Arab J Sci Eng (2017) 42:2777–2786
DOI 10.1007/s13369-017-2433-2

RESEARCH ARTICLE - ELECTRICAL ENGINEERING

Modelling and Genetic Algorithm Based-PID Control of *H*-Shaped Racing Quadcopter

Ahmed Alkamachi[1] · **Ergun Erçelebi**[2]

Received: 24 June 2016 / Accepted: 19 January 2017 / Published online: 8 February 2017

Abstract This work presents a detailed mathematical modelling of *H*-shaped racing quadcopter. The complete nonlinear dynamic model is obtained by exploiting Newton–Euler method as a common technique used in quadcopter modelling. A trajectory tracking controller is proposed, in which four PID controllers are designed to stabilize the quadcopter and to achieve the required altitude and orientation. However, a nested loop PID controllers are designed to track the desired x and y position of the quadcopter. The PID coefficients for the aforementioned proposed controllers are tuned using genetic algorithm (GA). The objective function for the GA was set so as to minimize the absolute tracking error, peak overshoot, and settling time for a step inputs. A MATLAB/Simulink environment is used to conduct the system model and the designed controller. The closed loop system performance is depicted for individual step inputs and for a predefined trajectory. Simulation results show a perfect step response performance and excellent trajectory tracking capability with a very low error budget. Finally, the controller robustness is examined and it is shown that the designed controller is robust against sensor noise, external disturbances, and model parameters uncertainties.

Keywords Genetic algorithm (GA) · *H*-shaped quadcopter · Newton–Euler model · PID controller · Racing quadcopter

✉ Ahmed Alkamachi
aa19272@mail2.gantep.edu.tr; amrk1978@gmail.com

Ergun Erçelebi
ercelebi@gantep.edu.tr

[1] Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

[2] Department of Electrical and Electronics Engineering, University of Gaziantep, Gaziantep, Turkey

1 Introduction

Unmanned aerial vehicles (UAVs) have shown an increasing interest thanks to the revolution in technologies which made the quadcopter affordable to civilian applications. Quadcopters have received a significant attention from researchers due to their broad range of applications in both military and civil fields.

Racing-type quadcopters which fulfil several important applications are not well discovered yet, and they are still in their infancy. The translational speed of the UAV is of considerable importance in several application like rescue, emergency, civil defence, and martial field; and consequently, racing quadcopters are gathering people and researchers attention rapidly.

The racing quadcopter consists of "H"-shaped main body frame with four rotor groups distributed on its tips as shown in Fig. 1. Each rotor group consists of a brushless DC motor that drives a fixed pitch propeller directly. Two opposite propellers are spinning in clockwise (CW) direction, while the remaining are spinning in counter clockwise (CCW) direction. The reason behind this configuration is to zero the resultant drag torque of the spinning propellers so the quadcopter body will not revolve around itself during hover.

The quadcopter can control its six DOFs by speeding up or slowing down its rotors' speed individually. The altitude can be controlled by either increasing or decreasing the angular velocity of the four rotors simultaneously. The quadcopter body can be rotated around the vertical axis (yawing) by making a difference in revolving speed between the CW and CCW propellers while maintaining the overall generated thrust constant. The quadcopter body can also be leaned around the lateral axis (pitching) or around the longitudinal axis (rolling). For pitching, the speed of rotors 2 and 3 should be increased (decreased), while reducing (increas-

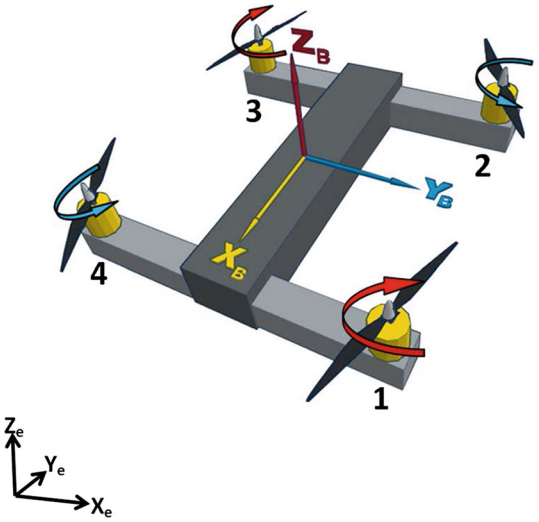

Fig. 1 Racing quadcopter configuration and reference frames

ing) that of rotors 1 and 4 with the same amount to maintain the quadcopter at the same altitude. In the same way, the quadcopter can make rolling motion by speeding up (slowing down) rotors 1 and 2 and slowing down (speeding up) rotors 3 and 4. The quadcopter can move by tilting its body to the desired direction. It can move along x direction if it is pitched and in y direction if it is rolled. In general, the more the quadcopter tilts, the faster it travels along the tilt direction.

Motivated by the UAVs advantages and applications, comprehensive modelling and control methods have been proposed. Quadcopter's dynamic models have been studied by the researchers who obtained their models using Newton–Euler and/or Lagrangian approach(s). In Erginer et al. [1], describe the quadcopter model mathematically and design a PD controller to stabilize it. In [2], the authors propose a dynamic model for X-type quadcopter. The gyroscopic and aerodynamic effect is considered through the modelling. A nonlinear model for a quadcopter is presented in [3], where the authors emphasize on the rotational motion control. Zemalache et al. propose a quadcopter with two of its rotors are bidirectional, and they derive a dynamic model for it in [4]. They make use of backstepping method to control the proposed quadcopter. Tayebi and McGilvray [5] propose a PD^2 feedback controller for exponential stabilization of quadcopter attitude. Bouabdallah et al. design and control an indoor micro quadcopter in [6]. They use a classical controller (PID) for a simplified model and a modern approach (LQ) based on a more complete dynamics. Both methods are tested in simulation environment and on a test bench [7]. Bresciani [8] uses Newton–Euler formalism to obtain the dynamic system model of a quadcopter helicopter. He designs a PID controller and evaluates the complete system by means of simulation and a test platform. Lagrangian mechanics is used to write a mathematical model for the hardware in the loop (HIL) test platform, and PD controller is designed to control the derived model in [9]. Çetinsoy [10] derives a model for holonomic quadcopter UAV and designs eight PID controller to control the quadcopter pose.

In the controller robustness point of view, some authors examined their designed controllers ability in rejecting disturbances, suppressing noise, and reducing the model uncertainties effects. Benallegue et al. [11] discuss the robustness of a feedback linearization-based controller against external disturbances and model uncertainties. A multivariable PD controller is designed to stabilize the quadcopter attitude, and the controller is tested for its robustness against the variation in the model parameters in [12]. Li and Li [13] test a PID controller for its robustness in regulating a quadcopter 6 DOFs.

The main contribution of this work is to derive a mathematical model for H-shaped racing quadcopter. Furthermore, a PID controller is proposed in which its parameters are tuned using GA with an objective function that compromise between the tracking error, peak overshoot, and the settling time of the output signal.

The rest of the paper sections is set as follows: Section 2 describes the static and dynamic mathematical model of the quadrotor type under consideration. The proposed control algorithm and design is presented in Sect. 3. Section 4 presents a trajectory tracking test, while the controller robustness is discussed in Sect. 5. Finally, the paper concludes in Sect. 6.

2 System Modelling

The dynamic model of a system is the mathematical equations that integrates all the forces that can act on a system at a given time [14].

The quadcopter is a multi-input multi-output (MIMO) system and has a complex structure with high nonlinear dynamics, thus obtaining its mathematical model is not an easy task [1,15]. In this section, the detailed mathematical model for the H-shaped quadcopter is developed.

2.1 Reference Frames

In this part of the work, two main reference frames are defined and the matrices used to switch between the reference frames are developed. These frames are the earth frame (F_e) and the body frame (F_b) as shown in Fig. 1.

The earth (inertial) frame is defined with ordered triplet vectors: X_e, Y_e, and Z_e which are pair-wise perpendicular. In what follows, the variables resolved to the earth frame are subscripted with letter "e". The body frame is described by the orthonormal basis vectors: X_b, Y_b, and Z_b. In the follow-

ing sections, a "b" letter subscript is used to denote variables resolved to the body frame. The earth and body frames differ in that the origin of the body frame can be drawn from that of the earth frame, and the axes of the body frame can change orientation with respect to those of the earth frame. To move from (F_e) to (F_b) and vice versa, a rotation matrix is required. The rotation matrix from (F_e) to (F_b) , also called direction cosine matrix, is a multiplication of the three canonical rotation matrices $Rx(\phi)$, $Ry(\theta)$, and $Rz(\psi)$ with a predefined sequence [16], i.e.,

$$R_i^b = Rx(\phi) * Ry(\theta) * Rz(\psi) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ -C_\phi S_\psi + S_\phi S_\theta C_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & S_\phi C_\theta \\ S_\phi S_\psi + C_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi & C_\phi C_\theta \end{bmatrix} \quad (1)$$

where C and S are the cosine and sine function, respectively. The sequence of rotation in (1) is $z\ y\ x$ (yaw with angle ψ about z axis) $\rightarrow$ (pitch with angle θ about y axis) $\rightarrow$ (roll with angle ϕ about x axis).

In order to go from the body to the earth coordinates, the inverse of R_i^b should be obtained. Since the matrices used to calculate R_i^b are orthogonal, then R_b^i is simply the transpose of R_i^b [17], that is,

$$R_b^i = \left(R_i^b\right)^T \quad (2)$$

The transformation from one frame to the other is necessary since not all the states are measured in on frame. For example, the thrust forces generated by the propellers are measured in the body frame, while the gravitational force is in the earth frame.

To express the relationship between the Euler rates $\dot{\eta} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$ which are measured in the earth frame and the body angular velocities $\omega_b = [p\ q\ r]^T$, the following transformation is used [18,19]:

$$\dot{\eta} = T * \omega_b \quad (3)$$

where

$$T = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \quad -\frac{\pi}{2} < \theta < \frac{\pi}{2}$$

2.2 Simplification Hypotheses

The following are some important assumptions that are used to simplify writing the system mathematical model. Without these hypotheses, the system equations of motion will be complicated.

– The structure supposed to be symmetrical and rigid.

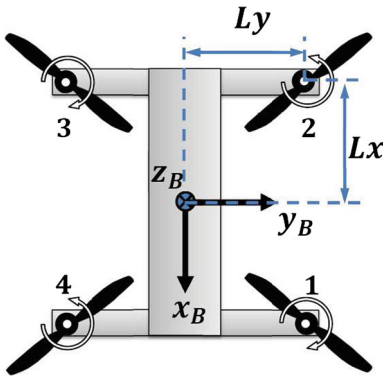

Fig. 2 Racing quadcopter schematic diagram

– The body frame origin attached to the centre of gravity (COG) of the quadrotor's body, with X_b along the forward movement direction as shown in Figs. 1 and 2.
– According to blade element theory (BET), the thrust and drag torque are assumed to be proportional to the square of the propeller revolving speed [20].

2.3 Mathematical Model

This portion presents a detailed derivation of the Newton–Euler 6 DOFs equations of motion for the quadcopter under consideration.

2.3.1 Static Model

The static model for the quadcopter is obtained by expressing the forces and torques acting on its body mathematically.

Forces The forces acting on the quadcopter are divided into two types: the propulsive force and the gravitational force. Each of the four propellers produces a propulsive force when spinning. The overall generated force is the algebraic sum of that produced by the four propellers. The propulsive force for the ith rotor expressed in the body frame is:

$$F_{Ri} = \begin{bmatrix} 0 \\ 0 \\ K_{th} * \Omega_i^2 \end{bmatrix} \quad \text{for} \quad i = 1, 2, 3, 4. \quad (4)$$

where K_{th} is the thrust coefficient and Ω_i is the spinning speed of the ith propeller.

Then the overall force of the four propellers is the vector sum of the individual forces.

$$F_T = \sum_{i=1}^{4} F_{\mathrm{Ri}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K_{\mathrm{th}} & K_{\mathrm{th}} & K_{\mathrm{th}} & K_{\mathrm{th}} \end{bmatrix} * \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{5}$$

Not only the propulsive force affects the quadcopter motion but also the gravitational force tries to pull it down to the ground. The gravitational force in the earth frame is determined by:

$$FG_{\mathrm{e}} = \begin{bmatrix} 0 \\ 0 \\ -M * g \end{bmatrix} \tag{6}$$

where M is the total mass of the quadcopter and g is the gravitational constant.

The total force applied on the quadcopter is a couple of the propulsive and gravitational forces:

$$F_{\mathrm{e}} = FG_{\mathrm{e}} + R_{\mathrm{b}}^{\mathrm{e}} * F_{\mathrm{T}} \tag{7}$$

In the above equation, the rotation matrix is inserted to transfer the propulsive force from the body to the earth coordinate so that, it can be added to the gravitational force.

Torques The torques that affect the rotational motion of the quadcopter are the reactive and the drag torques, where both of them are a result of propeller rotation. The torque due to the ith rotor is:

$$M_{\mathrm{Ri}} = L_{O_{\mathrm{b}} \to Pi} \times F_{\mathrm{Ri}} - M_{\mathrm{Di}} \tag{8}$$

where $L_{O_{\mathrm{b}} \to Pi}$ is a vector from the origin of the body frame to the centre of the ith propeller and is equal to:

$$\begin{aligned} L_{O_{\mathrm{b}} \to P1} &= [\ \ L_x, \ \ L_y, \ 0]^T, \\ L_{O_{\mathrm{b}} \to P2} &= [-L_x, \ \ L_y, \ 0]^T, \\ L_{O_{\mathrm{b}} \to P3} &= [-L_x, -L_y, \ 0]^T, \\ L_{O_{\mathrm{b}} \to P4} &= [\ \ L_x, -L_y, \ 0]^T \end{aligned}$$

and L_x, L_y are shown in Fig. 2.

The drag torque for the ith propeller is:

$$M_{\mathrm{Di}} = \begin{bmatrix} 0 \\ 0 \\ (-1)^i * K_{\mathrm{D}} * \Omega_i^2 \end{bmatrix} \quad \text{for} \quad i = 1, 2, 3, 4. \tag{9}$$

where K_{D} is the drag coefficient and $(-1)^i$ is negative for CW rotated propellers (1 and 3) and positive for CCW propellers (2 and 4).

The total torque due to the four rotating propellers is:

$$M_{\mathrm{T}} = \sum_{i=1}^{4} M_{\mathrm{Ri}} \tag{10}$$

substituting (9) in (8) and the resulting equation into (10) and let $L_x = L_y = L$, yield:

$$M_T = \begin{bmatrix} L * K_{\mathrm{th}} & L * K_{\mathrm{th}} & -L * K_{\mathrm{th}} & -L * K_{\mathrm{th}} \\ -L * K_{\mathrm{th}} & L * K_{\mathrm{th}} & L * K_{\mathrm{th}} & -L * K_{\mathrm{th}} \\ K_{\mathrm{D}} & -K_{\mathrm{D}} & K_{\mathrm{D}} & -K_{\mathrm{D}} \end{bmatrix} * \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{11}$$

2.3.2 Control Vector U

At this stage and for the purpose of better understanding the mathematical model of the quadcopter, it is important to define a control vector U that is consist of four inputs $U1, U2, U3$, and $U4$.

The first control input ($U1$) is the total forces generated by the rotors in the z axis direction. It is responsible of altitude control.

$$U1 = F_{\mathrm{Tz}} = K_{\mathrm{th}} * \left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \tag{12}$$

The second control input ($U2$) is the resulting moment around x axis and is responsible of roll angle control.

$$U2 = M_{\mathrm{Tx}} = L * K_{\mathrm{th}} * \left(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2\right) \tag{13}$$

The third control input ($U3$) represents the moments around y axis so it affects the pitch angle of the quadcopter.

$$U3 = M_{\mathrm{Ty}} = L * K_{\mathrm{th}} * \left(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right) \tag{14}$$

The fourth control input ($U4$) is in charge of controlling the yaw angle since it is simply the resultant moment around z axis.

$$U4 = M_{\mathrm{Tz}} = K_{\mathrm{D}} * \left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right) \tag{15}$$

Equations (12) through (15) can be combined into one matrix equation as follow:

$$\begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix} = K * \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{16}$$

where $K \in R^{4\times4}$ is the coefficient matrix and equal to:

$$K = \begin{bmatrix} K_{\text{th}} & K_{\text{th}} & K_{\text{th}} & K_{\text{th}} \\ L*K_{\text{th}} & L*K_{\text{th}} & -L*K_{\text{th}} & -L*K_{\text{th}} \\ -L*K_{\text{th}} & L*K_{\text{th}} & L*K_{\text{th}} & -L*K_{\text{th}} \\ K_{\text{D}} & -K_{\text{D}} & K_{\text{D}} & -K_{\text{D}} \end{bmatrix}$$

The rotation speed of the rotors for a specific control input vector can be calculated by:

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = \sqrt{K^{-1} * \begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix}} \tag{17}$$

2.3.3 Dynamic Model

The dynamic model of the quadcopter describes its behaviour over the time. Unlike terrestrial mobile robots in which it is possible to describe their models kinematically, the quadcopter modelling requires their dynamics to be considered which involves the gravitational effects [21]. There are two methods used to represent the dynamic model, the Lagrangian and the Newton–Euler method, and both are consistent in describing the quadcopter dynamic model. Nevertheless, Newton–Euler method is more continent in describing the physics of the system and so it is easy to be understood [22]. The dynamics of a standard quadcopter falls into two categories: a translational motion (x, y, and z) which is underactuated and a rotational motion (roll ϕ, pitch θ, and yaw ψ) which is fully actuated.

Translational motion equations Newton's second law of motion can be applied to describe the translational motion of the quadcopter in the earth frame.

$$F_e = M * \dot{v_{\text{b}}} \tag{18}$$

where $\dot{v_{\text{b}}} = [\ddot{x},\ \ddot{y},\ \ddot{z}]^T$ is the quadcopter linear acceleration.

By substituting (7), (12) into the above equation, we can get:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R_{\text{b}}^{i} * \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix} \tag{19}$$

and after some arrangement we can get:

$$R_i^b * \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z}+g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix} \tag{20}$$

then the following set of equations, which describes the translational motion of our model, can be obtained.

$$\ddot{x}C_\theta C\psi + \ddot{y}C_\theta S_\psi - (\ddot{z}+g)S_\theta = 0 \tag{21}$$

$$\ddot{x}(S_\phi S_\theta C_\psi - C_\phi S_\psi) + \ddot{y}(S_\phi S_\theta S_\psi + C_\phi C_\psi) + (\ddot{z}+g)S_\phi C_\theta = 0 \tag{22}$$

$$\ddot{x}(C_\phi S_\theta C_\psi + S_\phi S_\psi) + \ddot{y}(C_\phi S_\theta S_\psi - S_\phi C_\psi) + (\ddot{z}+g)C_\phi C_\theta = \frac{U1}{M} \tag{23}$$

Rotational motion equations According to Newton–Euler formalism, the rotational motion equations are derived and expressed in the body coordinates.

$$M_{\text{T}} = I * \dot{\omega_{\text{b}}} + (\omega_{\text{b}} \times I * \omega_{\text{b}}) \tag{24}$$

where $I \in R^{3\times3}$ is a moment of inertia matrix of the quadcopter. The quadcopter assumed to be symmetrical in Sect. 2.2, so the moment of inertia matrix is expected to be diagonal.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{25}$$

The set of rotational equation of motion can be obtained by means of (13–15) and (24) as follows:

$$\dot{\omega_{\text{b}}} = I^{-1} * \left(\begin{bmatrix} U1 \\ U2 \\ U3 \end{bmatrix} - (\omega_{\text{b}} \times I * \omega_{\text{b}}) \right) \tag{26}$$

To this end, we have obtained the complete dynamics for the H-shaped quadcopter and the next step is to propose and design a suitable trajectory tracking controller.

3 System Controller Design

The MATLAB/Simulink environment on a personal computer with 2.5 GHz processing speed and 6 GB RAM is used to verify the derived dynamical model and to carry out all the subsequent tests in the following sections. The quadcopter model parameters that are used during the simulation are shown in Table 1. The complete block diagram for the model with the controllers is shown in Fig. 3. The following is the detailed description for each individual controller.

3.1 Altitude Control

The altitude (z position) of the quadcopter is subtracted from the desired altitude Z_{d} to form the error signal as shown in Fig. 4. The resulting error signal is fed to a PID controller

Table 1 Quadcopter model parameters

Parameter	Value
M	1.2 kg
L	0.3 m
K_{th}	$7.5e$-05 N s^2/rad^2
K_{D}	$1.7e$-06 Nm s^2/rad^2
I_{xx}	0.01987 kg m^2
I_{yy}	0.01987 kg m^2
I_{zz}	0.03848 kg m^2
Ω_{min}	100 rad/s
Ω_{max}	750 rad/s

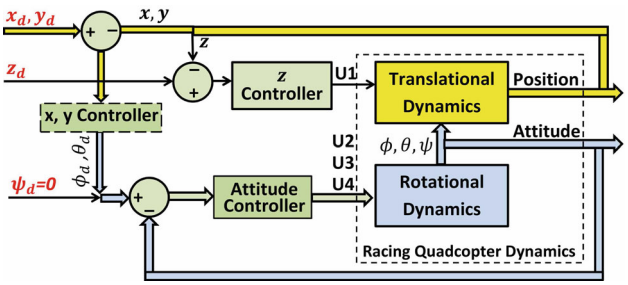

Fig. 3 Racing quadcopter overall controller block diagram

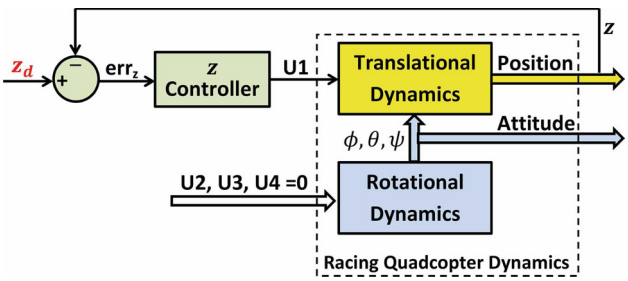

Fig. 4 Altitude feedback control block diagram

that will adjust the value of the control input U1 to achieve the desired altitude.

3.2 Attitude Control

The attitude controller is a combination of 3 PID controllers. These controllers are responsible of producing a control signal U2, U3, and U4 to orient the quadcopter to the required roll ϕ, pitch θ, and yaw ψ angle, respectively. Attitude control block diagram is shown in Fig. 5.

3.3 *X–Y* Position Control

In contrast to the altitude and attitude control, the x and y position cannot be controlled directly. The desired x and y position can be obtained by tilting the quadcopter body with a specific roll and pitch angles. The required roll and pitch

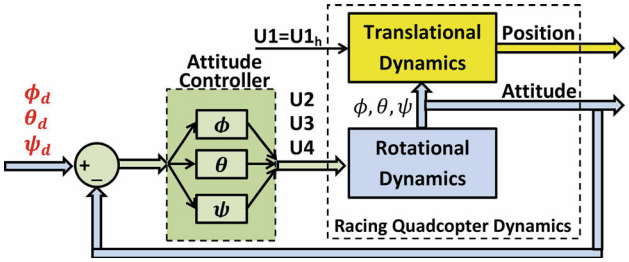

Fig. 5 Attitude feedback control block diagram

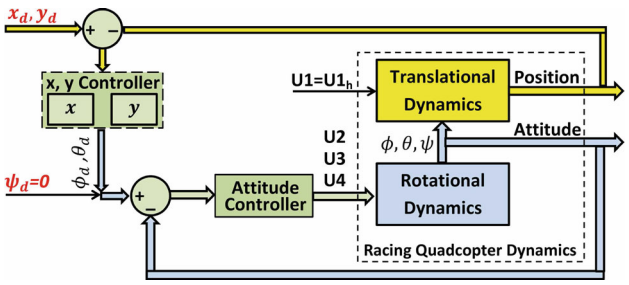

Fig. 6 $x - y$ position feedback control block diagram

angles should be calculated to achieve the desired $x - y$ position. Two PID controllers are used to produce the desired roll and pitch angles for a specific $x - y$ position as shown in Fig. 6.

Roll (ϕ) angle calculation The required roll angle ϕ can be obtained by subtracting (22) * cos(ϕ) from (23) * sin(ϕ) that results in:

$$\frac{1}{M} U1 \sin(\phi) = \ddot{x} \sin(\psi) - \ddot{y} \cos(\psi) \tag{27}$$

then the desired roll angle ϕ can be extracted by rearranging the above equation as:

$$\phi_{\text{d}} = \sin^{-1}\left(M * \frac{(\ddot{x} \sin(\psi) - \ddot{y} \cos(\psi))}{U1}\right) \tag{28}$$

Pitch (θ) angle calculation In our design, it is assumed that ϕ and θ angles are in the $\pm 90°$ range. With this in mind, (21) can be divided by $\cos(\theta) \neq 0$ then:

$$\ddot{x} \cos(\psi) + \ddot{y} \sin(\psi) - (\ddot{z} + g) \tan(\theta) = 0 \tag{29}$$

and the desired pitch angle (θ) can be obtained as:

$$\theta_d = \tan^{-1}\left(\frac{\ddot{x} \cos(\psi) + \ddot{y} \sin(\psi)}{(\ddot{z} + g)}\right) \tag{30}$$

In view of (20), we can get:

$$\left(R_i^b \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z}+g \end{bmatrix}\right)^T * \left(R_i^b \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z}+g \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix}^T * \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z}+g \end{bmatrix}^T {R_i^b}^T * R_i^b \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z}+g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix}^T * \begin{bmatrix} 0 \\ 0 \\ \frac{U1}{M} \end{bmatrix} \quad (32)$$

then

$$(\ddot{z}+g) = \sqrt{\left(\frac{U1}{M}\right)^2 - \ddot{x}^2 - \ddot{y}^2} \quad (33)$$

substitute (33) into (30) yield:

$$\theta_d = \tan^{-1}\left(\frac{\ddot{x}\cos(\psi) + \ddot{y}\sin(\psi)}{\sqrt{\left(\frac{U1}{M}\right)^2 - \ddot{x}^2 - \ddot{y}^2}}\right) \quad (34)$$

3.4 PID Parameters Tuning Using GA

GA is a search heuristic method that imitates the natural selection process [23]. The outline for the GA optimization algorithm is:

- Generate a random population. The population consists of individuals which are in our work the PID parameters (proportional gain K_p, integral gain K_i, and the derivative gain K_d). The PID controller configuration, that is used in this work, is shown in Fig. 7.
- The algorithm is then evaluate the individuals' fitnesses according to an objective function and it is in our case is set to be:

  $$\text{Obj} = A * \text{err} + B * t_s + C * M_P \quad (35)$$

  where err is the difference between the desired and the actual output response, t_s is the settling time, M_P is the peak overshoot of the response, A, B, C are positive constants subject to $A + B + C = 1$.
- A new population is then reproduced with their individuals using the following GA functions
  - Evolution
  - Crossover
  - Mutation
- An evaluation process is then repeated for the new individuals from the new population using (35) to test their merit.
- The fittest individuals from the first and second population are then chosen to generate a third population.
- The above steps are repeated as iterations until the termination criterion is met, and it is in our case set to be the maximum number of generations.

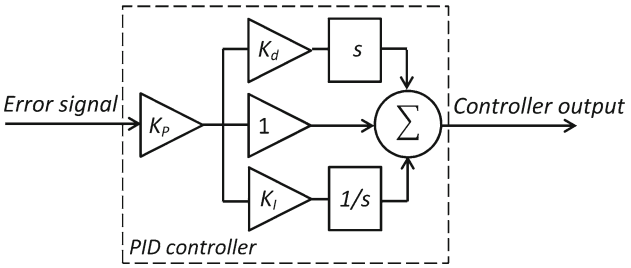

Fig. 7 PID controller configuration

Table 2 PID controller parameters

Control	K_p	K_i	K_d
z position ($z_d = 1$ m)	14.88	0.7994	0.4734
Yaw ($\psi_d = 20°$)	0.4768	0.00864	0.512
Roll ($\phi_d = 10°$)	2.151	0.0019	0.1723
Pitch ($\theta_d = 10°$)	1.454	0.00239	0.2134
x position ($x_d = 1$ m)	1.0083	0.0096	0.4965
y position ($y_d = 1$ m)	9.7023	0.0099	0.5420

In our work, the following values are used to initialize the GA process: $A = 0.4$, $B = 0.4$, $C = 0.2$, population size = 100, and the number of generation for termination = 100.

Another code is added to the above algorithm to ensure that the GA tuning process converges to a controller parameters that produce actuators signals within the physical limits of the rotors:

$$(\Omega_{\min} < \Omega_i < \Omega_{\max} \text{ for } \quad i = 1, 2, 3, 4).$$

It means that if the selected PID parameters cause the controller output to exceed the actuators' limits, then its objective function value is assigned to an extremely high weight so that it will not be reselected again. An algorithm is written in MATLAB to calculate the objective function for every set of PID parameters. The code takes the response data from the quadcopter Simulink model to calculate the individuals' fitnesses. The optimization process is repeated for the 6 PID controllers, and the resulting tuned parameters and step response are shown in Table 2. The response to a desired step values in Table 2 is shown in Fig. 8 with their associated settling times and percentage peak overshoots.

4 Trajectory Tracking Test

As a trajectory tracking test, we assumed that the quadcopter initially at hover ($z = 1$ m) and it should follow a horizontal infinity like shape (∞) trajectory. The desired position profile is shown in Fig. 9a and can be expressed as:

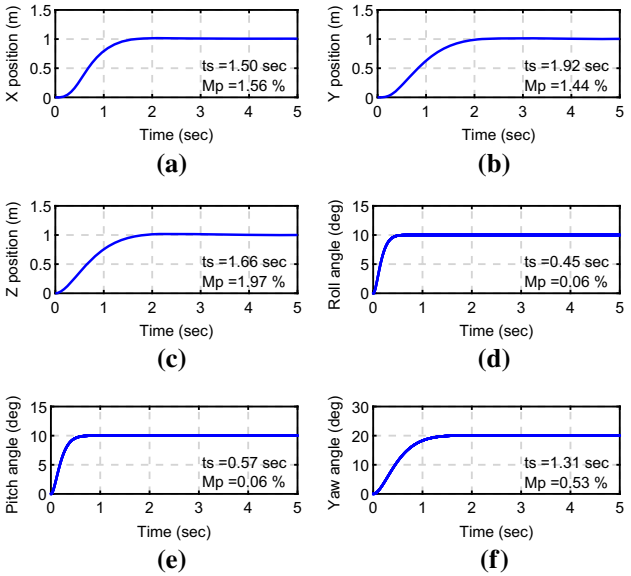

Fig. 8 Simulation results for step inputs **a** x position response, **b** y position response, **c** z position response, **d** roll ϕ response, **e** pitch θ response, **f** yaw ψ response

$$x_{\mathrm{d}} = 1 - \rho * \cos(0.2 * (t))$$
$$y_{\mathrm{d}} = \rho * \sin(0.4 * (t))$$
$$z_{\mathrm{d}} = 1$$

where $\rho(t) = 2/(3 - \cos(0.4 * (t)))$.

The simulation result shows that the maximum tracking error percentage in $x - y$ position is approximately 1.5%.

The other important quantities are:

- The peak linear velocity of the quadcopter body is $\dot{x} = 0.18$ and $\dot{y} = 0.43\,\mathrm{m/s}$;
- The maximum linear acceleration during the path following is $\ddot{x} = 0.57$ and $\ddot{y} = 0.21\,\mathrm{m/s^2}$.

5 Robustness Test

Simulation environment is used to implement three tests to verify the robustness of the designed controller.

5.1 Random Noise

A white Gaussian random noise (shown in Fig. 10a) is generated with a maximum absolute amplitude of 0.01. It is then injected in the feedback loops to simulate the sensor noise. In order to simulate the rotors' actuator, a saturation block is added to limit the calculated rotor' upper and lower speed to its physical limits stated in Table 1. Simulation results in Fig. 10b, c show that for the infinity shape trajectory tracking in the previous section, the maximum error in x and y position is 43 mm which implied the controller capability of suppressing noise.

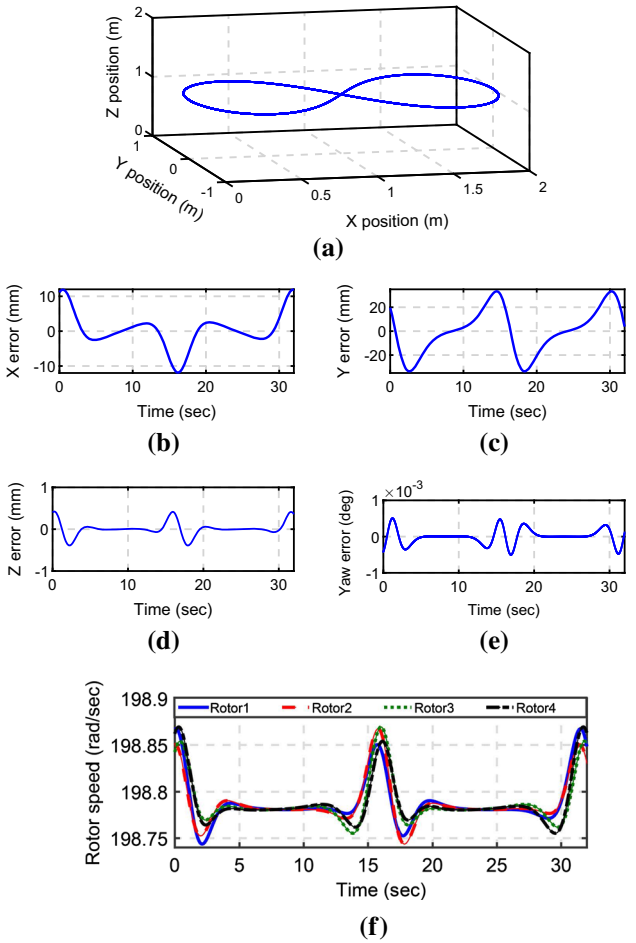

Fig. 9 Trajectory tracking results **a** 3D desired position trajectory, **b** tracking error in x position, **c** tracking error in y position, **d** z position drift, **e** yaw ψ tracking error, **f** rotors behaviour

5.2 Model Uncertainties

For the model uncertainty test, we assume that the quadcopter weight is increased by 35% due to an added payload. A simulation is implemented with this uncertainty, and the quadcopter is ordered to follow the infinity shape trajectory in Fig. 9a. The position tracking error is shown in Fig. 11a, b. It can be seen from the rotors behaviour in Fig. 11c that the actuators did not exceed their upped and lower limits.

5.3 External Disturbances

In this test, we assume that the racing quadcopter received a sudden disturbance force while it is at hover ($z = 1\,\mathrm{m}$). The disturbance force of 1N is applied in the upward direction

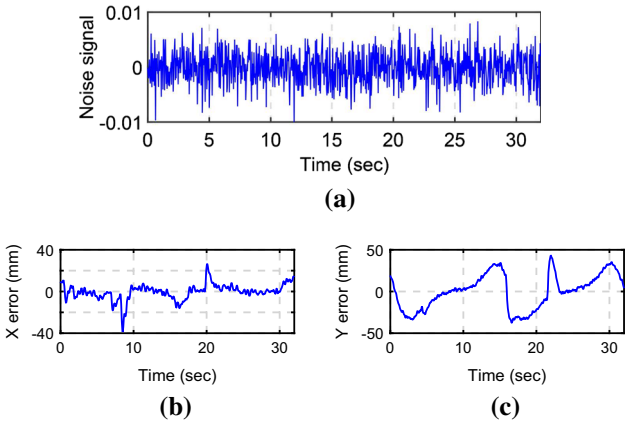

Fig. 10 Random noise test results **a** noise signal, **b** x position error, **c** y position error

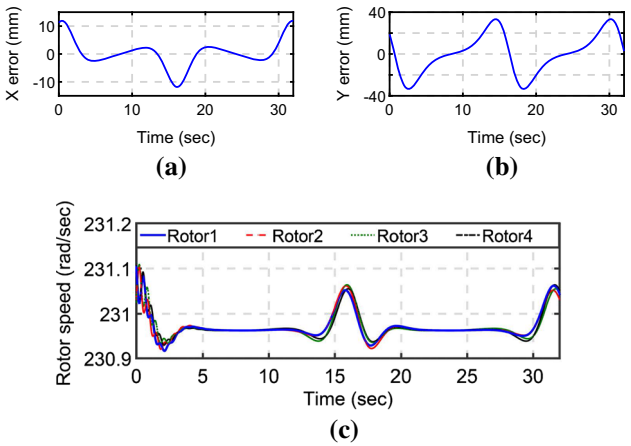

Fig. 11 Model uncertainty test results **a** x position error, **b** y position error, **c** rotors behaviour

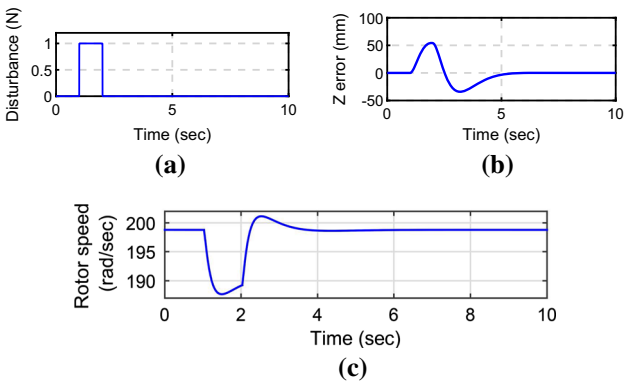

Fig. 12 Disturbance test results **a** disturbance input, **b** z position error, **c** rotors behaviour

directly under the COG point and it is continued for 1 s. The controller shows the capability to reject the disturbance and maintain the stability and the position of the system. The simulation results are shown in Fig 12.

6 Conclusion

Racing quadcopters are of great importance for their several serious application related to human life rescue. In contrast with this quadcopter type importance, there is no attention for its model derivation and control. In this paper, a complete mathematical model for H-Shaped racing quadcopter has been obtained. The dynamics of the model have been described in a comprehensive and systematic way. A PID control scheme has been proposed to control the six DOFs of the quadcopter. Besides, the PID coefficients have been tuned using GA with a novel objective function that compromises between three important step response characteristics. The GA set the PID coefficients to achieve the best performance that minimize the tracking error, the settling time, and the peak overshoot while maintaining the actuators in the normal operation limits. The derived model along with the designed controller has been implemented in the Simulink environment. The system has been commanded to follow a desired infinity-shaped trajectory, and the simulation results have shown a perfect trajectory tracking performance with a very low tracking error. In the final analysis, the controller has been tested for its robustness in terms of sensor noise suppression, disturbance rejection, and the sensitivity to model parameters uncertainties. The robustness test results revealed the designed controller effectiveness and performance.

References

1. Erginer, B.; Altug, E.: Modeling and PD control of a quadrotor VTOL vehicle. In: IEEE Intelligent Vehicles Symposium, pp. 894–899. IEEE (2007)
2. Hamel, T.; Mahony, R.; Lozano, R.; Ostrowski, J.: Dynamic modelling and configuration stabilization for an X4-flyer. IFAC Proc. Vol. **35**(1), 217–222 (2002)
3. Mokhtari, A.; Benallegue, A.: Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA'04), vol. 3, pp. 2359–2366 (2004)
4. Zemalache, K.M.; Beji, L.; Marref, H.: Control of an underactuated system: application a four rotors rotorcraft. In: IEEE International Conference on Robotics and Biomimetics-ROBIO, pp. 404–409 (2005)
5. Tayebi, A.; McGilvray, S.: Attitude stabilization of a VTOL quadrotor aircraft. IEEE Trans. Control Syst. Technol. **14**(3), 562–571 (2006)
6. Bouabdallah, S.; Murrieri, P.; Siegwart, R.: Design and control of an indoor micro quadrotor. In: IEEE International Conference on Robotics and Automation (ICRA '04), vol. 5, pp. 4393–4398 (2004)
7. Bouabdallah, S.; Noth A.; Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. International Conference on Intelligent Robots and Systems (IROS 2004), vol. 3, pp. 2451–2456 (2004)
8. Bresciani, T.: Modelling, Identification and Control of a Quadrotor Helicopter. In: Department of Automatic Control. Master Thesis Lund: Lund University (2008)

9. Bayrakceken, M.; Kemal, M.; Kursat Y.; Aydemir A.; Abdurrahman K.: HIL simulation setup for attitude control of a quadrotor. In: IEEE International Conference on Mechatronics (ICM), pp. 354–357. IEEE, Havard (2011)
10. Çetinsoy, E.: Design and flight tests of a holonomic quadrotor UAV with sub-rotor control surfaces. In: IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1197–1202. IEEE (2013)
11. Benallegue, A.; Moktari, A.; Fridman, L.: Feedback linearization and high order sliding mode observer for a quadrotor UAV. In: Proceedings of the 2006 International Workshop on Variable Structure Systems (2006)
12. Lara, D.; Romero, G.; Sanchez A.; Lozano, R.: parametric robust stability analysis for attitude control of a four-rotor mini-rotorcraft. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 4351–4356. San Diego (2006)
13. Li, J.; Li, Y.: Dynamic analysis and PID control for a quadrotor. In: IEEE International Conference on Mechatronics and Automation, pp. 573–578. Beijing (2011)
14. Gupte, S.; Mohandas, P.I.T.; Conrad, J.M.: A survey of quadrotor unmanned aerial vehicles. In: Proceedings of IEEE Southeastcon, pp. 1–6 (2012)
15. Stingu, E.; Lewis, F.: Design and implementation of a structured flight controller for a 6DoF quadrotor using quaternions. In: IEEE 17th Mediterranean Conference on in Control and Automation MED'09, pp. 1233–1238 (2009)
16. Gruber, D.: The mathematics of the 3D rotation matrix. In: The Xtreme Game Developers Conference (2000)
17. Padfield, G.D.: Helicopter Flight Dynamics. Wiley, New York (2008)
18. Cai, G.; Chen, B.M.; Lee, T.H.: Unmanned Rotorcraft Systems. Springer, Berlin (2011)
19. Leishman, R.C.; Macdonald, J.C.; Beard, R.W.; McLain, T.W.: Quadrotors and accelerometers: state estimation with an improved dynamic model. IEEE Control Syst. **34**(1), 28–41 (2014)
20. Mahony, R.; Kumar, V.; Corke, P.: Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. IEEE Robot. Autom. Mag. **19**(3), 20–32 (2012)
21. Salih, A.L.; Moghavvemi, M.; Mohamed, H.A.; Gaeid, K.S.: Flight PID controller design for a UAV quadrotor. Sci. Res. Essays **5**(23), 3660–3667 (2010)
22. Zhang, X.; Li, X.; Wang, K.; Lu, Y.: A survey of modelling and identification of quadrotor robot. Abstr. Appl. Anal. **2014**, 1–16 (2014)
23. Mitchell, M.: An Introduction to Genetic Algorithms. MIT press, Cambridge (1996)