University of Baghdad
College of Engineering
JE
JOURNAL OF ENGINEERING

Journal of Engineering
journal homepage: www.jcoeng.edu.iq
Number 5 Volume 24 May 2018

Electrical, Electronics and communications, and Computer Engineering

Design of an Optimal Integral Backstepping Controller for a Quadcopter

Laith Jasim Saud*
Assistant Professor
Control & Systems Eng. Dept.
University of Technology
E-mail: Laithjasim15@yahoo.com

Alaq Falah Hasan
M.Sc. Student
Control & Systems Eng. Dept.
University of Technology
E-mail: alaq.falah@yahoo.com

ABSTRACT

In this paper, an Integral Backstepping Controller (IBC) is designed and optimized for full control, of rotational and translational dynamics, of an unmanned Quadcopter (QC). Before designing the controller, a mathematical model for the QC is developed in a form appropriate for the IBC design. Due to the underactuated property of the QC, it is possible to control the QC Cartesian positions (X, Y, and Z) and the yaw angle through ordering the desired values for them. As for the pitch and roll angles, they are generated by the position controllers. Backstepping Controller (BC) is a practical nonlinear control scheme based on Lyapunov design approach, which can, therefore, guarantee the convergence of the position tracking error to zero. To improve controller capability in the steady state against disturbances, an integral action is used with the BC. To determine the optimal values of the IBC parameters, the Particle Swarm Optimization (PSO) is used. In the algorithm, the controller parameters are computed by minimizing a cost function that depends on the Integral Time Absolute Error (ITAE) performance index.

Finally, different numerical simulations are provided in order to illustrate the performances of the designed controller. And for comparison purposes, a PID controller is designed and optimized using the PSO to control the quadcopter. The obtained results indicated a superiority in performance for the IBC over the PID controller based on some points among which are: a 13.3% and 30.5% lesser settling times for X and Y consequently, the ability to perform critical maneuvers that the quadcopter failed to do using the PID controller, and the capability of fast following up and conforming the changes of pitch (θ) angle (within 0.26 seconds) and roll (φ) angle (within 0.26 seconds), while the PID controller indicated a lag between the actual and the desired angles which reached 83.6% of the desired θ and 35.6% of the desired φ. In addition, the results showed the robustness of the designed IBC controller against external disturbances which represent the effect of running a quadcopter in an outdoor environment.

Keywords: quadcopter, integral backstepping control, particle swarm optimization, position control, attitude control.

*Corresponding author

Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2018.05.04

تصميم وحدة تحكم مثلى نوع الخطو الخلفي التكاملي لمروحية رباعية

الأستاذ المساعد الدكتور ليث جاسم سعود
قسم هندسة السيطرة والنظم
الجامعة التكنولوجية

الق فلاح حسن
قسم هندسة السيطرة والنظم
الجامعة التكنولوجية

الخلاصة

في هذا البحث، تم تصميم وحدة تحكم ذات خطو خلفي تكاملي للتحكم الكامل في الديناميكا الدورانية والخطية لمروحية رباعية بدون طيار. قبل تصميم وحدة التحكم، يتم تطوير نموذج رياضي للمروحية الرباعية في شكل مناسب لتصميم وحدة التحكم. بسبب عدد المحركات الاقل من درجات حرية الحركة للمروحية الرباعية، فمن الممكن التحكم في موقع المروحية الرباعية في المحورالديكارتي (X،Y ،Z) وزاوية الانعراج من خلال تحديد القيم المطلوبة بالنسبة لهم. أما بالنسبة لزوايا العطوف والخطران، فيتم توليدها بواسطة وحدات تحكم الموقع. المتحكم ذو الخطو الخلفي هو متحكم غير خطي مرتكز على نهج تصميم ليابونوف، هذا النهج الذي يمكن من ضمان اقتراب خطأ تتبع الموقع إلى الصفر. لتحسين قدرة وحدة التحكم في حالة الاستقرار ضد الاضطرابات، يتم استخدام إجراء تكاملي مع وحدة التحكم ذات الخطو الخلفي. لتحديد القيم المثلى لمعاملات المتحكم التكاملي ذو الخطو الخلفي، تستخدم خوارزمية سرب الجسيمات للتحسين الامثل. في هذه الخوارزمية، يتم حساب معاملات وحدة التحكم عن طريق تقليل دالة التكلفة التي تعتمد على مؤشر أداء تكامل مطلق الخطأ. ينتهي العمل بتنفيذ عمليات محاكاة رقمية مختلفة لتوضيح أداء وحدة التحكم المصممة. ولأغراض المقارنة، تم تصميم وحدة تحكم PIDوتحسينها باستخدام خوارزمية سرب الجسيمات للتحسين الامثل للسيطرة على المروحية الرباعية. وأظهرت النتائج التي تم الحصول عليها تفوق في الأداء ل المسيطر ذو الخطو الخلفي التكاملي على وحدة تحكم PIDاستنادا إلى بعض النقاط منها: زمن استقرار أقل بنسبة 13.3٪ و30.5٪ ل X وY على التوال ،القدرة على أداء مناورات حرجة المروحية الرباعية فشلت في القيام به باستخدام وحدة تحكم PID والقدرة على متابعة سريعة ومطابقة التغيرات في زاوية الخطران (θ) خلال (0.26 ثانية) وزاوية الانعراج (φ) خلال(0.26 ثانية)، في حين أشارت وحدة تحكم PID إلى تأخر بين الزوايا الفعلية والمطلوبة التي وصلت إلى 83.6٪ من زاوية الخطران المطلوبة و 35.6٪ من زاوية الانعراج المطلوبه.بالإضافة إلى ذلك، أظهرت النتائج متانة وحدة تحكم المقترحة ضد الاضطراب الخارجي الذي يمثل تأثير تشغيل المروحية الرباعية في بيئة الخارجية.

الكلمات الرئيسية: المروحية الرباعية، المسيطر ذو الخطو الخلفي التكاملي، سرب الجسيمات للتحسين الامثل، متحكم الموقع، متحكم التوجه.

1. INTRODUCTION

Quadcopters, among Multi copters, got and still are getting increasing interest due to their importance and growing use in many vital applications. For this and for the fact that a quadcopter system is characterized by being nonlinear, underactuated, and strongly coupled, quadcopters control represents a challenge for researchers and so bringing more interest and research. A quadcopter is an unmanned aerial helicopter with four rotors. The rotors are directed upwards and they have positioned in a square formation with equal space from the center of mass of the quadcopter.

The quadcopter is controlled by changing the angular velocities of the rotors. The major forces and moments acting on a quadcopter are those created by rotors. The four rotors form two pairs (front and back) and (left and right). One pair rotates clockwise, however, the other pair rotates counter-clockwise to balance the torques exerted upon the body of the quadcopter **Brito, 2009**. The free body diagram and axes of a quadcopter are shown in **Fig.1**. Increasing or decreasing the speed of the four rotors together generates vertical motion. Forward (backward)motion, which is related to the pitch (θ) angle of rotation about the y-axis, can be obtained by increasing the back (front) rotor speed and decreasing the front (back) rotor speed. A sideway motion, which is related to the roll (φ) angle of rotation about the x-axis, can be achieved by increasing the left (right) rotor speed and decreasing the right (left) rotor speed. Finally, the yaw motion given by angle (ψ) which represents the rotation about the z-axis, is obtained from the difference in the counter torque between each pair of rotors. Thus way, the quadcopter has six degrees of freedom, X, Y, Z, θ, φ, and ψ.Due to the property of the QC of being underactuated, it is possible to control the QC Cartesian positions (X, Y, and Z) and the yaw angle through ordering the desired values for them. As for the pitch and roll angles, they are generated by the position controllers.

The dynamical model of the quadcopter is the starting point for all studies related to quadcopter control. The mathematical model for the quadcopter dynamics and motion could be obtained either using Euler -Lagrange equations or Newton-Euler equations. Same results are obtained in both cases, **Bouabdallah, 2007,** and**, Bresciani, 2008.** In this work, Newton-Euler formulation is used.

Several control techniques can be used to control the quadcopter. Such techniques vary from the classical linear to the nonlinear ones. Some examples of these techniques are the Proportional-Integral-Derivative (PID), Linear Quadratic (LQ) controllers, **He, and Zhao, 2014,** and**, Bouabdallah, et al., 2004.** Backstepping control and sliding-mode controllers, **Bouabdallah, and Siegwart, 2005,** and, **Swarup,** and **Sudhir, 2014**, fuzzy control, **Rabhi, et al., 2011**, fuzzy PID controller, **Seidabad, et al., 2014,** and neural network control, **Burman, 2016**.

In this work, an IBC technique based on the Lyapunov stability theory is developed to stabilize the system on the desired trajectory. The reasons behind this choice are:

- It provides a systematic and recursive design methodology for nonlinear feedback control.
- It has sort of robustness against external disturbance and parameters uncertainty.
- It allows the system operating outside linear region (the hovering condition), thus does not need to simplify the dynamical model (ignoring coupling terms) as in the design of linear controllers which they suffer from a huge performance degradation whenever the quadcopter leaves the nominal conditions or performs aggressive maneuvers.

The idea of the IBC design is to select recursively some appropriate state variables as virtual inputs for lower dimension subsystems of the overall system and the Lyapunov functions are designed for each stable virtual controller. Therefore, the designed final actual control law can guarantee the stability of the total control system, **Bouabdallah, 2007.** Although the IBC method can provide a systematic process for controller design, a stable and satisfactory performance is not achieved without proper values for the IBC parameters. To get beyond satisfactory response, IBC parameters optimization is necessary, and thus in this work, PSO is used to off-line compute the optimal parameters for the IBC.

In addition to evaluating the performance of the proposed controller itself, its performance is compared to a PID controller designed and tuned with PSO. The reason for the selection of this controller as a reference for comparison is that a huge number of researchers dealt with it and indicated a fair response to it. The mechanism of control of the quadcopter with the PID controller is covered by many researchers among which are **Bresciani, 2008,** and **Mohamed, 2014**.

2. MATHEMATICAL MODEL

A nonlinear model for the kinematics and dynamics for the quadcopter is given here and based on Newton-Euler formalism**.** To develop the mathematical model of the quadcopter, sensible assumptions are established for the quadcopter to accommodate the controller design**, Bouabdallah, 2007**. The assumptions are as follows:

- The structure is assumed rigid.
- The structure is assumed symmetrical.
- The propellers are assumed rigid.
- The center of gravity and the body fixed frame origin are supposed to coincide.

- Thrust and drag are proportional to the square of rotor's speed.

2.1 Kinematic Model

In order to model the quadcopter kinematics, two frames have to be defined and as shown in **Fig.1**. In this figure, B represents the body coordinate system and E represents the earth coordinate system. The earth frame (E) is used to define the linear position ($\mathit{\Gamma}_E$ [m]), while the body frame (B) is used to define the forces (F_i[N]), the torques ($\tau_B[N.m]$), and the angular position (or attitude) $\mathit{\Theta}_B$ of the quadcopter. The quadcopter motion can be divided into two motions: the linear translational motion and the angular rotational motion. Thus, the model is described, respectively, in translational and rotational subsystems by Eq.(1) and Eq.(2):

$$\mathit{\Gamma}_E = [X \quad Y \quad Z]^T \qquad (1)$$

$$\mathit{\Theta}_B = [\varphi \quad \theta \quad \psi]^T \qquad (2)$$

The translational and rotational kinematic equations is obtained by means of the rotation R and transfer T matrices respectively. The expression of the rotation R and transfer T matrices are and given consequently by Eq.(3) and Eq.(4), **Bresciani, 2008.**

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} \qquad (3)$$

$$\mathrm{T} = \begin{bmatrix} 1 & t\theta s\varphi & t\theta c\varphi \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi/c\theta & c\varphi/c\theta \end{bmatrix} \qquad (4)$$

where *s*, *c*, and *t* are abbreviations for *sin*, *cos*, and *tan* respectively. The translational kinematic is written as:

$$\dot{\mathit{\Gamma}}_E = R\, V \qquad (5)$$

where $\dot{\mathit{\Gamma}}_E$ and V are respectively, the linear velocity vector with respect to the earth frame *E* and body frame *B*. The rotational kinematics can be defined as follows:

$$\dot{\xi} = \mathrm{T}\,\dot{\Theta}_{\mathrm{B}} \qquad (6)$$

where $\dot{\xi}$ and $\dot{\Theta}_{\mathrm{B}}$ are the angular velocity vectors with respect to the earth frame *E* and body frame *B*, respectively.

2.2 Dynamic Model

The dynamic model of the quadcopter is derived using Newton– Euler approach. It is useful to express the translational dynamic equations with respect to the earth frame E and rotational dynamic equations with respect to the body frame B, **Bresciani, 2008**. According to the Euler's first law of motion for rigid body dynamics, the translational dynamic equations of the quadcopter is written as follows:

$$\mathrm{m}\begin{bmatrix}\ddot{X}\\ \ddot{Y}\\ \ddot{Z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -\mathrm{m}g\end{bmatrix} + R\,U_1 \qquad (7)$$

where m represents the quadcopter mass, g is the gravity acceleration and $U_1$is the total thrust generated by the four rotors:

$$U_1 = \sum_{i=1}^{4} F_i = b\sum_{i=1}^{4} {w_i}^2 \qquad (8)$$

where *b [N s^2]* is the thrust coefficient and w_i (rad/sec) is the angular velocity of rotor i.The rotational motion equations are derived according to the Newton-Euler formalism:

$$J\ddot{\Theta}_B = \tau_B - \dot{\Theta}_B \times J\,\dot{\Theta}_B - \dot{\Theta}_B \times [0 \quad 0 \quad J_r\Omega_r] \qquad (9)$$

where J is an inertia matrix of the quadcopter, J_r is an inertia of the rotors, Ω_r relative speed and τ_B is the moments acting on the quadcopter in the body frame.

$$J = \begin{bmatrix} I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix} \qquad (10)$$

$$\Omega_r = (-w_1 - w_3 + w_2 + w_4) \qquad (11)$$

$$\tau_B = \begin{bmatrix}\tau_\varphi\\ \tau_\theta\\ \tau_\psi\end{bmatrix} = \begin{bmatrix}U_2\\ U_3\\ U_4\end{bmatrix} = \begin{bmatrix} bl({w_4}^2 - {w_2}^2)\\ bl({w_3}^2 - {w_1}^2)\\ d(-{w_1}^2 + {w_2}^2 - {w_3}^2 + {w_4}^2)\end{bmatrix} \qquad (12)$$

where l [m] is the distance between the center of the quadcopter and the center of a propeller and d [*N m s^2*] is the drag coefficient. Using Eq.(7) and Eq.(9), the motion equations of quadcopter can be derived as follows:

$$\ddot{\varphi} = \frac{U_2}{I_{xx}} + \frac{\dot{\theta}\dot{\psi}(I_{yy} - I_{zz})}{I_{xx}} - \frac{J_r}{I_{xx}}\Omega_r\dot{\theta} \qquad (13)$$

$$\ddot{\theta} = \frac{U_3}{I_{yy}} + \frac{\dot{\varphi}\dot{\psi}(I_{zz} - I_{xx})}{I_{yy}} + \frac{J_r}{I_{yy}}\Omega_r\dot{\varphi} \qquad (14)$$

$$\ddot{\psi} = \frac{U_4}{I_{zz}} + \frac{\dot{\theta}\dot{\varphi}(I_{xx} - I_{yy})}{I_{zz}} \qquad (15)$$

$$\ddot{Z} = -g + \frac{1}{m}(\cos\theta\ \cos\varphi)U_1 \qquad (16)$$

$$\ddot{X} = (\sin\psi\ \sin\varphi + \cos\psi\sin\theta\cos\varphi)\frac{U_1}{m} \qquad (17)$$

$$\ddot{Y} = (-\cos\psi\sin\varphi + \sin\psi\sin\theta\cos\varphi)\frac{1}{m}U_1 \qquad (18)$$

The second term in the rotational subsystem Eq.(13) to Eq.(15) is the gyroscopic effect resulting from the rigid body rotation in space and the third term in Eq.(13) and Eq.(14) is due to the gyroscopic effect resulting from the rotation of the propeller. With the renaming of the control inputs as:

$$U=\begin{bmatrix}U_1\\U_2\\U_3\\U_4\end{bmatrix}=\begin{bmatrix}b({w_1}^2+{w_2}^2+{w_3}^2+{w_4}^2)\\bl({w_4}^2-{w_2}^2)\\bl({w_3}^2-{w_1}^2)\\d(-{w_1}^2+{w_2}^2-{w_3}^2+{w_4}^2)\end{bmatrix} \tag{19}$$

The rotors velocities are required to be calculated from the control inputs, an inverse relationship between the control inputs and the rotors' velocities are required.

$$\begin{cases}{w_1}^2=\frac{1}{4b}U_1-\frac{1}{2bl}U_3-\frac{1}{4d}U_4\\{w_2}^2=\frac{1}{4b}U_1-\frac{1}{2bl}U_2+\frac{1}{4d}U_4\\{w_3}^2=\frac{1}{4b}U_1+\frac{1}{2bl}U_3-\frac{1}{4d}U_4\\{w_4}^2=\frac{1}{4b}U_1+\frac{1}{2bl}U_2+\frac{1}{4d}U_4\end{cases} \tag{20}$$

3. THE CONTROL STRATEGY

The system equations are rewritten in state space representation for controller design purposes. The state space model adopted by the control system is $\dot{X}_s=f(X_s,U)$, where X_s is the state vector and U is the control input vector. The state vector is selected as $X_s=[\varphi\ \dot{\varphi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\ Z\ \dot{Z}\ X\ \dot{X}\ Y\ \dot{Y}]$. In the design of the controller the state variables are selected as: $x_1=\varphi$, $x_2=\dot{\varphi}$, $x_3=\theta$, $x_4=\dot{\theta}$, $x_5=\psi$, $x_6=\dot{\psi}$, $x_7=Z$, $x_8=\dot{Z}$, $x_9=X$, $x_{10}=\dot{X}$, $x_{11}=Y$, $x_{12}=\dot{Y}$.

The state equations can be described as:

$$f(X_s,U)=\begin{bmatrix}x_1\\b_1U_2+x_4x_6a_1+a_2\Omega_rx_4\\x_3\\b_1U_3+x_2x_6a_3+a_4\Omega_rx_2\\x_5\\b_3U_4+x_2x_4a_5\\x_7\\-g+\frac{U_1}{m}cosx_1cosx_3\\x_9\\\frac{U_1}{m}u_x\\x_{11}\\\frac{U_1}{m}u_y\end{bmatrix} \tag{21}$$

where

$$a_1=\frac{I_{yy}-I_{zz}}{I_{xx}},a_2=\frac{-J_r}{I_{xx}},a_3=\frac{I_{zz}-I_{xx}}{I_{yy}},a_4=\frac{J_r}{I_{yy}},a_5=\frac{I_{xx}-I_{yy}}{I_{zz}},$$

$$b_1=\frac{1}{I_{xx}},b_2=\frac{1}{I_{yy}},b_3=\frac{1}{I_{zz}}$$

$$u_x=(\sin x_5\sin x_1+\cos x_5\sin x_3\cos x_1) \tag{22}$$

$u_y = (-\cos x_5 \sin x_1 + \sin x_5 \sin x_3 \cos x_1)$ (23)

From the dynamic model, it can be seen that the quadcopter is a multivariate, nonlinear, underactuated (6 DOF and only 4 inputs), with strong coupling system. The rotational motions do not rely on translational motion while the opposite is not true. Thus, double-loop control architecture is designed for the flying quadcopter's attitude and position control. The inner control loop is designed for stability and following of desired attitude. While the outer control loop is intended for quadcopter position control. The structure diagram is presented in **Fig .2**.

3.1 Integral Backstepping Controller Design

The control aim is to design an appropriate control law so that the state trajectory X_{sd}=$[\varphi_d \quad \theta_d \quad \psi_d \quad Z_d \quad X_d \quad Y_d]^T$ of the quadcopter system can track a desired reference trajectory. The description of the control system design of the quadcopter is similar for each one of the six controllable degrees of freedom (DOF), for simplicity only one DOF is considered.

The methodical design of the (IBC) is described as follows:

Step 1: Defining the tracking error:

$e_1 = \varphi_d - \varphi$ (24)

where φ_d represents a desired trajectory that is specified by a reference model. Then, the derivative of the tracking error can be represented as:

$\dot{e}_1 = \dot{\varphi}_d - \dot{\varphi}$ (25)

The first Lyapunov function is selected as:

$V_1(e_1, \chi_1) = \frac{1}{2}{e_1}^2 + \frac{1}{2}\lambda_1{\chi_1}^2$ (26)

where

$\chi_1 = \int_0^t e_1(\tau)d\tau$ (27)

The derivative of V_1 is:

$\dot{V}_1(e_1) = e_1\dot{e}_1 + \lambda_1\chi_1 e_1 = e_1(\dot{\varphi}_d - \dot{\varphi} + \lambda_1\chi_1)$ (28)

If we set the virtual control$\dot{(\varphi)}_d$ of $\dot{\varphi}$ as:

$\dot{(\varphi)}_d = \dot{\varphi}_d + \lambda_1\chi_1 + c_1 e_1$ (29)

where c_1 and $\lambda_1$are positive constants, then:

$\dot{V}_1 = -c_1{e_1}^2 \leq 0$ (30)

Step 2: Set the tracking-error of $\dot{\varphi}$ as:

$e_2 = \dot{(\varphi)}_d - \dot{\varphi} = \dot{\varphi}_d + \lambda_1\chi_1 + c_1 e_1 - \dot{\varphi}$ (31)

$\dot{e}_1 = \dot{\varphi}_d - \dot{\varphi} = -\lambda_1\chi_1 - c_1 e_1 + \dot{(\varphi)}_d - \dot{\varphi} = -\lambda_1\chi_1 - c_1 e_1 + e_2$ (32)

The derivative of e_2 is expressed as:

$\dot{e}_2 = \ddot{\varphi}_d + c_1(-\lambda_1\chi_1 - c_1 e_1 + e_2) + \lambda_1 e_1 - b_1 U_2 - \mathrm{a}_2\Omega_r x_4 - x_4 x_6 a_1$ (33)

The second Lyapunov function is chosen as:

$$V_2(e_1, e_2, \chi_1) = \frac{1}{2}{e_1}^2 + \frac{1}{2}{e_2}^2 + \frac{1}{2}\lambda_1{\chi_1}^2 \qquad (34)$$

The derivative of V_2 is:

$$\dot{V}_2 = -c_1{e_1}^2 + e_2[(1 + \lambda_1 - {c_1}^2)e_1 + \ddot{\varphi}_d - c_1\lambda_1\chi_1 + c_1e_2 \ - b_1U_2 - \mathrm{a}_2\Omega_r x_4 - x_4x_6a_1)] \qquad (35)$$

Step 3: For satisfying $\dot{V}_1(e_1, e_2, \chi_1) \leq 0$, the control input U_2 is selected as:

$$U_2 = \frac{1}{b_1}((1 + \lambda_1 - {c_1}^2)e_1 + \ddot{\varphi}_d - c_1\lambda_1\chi_1 + (c_1 + c_2)e_2 - \mathrm{a}_2\Omega_r x_4 - x_4x_6a_1) \qquad (36)$$

$$\dot{V}_1(e_1, e_2, \chi_1) = -c_1{e_1}^2 - c_2{e_2}^2 \qquad (37)$$

where c_2 a positive constant and the term c_2e_2 is added to stabilize the tracking error e_1. $\dot{V}_2(e_1, e_2, \chi_1) \leq 0$, $\dot{V}_1(e_1, e_2, \chi_1)$ is a negative semi-definite. Therefore , the control law will asymptotically stabilize the system.

The same steps are followed to extract U$_3$, U$_4$, U$_1$, u_x and u_y.

For pitch control (θ)

$$\begin{cases} e_3 = \theta_d - \theta \\ e_4 = \left(\dot{\theta}\right)_d - \dot{\theta} = \dot{\theta}_d + \lambda_2\chi_2 + c_3e_3 + \dot{\theta} \\ \chi_2 = \int_0^t e_3\,(\tau)\,d\tau \end{cases} \qquad (38)$$

$$U_3 = \frac{1}{b_2}((1 + \lambda_2 - {c_3}^2)e_3 + \ddot{\theta}_d - c_3\lambda_2\chi_2 + (c_3 + c_4)e_4 - \mathrm{a}_4\Omega_r x_2 - x_2x_6a_3) \qquad (39)$$

For yaw control (ψ)

$$\begin{cases} e_5 = \psi_d - \psi \\ e_6 = \left(\dot{\psi}\right)_d - \dot{\psi} = \dot{\psi}_d + \lambda_3\chi_3 + c_5e_5 + \dot{\psi} \\ \chi_3 = \int_0^t e_5\,(\tau)\,d\tau \end{cases} \qquad (40)$$

$$U_4 = \frac{1}{b_3}((1 + \lambda_3 - {c_5}^2)e_5 + \ddot{\psi}_d - c_5\lambda_3\chi_3 + (c_5 + c_6)e_6 - x_2x_4a_5) \qquad (41)$$

For linear Z motion control

$$\begin{cases} e_7 = Z_d - Z \\ e_8 = \left(\dot{Z}\right)_d - \dot{Z} = \dot{Z}_d + \lambda_4\chi_4 + c_7e_7 + \dot{Z} \\ \chi_4 = \int_0^t e_7\,(\tau)\,d\tau \end{cases} \qquad (42)$$

$$U_1 = \frac{m}{cosx_1cosx_3}((1 + \lambda_4 - {c_7}^2)e_7 + \ddot{z}_d - c_7\lambda_4\chi_4 + (c_7 + c_8)e_8 + g) \qquad (43)$$

For linear X motion control

$$\begin{cases} e_9 = X_d - X \\ e_{10} = \left(\dot{X}\right)_d - \dot{X} = \dot{X}_d + \lambda_5 \chi_5 + c_9 e_9 + \dot{X} \\ \chi_5 = \int_0^t e_9\,(\tau)\,d\tau \end{cases} \tag{44}$$

$$u_x = \frac{m}{U_1}((1 + \lambda_5 - c_9{}^2)e_9 + \ddot{x}_d - c_9\lambda_5\chi_5 + (c_9 + c_{10})e_{10}) \tag{45}$$

For linear Y motion control

$$\begin{cases} e_{11} = Y_d - Y \\ e_{12} = \left(\dot{Y}\right)_d - \dot{Y} = \dot{Y}_d + \lambda_6 \chi_6 + c_{11} e_{11} + \dot{Y} \\ \chi_6 = \int_0^t e_{11}\,(\tau)\,d\tau \end{cases} \tag{46}$$

$$u_y = \frac{m}{U_1}((1 + \lambda_6 - c_{11}{}^2)e_{11} + \ddot{y}_d - c_{11}\lambda_6\chi_6 + (c_{11} + c_{12})e_{12}) \tag{47}$$

where $(c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6)$ are positive constants. From Eq.(22) and Eq.(23) φ_d and θ_d can be found:

$$\varphi_d = \arcsin(u_x\, sin\psi_d - u_y\, cos\psi_d) \tag{48}$$

$$\theta_d = \arcsin\left(\frac{u_x\, cos\psi_d + u_y\, sin\psi_d}{cos\varphi_d}\right) \tag{49}$$

3.2 Tuning Using PSO

The PSO is a kind of swarm intelligence methods and a population-based algorithm that is normally used as an optimization tool. Each particle of the population is a candidate solution. In PSO, each particle navigates around the search (solution) space by updating their velocity according to its own and the other particles searching experience. Each particle tries to imitate traits from their successful peers to improve themselves. Further, each particle has a memory to keep tracking the previous best position (known as pbest) and corresponding fitness. The particle with the greatest fitness in the population is called gbest. Three steps are involved in the basic PSO algorithm, namely, generating particles' positions and velocities, velocity update, and finally, position update. First, by using the design upper, x_{max} and lower, x_{min} bound values, the initial positions $x_i^k$and velocities v_i^k of particles are randomly produced, as expressed in the following equations**, Rini, et al., 2011.**

$$x_i^k = x_{min} + rand(x_{max} - x_{min}) \tag{50}$$

$$v_i^k = x_{min} + rand(x_{max} - x_{min}) \tag{51}$$

In Eq.(50) and Eq.(51), the subscript and the superscript mean the *i*th particle at iteration k, respectively, while *rand* is a uniformly distributed random variable that can take any value between 0 and 1. The second step is to update the velocities of all particles according to the following expression:

$$v_i^{k+1} = w.v_i^k + n_1.rand.\left(pbest - x_i^k\right) + n_2.rand.\left(gbest - x_i^k\right) \quad (52)$$

Three weight factors, namely: inertia weight factor w, self-confidence factor n_1, and swarm confidence factor n_2, are combined in Eq.(52) to affect the particles direction. Lastly, the position of each particle is updated using its velocity vector as Eq.(53) and illustrated in **Fig.3**.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (53)$$

Repeat the three steps of (i) velocity update, (ii) position update, and (iii) fitness calculations until a stopping criterion is reached, **Rini, et al., 2011.** The flowchart of the PSO program is shown in **Fig. 4**.

In this work, the control parameters are calculated by minimizing a cost function defined by using the Integral Time Absolute Error (ITAE) performance index.

$$ITAE = \int_0^T t|e(t)|dt \quad (54)$$

The ITAE performance index has the benefits of producing smaller overshoots and oscillations than the IAE (integral of the absolute error) or the ISE (integral square error) performance indices. In this work, the following values are assigned for controller parameter optimization:

- Population/swarm size = 30.
- The number of maximum iterations = 30.
- The self, swarm confident and inertia weight factors, $n_1$and n_2 = 2 and *w* =1.5.
- The simulation time is equal to 10 seconds.

The variation of the fitness function with the number of iterations for IBC is shown in **Figs.5** to **9**. The optimal parameters of the controllers of the quadcopter system are listed in **Table.1**.

4. THE SIMULATION RESULTS AND DISCUSSION

In this section, the quadcopter model and the designed controllers will be simulated. The work is implemented in the Matlab/Simulink simulation environment. The quadcopter system is modeled with Simulink and the PSO algorithm is applied in Matlab. The model parameters values of the quadcopter system are recorded in **Table. 2**.

To judge the effectiveness of the designed optimal integral backstepping controller, three simulation tests have been performed on the quadcopter. In the first simulation test, the control goal is to take the quadcopter to a specific point in the space. The desired position/yaw is given by (X_d,Y_d,Z_d,ψ_d) = (1, 1, 1, 0). As mentioned earlier, the pitch and roll desired angles are generated by the position controllers. The performance of the designed IBC controller is compared with the performance of a PID controller. The PID is not the main issue here and it is designed for comparison purposes, and its parameters are tuned using PSO. **Figs. 10** to **14** show the responses for attitude states. In the transient response of φ and θ angles, the controllers show different behaviors. In case of the IBC controller, the actual trajectory conformed to the desired trajectory after 0.26 second, while in case of the PID controller, there was a lag between the actual trajectory and the desired one. The two controllers showed almost identical responses for ψ angle. **Figs. 15** to **17** show the responses for position states. A comparison using these figures between the IBC and the PID indicated that the IBC is with 13.3% and 30.5% lesser settling time for X and Y consequently and indicated that the PID is with 40% lesser settling time for Z. Moreover, both controllers gave zero overshoot and zero steady state error. The responses characteristics are summarized in **Table 3**. The control actions (U_1 through U_4) which represent the desired thrust force and torques for are shown in **Figs. 18** to **20**. In the second test, the objective is to test the ability of the proposed controller to track the path whether this path is linear, as the path shown in **Fig.21** which took 35 seconds, or curved (critical maneuvers) as show in **Fig. 22.** The desired trajectory is generated using command signals for X, Y, and Z and as shown in **Figs.23** to **25**. These figures also show the actual responses for Z, X and Y from which the actual path is composed. After 1, 2.5 and 1.5 seconds from ordering the trajectory, the actual responses of Z, X and Y, consequently, settled for the IBC controller. The PID controller could not perform this critical maneuver. Finally, the performance of the scheme is investigated in the presence external disturbance problem. Here, a disturbance is added to the quadcopter model in the form of additional forces acting on it to give the effect of running a quadcopter in an outdoor environment. The forces are added to the right hand side of the system's translational equations of motion Eq.(7) as Gaussian noise with zero mean and with a maximum value of 1 Newton. The simulation result of position tracking for the IBC approach in the presence of external disturbances is shown in **Fig. 26**. The results of the last test showed the robustness of the proposed controller against an external disturbance which represents the effect of running a quadcopter in an outdoor environment.

5. CONCLUSIONS

In this paper, an integral backstepping control algorithm has been designed to realize position and attitude control of a quadcopter. The controller consists of two portions: position control and attitude control. The position controller is used to track the desired trajectory in the Cartesian coordinate while the attitude controller is used to track the desired angles (φ_d and θ_d) obtained from the position controller and the desired yaw angle. The particle swarm optimization (PSO) has been used to determine the optimal values of the IBC conntroller parameters. Also, a PID controller has been designed and optimized using PSO to be used for performance comparison purposes with the IBC controller. The obtained results indicated:

- The ability of the IBC controller to control the QC stably when working in the near hovering case or in critical maneuvering case. In contrast, the PID controller failed in the maneuvering case. Compared to the PID controller, the IBC controller allows the system to operate outside the linear region (the hovering condition), as it is itself a nonlinear controller, and so does not need to simplify the dynamical model (ignoring coupling terms) as the case for the design of linear controllers.
- A better capability for the IBC controller in following up changes in pitch (θ) and roll (φ) angles. For the given tests, the IBC controller managed to follow up and conforming the changes of pitch (θ) angle within 0.26 seconds and roll (φ) angle within 0.26 seconds. On the other hand, the PID controlled failed in the follow up process and indicated a lag between the actual pitch (θ) and roll (φ) angles and the desired ones. For the given tests, this lag for example reached 83.8 % of the desired and 35.6 % of the desired φ.
- The given tests for step responses indicated 13.3% and 30.5% lesser settling time for X and Y, consequently, for the IBC controller compared to those for the PID controller.

- The robustness of the designed IBC controller against external disturbances which represent the effect of running a quadcopter in an outdoor environment.

6. REFERENCES

- Bouabdallah, S. and Siegwart, R., 2005, *Backstepping and Sliding mode Techniques Applied to an Indoor Micro Quadrotor*, Proceeding of IEEE International Conference on Robotics and Automation Barcelona, Spain, pp. 2259-2264.
- Bouabdallah, S., 2007, *Design and Control of Quadrotors with Application to Autonomous Flying*, Ph.D. thesis, Ecole Polytechnique Federale de Lausanne.
- Bouabdallah, S., Noth, A., and Siegwart, R., 2004, *PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. 2451–2456.
- Bresciani, T., 2008, *Modeling, Identification, and Control of a Quadrotor Helicopter*, Master’s thesis, Lund University.
- Brito, J. M., 2009, *Quadrotor Prototype*, Master’s thesis, Technical University in Lisbon.
- Burman, P., 2016, *Quadcopter Stabilization with Neural Network*, Project Report, the Faculty of the Graduate School of The University of Texas at Austin.
- He, Z. and Zhao, L., 2014, *A Simple Attitude Control of Quadrotor Helicopter Based on Ziegler- Nichols Rules for Tuning PD Parameters*, The Scientific World Journal.
- Mohamed, H. T., 2014, *Dynamic Modeling, and Control of a Quadrotor Using Linear and Nonlinear Approaches*, Master’s thesis, The American University in Cairo.
- Rabhi, A., Chadli, M., and Pegard, C., 2011, *Robust Fuzzy Control for Stabilization of a Quadrotor*, the 15th International Conference on IEEE, pp. 471-475.
- Rini, D. P., Shamsuddin, S. M. and Yuhaniz, S. S., 2011, *Particle Swarm Optimization: Technique, System, and Challenges*, International Journal of Computer Applications, Vol. 14, pp. 19-26.
- Seidabad, E. A., Vandaki, S. A., and Kamyad, V., 2014, *Designing Fuzzy PID Controller for Quadrotor*, International Journal of Advanced Research in Computer Science and Technology, Vol. 2, pp. 221-227.
- Swarup, A., and Sudhir. , 2014, *Comparison of Quadrotor Performance Using Backstepping and Sliding Mode Control,* Proceedings of the 2014 International Conference on Circuits, Systems and Control.

Table1. Optimal parameters of controllers.

Controller / Signal	IBC Parameters	PID Parameters
X- Signal	$c_9 = 2.0386$ $c_{10} = 1.9000$ $\lambda_5 = 0.0007$	P_1 = 14.4235 I_1 = 0.0014 D_1 = 8.2480
Y- Signal	$c_{11} = 2.4006$ $c_{12} = 2.2324$ $\lambda_6 = 0.0027$	P_2 = 15.8556 I_2 =0.0462 D_2 =11.0861
Z- Signal	$c_7 = 3.5462$ $c_8 = 3.3167$ $\lambda_4 = 0.0005$	P_3 = 19.6252 I_3 = 0.0051 D_3 = 6.9100

Phi- Signal	$c_1 = 17.8583$ $c_2 = 19.9880$ $\lambda_1 = 0.0019$	$P_4 = 19.3134$ $I_4 = 0.0002$ $D_4 = 10.5147$
Theta- Signal	$c_3 = 18.7340$ $c_4 = 19.2909$ $\lambda_2 = 0.0008$	$P_5 = 19.2005$ $I_5 = 0.0012$ $D_5 = 9.9962$
Psi- Signal	$c_5 = 2$ $c_6 = 1.2$ $\lambda_3 = 0.05$	$P_6 = 15.5$ $I_6 = 0.01$ $D_6 = 26$

Table 2. Parameters of the quadcopter.

Parameter	Description	Value	Unit
I_{xx}	Inertia on x-axis	7.5e-3	$kg.m^2$
I_{yy}	Inertia on y-axis	7.5e-3	$kg.m^2$
I_{zz}	Inertia on z-axis	1.3e-2	$kg.m^2$
l	Arm length	0.23	m
M	Quadcopter mass	0.650	kg
b	Thrust coefficient	3.13e-5	$N.s^2$
d	Drag coefficient	7.5e-7	$N.m.s^2$

Table 3. The response characteristics values.

IBC	X	Y	Z	PID	X	Y	Z
Settling time (sec)	2.6	2.5	2	**Settling time (sec)**	3	3.6	1.2
Overshoot (%)	0	0	0	**Overshoot (%)**	0	0	0
Steady state error	0	0	0	**Steady state error**	0	0	0

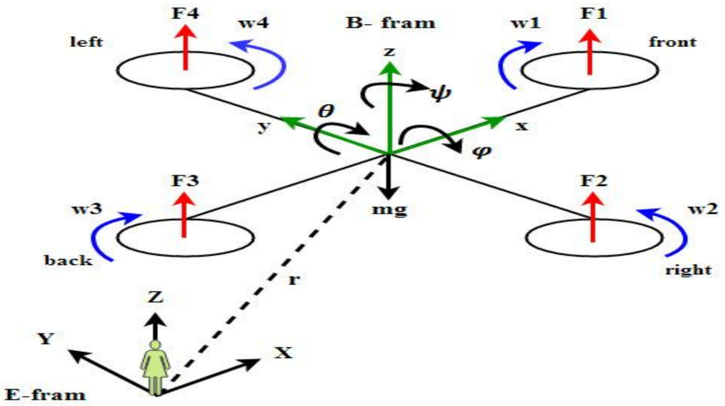

Figure 1. Quadcopter configuration.

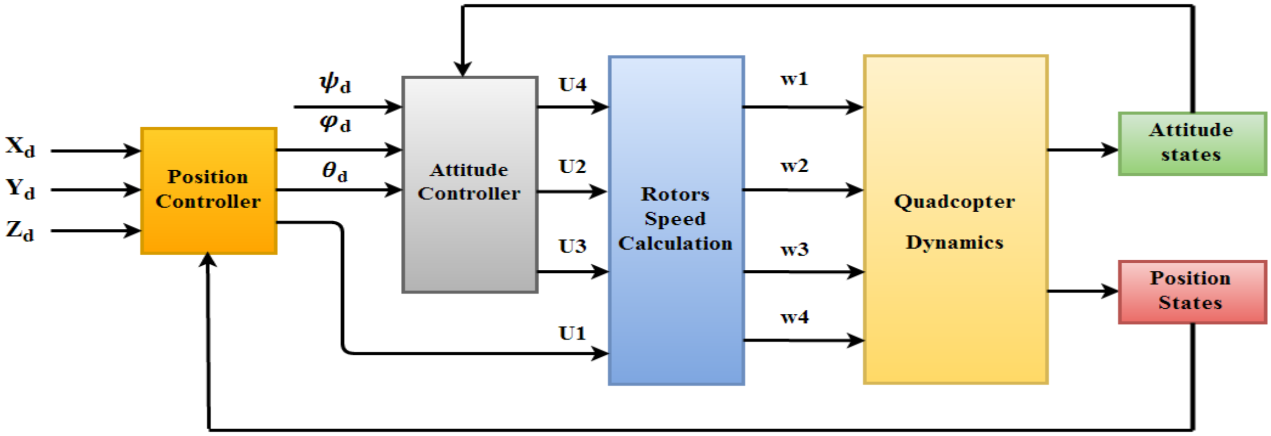

Figure 2. The quadcopter closed loop system.

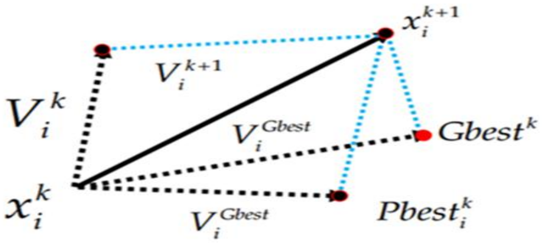

Figure 3. The depiction of the velocity and position updates in PSO.

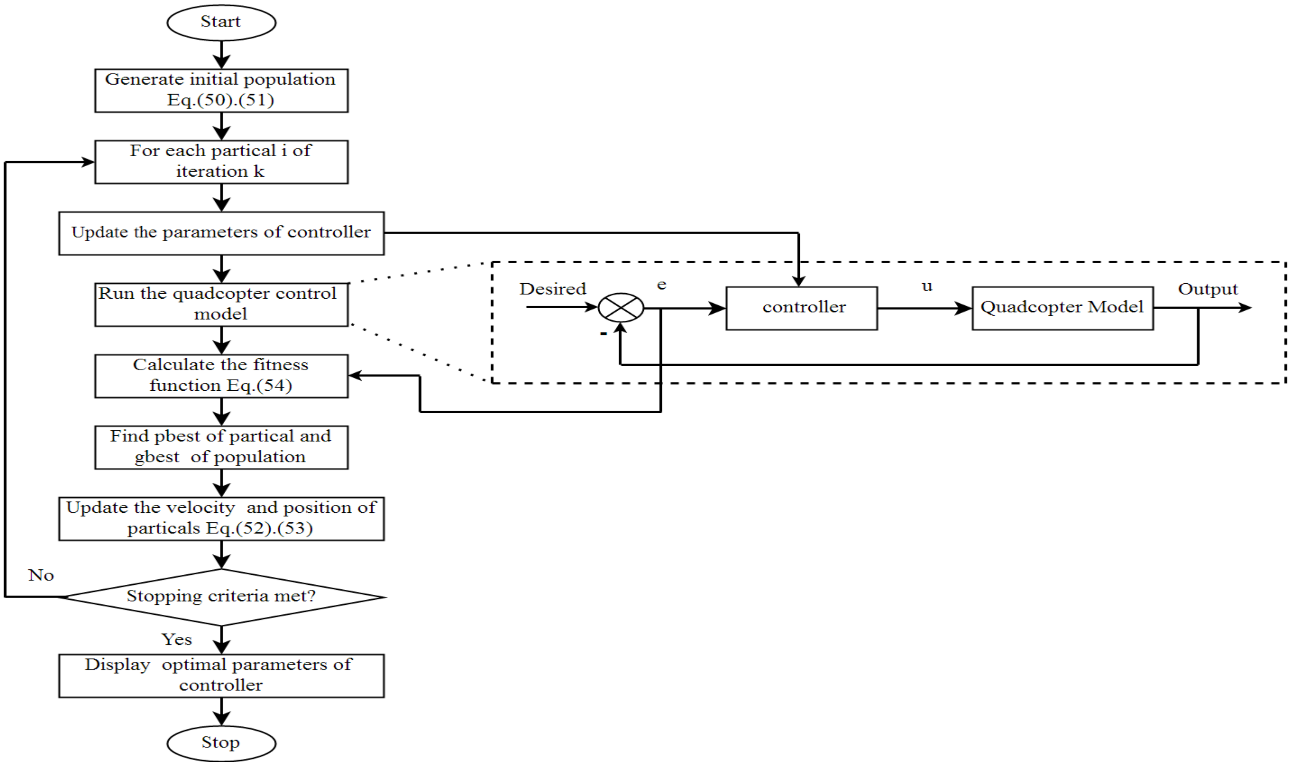
Start
Generate initial population Eq.(50).(51)
For each partical i of iteration k
Update the parameters of controller
Run the quadcopter control model
Calculate the fitness function Eq.(54)
Find pbest of partical and gbest of population
Update the velocity and position of particals Eq.(52).(53)
No
Stopping criteria met?
Yes
Display optimal parameters of controller
Stop
Desired
e
-
controller
u
Quadcopter Model
Output

Figure 4. The flowchart of the PSO program.

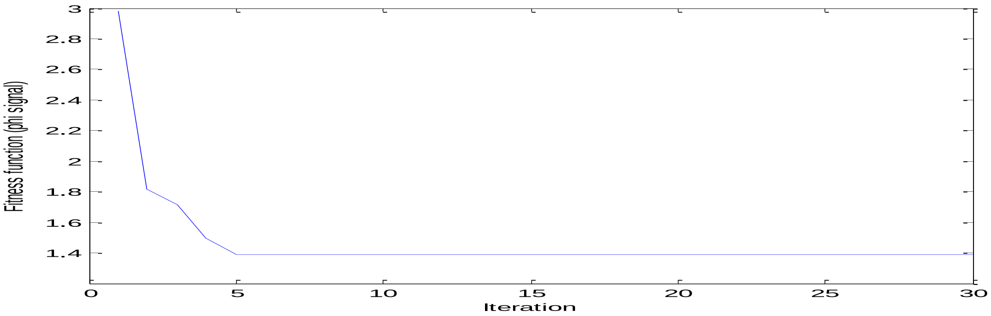

Figure 5. The convergence of fitness function for the phi signal in case of IBC with the number of iterations.

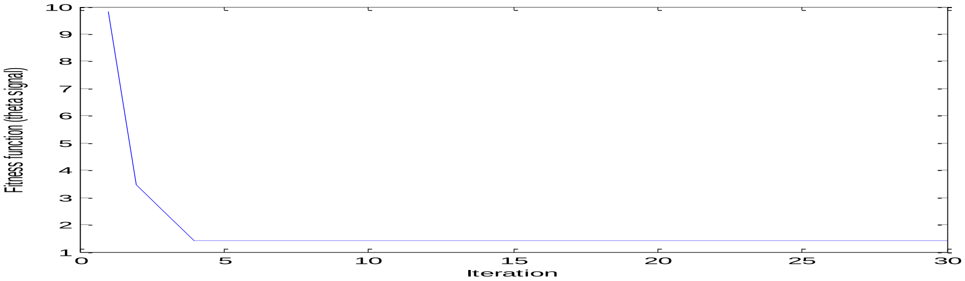

Figure 6. The convergence of fitness function for the theta signal in case of IBC with the number of iterations.

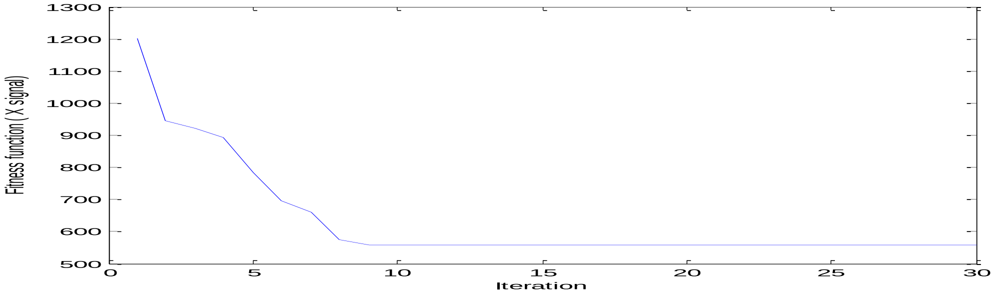

Figure 7. The convergence of fitness function for the X signal in case of IBC with the number of iterations.

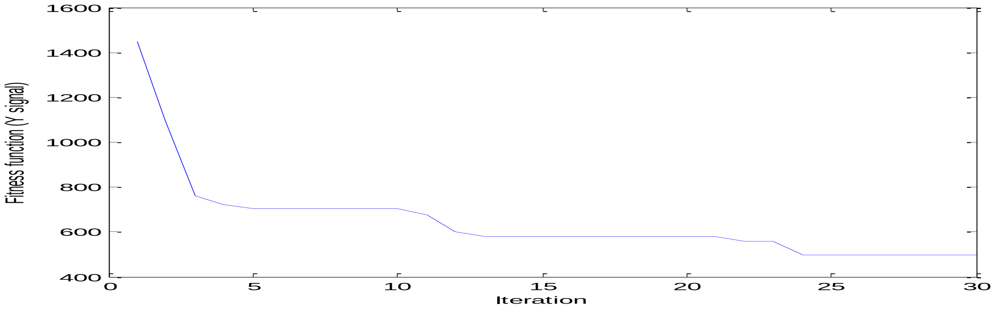

Figure 8. The convergence of fitness function for the Y signal in case of IBC with the number of iterations.

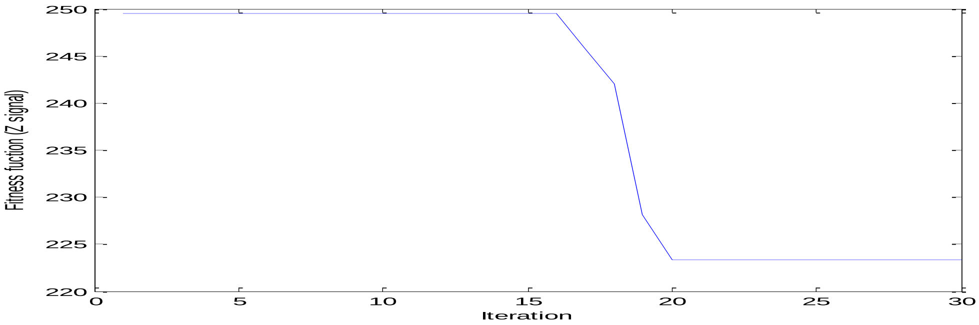

Figure 9. The convergence of fitness function for the Z signal in case of IBC with the number of iterations.

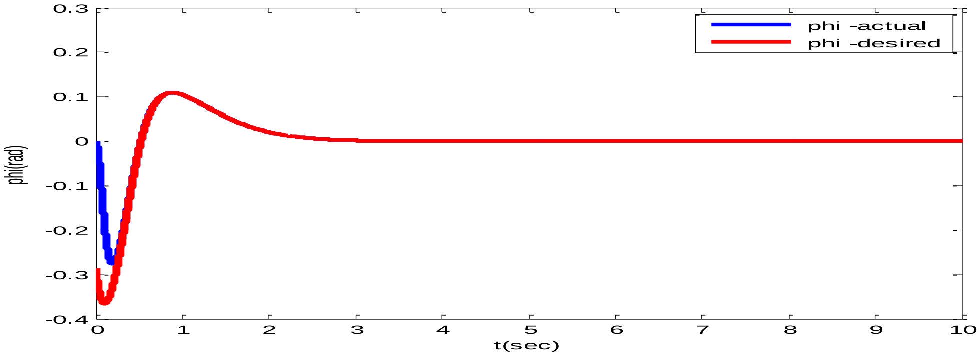

Figure 10. The phi signal response for the IBC controller.

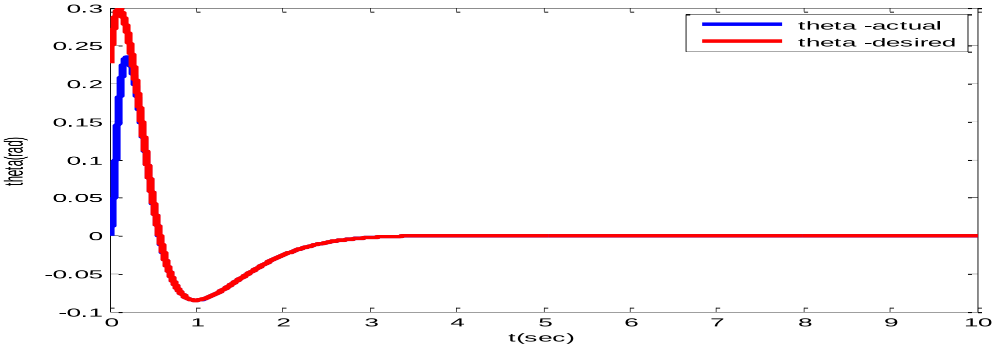

Figure 11. The theta signal response for the IBC controller.

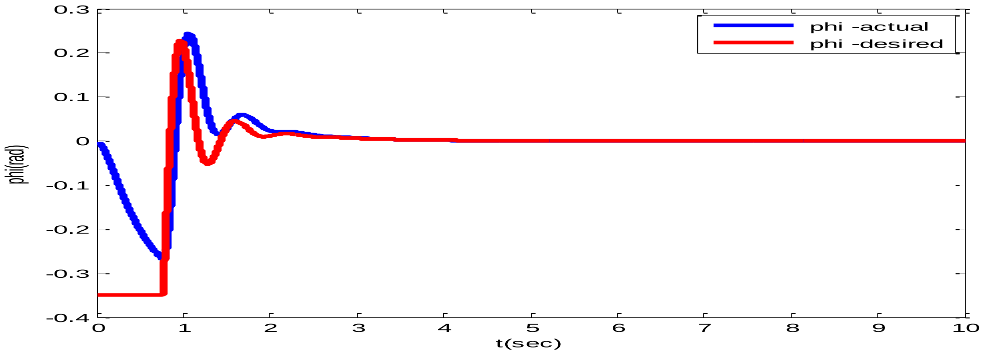

Figure 12. The phi signal response for the PID controller.

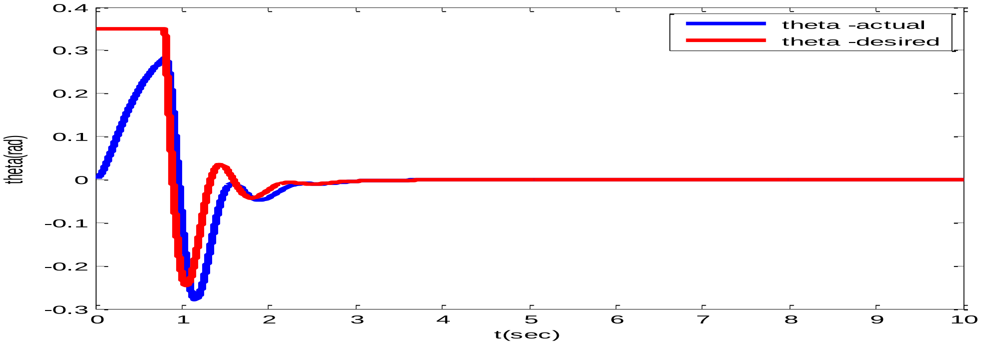

Figure 13. The theta signal response for the PID controller.

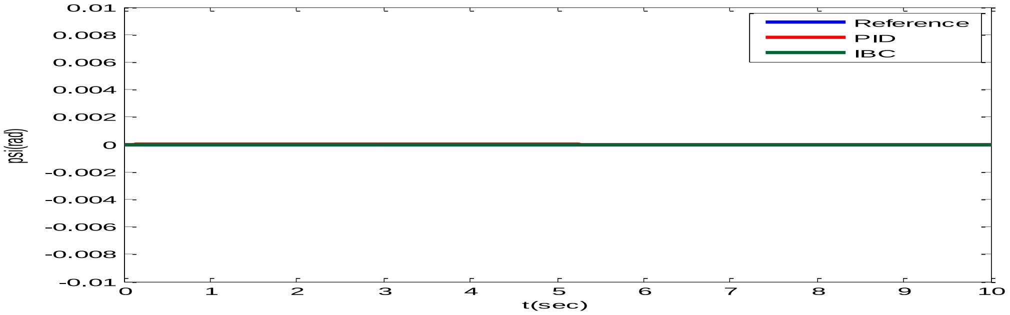

Figure 14. The psi signal response for the PID and the IBC controllers.

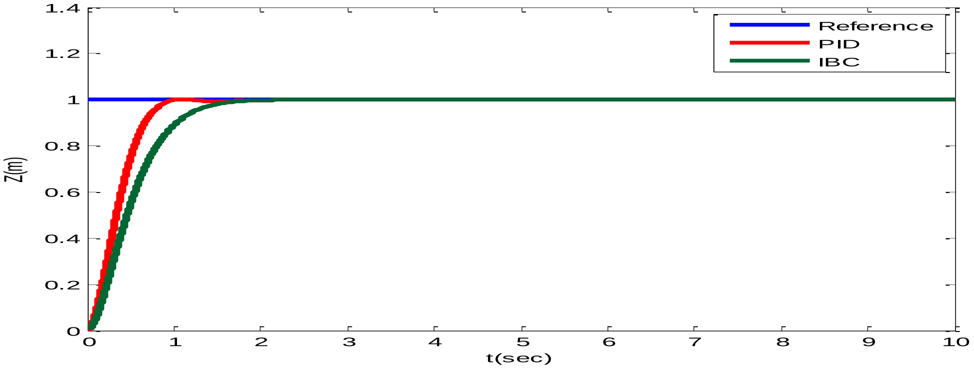

Figure 15. The Z signal response for the PID and the IBC controllers.

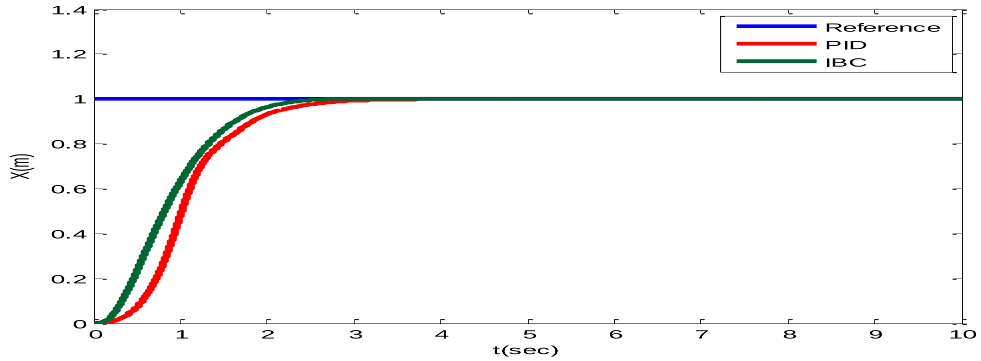

Figure 16. The X signal response for the PID and the IBC controllers.

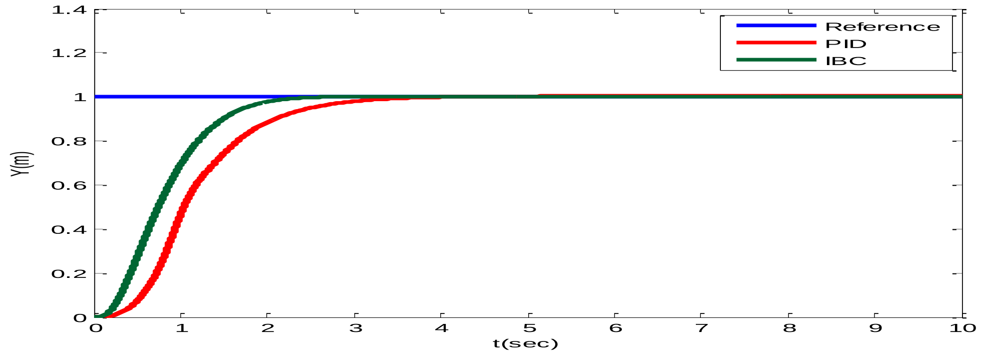

Figure 17. The Y signal response for the PID and the IBC controllers.

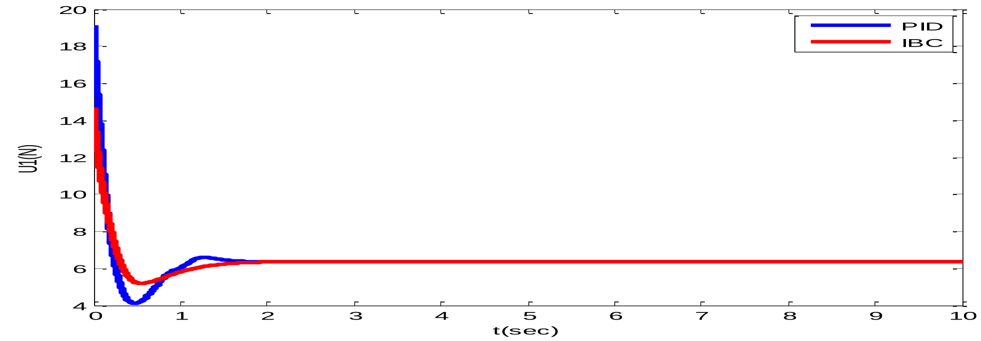

Figure 18. Control action U1 for the PID and the IBC controllers.

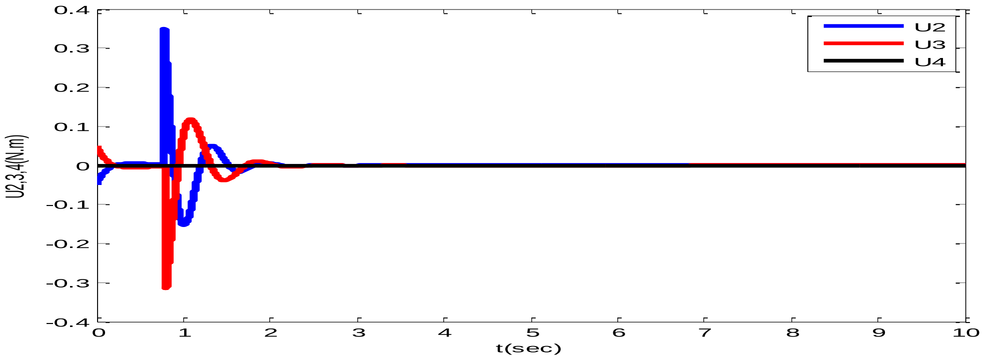

Figure 19. Control action U2, U3, and U4 for PID controller.

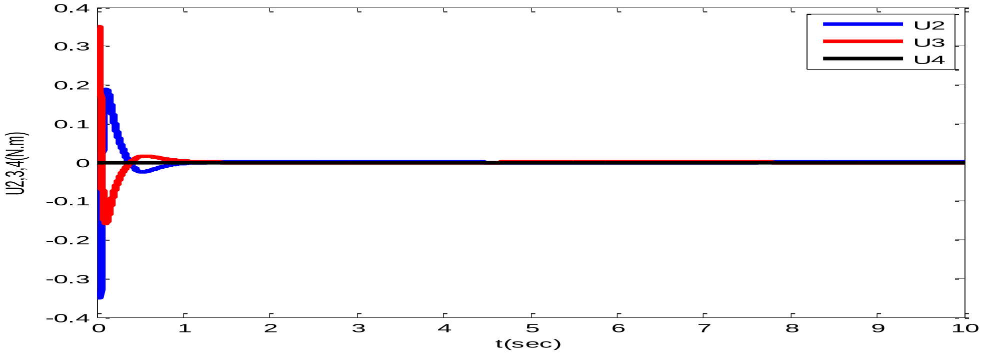

Figure 20. Control action U2, U3, and U4 for IBC.

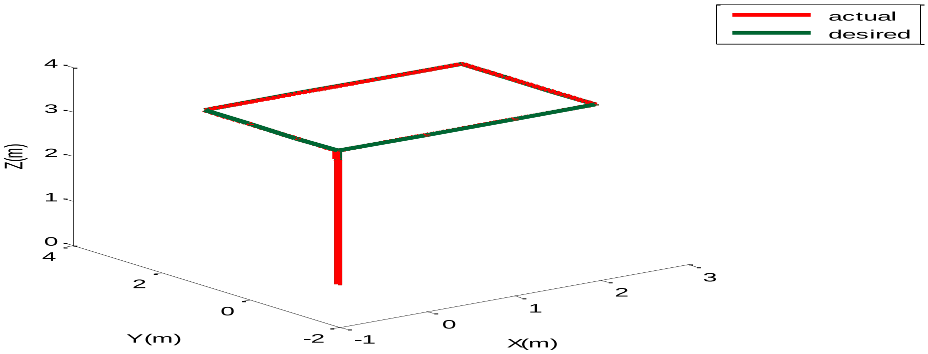

Figure 21. The first trajectory of the quadcopter in 3D space in case of the IBC.

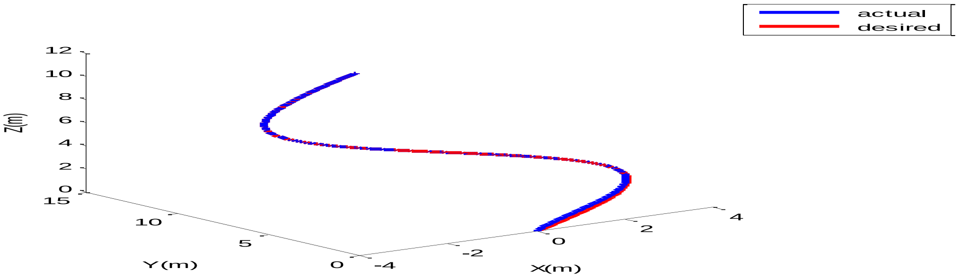

Figure 22. The second trajectory of the quadcopter in 3D space in case of the IBC.

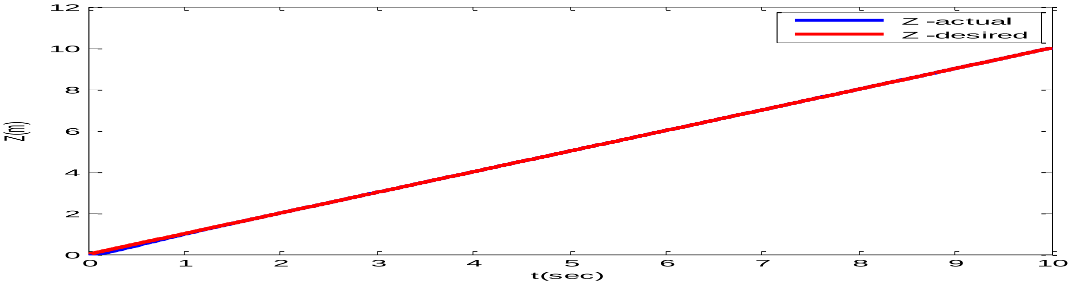

Figure 23. The Z signal response for the second trajectory of the quadcopter in case of the IBC.

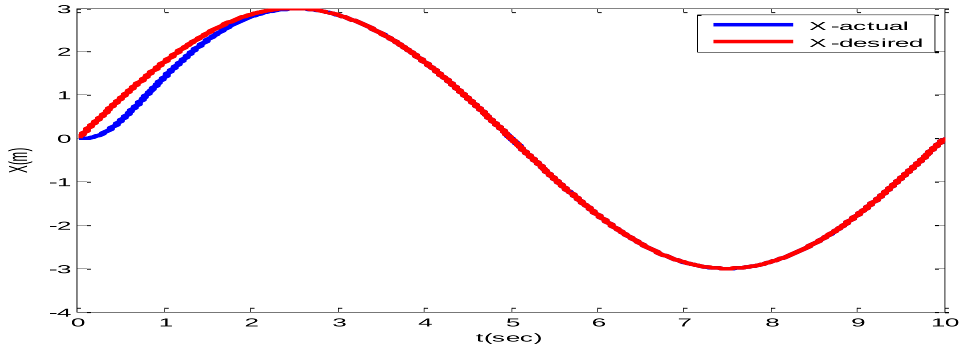

Figure 24. The X signal response for the second trajectory of the quadcopter in case of the IBC.

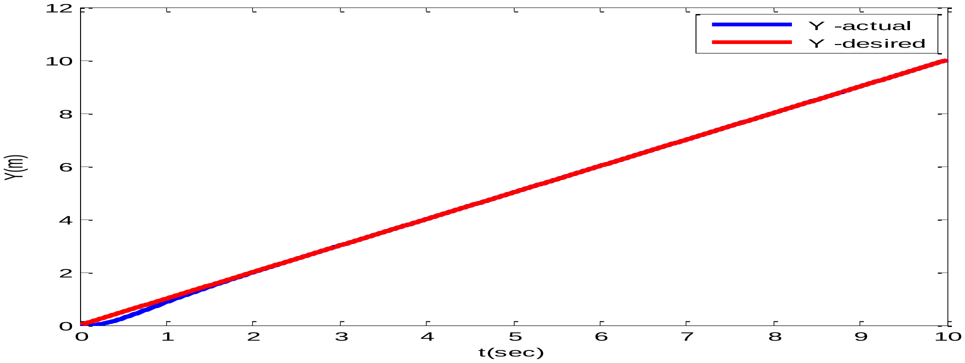

Figure 25. The Y signal response for the second trajectory of the quadcopter in case of the IBC.

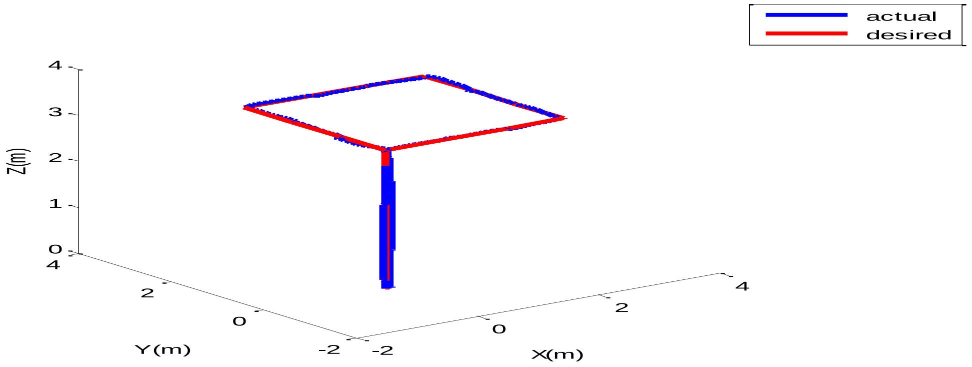

Figure 26. Trajectory of the quadcopter in the presence of external disturbance in case of the IBC.