستخدم في المجالات البحثية والتجارية والعسكرية. هذا العمل يقدم طريقة بناء نموذج رياضي مفصل عن طبيعة النظام الميكانيكي الغير خطي وهي طريقة نيوتن-اويلر. بالرغم من ان نظام الطائرات الرباعية هي سداسي الابعاد ولايمكن السيطرة الكلية عليه الا ان النظام الثانوي الدوراني يمكن السيطرة عليهة بصورة كلية ولكن النظام الخطي لا يمكن السيطرة الكلية عليه. بعد الاشتقاق الرياضي للنموذج الذي يمثل الطائرة الرباعية نبدا بتصميم المتحكم الذي يسيطر على ارتفاع و وضع واتجاه الطائرة في الفضاء . الطريقة المستخدمة في السيطرة تعتمد على المتحكم التناسبي – التفاضلي – التكاملي الخطي ،

وبالتالي سنحتاج الى ان نقوم بتعديل نموذج الطائرة الى الخطي ايضاً . القيم المستحصلة لضبط هذا النوع من المتحكمات تكون بواسطة ألخوارزميات ألامثلية للحصول على القيم الامثل والافضل للتحكم لغرض الحصول على افضل استجابة . في هذا البحث قمنا بتطبيق خوارزمية تنافس الامبراطوريات وهي من ضمن الخوارزميات التطورية وايضا قمنا بتطويرها الى خوارزمية افضل (خوارزمية التبادل المعرفي في تنافس الامبراطوريات) وعند المقارنة حصلنا على نتائج افضل بنسبة تحسين (23.91%) بتثبيت متغيرات الخوارزميتين عند المقارنة بينهما.

الكلمات الرئيسية: الطائرات الرباعية، المتحكم التناسبي-التفاضلي-التكاملي، الربوتات الطائرة، خوارزمية تنافس الامبراطوريات، خوارزمية التبادل المعرفي في تنافس الامبراطوريات.

1. INTRODUCTION

The quadrotor unmanned aerial vehicle (UAV) has been an increasingly popular research topic in recent years due to their low cost, maneuverability, and ability to perform a variety of tasks. They are equipped with four motors typically designed in an “X” configuration, every two pairs of opposite motors rotating clockwise and the other motor pair rotating counter-clockwise to balance the torque. Compared with the conventional rotor-type aircraft, it has no tail, quadrotors advantage compared with normal airplanes is that it has more compact structure and it can hover. The quadrotor does not have complex mechanical control system because it relies on fixed pitch rotors and uses the variation in motor speed for vehicle control. However, these advantages come with a disadvantage. Controlling a quadrotor is not easy because of the coupled dynamics between translational and rotational dynamics thus making it under-actuated system. Also, it uses more energy battery to keep its position in the air. In addition, the dynamics of the quadrotor are highly non-linear and several effects are encountered during its flights, thus making its control a good research field to explore. This opened the way to several control algorithms and different optimization technique for tuning that are proposed in the literature.

Going through the literature, one can see that most of the papers use two approaches for modeling the mechanical model. They are Newton-Euler and Newton-Lagrange but the most used one and it’s most familiar and better is Newton-Euler which as in **Benić, et al. 2016** and **Bouabdallah, 2006.** Also regarding the control of quadcopter, there is a good amount of research done both in linear and nonlinear methods, **ElKholy, 2014**. However, for practicality the linear controller especially PID perform better in practical implementation and that is the reason PID was used as the controller for the quadcopter **Bouabdallah, 2004** and **Li,** and **Li, 2011**. Also, some researchers work on the stabilization of quadcopter that means inner loop, **Salih, et al., 2010** and **Magsino, et al., 2014**. Other researchers include also the position control of quadcopter to cover all topics in quadcopter control **Pipatpaibul,** and **Ouyang, 2011**. Tuning the parameters of the PID controllers is very important to get the best performance of quadcopter. Thus in order to get the best (optimal) setting to achieve the objective, an optimization algorithm must be used. In this work the Imperialist Competitive Algorithm was applied which is the same as **Atashpaz-Gargari**, and **Lucas**, **2007**, then it was improved to generate new algorithm called cultural exchange imperialist competitive Algorithm which achieved better results. As much of research done in this field there is still optimization algorithms, which is not tested in quadcopter control systems, imperialist competitive algorithms, is one of them. And also this work explores new optimization algorithm which performed better. Both algorithms were measured statically and it turns out that the new algorithm is performing better than the normal algorithm always in getting lower objective value.

This paper discusses the formulation of mathematical model of quadcopter in section 2. After that, the control system of the quadcopter which is PID and how to apply it to stabilize the quadcopter and control its position in section 3. Section 4 will be about the optimization techniques used tune the PID parameter Kp, Ki, and Kd and how to enhance the algorithm for tuning the parameters more effectively and better than the normal one. Section 5 will contain the simulation results and the conclusion is in section 6.

2. MATHEMATICAL MODEL

Designing a control system for physical systems is commonly started by building a mathematical model. The model is very important because it gives an explanation of how the system acts to the inputs given to it. In this research, the mathematical model equations of motion are derived using a full quadcopter with body axes as shown in **Fig. 1**.

2.1 Quadcopter Fundamentals

The quad-rotor is a helicopter equipped with four motors and propellers mounted on them. It is very well modeled with a cross (X) configuration style. Each opposite motors rotate in the same direction counter-clockwise and clockwise. This configuration of opposite pairs' directions completely eliminates the need for a tail rotor, which is needed for stability in the conventional helicopter structure. **Fig. 2** shows the model in a stable hover, where all the motors rotate at the same speed, thus generating equal lift from the four motors and all the tilt angles are zero **Bouabdallah, 2006**. Four basic movements govern the quadcopter orientation and displacement, which allow the quadcopter to reach a certain altitude and attitude:

a)Throttle
It is provided by concurrently increasing or decreasing all propeller speeds with the same amount and rate. This generates a cumulative vertical force from the four propellers, with respect to the body-fixed frame. As a result, the quadcopter is raised or lowered by a certain value.

b)Roll
It is provided by concurrently increasing or decreasing the left propellers speed and the opposite for the right propellers speed at the same rate. It generates a torque with respect to the x-axis which makes the quadcopter to tilt about the axis, thus creating a roll angle. The total vertical thrust is maintained as in hovering; thus this command leads only to a roll angular acceleration.

c)Pitch
The pitch and roll are very similar. It is provided by concurrently increasing or decreasing the speed of the rear propellers and the opposite for the speed of the front propellers at the same rate. This generates a torque with respect to the y-axis which makes the quad-rotor to tilt about the same axis, thereby creating a pitch.

d)Yaw
This command is provided by increasing (or decreasing) the opposite propellers' speed and by decreasing (or increasing) that of the other two propellers. It leads to a torque with respect to the z-axis which makes the quadrotor turn clockwise or anti clockwise about the z-axis.

2.2 Quadcopter Dynamics

There are two coordinate systems to be considered in quadcopter dynamics, shown in **Fig.3**:

- The earth inertial frame (E-frame)
- The body-fixed frame of the vehicle (B-frame)

They are related through three successive rotations:

- Roll: Rotation of φ around the x-axis $\mathbf{R_x}$;
- Pitch: Rotation of θ around the y-axis $\mathbf{R_y}$;
- Yaw: Rotation of ψ around the z-axis $\mathbf{R_z}$.

Switch between the coordinates system using rotation matrix **R** which is a combination of rotation about 3 axis rotations also **Fig. 4** shows how the rotation is applied **Pipatpaibul, Ouyang, 2011**:

$$\boldsymbol{R} = Rx\,.Ry\,.Rz = \begin{pmatrix} \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi & \cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi & \sin\psi\sin\theta\cos\phi-\sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix} \quad (1)$$

The developed model in this work assumes the following:

- The structure is supposed rigid.
- The structure is supposed symmetrical.
- The center of gravity and the body fixed frame origin are assumed to coincide.
- The propellers are supposed rigid.
- Thrust and drag are proportional to the square of propeller's speed.

Using the Newton-Euler formalism to create a nonlinear dynamic model for the quadrotor, in this research, the Newton-Euler method is employed for both the main body and rotors. The general form of Newton-Euler equation is expressed as:

$$\begin{bmatrix} mI_{3*3} & 0 \\ 0 & I \end{bmatrix}\begin{Bmatrix} \dot{V} \\ \dot{\omega} \end{Bmatrix} + \begin{Bmatrix} \omega \times mV \\ w \times I\omega \end{Bmatrix} = \begin{Bmatrix} F \\ \tau \end{Bmatrix} \quad (2)$$

Eq. (2) is a generic form of equations of motion. It can be applied to any position in the coordinate system. In this case, the main point is the center of mass of the quadcopter. Referring to the body frame (B) in **Fig. 3**, Eq.(2) cut-down to **Pipatpaibul, Ouyang, 2011**:

$$\begin{bmatrix} mI_{3*3} & 0 \\ 0 & I \end{bmatrix}\begin{Bmatrix} \dot{V} \\ \dot{\omega} \end{Bmatrix} + \begin{Bmatrix} 0 \\ w \times I\omega \end{Bmatrix} = \begin{Bmatrix} F \\ \tau \end{Bmatrix} \quad (3)$$

Regarding the main body of the quadcopter and Eq. (3), the translational dynamics of the quadcopter in the body frame (B) is defined as **Pipatpaibul,** and **Ouyang, 2011**:

$$m\begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix}_B = \begin{Bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} b\Omega_i^2 \end{Bmatrix} - R_{xyz}^{-1}\begin{Bmatrix} 0 \\ 0 \\ mg \end{Bmatrix} \quad (4)$$

described in the earth frame (E) **Fig. 3** through Eq. (4) as **Pipatpaibul,** and **Ouyang, 2011**:

$$m\begin{Bmatrix}\ddot{X}\\ \ddot{Y}\\ \ddot{Z}\end{Bmatrix}_E = R_{xyz}m\begin{Bmatrix}\ddot{X}\\ \ddot{Y}\\ \ddot{Z}\end{Bmatrix} = R_{xyz}\begin{Bmatrix}0\\ 0\\ \sum_{i=1}^{4} b\Omega_i^2 - R_{xyz}^{-1}mg\end{Bmatrix}_B$$

$$= R_{xyz}\begin{Bmatrix}-sin\theta\cos\phi \sum_{i=1}^{4} b\Omega_i^2\\ sin\varphi \sum_{i=1}^{4} b\Omega_i^2\\ -mg + cos\theta\cos\phi \sum_{i=1}^{4} b\Omega_i^2\end{Bmatrix} \qquad (5)$$

From Eq. (3), the main body rotational dynamics can be described in the body frame (B) as **Pipatpaibul,** and **Ouyang, 2011**:

$$\begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}\begin{Bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{Bmatrix} + \begin{Bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{Bmatrix} \times \begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}\begin{Bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{Bmatrix} = \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix} \qquad (6)$$

$$\begin{Bmatrix}I_{xx}\ddot{\phi}\\ I_{yy}\ddot{\theta}\\ I_{zz}\ddot{\Psi}\end{Bmatrix} + \begin{Bmatrix}\dot{\theta}\dot{\Psi}\left(I_{zz} - I_{yy}\right)\\ \dot{\phi}\dot{\Psi}\left(I_{xx} - I_{zz}\right)\\ \dot{\phi}\dot{\theta}\left(I_{yy} - I_{xx}\right)\end{Bmatrix} = \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix}_e + \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix}_r \qquad (7)$$

Where subscripts e and r refer to moments due to external forces, which ultimately caused by thrust and drag from rotors, and moments due to rotor gyro effect, respectively.

2.3 Rotor dynamics

The rotor dynamics can be described using Eq. (2) by considering a coordinate system of each rotor, which is the same plane as the body frame for X and Y axes while Z-axis coincides with the rotation of the rotor. Considering rotational dynamics of each rotor in the form of Newton-Euler **Pipatpaibul,** and **Ouyang, 2011**:

$$\begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}_{r,i}\begin{Bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{Bmatrix}_{r,i} + \begin{Bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{Bmatrix}_{r,i} \times \begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}_{r,i}\begin{Bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{Bmatrix}_{r,i} = \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix}_{r,i} \qquad (8)$$

$$\begin{Bmatrix}I_{xx}\ddot{\phi}\\ I_{yy}\ddot{\theta}\\ I_{zz}\ddot{\Psi}\end{Bmatrix}_{r,i} + \begin{Bmatrix}\dot{\theta}\dot{\Psi}\left(I_{zz} - I_{yy}\right)\\ \dot{\phi}\dot{\Psi}\left(I_{xx} - I_{zz}\right)\\ \dot{\phi}\dot{\theta}\left(I_{yy} - I_{xx}\right)\end{Bmatrix}_{r,i} = \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix}_{r,i} \qquad (9)$$

for i= 1,2,3,4 and denotes i^{th} rotor. Since the rotors always rotate about their Z-axes at the rate of Ω with the moment of inertia about Z-axis of J_r and have very low masses, $I_{xx}$and I_{yy} can then be omitted and the dynamics of each rotor reduces to **Pipatpaibul,** and **Ouyang, 2011**:

$$\begin{Bmatrix}0\\ 0\\ J_r\dot{\Omega}\end{Bmatrix}_i + \begin{Bmatrix}\dot{\theta}\Omega J_r\\ -\dot{\phi}\Omega J_r\\ 0\end{Bmatrix}_i = \begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix}_{r,i} \qquad (10)$$

Note that Eq. (10) is a function of rotor speed Ω. Since rotor 1 and 3 rotate in the opposite direction of rotor 2 and 4, one can define the total rotor speed as **Pipatpaibul,** and **Ouyang, 2011**:

$\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$

From Eq. (9) and (10), the total moment due to gyro effect from all rotors can be expressed as **Pipatpaibul,** and **Ouyang, 2011** :

$$\begin{Bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{Bmatrix}_r = \sum_{i=1}^{4} \begin{Bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{Bmatrix}_r \quad = \sum_{i=1}^{4} \left[\begin{Bmatrix} 0 \\ 0 \\ J_r\dot{\Omega} \end{Bmatrix}_i + \begin{Bmatrix} \dot{\theta}\Omega J_r \\ -\dot{\phi}\Omega J_r \\ 0 \end{Bmatrix}_i \right] \tag{11}$$

$$\begin{Bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{Bmatrix}_r = \begin{Bmatrix} \dot{\theta}\Omega_r J_r \\ -\dot{\phi}\Omega_r J_r \\ J_r\dot{\Omega}_r \end{Bmatrix} \tag{12}$$

2.4 Equation of motion (EoM)

Now that all necessary dynamics of the entire model has been established, one can write the complete equations of motion of the quadrotor. Combining Eq. (4), (6), (8), (10), (11) and (12) yields:

$$\begin{cases} translational\ dynamics \begin{cases} m\ddot{X} = -sin\theta cos\phi \sum_{i=1}^{4}[b\Omega_i^2] \\ m\ddot{Y} = sin\phi \sum_{i=1}^{4}[b\Omega_i^2] \\ m\ddot{Z} = -mg + cos\theta cos\phi \sum_{i=1}^{4}[b\Omega_i^2] \end{cases} \\ Rotational\ dynamics \begin{cases} I_{xx}\ddot{\phi} = \dot{\theta}\dot{\Psi}(I_{zz} - I_{yy}) - \dot{\theta}\Omega_r J_r + lb(-\Omega_2^2 + \Omega_4^2) \\ I_{yy}\ddot{\theta} = \dot{\phi}\dot{\Psi}(I_{xx} - I_{zz}) + \dot{\phi}\Omega_r J_r + lb(\Omega_1^2 - \Omega_3^2) \\ I_{zz}\ddot{\Psi} = \dot{\phi}\dot{\theta}(I_{yy} - I_{xx}) + J_r\dot{\Omega}_r + \sum_{i=1}^{4}[(-1)^i d\Omega_i^2] \end{cases} \end{cases} \tag{13}$$

2.5 Model Parameters

The chosen parameters in this research for simulation was taken from the datasheet of real components that are used in building quadcopter shown in **Table 1.**

3. PID CONTROL

PID controllers have been used in a broad range of controller applications. It is for sure the most applied controller in the industry. The PID controller shown in **Fig. 5,** has the advantage that parameters (K_p, K_i, K_d) are easy to tune, which is simple to design and has good robustness. However, the quadcopter includes non-linearity in the mathematical model and may include some inaccurate modeling of some of the dynamics, which will cause bad performance of the control system, thus it is needed form the designer to be careful when neglecting some effects in the model or simplifying the model **Kotarski, et al., 2016.**

Stabilization is very important for an under-actuated system like a quadrotor, as it is inherently unstable due to its six degrees of freedom and four actuators. A control system is modeled for the quadcopter using four PID controllers to control the Attitude (Roll, pitch, and yaw) and the Altitude (Z height) to introduce stability these four controllers form the inner loop of control for the quadcopter system. And then two more PID controllers are used to control the position of the quadcopter (X and Y axes) and the output of these two controllers will be input to the roll and pitch controllers these two PID's will form the outer control loop.

3.1 Inner Control Loop

Inner control loop controls quadcopter altitude and attitude. Input variables for inner loop can be divided into two parts, desired and sensor signals. Desired signals are obtained from the control signals coming directly from the pilot or the autopilot program, these signals are the Height (altitude) and pointing (Yaw) of the quadcopter the other two signals desired roll and pitch comes for the output of the outer loop control since they are translated from the desired x and y position in the outer PID's. **Fig. 6** shows the complete control system for the quadcopter including the inner loop and the outer loop.

Altitude control
The equation for the thrust force control variable $U1$ is:

$$U_1 = K_{PZ}e_Z + K_{IZ}\int e_Z - K_{DZ}\frac{d}{dt}e_Z \tag{14}$$

where K_{PZ}, K_{IZ} and K_{DZ} are three altitude PID controller parameters. e_Z is the altitude error, where $e_Z = Z_{des} - Z_{mes}$. Z_{des} : desired altitude and Z_{mes} : measured altitude.

Roll control
The equation for the roll moment control variable $U2$ is:

$$U_2 = K_{P\phi}e_\phi + K_{I\phi}\int e_\phi - K_{D\phi}\frac{d}{dt}e_\phi \tag{15}$$

where $K_{P\phi}$, $K_{I\phi}$ and $K_{D\phi}$ are three roll angle PID controller parameters. e_ϕ is the roll angle error, where $e_\phi = \phi_{des} - \phi_{mes}$. ϕ_{des} : desired roll angle and ϕ_{mes} : measured roll angle.

Pitch control
The equation for the pitch moment control variable $U3$ is:

$$U_3 = K_{P\theta}e_\theta + K_{I\theta}\int e_\theta - K_{D\theta}\frac{d}{dt}e_\theta \tag{16}$$

Similarly, $K_{P\theta}$, $K_{I\theta}$ and $K_{D\theta}$ are three pitch angle PID controller parameters. e_θ is the pitch angle error, where $e_\theta = \theta_{des} - \theta_{mes}$. θ_{des} : desired pitch angle and θ_{mes} : measured pitch angle.

Yaw control
The equation for the yaw moment control variable $U4$ is:

$$U_4 = K_{P\Psi}e_\Psi + K_{I\Psi}\int e_\Psi - K_{D\Psi}\frac{d}{dt}e_\Psi \tag{17}$$

where $K_{P\Psi}$, $K_{I\Psi}$ and $K_{D\Psi}$ are three yaw angle PID controller parameters. e_Ψ is the yaw angle error, where $e_\Psi = \Psi_{des} - \Psi_{mes}$. Ψ_{des} is the desired yaw angle and Ψ_{mes} is the measured yaw angle.

3.2 Outer Control Loop

The outer control loop is applied since the quadcopter is under-actuated system and it is not applicable to control all of the quadcopter 6-DOF straightly. As mentioned earlier, the inner loop directly controls 4-DOF (three angles and altitude). To be able to control X and Y position, outer loop is implemented. The outer control loop outputs are desired roll and pitch angles which are the inputs to the inner loop for the desired X and Y position.

Eq. (18) and **(19)** are the equations for the quadcopter X and Y linear accelerations **ElKholy, 2014**:

$$\ddot{X} = (cos\Psi sin\theta_d cos\phi_d + sin\Psi sin\phi_d)\frac{U_1}{m} \tag{18}$$

$$\ddot{Y} = (sin\Psi sin\theta_d cos\phi_d - cos\Psi sin\phi_d)\frac{U_1}{m} \tag{19}$$

Quadcopter dynamics of the X and Y linear accelerations can be simplified since the quadcopter around hovering that means the values of roll and pitch are small thus it can be approximated to ($sin\theta_d \equiv \theta_d$, $sin\phi_d \equiv \phi_d$, $cos\phi_d = 1$), **ElKholy, 2014**:

$$\ddot{X} = (\theta_d cos\Psi + \phi_d sin\Psi)\frac{U_1}{m} \tag{20}$$

$$\ddot{Y} = (\theta_d sin\Psi - \phi_d cos\Psi)\frac{U_1}{m} \tag{21}$$

Now Eq. (20) and (21) are put in matrix notation:

$$\begin{bmatrix}\ddot{X}\\ \ddot{Y}\end{bmatrix} = \frac{\mathrm{U_1}}{\mathrm{m}}\begin{bmatrix} sin\Psi & cos\Psi \\ -cos\Psi & sin\Psi\end{bmatrix}\begin{bmatrix}\phi_d\\ \theta_d\end{bmatrix} \tag{22}$$

the desired pitch and roll:

$$\begin{bmatrix}\phi_d\\ \theta_d\end{bmatrix} = \frac{\mathrm{m}}{\mathrm{U_1}}\begin{bmatrix} -sin\Psi & -cos\Psi \\ cos\Psi & -sin\Psi\end{bmatrix}^{-1} \tag{23}$$

the complete control system with the dynamic system is applied in Matlab-Simulink can be shown in **Fig. 7**.

4. OPTIMIZATION ALGORITHM

Optimization is the process of making something better. Optimization is a process of finding the best solution (setting) for a problem to achieve a certain goal (cost function) by trying variations on an initial solution and using the information gained to reach global optimum.

4.1 Cost function

Cost function or objective function formulation is very important in optimization since it is the main parameter to measure the performance of the optimization technique and decide whether the solution will suit the problem or not. In our work, the objective function was built to achieve two goals optimize the consumption of battery and speed up the response time. After researching about the battery consumption, it was found that reducing the oscillation and the overshoot of the response of the quadcopter as much possible will reduce unnecessary power consumption **Oscar, 2015**. An error objective function was used to make sure that the response is fast to follow the control signal. Integral square error (ISE) was used in the objective function, the total objective function that was implemented in Matlab m-file in both ICA and CEICA programs is shown in **Eq. (24):**

$$CostFunction =$$
$$\alpha_\phi Mp_\phi + \beta_\phi \int |e_\phi|^2 \, dt + \alpha_\theta Mp_\theta + \beta_\theta \int |e_\theta|^2 \, dt + \alpha_\Psi Mp_\Psi + \beta_\Psi \int |e_\Psi|^2 \, dt + \alpha_z Mp_z + \beta_z \int |e_z|^2 \, dt \quad (24)$$

Where $= \frac{u_d - u_a}{u_a} \times 100$, u_d: desired input , u_a: actual input.

4.2 Imperialist Competitive Algorithm

Esmaeil Atashpaz-Gargari and Caro Lucas invented the imperialist competitive algorithm. It is inspired by the competition of empires. It begins with an initial-population (countries in the world). A group of the fittest countries in the initial population is chosen to be the imperialists and the others, form the colonies of these imperialists.

$$country = [p_1, p_2, p_3, \dots, p_{N_{var}}] \quad (25)$$

All the colonies of initial-population are divided between the mentioned imperialists based on their power (fitness). the normalized cost of an imperialist:

$$C_n = c_n - \max\{c_i\} \quad (26)$$

The power of an empire is the inverse of the cost which means the imperialist competitive is a minimization technique.

$$\mathrm{p_n} = \left| \frac{\mathrm{C_n}}{\sum_{\mathrm{i=1}}^{\mathrm{N}_{imp}} \mathrm{C_i}} \right| \quad (27)$$

After dividing all colonies among imperialists, these colonies start moving toward their relevant imperialist country. The total power of an empire depends on both the power of the imperialist country and the power plus a percentage of mean power of its colonies:

$$TC_n = Cost(imperialist_n) + \xi\{Cost(colonies\ of\ empire_n)\} \quad (28)$$

Then the imperialistic competition begins among all the empires. Any empire that is not able to succeed in this competition and can't increase its power (or at least prevent decreasing its power) will be eliminated from the competition. The imperialistic competition will gradually result in an increase in the power of powerful empires and a decrease in the power of weaker ones. The competition starts with finding the possession probability, the normalize total cost is:

$$NTC_n = TC_n - m_{ax_i}\{TC_i\} \quad (29)$$

The possession probability of each empire

$$p_{p_n} = \left| \frac{NTC_n}{\sum_{i=1}^{Nimp} NTC_i} \right| \quad (30)$$

Then do roulette wheel is carried out to assign the weakest colony of the weakest empire to another empire. Weak empires will lose their power and ultimately they will collapse. The movement of colonies toward their relevant imperialists along with competition among empires and also the collapse mechanism will hopefully cause all the countries to converge to a state in which there exists just one empire in the world and all the other countries are colonies of that empire. In this ideal new world colonies, have the same position and power as the imperialist **Atashpaz-Gargari**, and **Lucas**, **2007**. All the steps mentioned above are illustrated in flow chart **Fig. 8**.

4.3 Cultural Exchange Imperialist Competitive Algorithm (CEICA)

As mentioned earlier, the normal ICA tuning did not give the desired performance for 60 iterations and 40 countries, which is long simulation time 1h 45min (it's was found that increasing the iteration number more than 60 iterations will not generate a much better solution and if the iteration number less than 60 the solution will not be optimal always). During the work with ICA and PID's, it was noticed that ICA is a very random process and it could be improved by adding an additional step to the algorithm. This step is called cultural exchange which is also obvious in the name of the new algorithm Cultural Exchange Imperialist Competitive Algorithm (CEICA) this modification is applicable to any application which has more than one parameter to form the objective function. The detailed concept behind this modification is that there is 12 variable, which is the country size, but every 3 variables which are the Kp, Ki, Kd of a PID will result in a specific response. Let's say the roll angle response is enhanced and the pitch angle response is getting worst by the same value, the total changed in objective function will not be noticed in ICA, so an indicator is used to show if the response of partitions (in this case 4 partitions) is improved. A comparing process is carried out between the imperialist and the colonies on each partition if the partition of the colony is better than of the imperialist then a switch process is carried out. Every partition represents a cultural value which improves the community and results in empowering the empire if adopted. Reflecting the process of equations:

$$country = [|p_1, p_2, p_3|, |p_4, p_5, p_6|, \dots, |p_{N_{var}-2}, p_{N_{var}-2}, p_{N_{var}}|] \quad (31)$$

Now calculate the cultural values of each colony and the imperialist:

$$\sum_{i=1}^{N_{var}/N_{cv}} cv_i = f(|p_{1+i}, p_{2+i}, p_{3+i}) \quad (32)$$

After that, a comparison and switching process is carried between colonies and imperialist. Cultural exchange step will note be add in every iteration but it was found out that by trial and error that each 4 iterations will give the best results decreasing the number will not allow the colonies to explore the search space and increasing it will not speed the convergence of finding the optimal solution.

5. SIMULATION RESULTS AND DISCUSSION

5.1 Simulation Parameters

The quadcopter parameters was calculated practically by taking the measurements from the DJI F-450 frame and then applying two tests on a2212 13t 1000kv(stator 22mm by 12mm, 13truns in each coil) brushless motor to calculated thrust factor and drag factor, also the total weight was calculated all the results can be seen in **Table 1**. Regarding the algorithm, all the setting found in **Table 2**.

5.2 Simulation Results

The imperialist competitive algorithm was applied using m-file code to the simulation model in Simulink in order to tune the PID's gains of the inner loop and then the outer loop. The inner loop which is 4 PID's (12 parameter) then the algorithm is applied a second time to tune the outer control loop 2 PID (6 parameters). After the iteration of the simulation, the following results were achieved considering that a number of iteration is 60 and number of population is 40, the step response for the Roll, pitch, and yaw which is essential to check for the stability of the quadcopter and the objective function. The step response is shown in **Fig. 9**. In addition, **Table 3** is showing the values of PID parameters of the optimal solution after the iteration is finished and the corresponding overshoot and Integral square error ISE.

The tracking of the desired trajectory in the space is shown in **Fig. 10**. It is noticed that the normal ICA optimization is not following the exact path of the trajectory and it is because that the algorithm is fully random and explore the search space blindly without clues. **Fig. 11** shows the tracking in each single axes to check each axis independently. Testing the outer PID controller with step response it shows that it needs more iteration to get better results or improve the algorithm to achieve a better result for the same setting.

The new Algorithm was tested also to tune the quadcopter control system. The Cultural Exchange Imperialist Competitive Algorithm was applied to the complete model in Simulink to tune the PID's gains of the inner (loop 4 PID's, 12parameters), then the algorithm was applied a second time to tune the outer control loop (2 PID's, 6 parameters). The following results were achieved considering that a number of iteration is 60 and number of population is 40, the step response for the Roll pitch and yaw which is essential to check for the stability of the quadcopter the step response is shown in **Fig. 12** was better than the previous. The tracking of the desired trajectory in the space is shown **Fig. 13** was very good the desired track and actual track was almost the same. It is because that the algorithm is has the indicators (check the cultural value) which enable the algorithm to converge fast to the optimum solution, **Fig. 14** shows the tracking in each single axes.

Ten experiment was carried out for the previous algorithm and the enhanced new algorithm because judgment cannot be based on one test since the algorithms are random procedures so it must be verified statistically and prove by statistics that the enhancement is better than the standard algorithm. As shown in **Table 5**, the normal algorithm (ICA) obtain (129.1791) as an average of 10 samples for the objective function where the improved algorithm (CEICA) has achieved (98.29343) as an average of 10 samples for the objective function. That means (23.91%) improvement in CEICA algorithm against the ordinary algorithm ICA.

6. CONCLUSIONS

Form the work done on the quadcopter system the PID controller is a suitable controller because it is simple and at the same time it can control the quadcopter and produce fast control signals fast to follow the sampling speed of the sensors and refresh rate of ESC. Optimization algorithms are essential for tuning the PID's, ICA algorithm in the literature was not used to tune multiple PID's simultaneously but in this work, it was tested and then improved to generate enhanced algorithm which handles multi PID tuning in a very efficient way. Cultural exchange Imperialist competitive algorithm (CEICA) was inspired from the fusion of cultural values exchanged between the colonies and imperialist in order to make the solution converge directly to the optimal point. The CEICA average objective function was (98.29343) and compared to ICA (129.1791) thus achieved enhancement by (23.91%).

7. REFERENCES

- Atashpaz-Gargari, E., and Lucas C., 2007, *Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition*, IEEE Congress on Evolutionary Computation, Suntec, Singapore.
- Beard, R.W., 2008, *Quadrotor Dynamics and Control*, Report, BYU ScholarsArchive, Brigham Young University, Provo.
- Benić, Z., Piljek, P., and Kotarski, D., 2016, *Mathematical modeling of unmanned aerial vehicles with four rotors*. Interdisciplinary Description of Complex Systems Vol.14, No.1, pp. 88-100.
- Bresciano, T., 2008, *Modelling, Identification, and Control of a Quadrotor Helicopter*, M.Sc. Thesis. Lund University – Department of Automatic Control, Lund, Sweden.
- Bouabdallah, S., 2004, *PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendal, Japan, pp. 2451–2456.
- Bouabdallah, S., 2006, *Design and Control of Quadrotors with Application to Autonomous Flying*, Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland.
- ElKholy, M., N., 2014, *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*, Master Thesis, School of Sciences and Engineering, American University in Cairo, Egypt.
- Kotarski, D., Benić, Z., and Krznar, M., 2016, *Control Design For Unmanned Aerial Vehicles With Four Rotors*, Interdisciplinary Description of Complex Systems Vol. 14, No. 2, pp. 236-245.
- Li, J., Li, Y., 2011, *Dynamic Analysis and PID Control for a Quadrotor*, IEEE International Conference on Mechatronics and Automation, Beijing, China.
- Magsino, E.R., Dollosa C.M., Gavinio, S., Hermoso, G., Laco, N., and Roberto L.A., 2014, *Stabilizing Quadrotor Altitude and Attitude through Speed and Torque Control of BLDC Motors*, 13th International Conference on Control, Automation, Robotics & Vision, IEEE, Marina Bay Sands, Singapore pp. 1825–1838.
- Oscar A., 2015, *Get Longer Flight Time On Multicopter*, website article https://oscarliang.com/maximize-flight-time-quadcopter/
- Pipatpaibul**,** P., and Ouyang P. R., 2011, *Quadrotor UAV control : online learning approach*, Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2011 August 28-31, 2011, Washington, DC, USA.
- Salih A. L., Moghavvemi M., Haider, Mohamed A. F., and Gaeid K. S., 2010, *Flight PID controller design for a UAV quadrotor*, Scientific Research and Essays Vol. 5, No. 23, pp. 3660-3667.

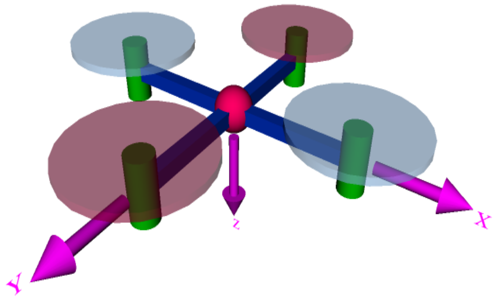

Figure 1. General Configuration for a quadcopter.

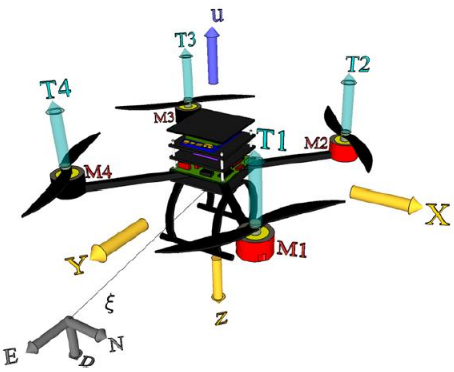

Figure 2. Quadcopter at hovering state with the main acting forces and torques.

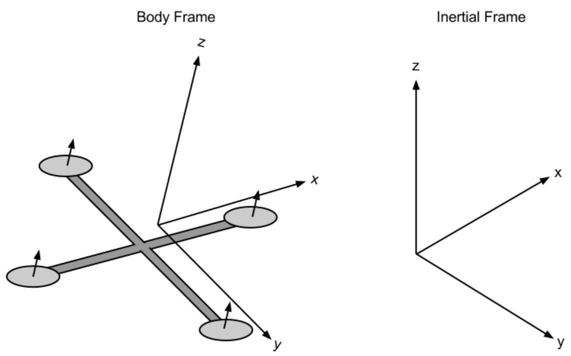

Figure 3. Inertial frame and body frame of the quadcopter.

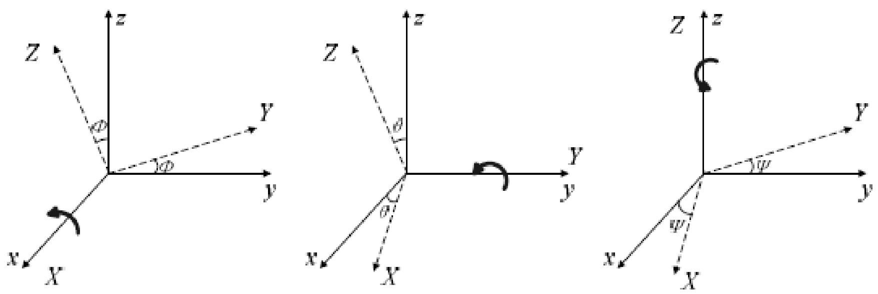

Figure 4. Rotations about X, Y, and Z axes.

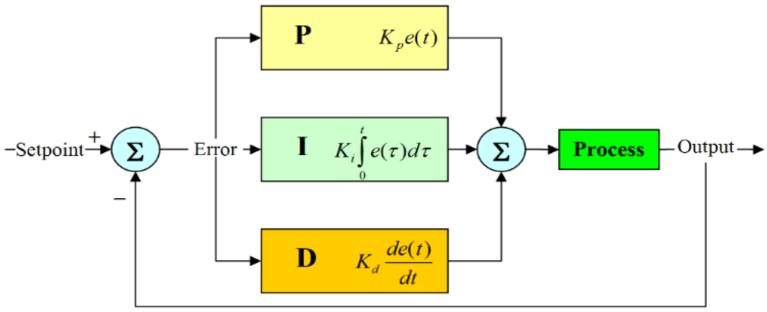

Figure 5. The general structure of PID controller.

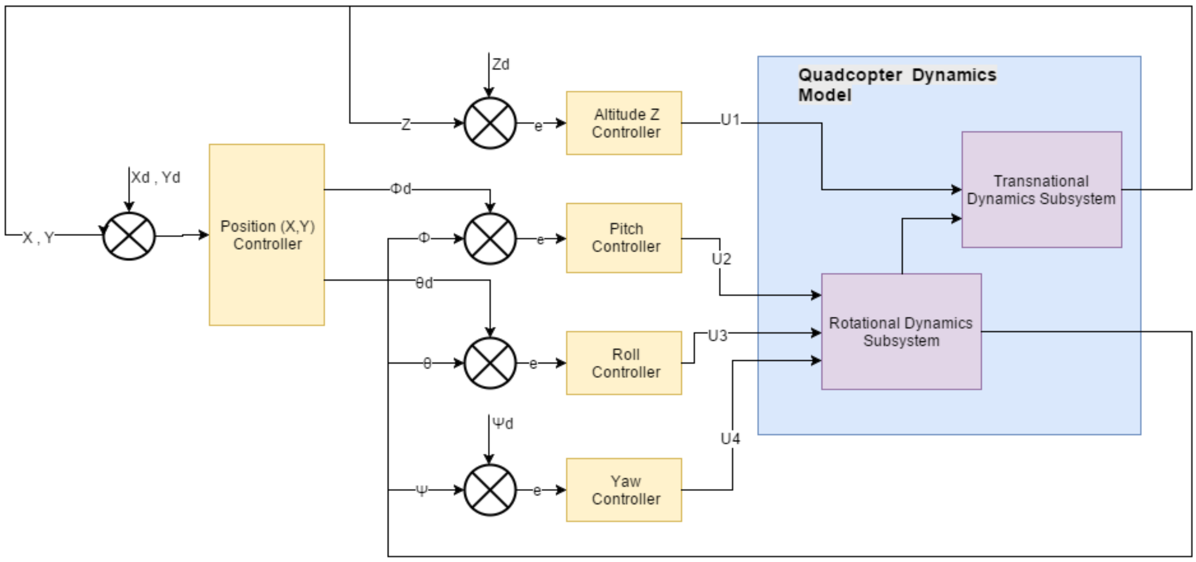

Figure 6. Block diagram of the complete quadcopter control system.

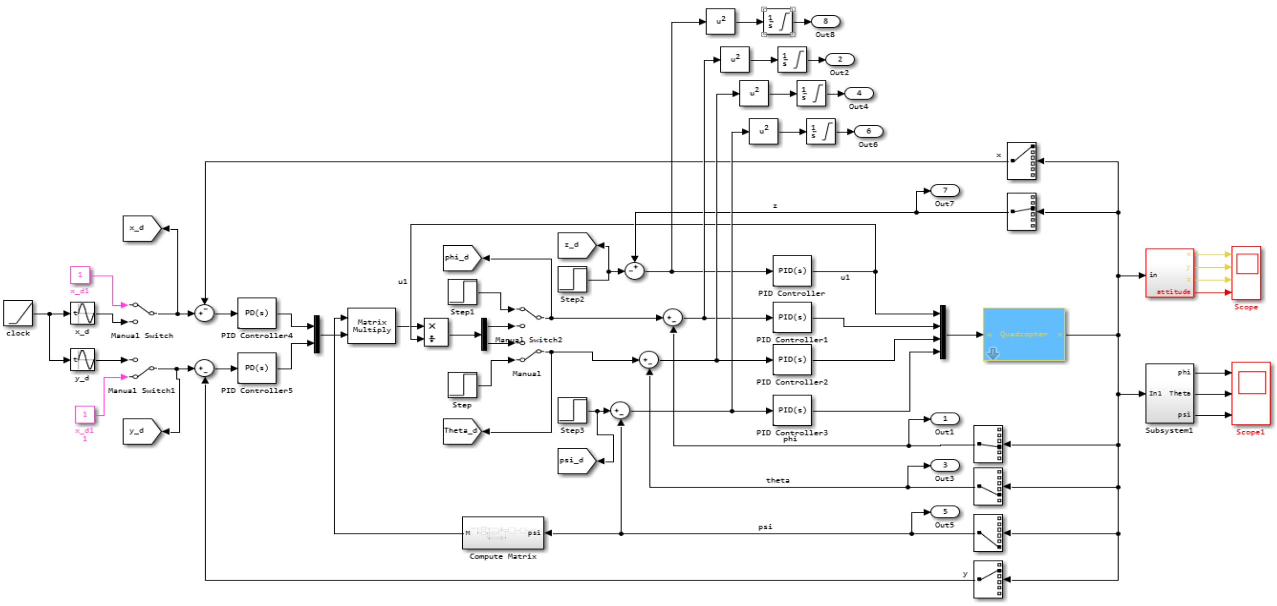

Figure 7. Complete Simulink model and control system for quadcopter (Noting that the rounded squares are the outputs to the M-file in Mat-lab).

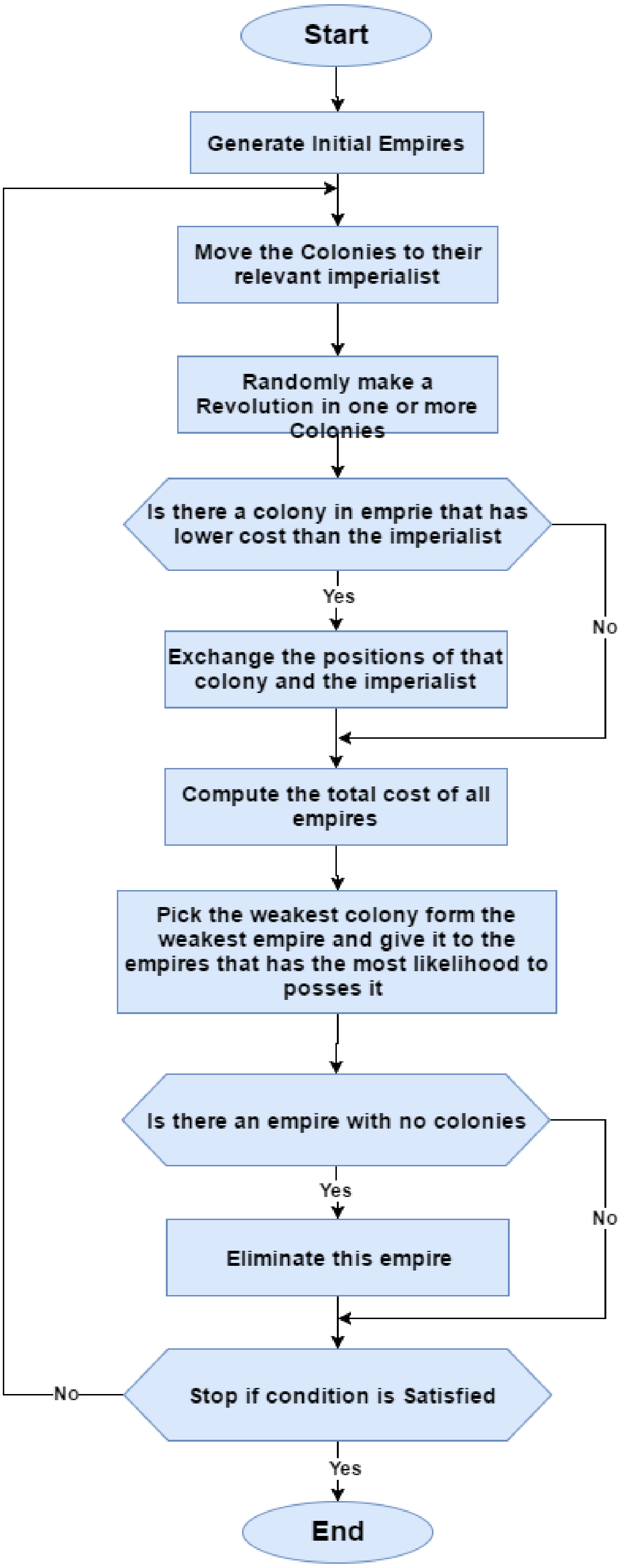

Figure 8. Flowchart of the imperialist competitive algorithm.

Figure 9. Step response of roll pitch and yaw angles in degrees after tuning the PID's with ICA (1: refers to a desired signal, 2: refers to actual signal).

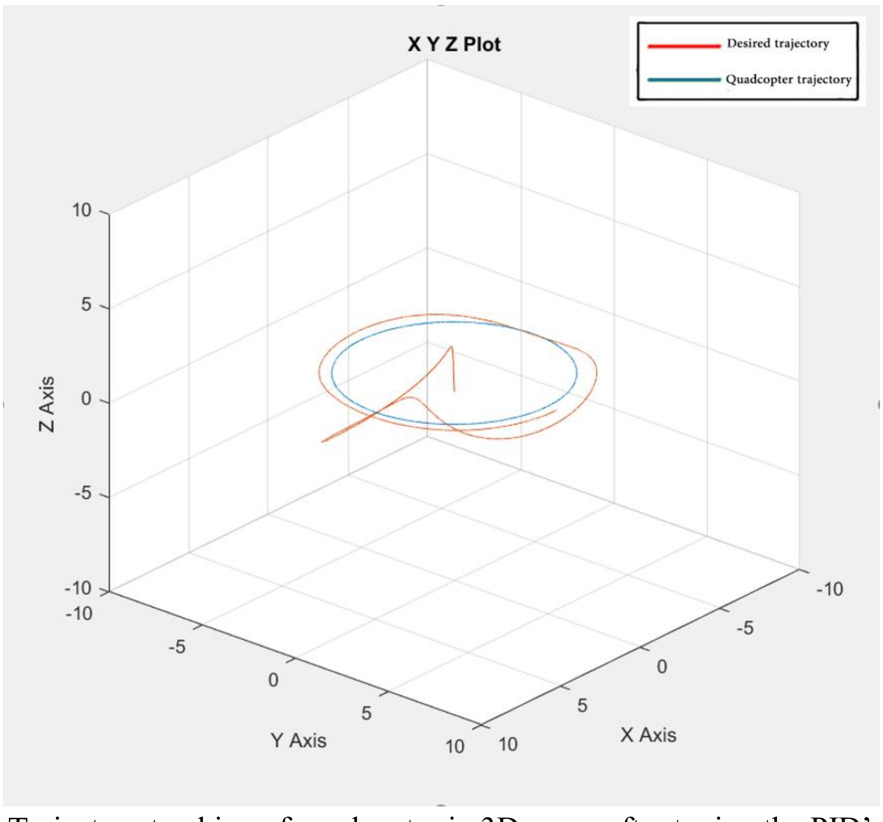

Figure 10. Trajectory tracking of quadcopter in 3D space after tuning the PID's with ICA.

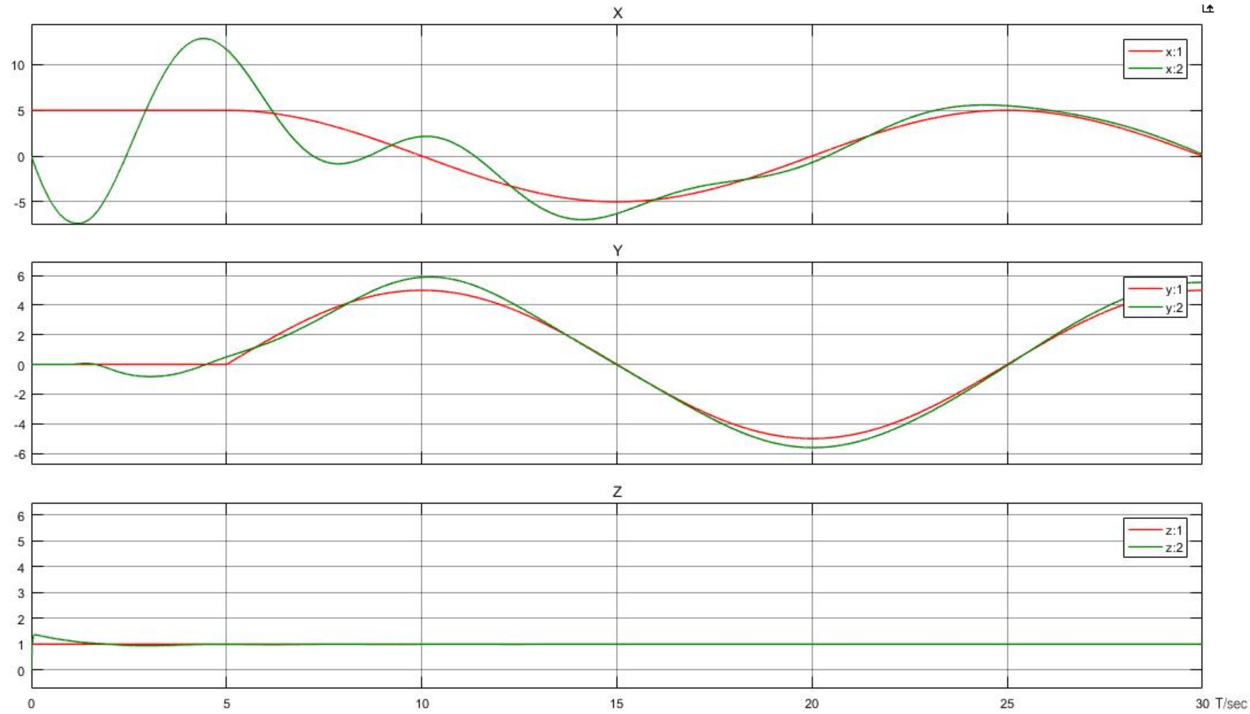

Figure 11. Position response X, Y and Z axis in meters response on single axes after tuning the PID's with ICA (1: refers to a desired signal, 2: refers to actual signal).

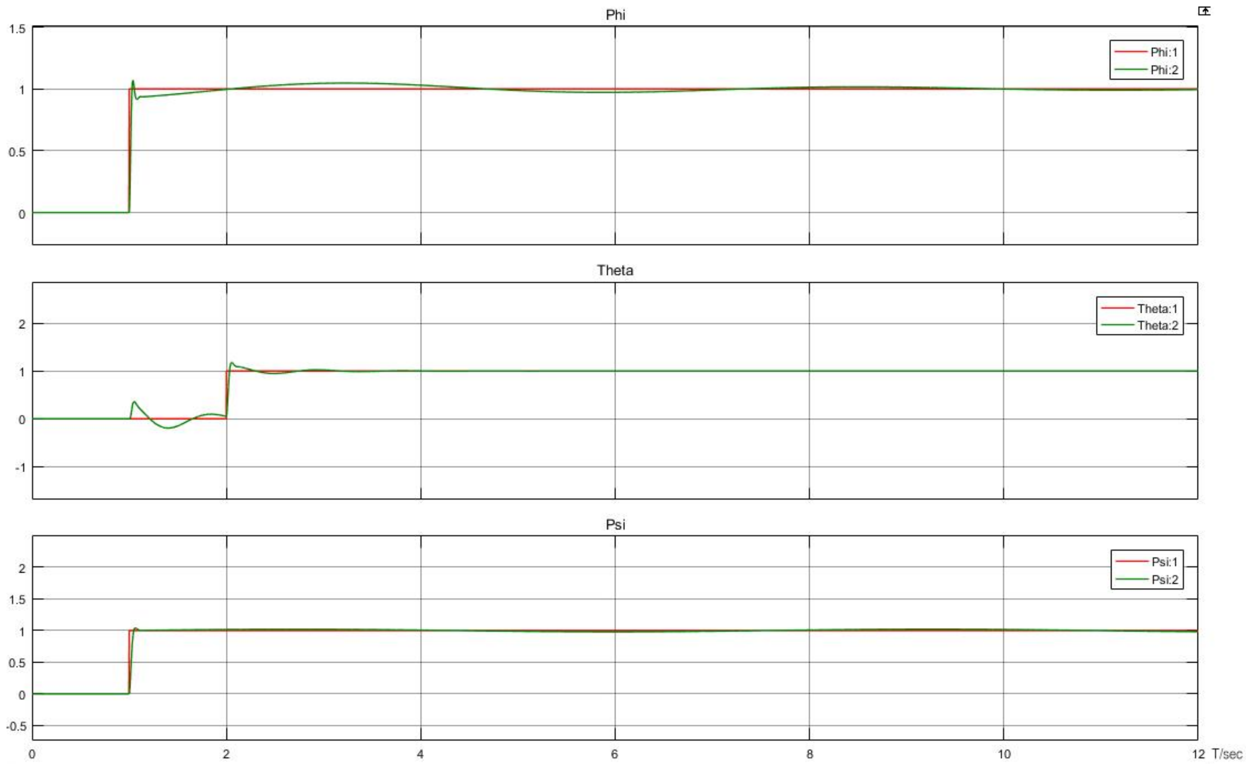

Figure 12. Step response of roll pitch and yaw angles in degrees after tuning the PID's with CEICA (1: refers to a desired signal, 2: refers to actual signal).

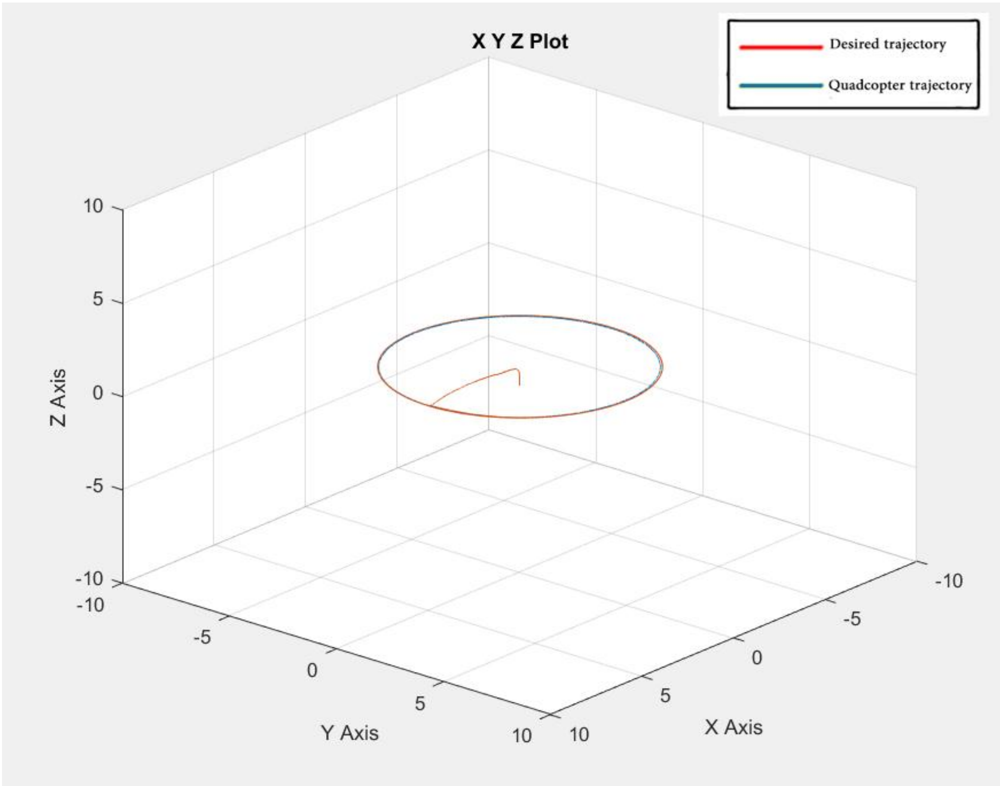

Figure 13. Trajectory tracking of quadcopter in 3D space after tuning the PID's with CEICA.

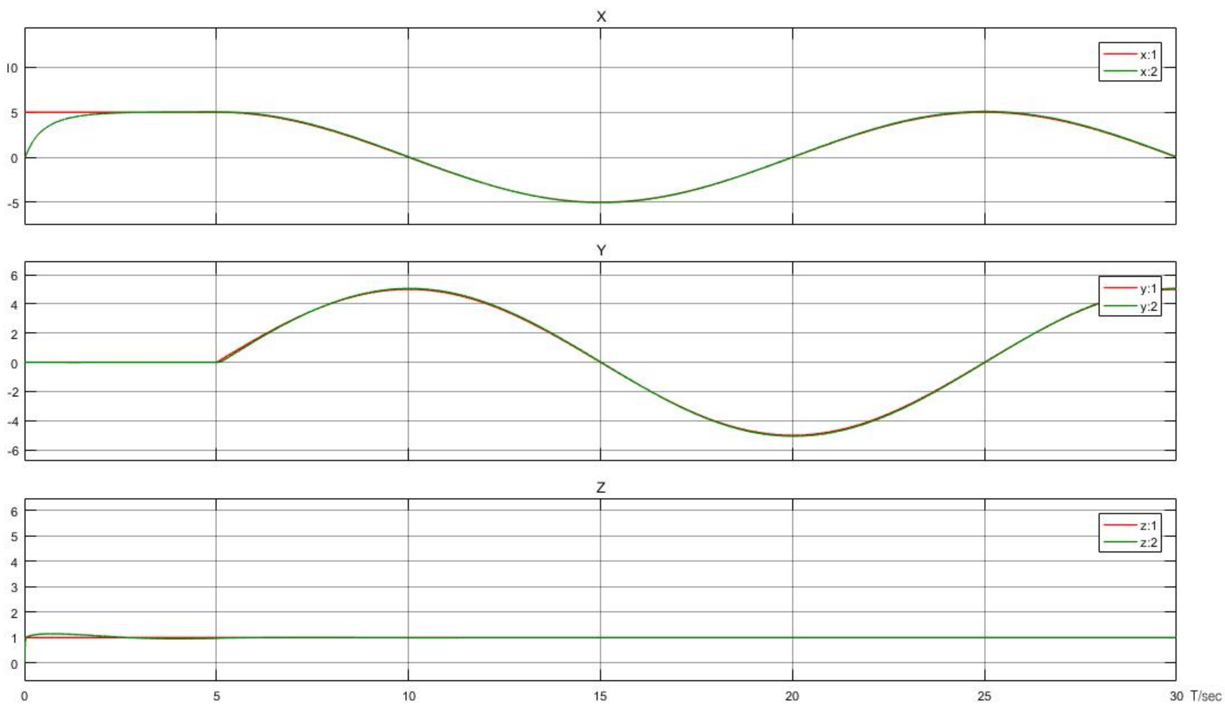

Figure 14. Position response X,Y, and Z in meters for step input signal after tuning the PID's with CEICA (1: refers to desired signal, 2: refers to actual signal).

Table 1. Model parameters, taken from real quadcopter system datasheets and experiment (DJI F-450 frame and Brushless motors a2212 13t 1000Kv).

Symbol	Value	Description
m	0.964 Kg	Total mass of quadcopter
l	0.22 m	Distance from center of quadcopter to the motor
I_{xx}	8.5532×10^{-3}	Quadcopter moment of inertia around X axes
I_{yy}	8.5532×10^{-3}	Quadcopter moment of inertia around Y axes
I_{zz}	1.476×10^{-2}	Quadcopter moment of inertia around Z axes
J_r	5.225×10^{-5}	Rotational moment of inertia around the propeller axis
b	7.66×10^{-5}	Thrust coefficient
d	5.63×10^{-6}	Drag coefficient

Table 2. Imperialist competitive setting.

Symbol	Value	Description
α	1	Selection Pressure
β	1.5	Assimilation Coefficient
Max_it	60	Maximum Number of Iterations
nPop	40	Population Size
nEmp	15	Number of Empires/Imperialists
pRevolution	0.05	Revolution Probability
mu	0.1	Revolution Rate
ξ	0.2	Colonies Mean Cost Coefficient

Table 3. PID values for roll, pitch, yaw and attitude and their corresponding overshot and ISE tuned by ICA.

Roll				
Kp	Ki	Kd	Overshoot %	ISE
8.514	46.06625	2.3264	7.5922	0.006646
Pitch				
Kp	Ki	Kd	Overshoot %	ISE
36.9288	28.0083	4.603	57.3633	0.006956
Yaw				
Kp	Ki	Kd	Overshoot %	ISE
-8.4373	31.4505	20.8355	66.9892	0.010992
Z-Altitude				
Kp	Ki	Kd	Overshoot %	ISE
81.0536	60	48.7586	12.8603	0.037952
X-position				
Kp	Ki	Kd	Overshoot %	ISE
4.0701	0.231	10.753	29.74198	0.3647
Y-position				
Kp	Ki	Kd	Overshoot %	ISE
5.5898	0.327	7.5787	15.4514	0.12848

Table 4. PID values for roll, pitch, yaw and attitude and their corresponding overshot and ISE tuned by CEICA.

Roll				
Kp	Ki	Kd	Overshoot %	ISE
0.29177	1.061	0.72886	7.1135	0.015718
Pitch				
Kp	Ki	Kd	Overshoot %	ISE
1.7825	26.0016	0.50338	18.3795	0.031516
Yaw				
Kp	Ki	Kd	Overshoot %	ISE
-0.00546	0.66847	0.73394	4.305	0.017597
Z-Altitude				
Kp	Ki	Kd	Overshoot %	ISE
44.4423	59.3334	59.9424	25.826	0.14864
X-position				
Kp	Ki	Kd	Overshoot %	ISE
8.0701	0.1	6.753	0.74198	0.2697
Y-position				
Kp	Ki	Kd	Overshoot %	ISE
6.5898	0.12859	9.5847	1.4514	0.30238

Table 5. Comparison between ICA and CEICA with 10 sample to prove the enactment statistically.

	ICA	**CEICA**
	129.4325	103.4353
	142.9757	95.841
	97.897	97.7699
	127.8194	136.046
	128.4142	77.5138
	152.5955	107.2114
	136.6101	76.4164
	123.8592	103.6134
	104.8666	89.2461
	147.3208	95.841
Statistical average	129.1791	98.29343