Decentralized PID Control by Using GA Optimization Applied to a Quadrotor

Submitted: 26th February 2018; accepted: 14th May 2018

Seif-El-Islam Hasseni: *Latifa Abdou*

DOI: 10.14313/JAMRIS_2-2018/9

Abstract:
Quadrotors represent an effective class of aerial robots because of their abilities to work in small areas. We suggested in this research paper to develop an algorithm to control a quadrotor, which is a nonlinear MIMO system and strongly coupled, by a linear control technique (PID), while the parameters are tuned by the Genetic Algorithm (GA). The suggested technique allows a decentralized control by decoupling the linked interactions to effect angles on both altitude and translation position. Moreover, the using a meta-heuristic technique enables a certain ability of the system controllers design without being limited by working on just the small angles and stabilizing just the full actuated subsystem. The simulations were implemented in MATLAB/Simulink tool to evaluate the control technique in terms of dynamic performance and stability. Although the controllers design (PID) is simple, it shows the effect of the proposed technique in terms of tracking errors and stability, even with large angles, subsequently, high velocity response and high dynamic performances with practically acceptable rotors speed.

Keywords: *quadrotor, non-linear systems, decentralized control, PID, optimization, genetic algorithm.*

Abbreviations:

BMW:	Best-Mates-Worst technique
DE:	Differential Evolution
ISE:	Integral Squared Error
GA:	Genetic Algorithm
MIMO:	Multi-Input Multi-Output system
LQR:	Linear Quadratic Regulator
OS4:	**O**mni-directional **S**tationary **F**lying **OU**tstretched **R**obot
PID:	Proportional Integral Derivative Controller
PSO:	Particle Swarm Optimization
RD:	Rotor Dynamic
UAV:	Unmanned Aerial Vehicle

Nomenclature:

E:	Earth frame
B:	Body frame
R_B^E:	Transformation coordinates matrix from body frame (B) to earth frame (E)
x:	Translation coordinate in x axis
y:	Translation coordinate in y axis
z:	Altitude coordinate
φ, θ, ψ:	Roll, Pitch and Yaw Euler-angles, respectively.
V:	Vector of linear velocity
Ω:	Vector of angular velocity
q_i:	Coordinate of degrees of freedom = $[x, y, z, \varphi, \theta, \psi]$
L:	Lagrangian term
E_k:	Kinetic Energy
E_{kr}:	Kinetic-Rotation Energy
E_{kt}:	Kinetic-Translation Energy
E_p:	Potential Energy
Γ:	Vector of non-conserved forces and torques
F_x, F_y, F_z:	Force on x, y and z axis, respectively.
τ:	Vector of torques
τ_f:	Vector of forces torque
τ_g:	Vector of gyroscopic torque
$\tau_\varphi, \tau_\theta, \tau_\psi$:	Torque on x, y and z axis, respectively.
f_i:	Force of ith rotor
T_i:	Thrust force of ith rotor (countre-torque)
J:	Inertia matrix
I_x, I_y, I_z:	Inertia on x, y and z axis, respectively.
ω_i:	Angular speed of ith rotor
ω_{di}:	Desired angular speed of ith rotor
Ω_r:	$\omega_1 - \omega_2 + \omega_3 - \omega_4$
J_r:	Rotor inertia
b:	Thrust factor
d:	Drag factor
l:	Arm length
m:	Quadrotor mass
g:	Gravity constant
U_i:	The ith control input
U_{di}:	The desired ith control input
u_x, u_y:	Virtual inputs of x and y subsystems
k:	Adaptive decoupler factor
k_p, k_i, k_d:	Proportional, Integral and Derived gains of PID, respectively.

1. Introduction

The quadrotor is an Unmanned Aerial Vehicle (UAV) that is considered as an aircraft without a pilot, and which can be driven by a human at a ground control station or can fly autonomously by known flight trajectory. Quadrotors were used at the beginning of their appearances in the military domain for monitoring or reconnaissance missions. Civil applications made their appearances later, as it is the case of the monitoring of the road traffic. The quadrotor has some advantages compared with other UAVs, as vertical take-off landing, stationary and low-speed flight, which make it highly maneuverable, with precise navigation in difficult or dangerous areas. For these reasons, the researchers were carried about this aerial robot to develop the control algorithms with successful comportments. However, the quadrotor is a characteristically nonlinear, underactuated mechanical system; with six degrees of freedom (three for rotation and three for translation) and just four inputs, its dynamic is complex because of its strong coupling between the translational and rotational subsystems, which don't allow releasing its controllers with simple ways.

This nonlinear system motivates to design a complex nonlinear control algorithm. A large number of control methods are used to solve the attitude stabilization and trajectory tracking problems. In [1] a sliding mode control had been developed for the full system, a robust terminal sliding mode is used for the full actuated subsystem (rotational) and the simple sliding mode for the under-actuated subsystem (translational). Both integral backstepping and sliding mode have been used in thesis [2], [3] and both of them conclude that the combination between backstepping and integral action is the best for quadrotor stability against disturbances and model uncertainties. A significant number of researches were based on combination between two techniques to design the full controller; in [4], a combination of sliding mode and integral backstepping is used on both of rotational and translational subsystems. In [5] an adaptive sliding mode control is developed against the actuators failures. In [6] a sliding mode control was applied with switching gains adaptively by fuzzy logic system based on information from the sliding surfaces, but just for stabilizing the attitude subsystem. An online optimization was associated to improve the backstepping designed control [7]. The predictive control is one of the effective techniques of the nonlinear systems, it is developed in [8], [9], where the authors have used the piecewise affine systems to derive the predictive model. The optimal control is also a very effective way of controller's synthesize of unmanned aerial vehicles [10], [11], in [12] a linearized model of the quadrotor rotational subsystem was derived for applied the LQR control. The most classical linear controller, which is the PID have been used in the unmanned aerial vehicles [13] because of its simplicity. In case of quadrotor, it is used in [12], [14], [15], [16], the linearized model was inquired, subsequently it is used for just the stabilization of the attitude subsystem, so unfortunately only for the small range of angles and low dynamics. Some works are not based on classical sensors; GPS, accelerometer and gyroscope, but, they applied vision-based control [17], [18] where the only sensor is the camera and which use the image processing techniques to measure the velocity and make more precise movement.

In this paper the classical controller PID is proposed to control the full system. In contrast to other works in literatures, our dynamic system isn't limited to small range of angles but can exceed to nonlinear range of angles for getting a high dynamic. In addition we are motivated to use one of effective meta-heuristic optimization approach; Genetic Algorithms [19], which are applied in a large way of off-line controllers' design, like control structure design in fuzzy logic controller or PID parameters tuning. In this case, the synthesized structure is kept as classical one, but the parameters are optimized by GA.

The contribution of this paper can be presented as follows;

a) A decentralized control is applied; we designed each controller separately from others.
b) To realize that, we need to decouple the interactions between the translation and rotation coordinates.
c) For the decoupling between the altitude and the attitude, we added an adaptive online factor in the altitude input. Thereafter we designed the four controllers separately, the parameters tuning is used by Genetic Algorithm, no need to linearize the model and limit the system to work on small angles, subsequently limit on low dynamic performances. The optimization process is done with both small and big angles.
d) After tuning the four controllers of the actuated subsystem, the inner loop was locked with a view to decouple the influence of the rotation on the translation position and design the controllers of underactuated subsystem, which is presented by the x and y position, their controllers were also tuned by the Genetic Algorithm. This virtual control laws are used to compute the desired roll and pitch angles and used them on the full actuated subsystem.

The paper's structure is as follows: in the next section, the mathematical model is obtained by using Euler-Lagrange approach, including the aerodynamic effects and the rotors dynamics. In section 3, we present the proposed controller structure and the strategy of decoupling the interactions, to permit the application of decentralized control. In section 4, we introduce the use of GA as a solution to optimize some parameters in order to improve the quality of our controller. In section 5, we present and discuss the obtained results by simulating 3D trajectory tracking and the rotors speeds states, and lastly our conclusion in section 6.

2. System Modeling

Before the control development, the mathematical model for this mobile robot is needed. In this work, we propose to use Euler-Lagrange technique because of its advantages, related to the fact that it is based on potential and kinetic energies, and also dynamic equations in symbolic closed form. This technique is the best for study of dynamic properties and analysis of control schemes [20], where the robot is supposed as a rigid closed system:

In the quadrotor, we can find two frames: the inertial frame (E: earth) and the mobile frame, who is fixed in the quadrotor (B: body).

The distance between the earth frame and the body frame describes the absolute position of the mass center of the quadrotor $r = [x\, y\, z]^T$. The rotation R from the body frame to the inertial frame describes the orientation of the quadrotor. The orientation of the quadrotor, is described using roll, pitch and yaw

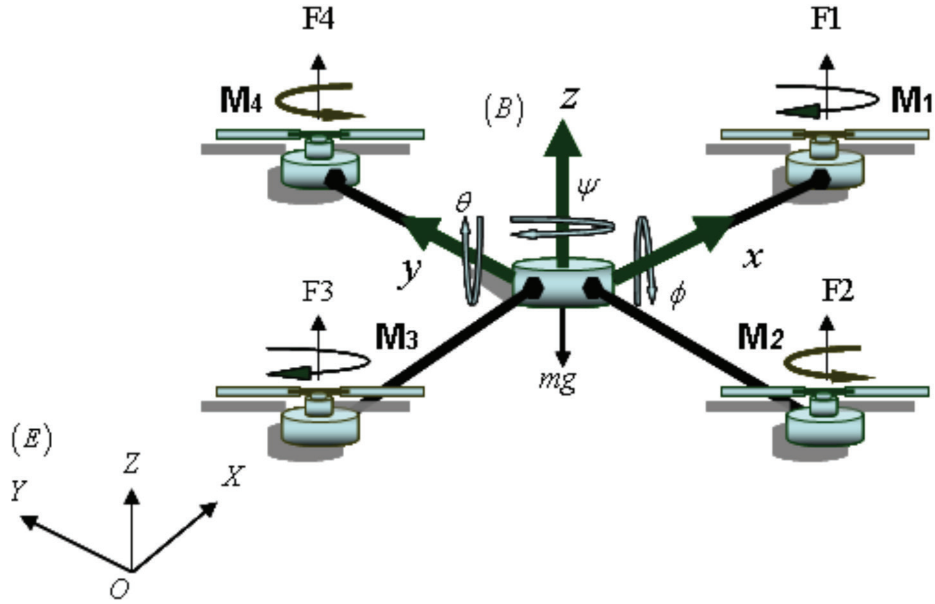

Fig. 1. Reference frames of Quadrotor

angles (φ, θ and ψ) which represent the orientation about the x, y and z axis respectively. The matrix of rotation is defined by [21]:

$$\boldsymbol{R}_B^E = R(z,\psi)R(y,\theta)R(x,\varphi) \tag{1}$$

Where:

$$\boldsymbol{R}(x,\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{pmatrix} \tag{2}$$

$$\boldsymbol{R}(y,\theta) = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \tag{3}$$

$$\boldsymbol{R}(z,\psi) = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Then:

$$\boldsymbol{R}_B^E = \begin{pmatrix} c\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ c\theta s\psi & cos\varphi cos\psi + s\varphi s\theta s\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{pmatrix} \tag{5}$$

Where s and c denote *sin* and *cos* respectively.

The vector of angular velocities of quadrotor is defined in body frame $\boldsymbol{\Omega} = [\omega_x \omega_y \omega_z]^T$. It is given depending on the derived Euler angles $[\dot{\varphi}\,\dot{\theta}\,\dot{\psi}]^T$ that are measured in the inertial frame. This transformation is derived as follows [s]:

$\boldsymbol{\Omega}$ – angular velocities

$$\boldsymbol{\Omega} = \begin{pmatrix} \dot{\varphi} \\ 0 \\ 0 \end{pmatrix} + \left(R(x,\varphi)\right)^{-1} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \left(R(x,\varphi)\right)^{-1} \left(R(y,\theta)\right)^{-1} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \tag{6}$$

$$\boldsymbol{\Omega} = \begin{pmatrix} \dot{\varphi} - \dot{\psi}\, sin\theta \\ \dot{\theta} cos\varphi + \dot{\psi}\, sin\varphi cos\theta \\ \dot{\psi}\, cos\varphi cos\theta - \dot{\theta} sin\varphi \end{pmatrix} \tag{7}$$

Lagrange law is given as [20]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \Gamma_i \tag{8}$$

Where:

q_i – the measured variable, which represents each one of the output variables: x, y, z, φ, θ and ψ.

Γ_i – the vector of non-conserved forces or torques (F_x, F_y, F_z, τ_φ, τ_θ and τ_ψ).

2.1. The Lagrangian

Defined with kinetic energy and potential energy as follow:

$$L = E_k - E_p = E_{kt} + E_{kr} - E_p \tag{9}$$

E_k – the kinetics energy, E_{kt} – for translation kinetic energy and E_{kr} – for rotation kinetic energy, E_p – the potential energy, then we have:

$$L = \frac{m}{2}\boldsymbol{V}^2 + \frac{1}{2}\boldsymbol{J\Omega}^2 - mgz \tag{10}$$

J – the matrix symmetric of inertia, I_x, I_y, and I_z are inertias on x, y and z axis, respectively.

V – the vector of translation velocities.

m – the mass of quadrotor.

g – gravitional constant.

$$\boldsymbol{J} = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \tag{11}$$

By applying (7) and (11) in (10), we find:

$$L = \frac{m}{2}\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) + \frac{1}{2} I_x \left(\dot{\varphi} - \dot{\psi} s\theta\right)^2 + \\ + \frac{1}{2} I_y \left(\dot{\theta} c\varphi + \dot{\psi} s\varphi c\theta\right)^2 + \frac{1}{2} I_z \left(\dot{\psi} c\varphi c\theta - \dot{\theta} s\varphi\right)^2 - mgz \tag{12}$$

2.2. Non-Conserved Forces and Torques

Returning to Fig. 1, the motors (M_1, M_3) turning the opposite direction of the motors (M_2, M_4) and each motor generates a thrust force $f_i = b.\omega_i$ and a drag torque $T_i = d.\omega_i$. Where b and d are thrust and drag coefficients respectively. ω_i: Is the rotation speed of motor M_i.

2.2.1. Non-conserved forces:

The thrust of quadrotor is the sum of forces f_i.

$$F = \sum_{i=1}^{4} f_i = b.\sum_{i=1}^{4}\omega_i^2$$

$$\Rightarrow F = b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right) \tag{13}$$

$$\boldsymbol{F_B} = \begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} \tag{14}$$

$\boldsymbol{F_B}$ – the vector of forces in the body frame.

Because, we need the forces in earth frame, we get it by transformation matrix: $F_E = R.F_B$, then:

$$\boldsymbol{F_E} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \begin{pmatrix} (c\varphi s\theta c\psi + s\varphi s\psi)\, b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right) \\ (c\varphi s\theta s\psi - s\varphi c\psi)\, b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right) \\ (c\varphi c\theta)\, b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right) \end{pmatrix} \tag{15}$$

2.2.2. Non-conserved torques:

Represented by:

$$\boldsymbol{\tau} = \boldsymbol{\tau_f} - \boldsymbol{\tau_g} \tag{16}$$

Where:

$\boldsymbol{\tau_f}$: Is the vector of forces torques, which contains:

- Torque according to x (roll), that depends on the difference between f_4 and f_2.
- Torque according to y (pitch), that depends on the difference between f_3 and f_1.
- Torque according to z (yaw), which is obtained by the difference between the sums of torques T_1 and T_3, and that of T_2 and T_1.

And

$\boldsymbol{\tau_g}$ – the gyroscopic torque.

$$\tau_f = \begin{pmatrix} l(f_4 - f_2) \\ l(f_3 - f_1) \\ T_1 - T_2 + T_3 - T_4 \end{pmatrix} = \begin{pmatrix} l\, b\left(\omega_4^2 - \omega_2^2\right) \\ l\, b\left(\omega_3^2 - \omega_1^2\right) \\ d\left(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2\right) \end{pmatrix} \tag{17}$$

$\boldsymbol{\tau_g}$ – Each rotor may be considered as a rigid disc in rotation around his own vertical axis with a rotation ω_i. The vertical axis itself moves during the rotation of the quadrotor around of one of their three axes. This action product an extra torque called gyroscopic given as follow [17]:

$$\boldsymbol{\tau_g} = \sum_{i=1}^{4} \boldsymbol{\Omega} \wedge J_r \begin{pmatrix} 0 \\ 0 \\ (-1)^{i+1}\omega_i \end{pmatrix} = \begin{pmatrix} \dot{\theta} J_r \Omega_r \\ -\dot{\varphi} J_r \Omega_r \\ 0 \end{pmatrix} \tag{18}$$

Where:

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{19}$$

And J_r is the inertia of the rotor.

We apply (17) and (18) in (16) we get:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} l\, b\left(\omega_4^2 - \omega_2^2\right) - \dot{\theta} J_r \Omega_r \\ l\, b\left(\omega_3^2 - \omega_1^2\right) + \dot{\varphi} J_r \Omega_r \\ d\left(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2\right) \end{pmatrix} \tag{20}$$

From (15) to (20) and by applying the Lagrangian law presented in (8), we obtain after simplification the following mathematical model.

$$f(X,U) = \begin{cases} \ddot{x} = \frac{1}{m}(c\varphi s\theta c\psi + s\varphi s\psi)U_1 \\ \ddot{y} = \frac{1}{m}(c\varphi s\theta s\psi - s\varphi c\psi)U_1 \\ \ddot{z} = \frac{1}{m}(c\varphi c\theta)U_1 - g \\ \ddot{\varphi} = \frac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} + \frac{l}{I_x}U_2 - \frac{J_r\Omega_r}{I_x}\dot{\theta} \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y}\dot{\varphi}\dot{\psi} + \frac{l}{I_y}U_3 + \frac{J_r\Omega_r}{I_y}\dot{\varphi} \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4 \end{cases} \tag{21}$$

Where:

The quadrotor is controlled by the rotation speed of motors ($\omega_1\, \omega_2\, \omega_3$ and ω_4), then the vector of control inputs is expressed as a function of rotation speed as follows:

$$\begin{cases} U_1 = b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2\right) \\ U_2 = b\left(\omega_4^2-\omega_2^2\right) \\ U_3 = b\left(\omega_3^2-\omega_1^2\right) \\ U_4 = d\left(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2\right) \end{cases} \tag{22}$$

The inputs of the system are altitude control (U_1), roll control (U_2), pitch control (U_3) and yaw control (U_4).

The states vector X:

$$X = \left[x, \dot{x}, y, \dot{y}, z, \dot{z}, \varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}\right]^T \tag{23}$$

The inputs vector U:

$$U = \left[U_1, U_2, U_3, U_4\right]^T \tag{24}$$

The equilibrium points of (21) satisfy:

$$f\left(X_{eq}, U_{eq}\right) = 0 \tag{25}$$

Where:

$$X_{eq}=\left[x_{eq}, \dot{x}_{eq}, y_{eq}, \dot{y}_{eq}, z_{eq}, \dot{z}_{eq}, \varphi_{eq}, \dot{\varphi}_{eq}, \theta_{eq}, \dot{\theta}_{eq}, \psi_{eq}, \dot{\psi}_{eq}\right]^T \quad (26)$$

$$U_{eq}=\left[U_{1eq}, U_{2eq}, U_{3eq}, U_{4eq}\right]^T \quad (27)$$

Solving (25) results in the stationary points

$$X_{eq}=\left[x_{ss}, 0, y_{ss}, 0, z_{ss}, 0,0,0,0,0, \psi_{ss}, 0\right]^T \quad (28)$$

$$U_{eq}=\left[g\ m, 0,0,0\right]^T \quad (29)$$

Where: x_{ss}, y_{ss}, z_{ss}, $\psi_{ss} \in R$, we notice *ss* to mean *steady state*.

In addition, there is a limit in roll (φ) and pitch (θ) angles:

$$\frac{-\pi}{2} \leq \varphi, \theta \leq \frac{\pi}{2} \quad (30)$$

As a simulation, we use the parameters of the OS4 project (**O**mni-directional **S**tationary **F**lying **OU**tstretched **R**obot) [2]:

Table 1. OS4 parameters

Parameter	Name	Value	Unit
m	Mass	0.65	Kg
l	Arm length	0.23	m
b	Thrust coefficient	$3.13\ 10^{-5}$	$N.s^2$
d	Drag coefficient	$7.5\ 10^{-7}$	$N.m.s^2$
I_x	Inertia on x axis	$7.5\ 10^{-3}$	$Kg.m^2$
I_y	Inertia on y axis	$7.5\ 10^{-3}$	$Kg.m^2$
I_z	Inertia on z axis	$1.3\ 10^{-2}$	$Kg.m^2$
J_r	Rotor inertia	$6\ 10^{-5}$	$Kg.m^2$
g	Gravity constant	9.8	$m.s^{-2}$

3. System Control

3.1. Modeling for Control

In fact, each rotor has a non-real time dynamic response, so we have to consider their dynamic. In OS4 project [2] the rotors have a dynamic transfer function (RD) as:

$$RD(s)=\frac{\omega_i}{\omega_{di}}=\frac{0.936}{0.178s+1} \quad (31)$$

Where:
ω_i – The actual angular velocity for each rotor.
ω_{di} – The desired angular velocity for each rotor.

Most researchers who work in control theory, applied on quadrotor and its simulation, didn't consider rotors dynamic in their works. We notice the desired inputs U_{di} which will be converted to rotors inputs as follow:

$$\begin{cases} \omega_{d1}=\sqrt{\frac{1}{4b}U_{d1}-\frac{1}{2b}U_{d3}+\frac{1}{4d}U_{d4}} \\ \omega_{d2}=\sqrt{\frac{1}{4b}U_{d1}-\frac{1}{2b}U_{d2}-\frac{1}{4d}U_{d4}} \\ \omega_{d3}=\sqrt{\frac{1}{4b}U_{d1}+\frac{1}{2b}U_{d3}+\frac{1}{4d}U_{d4}} \\ \omega_{d4}=\sqrt{\frac{1}{4b}U_{d1}+\frac{1}{2b}U_{d2}-\frac{1}{4d}U_{d4}} \end{cases} \quad (32)$$

Unlike the altitude and orientation of the quadrotor, its x and y position can't be directly controlled using one of the four controls laws U_{d1} through U_{d4}. On the other hand, the x and y position can be controlled through the roll and pitch angles. Then we suppose a virtual controls u_x and u_y where:

$$\begin{cases} \ddot{x}=u_x=\frac{1}{m}U_{d1}\left(c\varphi s\theta c\psi+s\varphi s\psi\right) \\ \ddot{y}=u_y=\frac{1}{m}U_{d1}\left(c\varphi s\theta s\psi-s\varphi c\psi\right) \end{cases} \quad (33)$$

The desired roll and pitch angles (φ_d and θ_d) can be computed from the translational equations of motion [3]. The expression (33) is given by:

$$\begin{cases} u_x=\frac{1}{m}U_{d1}\left(\cos\varphi_d \sin\theta_d \cos\psi+\sin\varphi_d \sin\psi\right) \\ u_y=\frac{1}{m}U_{d1}\left(\cos\varphi_d \sin\theta_d \sin\psi-\sin\varphi_d \cos\psi\right) \end{cases} \quad (34)$$

After simplification, because we have considered that the quadrotor is operating hover flight mode, we obtain the translation to rotation converter as:

$$\begin{cases} \varphi_d=\frac{m}{U_{d1}}\left(u_x \sin\psi-u_y \cos\psi\right) \\ \theta_d=\frac{m}{U_{d1}}\left(u_x \cos\psi+u_y \sin\psi\right) \end{cases} \quad (35)$$

We have now two control loops. Inner loop which contains the altitude (z) and the three angles (φ, θ and ψ) and the outer loop which contains the translation position coordinates x and y (Fig. 2).

3.2. Control Using PID Technique

The control strategy proposed in this work is the PID technique. The factors that attracted industries to choose PID could be due to low cost, easy to maintain, as well as simplicity in control structure and easy to understand [22]. Once the set point has been changed, the error will be computed between the set point and the actual output. The error signal is used to generate the proportional, integral and derivative actions, with the resulting signals weighted and summed to form the control signal applied to the plant model [22].

The six controls laws applied to quadrotor are:

$$U_q=k_p\left(q_d-q\right)+k_d\left(\dot{q}_d-\dot{q}\right)+k_i\int\left(q_d-q\right)dt \quad (36)$$

Where: $\boldsymbol{q}=[x\ y\ z\ \varphi\ \theta\ \psi]$
and $\boldsymbol{U_q}=[u_x\ u_y\ U_1\ U_2\ U_3\ U_4]$.

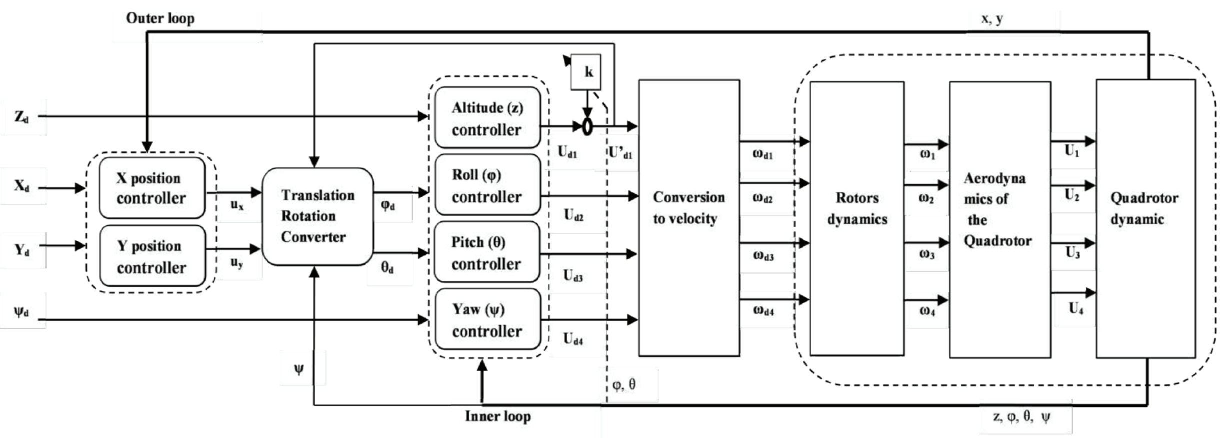

Fig. 2. Control structure of quadrotor

For each variable, we can design its controller independently of the others; for that, each loop of variable is being a mono-variable system. In this case, we can tune the parameters of each variable alone but, there are constraints related to the couples between outputs variables, so we use decentralized control.

3.3. Decentralized Control

The steps followed to control a MIMO system by decentralized control are: analyze the interactions, then decoupling them (for being weak) [23].

We can characterize two types of interactions:

(a). The interactions between the translation coordinates (x and y) with the roll and pitch angles, as we notice in section (3.1).

(b). The interactions between the variables of inner loop (Fig. 2), the altitude (z) and (φ, θ) are strongly coupled. The rest of interactions are negligible; they don't have direct influences.

For decoupling (a) we apply the sequential design [24], we design the controllers of the inner loop (altitude and three angles, as presented in Fig. 2), then the controllers of the outer loop (x and y) as a cascade design.

On the other hand, for (b), the influence of (φ, θ) in the altitude (z) is estimated ($\cos\varphi\cos\theta$). Then, to weaken this influence the idea is to replace the input U_{d1} by U'_{d1} such as:

$$U'_{d1} = kU_{d1} \tag{37}$$

Where: k is the adaptation coefficient:

$$k = \frac{1}{\cos\varphi\cos\theta} \tag{38}$$

We can now, apply the genetic algorithm technique for each controller.

4. Genetic Algorithm

For the reasons that the quadrotor system is non-linear and some outputs (x and y) cannot be directly controlled (they are underactuated), we cannot tune the PID parameters with one of classical approaches. Different methods can be used to obtain optimal control of robots [25], [26] in particular, optimal parameters of the PID, among them; we can find meta-heuristic techniques of optimization such as DE [27], PSO [28] and GA [27], [29] which were used in to solve some specific problems. We propose in this work, to use GA as an optimization technique to obtain the best PID's parameters. The use of such algorithm allows the control design of the system without the need of its linearization, which is not provided by using classical methods. Likewise, there are some degrees of freedom which are underactuated; as the position according x and y that have virtual controllers (u_x and u_y) and not direct inputs on the system. These last, can't be provided by classical methods also. The application of the proposed technique will improve the quality of our control system.

Genetic algorithms are group of operations which are extracted from selection evolution nature, proposed firstly by Holland [30]. They were used extensively in artificial intelligence with computer science problems. The process of GAs starts by creating a random population of solutions, which are encoded in chromosomes, then, they are evaluated according to the performance of the problem, by using particular criteria, called fitness function. In order to improve the quality of solutions, to find the best ones, there are some operations which must be applied on the population; selection, crossover and mutation.

4.1. Implementation of the GA

In this work, we create randomly an initial population, which contains fifty real encoding chromosomes. The fitness function is chosen as the Integral of the Squared Error (ISE).

$$fitness = ISE = \int_0^{\tau} e^2(t)\,dt \tag{39}$$

The selection of the parents to be crossed is applied by using BMW determinist scheme [31], [32]. In this case, the best solution, in the current population, mates the worst one and the less good mates the less worse. Because the encoding is real, a linear crossover presented by Wright in [33] is used. In a GA, after the crossover operation, a new population of children is obtained. This population must to be slowly disturbed by applying the Gaussian mutation operator with a probability of 4%. The next generation, is then made by taking the fifty best solutions of both populations of parents and offspring. This algorithm that includes all these operations is applied for thirty iterations, which the number of generations.

In this work, there is no determined formula for error function, so we use (*ode45.m*) function because the system is evolution. Then, the optimization program sends the chromosome parameters to this function and it receives its fitness function. However, we have to discrete the evolution (sample time T_s = 0.05 s, simulation time = 10 s).

Fig. 3 presents the diagram of control optimization with GA applied for each controlled variable.

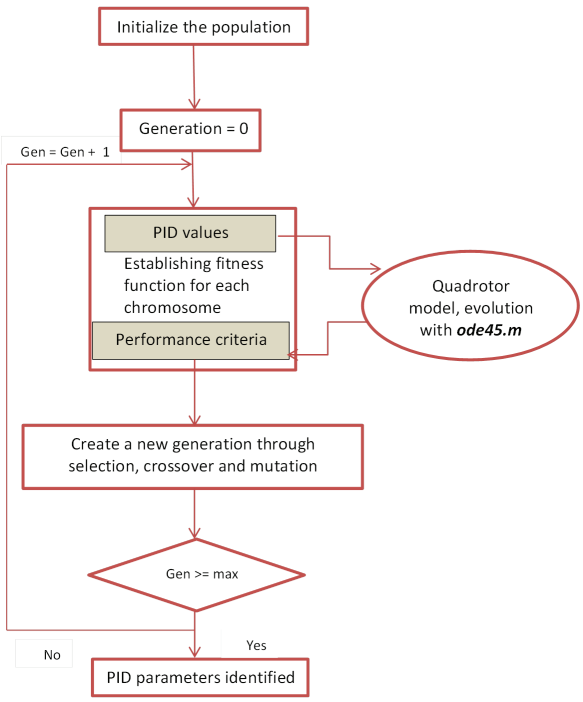

Fig. 3. Diagram of applied GA

4.2. Optimums Parameters obtained by GA

The GA algorithm is applied as an optimization strategy, under MATLAB environment, to the quadrotor model OS4 [2] with parameters presented in Table.1, for each variable except the yaw subsystem. This last, could be linear with hypothesis that the quadrotor is symmetric ($I_x = I_y$).

The equation from the model (21)

$$\left(\ddot{\psi} = \frac{\left(I_x - I_y\right)}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4\right) \text{ will be:}$$

$$\ddot{\psi} = \frac{1}{I_z}U_4 \tag{40}$$

The yaw has been controlled with classical PD.

As a result, we obtain a PID controller for the altitude and PDs for the others controllers, the optimum parameters are presented in Table 2. These PID's parameters are applied to simulate the control of the quadrotor under MATLAB/Simulink.

Table 2. Optimums PID's parameters obtained by GA

Variable	Kp	Ki	Kd
z	9.6135	4.0704	4.8897
φ, θ	0.4521	0	0.3622
ψ	0.15	0	0.15
x, y	4.8095	0	0.8967

5. Results

After the design of the controllers, we have to test their performance for the trajectory tracking. Table 3 presents the suggested trajectories' dynamics and their conditions.

Table 3. Trajectories' conditions and dynamics

		Mode 1	Mode 2
1st trajectory	Initial condition	$[x, y, z, \varphi, \theta, \psi]$ = [0,0,0,0,0,0]	$[x, y, z, \varphi, \theta, \psi]$ = [0,–2,2,0,0,0]
	Period	[0, 7]s	[7, 60]s
	Actuators' saturation	[0, 400] rad/s max = 370 rad/s	[0, 400] rad/s max = 296 rad/s
	Description Of Trajectory	$z_d = 2$ $x_d = 0$ $y_d = -2$ $\psi_d = 0$	$z_d = t$ $x_d = 2\sin\left(\frac{2\pi}{30}t\right)$ $y_d = 2\sin\left(\frac{2\pi}{30}t - \frac{\pi}{2}\right)$ $\psi_d = \pi\sin\left(\frac{2\pi}{30}t\right)$
	Attitude's saturation	$-20° < \varphi, \theta < 20°$	
2nd trajectory	Initial condition	$[x, y, z, \varphi, \theta, \psi]$ = [0,0,0,0,0,0]	$[x, y, z, \varphi, \theta, \psi]$ = [0,0,3,0,0,0,0]
	Period	[0, 7]s	[7, 37]s
	Actuators' saturation	[0, 400] rad/s max = 400 rad/s	[0, 400] rad/s max = 300 rad/s
	Description Of trajectory	$z_d = 3$ $x_d = 0$ $y_d = 0$ $\psi_d = 0$	$z_d = -0.1t$ $x_d = 4\sin\left(\frac{2\pi}{8}t\right)$ $y_d = 4\sin\left(\frac{2\pi}{8}t\right)$ $\psi_d = 0$
	Attitude's saturation	$-90° < \varphi, \theta < 90°$	

5.1. First trajectory

The suggested trajectory is a spiral, where the quadrotor makes a circle in 30 seconds and up (1 m/s). Fig. 4 shows the tracking of the desired trajectory for each position coordinate (z, x and y) and the rotation of the quadrotor with the yaw angle too. Fig. 5 shows the tracking of desired trajectory in 3D plan. The rotors angular speeds are shown in Fig. 6.

Firstly, from 0 to 7 seconds the quadrotor is stabilized in starter position $(x, y, z) = (0, -2, 2)$. Starting mode needs a big energy to up against the effect of

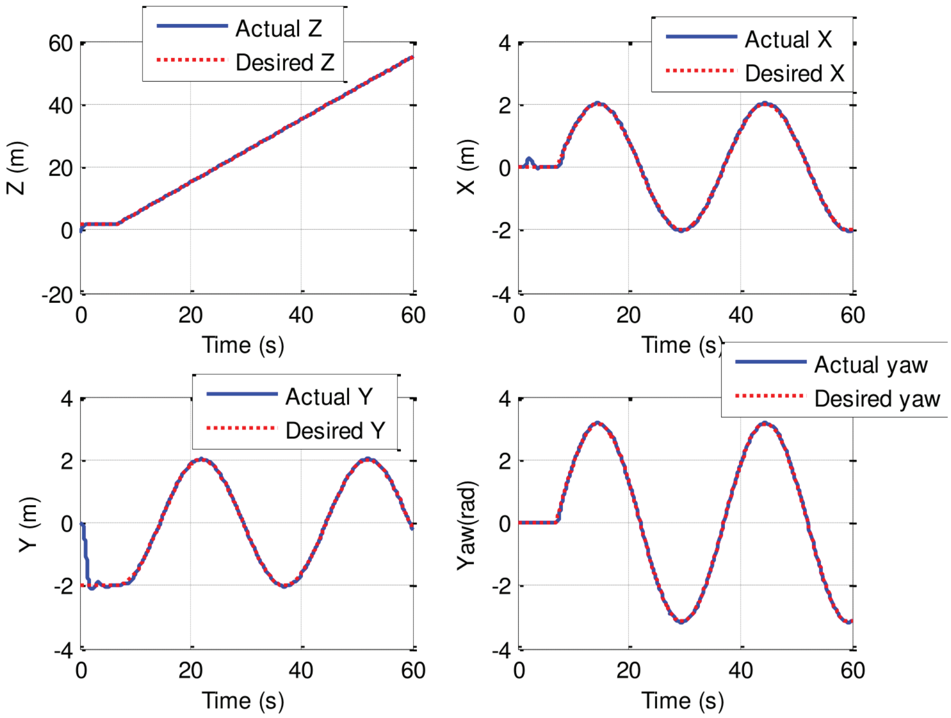

Fig. 4. z, x, y and yaw (ψ) responses (1st Trajectory)

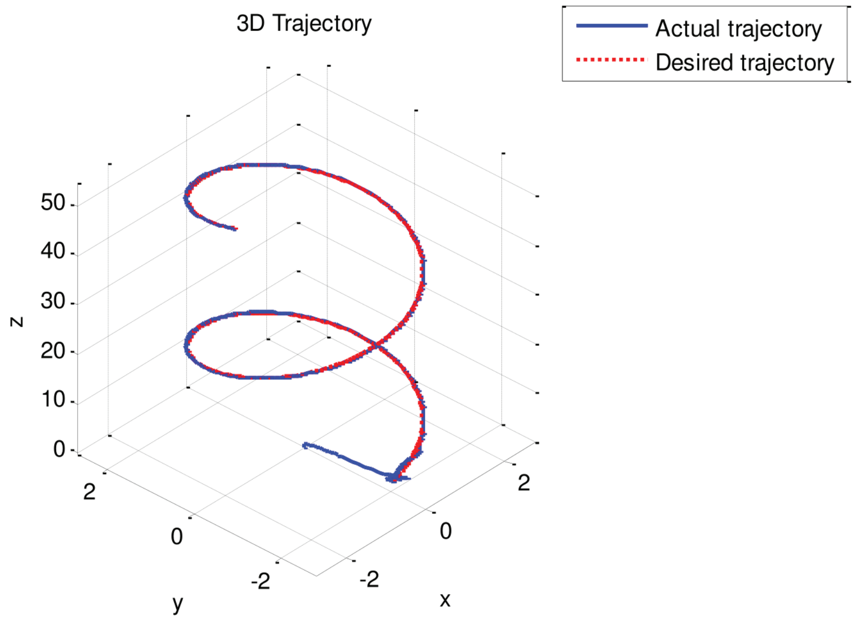

Fig. 5. Desired and actual trajectories in 3D (1st Trajectory)

gravity. Fig. 6 indicates that the angular speed of rotors in this mode can move up to 400 rad/s.

After that, the trajectory begins; a linear movement in altitude with a speed of about 1 m/s and for translation motion, the quadrotor makes a complete circle of 2 meters of radius in 30 seconds. In the same time, the quadrotor turns on around itself, and Fig. 4 shows the tracking of the desired trajectory with a precision at the end of the movement, of all variables (altitude z, translation motion of x and y, the yaw angle). In this mode, low energy is applied, as showed in Fig. 6, where the rotors turn on around 220 rad/s. These results show clearly the possibility of our system to follow trajectories and to design x and y controllers as an underactuated case.

5.2. Second Trajectory

Unlike the first trajectory which is hovering as a circle with small roll ant pitch angles, we will show the effect of the proposed technique in the case of big angles. The suggested trajectory is as follow; after the quadrotor rises in 3 meters of altitude, on the second 7 it begins to down slowly with 10 cm/s; in the same time, it balances diagonally between –4 meters and 4 meters in x axis and y axis, the period of bal-

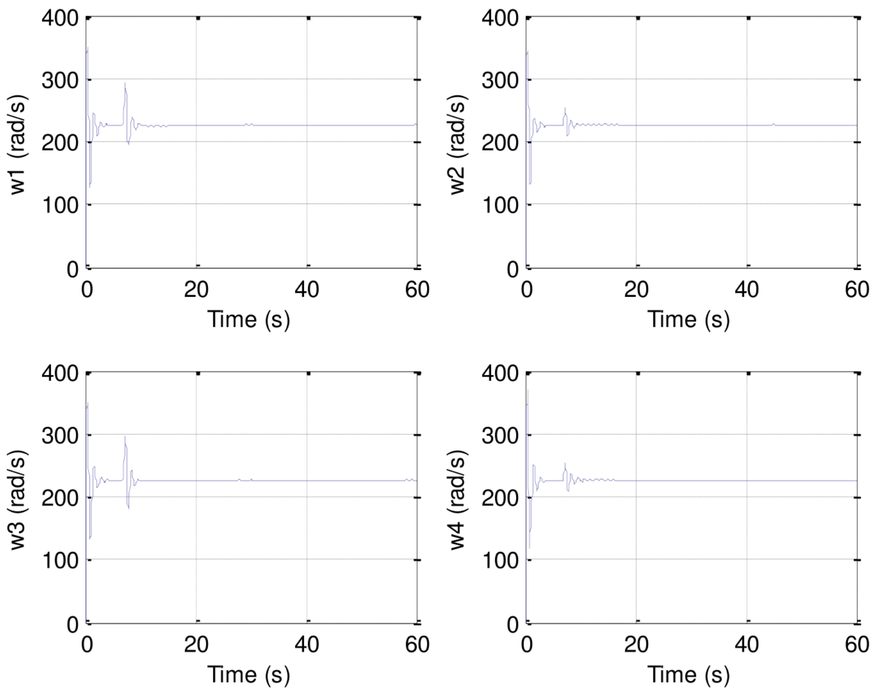

Fig. 6. Rotors angular speeds (ω_1, ω_2, ω_3 and ω_4) in rad/s (1st Trajectory)

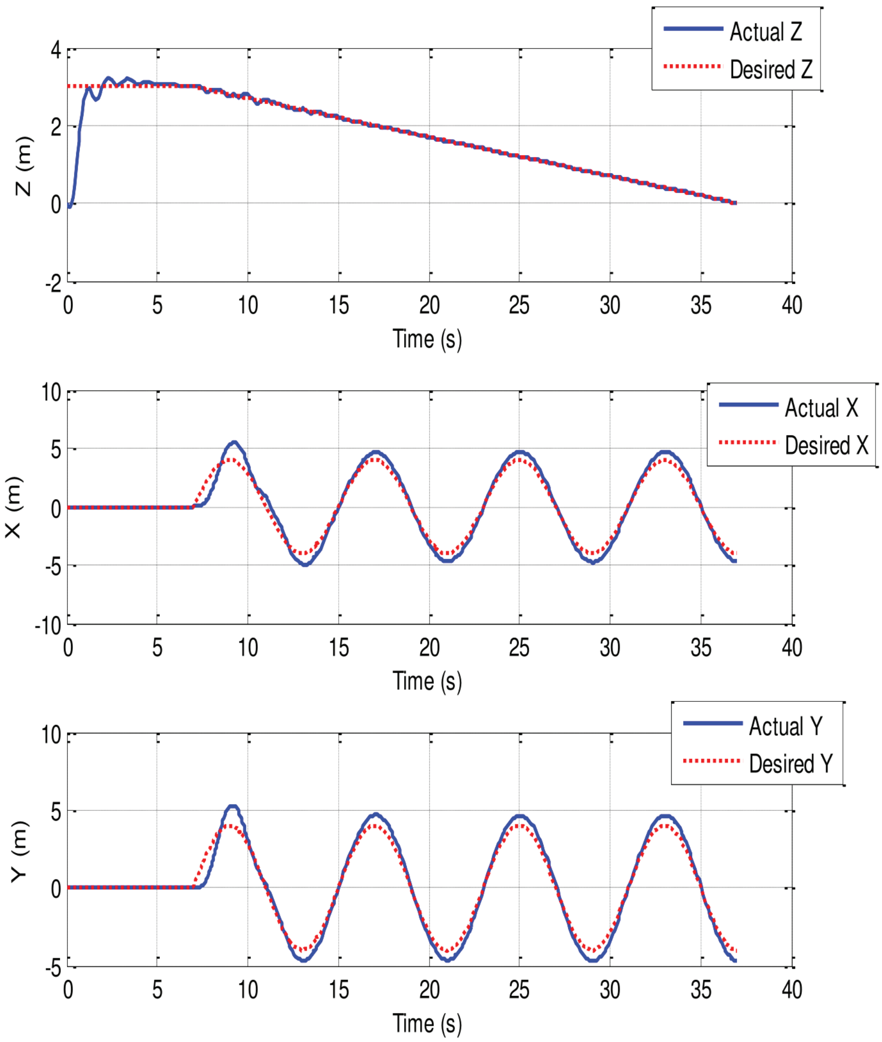

Fig. 7. z, x and y responses (2nd trajectory)

anced is just 8 seconds. Fig. 7 shows the tracking in *z*, *x* and *y* axis, where we notice that the error in *z* is null but in the *x* and *y* axis is not in the peaks. Fig. 8 shows the trajectory tracking in 3D plan to make the trajectory clearer. Fig. 9 shows the rotors' angular speed, which need a big energy in the starter mode (the speed reaches 400 rad/s) but when the quadrotor stabilizes in the balanced trajectory they are around

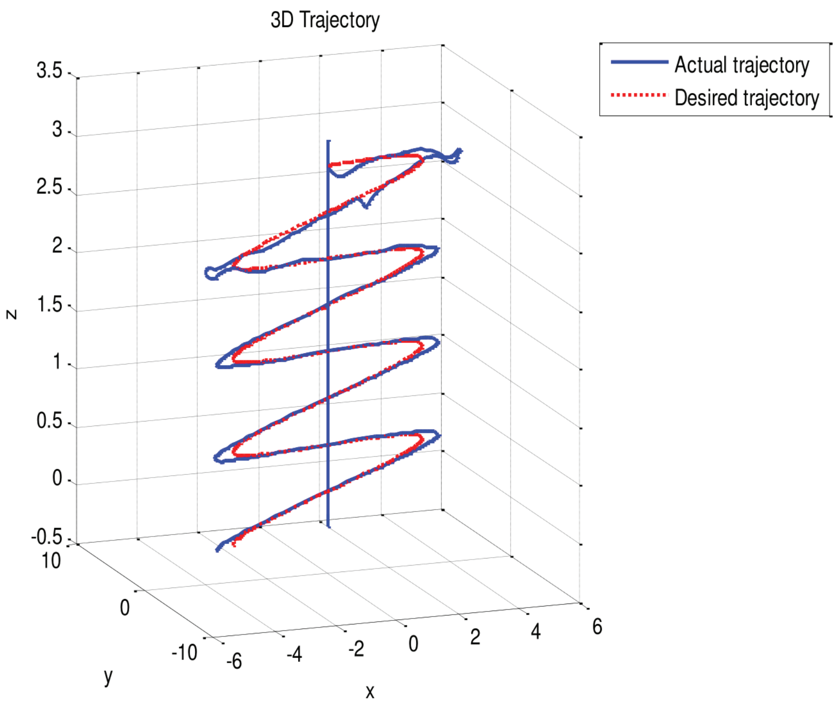

Fig. 8. Desired and actual trajectory in 3D plan (2nd trajectory)

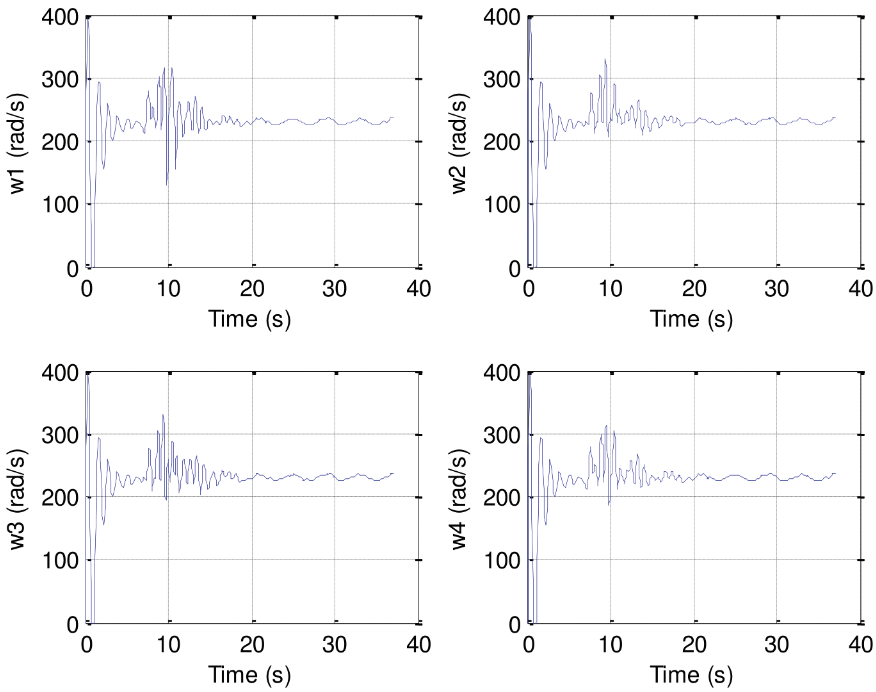

Fig. 9. Rotors'angular speeds ω_1, ω_2, ω_3 and ω_4 in rad/s (2nd trajectory)

220 rad/s. Fig. 10 shows the roll (φ) and pitch (θ) angles that the quadrotor makes this trajectory by them; we notice that they are between –55° and 50° which makes the system absolutely nonlinear because the linear interval is between –10° and 10°

6. Conclusion

In this paper, we have proposed the control of a non-linear, MIMO system, which is the quadrotor. The mathematical model of this system has been developed in details, including its aerodynamic effects and rotors dynamics. The quadrotor is a non-linear complex system, strongly coupled and under-actuated. A linear PID control technique was developed and synthesized, according to the decentralized control approach, for which, we have decoupled the interactions between the quadrotor variables. A complete simulation was implemented on MATLAB/Simulink tool relying on the derived mathematical model of the quadrotor system. The tuning of controllers parameters are done using GA, where the objective function is the dynamic response of the system in term of ISE. The response's simulation of the system presents a path tracking in 3D plan; where we notice that the tracking trajectory stabilizes, after a few seconds, this time is what the quadrotor needs to stabilize the altitude

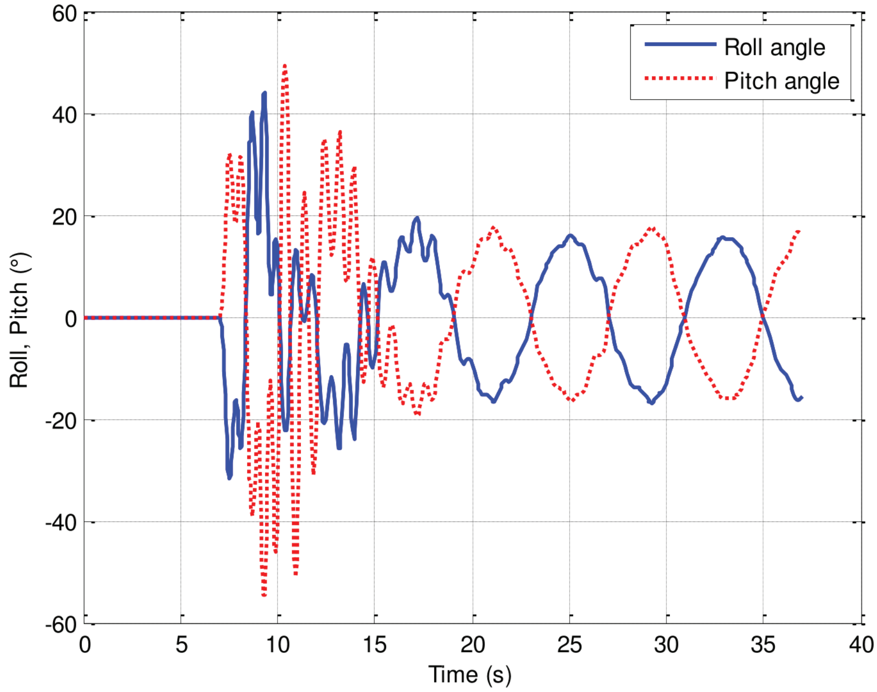

Fig. 10. The roll and pitch angles responses (2nd trajectory)

because a biggest energy is required in this period. Although the simplicity of the controllers' design (PID), the effect of proposed technique and designing the controllers by GA is shown in terms of tracking errors and stability, even with the big angles, subsequently, high velocities response and high dynamic performances, this is in contrast to works that used the PID with linearized model [12], [14], [15], [16], the variation of the angles is limited between –10° and 10° which is the linear range, so, limit the dynamics by low performance. Moreover, compared of [34], which is a similar work, when the GA was used to tune the PID's parameters, but in the evaluation's phase, they didn't use a high dynamic trajectory and the angles never exceeded the linear range. Even some works, where a nonlinear control is designed in, the used trajectory to evaluate the controller doesn't contain high dynamics with big range of angles; [1], [6], [7], and [35].

Finally, the shown rotors' speeds are acceptable in reality; so we have the possibility to apply these controllers in real system to get a trajectory tracking with a precision as presented in 3D trajectory and we don't need more powerful actuators.

AUTHORS

Seif-El-Islam Hasseni* – Energy Systems Modeling Laboratory (LMSE), Electrical Engineering Department; University of Biskra, BP 145 RP, 07000, Biskra, Algeria.E-mail: seif.hasseni@univ-biskra.dz.

Latifa Abdou – Identification, Command, Control and Communication Laboratory (LI3CUB), Electrical Engineering Department; University of Biskra, BP 145 RP, 07000, Biskra, Algeria.

*Corresponding author

REFERENCES

[1] J.J. Xiong, E.H. Zheng, "Position and attitude tracking control for quadrotor UAV", *ISA Trans.*, vol. 53, 2014, 725–731. DOI: 10.1016/j.isatra.2014.01.004.

[2] S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying*, PhD thesis, EPFL, Switzerland, 2007.

[3] H. Elkholy, *Dynamic modeling and control of a quadrotor using linear and nonlinear approaches*, Master thesis, American university in Cairo, Egypt, 2014.

[4] Z. Jia, J. Yu,Y. Mei *et al.*, "Integral backstepping sliding mode control for quadrotor helicopter under external uncertain disturbances", *Aerosp. Sci. Technol.*, vol. 68, 2017, 299–307. DOI: 10.1016/j.ast.2017.05.022.

[5] F. Chen, K. Zhang, Z. Wang *et al.*, "Trajectory tracking of a quadrotor with unknown parameters and its fault-tolerant control via sliding mode fault observer", *Proc. IMechE Part I: J Systems and Control Engineering*, 229, 2015, 279–292. DOI: 10.1177/0959651814566040.

[6] Y. Yang, Y. Yan, "Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control", *Aerosp. Sci. Technol.*, vol. 54, 2016, 208–217. DOI: 10.1016/j.ast.2016.04.005

[7] H. Lu, C. Liu, M. Coombes *et al.*, "Online optimisation-based backstepping control design with application to quadrotor", *IET Control Theory Appl.*, vol. 10, 2016, 1601–1611. DOI: 10.1049/iet-cta.2015.0976.

[8] D. Ma, Y. Xia, T. Li *et al.*, "Active disturbance rejection and predictive control strategy for a quadrotor helicopter", *IET Control Theory Appl.*, vol. 10, 2016, 2213–2222. DOI: 10.1049/iet-cta.2016.0125.

[9] K. Alexis, G. Nikolakopoulos, A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies", *IET Con-*

trol Theory Appl., vol. 6, 2012, 1812–1827. DOI: 10.1049/iet-cta.2011.0348.

[10] P. Zarafshan, S.B. Moosavian, S.A.A. Moosavian *et al*., "Optimal Control of an Aerial Robot". In: *Proc. of the 2008 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Xi'an, China, 2008, 1284–1289. DOI: 10.1109/AIM.2008.4601847.

[11] P. Zarafshan, S.A.A. Moosavian, M. Bahrami, "Comparative controller design of an aerial robot", *Aerosp. Sci. Technol*., vol. 14, 2010, 276–282. DOI: 10.1016/j.ast.2010.01.001.

[12] S. Bouabdallah, A. Noth, R. Siegwart, "PID vs LQ Control Techniques Applied to an indoor micro quadrotor". In: *Proc. of IEEE/RSJ Int. Conf. on intelligent robots and systems*, Sendal, Japan, 2004, 2451–2456. DOI:10.1109/IROS.2004.1389776.

[13] B. Kada, Y. Ghazzawi, "Robust PID Controller Design for an UAV Flight Control System". In: *Proc. of the World Congress on Engineering and Computer Sci*., vol. II, San Francisco, USA, 2011.

[14] S.J. Haddadi, P. Zarafshan, "Attitude Control of an Autonomous Octorotor", In: *Proc. of the 2nd RSI/ISM Int. Conf. on Robotics and Mechatronics*, Tehran, Iran, 2014, 540–545. DOI: 10.1109/ICRoM.2014.6990958.

[15] Q. Quan, G.X. Du, K.Y. Cai, "Proportional-Integral Stabilizing Control of a Class of MIMO Systems Subject to Nonparametric Uncertainties by Additive-State-Decomposition Dynamic Inversion Design", *IEEE/ASME Trans. Mech*., vol. 21, 2016, 1092–1101. DOI: 10.1109/TMECH.2015.2497258.

[16] N. Cao, A.F. Lynch, "Inner-Outer Loop Control for Quadrotor UAVs With Input and State Constraints", *IEEE Trans. Contr. Syst. Technol*., vol. 24, 2016, 1797–1804. DOI: 10.1109/TCST.2015.2505642.

[17] L. Carrillo, A. Lopez, R. Lozano *et al*., *Quad Rotorcraft Control*, London: Springer-Verlag, 2013.

[18] J. Ghommam, N. Fethalla, M. Saad, "Quadrotor circumnavigation of an unknown moving target using camera vision-based measurements", *IET Control Theory Appl*., vol. 10, 2016, 1874–1887. DOI: 10.1049/iet-cta.2015.1246.

[19] P.J. Fleming, R.C. Purshouse, *Genetic Algorithms in Control Systems Engineering*, Schiefled: Departement of Automatic Control and Systems Engineering, University of Schiefled, 2001.

[20] M.W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, New York: JohnWiley-&Sons, 2006.

[21] *ROMANSY 21 – Robot Design, Dynamics and Control – Proceedings of the 21st CISM-IFToMM Symposium*, V. Parenti-Castelli, W. Schiehlen (eds.) June 20–23, 2016. Udine, Italy, Springer.

[22] K.J. Astrom, T. Hagglund, *Automatic Tuning of PID Controllers*, Pennsylvania: Instrument Society of America, 1988.

[23] M.S. Mahmoud, *Decentralized Systems with Design Constraints*, London: Springer-Verlag, 2011.

[24] M. Hovd, S. Skogestad, "Sequential Design of Decentralized Controllers", *Automatica*, vol. 30, 1994, 1601–1607. DOI: 10.1016/0005-1098(94)90099-X.

[25] X. Dai, L. Jiang, Y. Zhao, "Cooperative exploration based on supervisory control of multi-robot systems", *Appl. Intell*., vol. 45, 2016, 18–29. DOI: 10.1007/s10489-015-0741-3.

[26] E. Cuevas, A. Luque, D. Zaldívar *et al*., "Evolutionary calibration of fractional fuzzy controllers", *Appl. Intell*., vol. 47, 2017, 291–303. DOI: 10.1007/s10489-017-0899-y.

[27] M.S. Saad, H. Jamaluddin, I.Z.M. Darus, "Implementation of PID Controller Tuning Using Differential Evolution and Genetic Algorithms", *Int. J. of Innovative Computing, Information and Control*, vol. 8, 2012, 7761–7779.

[28] J.S. Wang, C.X. Ning, Y. Yang, "Multivariable PID Decoupling Control Method of Electroslag Remelting Process Based on Improved Practical Swarm Optimisation (PSO) Algorithm", *Information*, vol. 5, 2014, 120–133. DOI: 10.3390/info5010120.

[29] K. Rajarathinam, J.B. Gomm, D.L. Yu *et al*., "PID Controller Tuning for a Multivariable Glass Furnace Process by Genetic Algorithm", *Int. J of Automation and Computing*, vol. 13, 2016, 64–72. DOI: 10.1007/s11633-015-0910-1.

[30] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Massachusetts: MIT Press, 1992.

[31] B.K. Yeo, Y. Lu, "Array Failure Correction with a Genetic Algorithm", *IEEE Trans. on Antennas and Propagation*, vol. 47, 1999, 823–828. DOI: 10.1109/8.774136.

[32] L. Abdou, F. Soltani, "OS-CFAR and CMLD Threshold Optimization with Genetic Algorithms". In: *Proc. of 3rd Int. Conf. on Systems, Signals &Devices*, vol. III Communication and Signal Processing, Sousse, Tunisia, 2005.

[33] A. Wright, *Genetic Algorithms for Real Parameter Optimization*, San Mateo, California: Morgan Kaufmann, 1991.

[34] A. Alkamachi, E. Erçelbi, "Modelling and Genetic Algorithm Based-PID Control of H-Shaped Racing Quadcopter", *Arab J. Sci. Eng*., vol. 42, 2017, 2777–2786. DOI: 10.1007/s13369-017-2433-2.

[35] Y. Zou, "Trajectory tracking controller for quadrotors without velocity and angular velocity measurements", *IET Control Theory Appl*., vol. 11, 2017, 101–109. DOI: 10.1049/iet--cta.2016.0647.