

# Design Tunable Robust Controllers for Unmanned Aerial Vehicle Based on Particle Swarm Optimization Algorithm

Baqir Nasser Abdul- Samed<sup>1</sup> and  
Ammar A. Aldair<sup>2</sup>  
<sup>1,2</sup>Electrical Engineering  
Department; University of Basrah,  
Iraq

**Abstract**— PID controller is the most popular controller in many applications because of many advantages such as its high efficiency, low cost, and simple structure. But the main challenge is how the user can find the optimal values for its parameters. There are many intelligent methods are proposed to find the optimal values for the PID parameters, like neural networks, genetic algorithm, Ant colony and so on. In this work, the PID controllers are used in three different layers for generating suitable control signals for controlling the position of the UAV (x,y and z), the orientation of UAV ( $\theta$ ,  $\phi$  and  $\psi$ ) and for the motors of the quadrotor to make it more stable and efficient for doing its mission. The particle swarm optimization (PSO) algorithm is proposed in this work. The PSO algorithm is applied to tune the parameters of proposed PID controllers for the three layers to optimize the performances of the controlled system with and without existences of disturbance to show how the designed controller will be robust. The proposed controllers are used to control UAV, and the MATLAB 2018b is used to simulate the controlled system. The simulation results show that, the proposed controllers structure for the quadrotor improve the performance of the UAV and enhance its stability.

**Keywords**— quadrotor, PID controller, PSO Algorithm, position control, orientation control.

## I. INTRODUCTION

The UAV is a small size helicopter and it is widely used in many different applications. The first quadrotor was invented in 1907 by Breguet-Richet Gyroplane. This quadrotor was manned and vertical takeoff and landing vehicle. Rapid development in technologies and electronic circuits makes the quadrotor developed very fast to become unmanned and even smaller with low cost and easy to use in many applications around the world [1].

Many researchers studied and designed PID controller for the quadrotor by using different strategies. Flight control system was developed by authors in reference [2] contains a controller with a robust PID controller. The system uses the deadbeat response and model reduction techniques to overcome the conventional short comings. In the reference [3] the author studied the flying performance of the Multirotor VTOL by using different control structures. Control strategy is framed as an angular stabilization of the four-rotor

structure. For better endurance of the quadrotor vehicle in hovering mode the cascade control system has been suggested. The design and the practical comprehension of the control architecture on the prototype vehicle are reviewed. In the reference [4], the authors studied attitude control of a quad rotor using PID controller, analytical method of PID tuning is employed Performance of the control structure is studied on several domain factors. In reference [5], the authors developed and constructed a quadrotor which is a type of an unmanned air vehicle with necessarily four motors which are capable for producing the movement of the air platform. Dynamic modeling based on the Newton-Euler method, experimental tests of a PID controller with microcontroller system project for the proposed structure and simulations schemes using Simulink/Matlab environment were presented. Mathematical modeling and simulation of an unmanned quadrotor vehicle was studied in the refence [6]. The authors reviewed the complex structure, the nonlinear dynamical modeling and the highly unstable nature of the vehicle. The developed model was later used to design a PID controller system to stabilize the roll, pitch and yaw angles of the quad rotor. Mainly, there are two configurations are used for the construction of quadrotor: the cross shape configuration or “X” and the plus shape configuration or “+”. Each configuration has different characteristics and application to



Figure 1 Configuration shape of quadrotor

be chosen as shown in Fig. 1 [1].

A quadrotor has four propellers and four motors that are attached to the its body frame, each two motors (not consecutive propellers) rotate in the same direction, the rotations of propellers 1 and 4 are clockwise direction while

the rotations of the other two propellers are counterclockwise direction to stabilize the quadrotor as shown in Fig. 2 [7].



Figure 2 Rotation of the four motors

A quadrotor has six-degree of freedom (6 DOF): three-degree of freedom represents the *rotational movements*, while the other three-degree of freedom represents the *translation movements* [7,8]. Quadrotor dynamics are simple and it depends on the changing speed of the four motors. The changing speed creates the following movements: forward, backward, up, down, right rotational and left rotational [8,9]. Fig. 3 shows the six-degree of freedom of quadrotor.

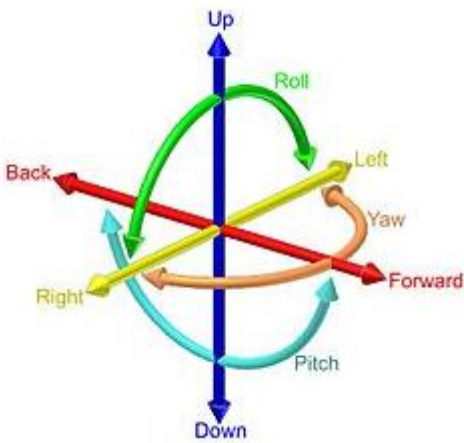


Figure 3 Six degree of freedom of quadrotor

The speed and direction of each motor should be controlled to construct different movements of quadrotor as described in Fig. 4:

**Rolling movement:**

- Change relative speed of the right and left motors

**Pitching movement**

- Change relative speed of the front and back motors.

**Yaw movement**

- Change speed of clockwise rotating pair and counter clockwise rotating pair.

**Vertical movement**

- Increasing or decreasing the speeds of all four rotors simultaneously by controlling the collective thrust [9].

Quadrotor is not easy to be controlled because of its inherent nonlinearity, that's why the designed controller should be good and robust to be able to make the quadrotor completely stable to achieve different missions without any mistakes.

Quadrotor becomes very important recently, it is applied and tested for saving lives, money and effort beside it might be pre provide information to prevent disasters that can probably hopped by observing the environments changing [9,10].

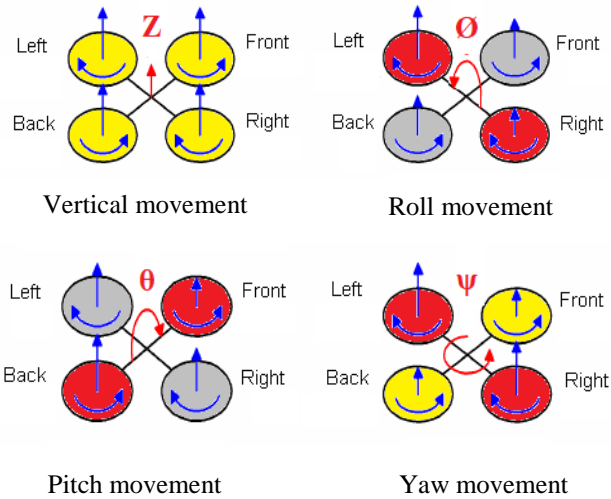


Figure 4 Quadrotor movements

II. QUADROTOR KINEMATIC AND DYNAMIC MODEL

• Kinematic Model

Quadrotor modeling starts with define the reference frames; these frames describe the motion of the quadrotor, earth frame, body frame, kinematic, aerodynamics and moments.

Earth frame define the E, N and D (east, north and downward) while the body frame defines the X, Y and Z which represent by the directions of motors 1,2 and ground. [7,10,11].

The distance between the Earth frame and the body frame describes the absolute position of the center of mass of the quadrotor  $r$  ( $r = [x \ y \ z]^T$ ) [5] as shown in Fig. 5.

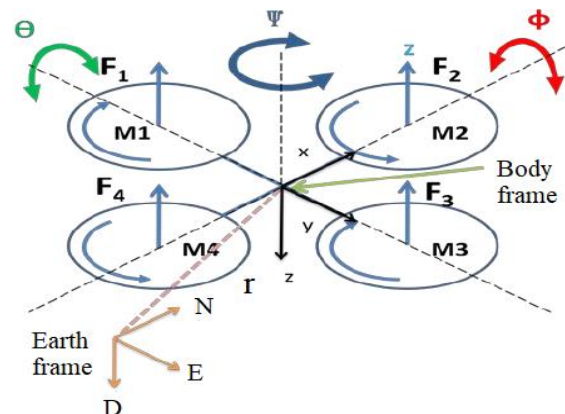


Figure 5 Reference frames

The rotation from the body frame to earth frame refers to the orientation of the quadrotor. The orientation of the quadrotor describes by the roll, pitch and yaw angles ( $\Theta, \emptyset$

and  $\psi$ ) which represent the rotation about X, Y and Z axes, respectively.

Consider that, the order of rotation to be roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) that leads to derive the rotation matrix R based on that angles order .

$$R = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\psi & S\phi S\theta S\psi + C\theta C\psi & C\phi S\theta S\psi - S\theta C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix}$$

where C and S refer to the sine and cosine function respectively .

The rotation matrix R is used to describe the dynamic model of the quadrotor. Some states need rotation about axes so that the rotation about ( X,Y and Z) should be found

- Rotation about X-axis

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

- Rotation about Y-axis

$$R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

- Rotation about Z -axis

$$R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To get the total conversion from the body frame to navigation, the above three matrices should be multiplied [12].

- **Dynamic model**

Quadrotor is characterized under actuated and coupled dynamics because it has six-degree of freedom: three-degree for rotational and three-degree for transitional. The three-degree of rotational are (roll, pitch and yaw) while the other three-degree are transitional (X, Y and Z). Quadrotor dynamic equation of motion can be derived by using Newton-Euler method

$$J \dot{\omega} + \omega \times J\omega + M_G = M_B \tag{1}$$

where J is quadrotor inertia matrix,

$\omega$  is angular body rate,

$M_G$  is Gyroscopic moments,

$M_B$  is moments acting on the quadrotor in the body frame.

$$M_G = \omega \times [0 \ 0 \ J_r \Omega_r]^T \tag{2}$$

Where  $J_r$  is Rotator's inertia ,

$\Omega_r$  is rotor's relative speed.

Equation (1) can be rewritten by using eq. (2) as shown in eq. (3):

$$J \dot{\omega} + \omega \times J\omega + \omega \times [0 \ 0 \ J_r \Omega_r]^T = M_B \tag{3}$$

$$\Omega_r = \Omega_1 + \Omega_2 + \Omega_3 + \Omega_4 \tag{4}$$

where  $\Omega_i$  is  $i^{th}$  angular velocity for  $i^{th}$  rotors as shown in Fig. 6.

The inertia matrix for the quadrotor is a diagonal matrix, the off diagonal elements, which are the product of inertia, are zero due to the symmetry of the quadrotor [7] as shown in eq. (5).

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{5}$$

where  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are the moments of inertia about the body frame axes X ,Y and Z respectively.

$$M_i = K_M \Omega_i^2 \tag{6}$$

$$F_i = K_f \Omega_i^2 \tag{7}$$

Where  $i$  refers to the number of motors ( $i = \{1,2,3,4\}$ ),

$M_i$  is  $i^{th}$  aerodynamic moment,

$F_i$  is  $i^{th}$  aerodynamic force,

$K_f$  and  $K_M$  are aerodynamic constants.

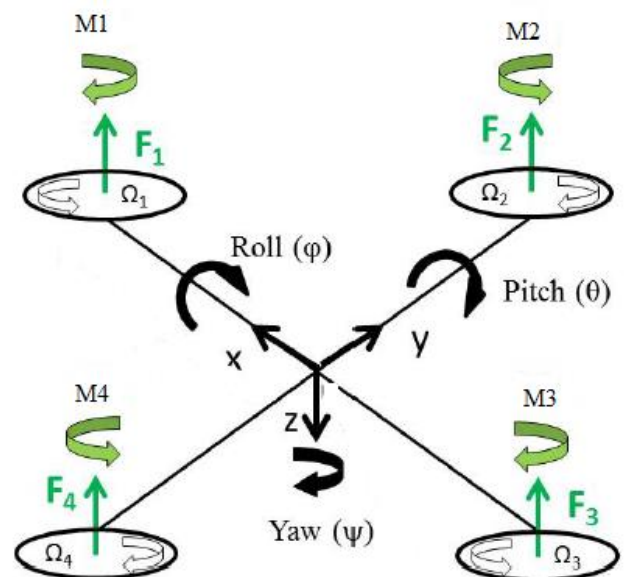


Figure 6 Moments, forces and velocity of rotors

Total moments about X-axes is given by:

$$M_x = -F_2l + F_4l = lK_f (-\Omega_2^2 + \Omega_4^2) \quad (8)$$

Total moments about Y-axes is given by:

$$M_y = F_1l + F_3l = lK_f (\Omega_1^2 - \Omega_3^2) \quad (9)$$

Total moments about Z-axes is given by:

$$M_z = M_1 - M_2 + M_3 - M_4 = K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (10)$$

The overall moments acting on the quadrotor in matrix  $M_B$  form can be found by combining ( 8,9, and 10) [6,7,8].

$$M_B = \begin{bmatrix} lK_f (-\Omega_2^2 + \Omega_4^2) \\ lK_f (\Omega_1^2 + \Omega_3^2) \\ K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (11)$$

The translation equation is found by using newton second Law:

$$m\ddot{r} = [0 \ 0 \ mg]^T + RF_B \quad (12)$$

where  $m$  is quadrotor mass,  
 $g$  is gravitational acceleration.

$F_B$  is nongravitational forces is given by

$$F_B = [0 \ 0 \ -K_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)] \quad (13)$$

### III. STATE SPACE MODEL

The quadrotor physical system with different inputs and outputs has been represented in a state-space form. State space representation is a mathematical model of a physical system as a set of inputs, outputs and state variables related by first order differential equations. The reason behind using state space is that, it will be easy to build a simulation model for the quadrotor [13,14].

The state vector forces

$$X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}] \quad (14)$$

where X is position along x-axes,  
Y is position along y-axes,  
Z is position along z-axes (height),  
 $\dot{x}$  is velocity along x-axes,  
 $\dot{y}$  is velocity along y-axes,  
 $\dot{z}$  is velocity along z-axes,  
 $\theta$  is pitch angle,  
 $\phi$  is roll angle,  
 $\psi$  is Yaw angle,  
 $\dot{\psi}$  is yaw rate,  
 $\dot{\theta}$  is pitch rate,  
 $\dot{\phi}$  is roll rate.

The control inputs vector law is

$$U = [U_1 \ U_2 \ U_3 \ U_4] \quad (15)$$

Where

$$U_1 = K_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (16)$$

$$U_2 = K_f (-\Omega_2^2 + \Omega_4^2) \quad (17)$$

$$U_3 = K_f (\Omega_1^2 - \Omega_3^2) \quad (18)$$

$$U_4 = K_m (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (19)$$

The four equations above can be expressed in matrix form

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_m & -K_m & K_m & -K_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (20)$$

$U_1$  : upward force of the four motors

$U_2$  : thrust difference between the 2 and 4 rotors which cause the roll rotation.

$U_3$  : thrust difference between the 1 and 3 rotors which cause pitch rotation.

$U_4$  : torque difference between the clockwise and anticlockwise motors.

The torque difference generates the yaw rotation [10,12]. It is important to choose the control law to get the desired rotation angle for the quadrotor (roll, pitch and yaw) and heading also.

The rotor velocity can be calculated from the knowledge of the control law by:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (21)$$

Taking square root: -

$$\Omega_1 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_3 + \frac{1}{4K_m} U_4} \quad (22)$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_2 - \frac{1}{4K_m} U_4} \quad (23)$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_3 + \frac{1}{4K_m} U_4} \quad (24)$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_2 - \frac{1}{4K_m} U_4} \quad (25)$$

To get the moment acting  $M_B$  in terms of  $U$  , substituting eqs. 21 in eq.11 :

$$M_B = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (26)$$

Now, the eq. 26 is substituted in eq. 3 with exchange each term in the eq 3 with prior definition to get the relation:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (27)$$

Eq. (3) is simplified to be as:

$$\begin{bmatrix} I_{xx} \ddot{\phi} \\ I_{yy} \ddot{\theta} \\ I_{zz} \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} I_{zz} \dot{\psi} - \dot{\psi} I_{yy} \dot{\theta} \\ \dot{\psi} I_{xx} \dot{\phi} - \dot{\phi} I_{zz} \dot{\psi} \\ \dot{\phi} I_{yy} \dot{\theta} - \dot{\theta} I_{xx} \dot{\phi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} J_r \Omega_r \\ -\dot{\phi} J_r \Omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (28)$$

Rewriting the eq. 28 to get (  $\ddot{\phi}$  ,  $\ddot{\theta}$  and  $\ddot{\psi}$  )

$$\ddot{\phi} = \frac{l}{I_{xx}} U_2 - \frac{J_r}{I_{xx}} \dot{\theta} \Omega_r + \frac{I_{yy}}{I_{xx}} \dot{\psi} \dot{\theta} - \frac{I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \quad (29)$$

$$\ddot{\theta} = \frac{l}{I_{yy}} U_3 - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_r + \frac{I_{zz}}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi} \quad (30)$$

$$\ddot{\psi} = \frac{1}{I_{zz}} U_4 + \frac{I_{xx}}{I_{zz}} \dot{\theta} \dot{\phi} - \frac{I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \quad (31)$$

With too many symbols is hard to dealing with the equations above , that's why we need to simplify them by assuming these assumptions

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} \quad a_2 = \frac{J_r}{I_{xx}}$$

$$a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad a_4 = \frac{J_r}{I_{yy}}$$

$$a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}} \quad b_1 = \frac{l}{I_{xx}}$$

$$b_2 = \frac{l}{I_{yy}} \quad b_3 = \frac{l}{I_{zz}}$$

Using these assumptions in the eqs. (29) ,(30) and (31) to get

$$\ddot{\phi} = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6 \quad (32)$$

$$\ddot{\theta} = b_2 U_3 + a_4 x_2 \Omega_r + a_3 x_2 \quad (33)$$

$$\ddot{\psi} = b_3 U_4 + a_5 x_2 x_4 \quad (34)$$

Where

$$\begin{aligned} \dot{x}_1 &= \dot{\phi} = x_2 \\ \dot{x}_2 &= \ddot{\phi} = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \\ \dot{x}_3 &= \dot{\theta} = x_4 \\ \dot{x}_4 &= \ddot{\theta} = x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \\ \dot{x}_5 &= \dot{\psi} = x_6 \end{aligned}$$

$$\begin{aligned} \dot{x}_6 &= \dot{\psi} = x_2 x_4 a_5 + b_3 U_4 \\ \dot{x}_7 &= \dot{z} = x_8 \\ \dot{x}_8 &= \ddot{z} = g - \frac{U_1}{m} (\cos x_1 \cos x_3) \\ \dot{x}_9 &= \dot{x} = x_{10} \\ \dot{x}_{10} &= \ddot{x} = -\frac{U_1}{m} (\sin x_1 \sin x_5 + \cos x_3 \cos x_5) \\ \dot{x}_{11} &= \dot{y} = x_{11} \\ \dot{x}_{12} &= \ddot{y} = \frac{U_1}{m} (\sin x_1 \cos x_5 - \cos x_1 \sin x_3 \sin x_5) \end{aligned}$$

#### IV. PID CONTROL

PID controller is the most usable controller in industrial applications due to its simple structure, good performance and low cost, so that it is still used from past till now as effective industrial controller. PID controller needs to adjust its parameters (gains) so that the system will do its job with minimum error, these parameters are not easy to adjust specially when the system is inherently nonlinear and complicated. Many algorithms are suggested to find the PID gains (Kp,Ki and Kd) like Ziegler-Nichols, Tyreus – Luyben , and other methods [11,12].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (35)$$

Where

$u(t)$  is the controlled signal

$e(t)$  is tracking error.

$K_p, K_i$  and  $K_d$  is the PID coefficients [15]

Figure 7 shows the general structure of the PID controller

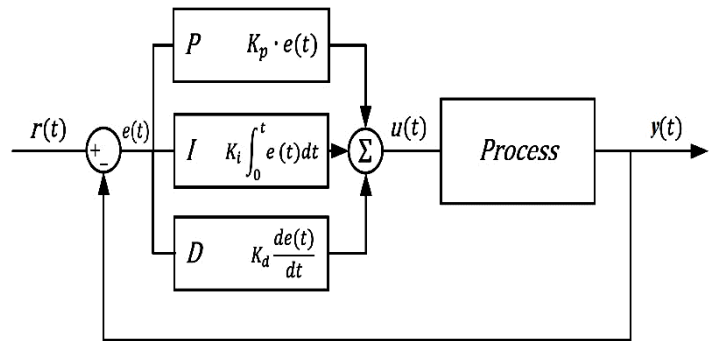


Figure7 General structure of PID controller

In this work, the proposed controllers have three layers. The first layer has three PID controllers for adjusting the position of UAV (x,y and z).The second layer also has three PID controllers for controlling the orientation of the quadrotor (  $\phi$  ,  $\theta$  and  $\psi$  ). Finally, The third layer has four PID controllers for generating the control signal for each motor to controlling the torque and speed of each one. Figure 8 shows the structure of proposed controllers' systems with quadrotor [16].

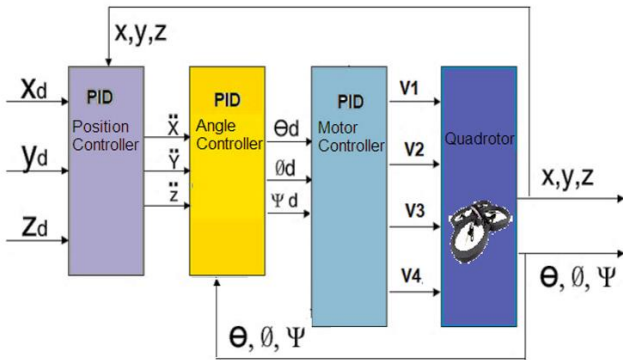


Figure8 PID controller for the quadrotor

The PID controllers for the position control loop can be given by:

$$\begin{cases} \ddot{x} = K_{px}(x_d - x) + K_{ix} \int (x_d - x)dt + k_{dx}(\dot{x}_d - \dot{x}) \\ \ddot{y} = K_{py}(y_d - y) + K_{iy} \int (y_d - y)dt + k_{dy}(\dot{y}_d - \dot{y}) \\ \ddot{z} = K_{pz}(z_d - z) + K_{iz} \int (z_d - z)dt + k_{dz}(\dot{z}_d - \dot{z}) \end{cases} \quad (36)$$

The PID controllers for the orientation loop can be given by:

$$\begin{cases} \ddot{\theta} = K_p(\theta_d - \theta) + K_i \int (\theta_d - \theta)dt + k_d(\dot{\theta}_d - \dot{\theta}) \\ \ddot{\phi} = K_p(\phi_d - \phi) + K_i \int (\phi_d - \phi)dt + k_d(\dot{\phi}_d - \dot{\phi}) \\ \ddot{\psi} = K_p(\psi_d - \psi) + K_i \int (\psi_d - \psi)dt + k_d(\dot{\psi}_d - \dot{\psi}) \end{cases} \quad (37)$$

The PID controller without optimal parameters is not an adaptable controller, which means that if any change happened in the system or the working condition, then the PID controller will not be able to control the system [17]. Therefore, the parameters of PID controllers need to be adjusted for the new working condition and this will need to stop the system and readjust the controller again, however, this is not likely happiness in some important systems which not need to stop because it is doing an important job. To solve this problem, one of intelligent method should be used to optimize the controller parameters by getting the optimal values. The following methods can be used to solve the mentioned problem [18]:

- Genetic algorithm optimization (GA)
- Particle swarm optimization (PSO)
- Neural Networks (NN)
- Ant colony (ACO)

In this work Particle Swarm Optimization (PSO) algorithm is suggested to optimize the parameters of PID controllers.

### V. PARTICLE SWARM OPTIMIZATION

The optimal values can be found in a large space of values in order to find these optimal values of the PID parameters through the huge search space it should be used an optimization technique. Many methods can be applied to find the best PID coefficient, one of these is the PSO.

The PSO technique is suggested by James Kennedy and Russell C. Eberhart in 1995, its inspiration from the social behavior of groups of birds and fish [19,20].

Many advantages of using the PSO like: -

- Simple to implement
- Parallel calculation
- Faster than other methods
- Fast convergence
- Easy to set the parameters
- Efficient for complex systems

The best advantage in this algorithm and different from the others algorithms is that the information can be sharing between particles. This sharing enables the particles to update the position and velocity to choose the optimal path or solution.

The PSO-based calculation is easy to implement and just requires the modification of a couple of parameters. It has been tested that this algorithm is more efficient than the other algorithms like neural networks or genetic algorithm [21,22]. Figure 9 shows the sharing information about the best solution in the swarm like birds, fishes ...etc.

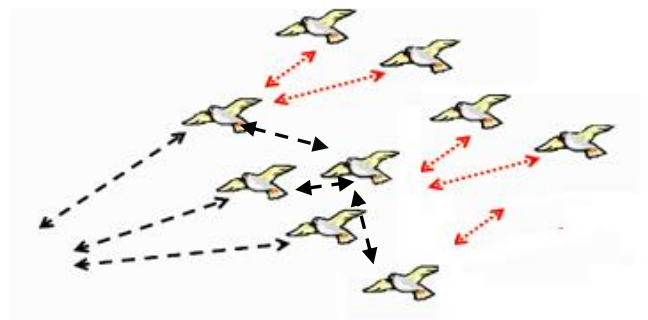


Figure 9 Sharing information in PSO

Another important property for the PSO can be applied on the continue and discrete or mixed systems with benefit of dealing with multiple local minima in the search space as shown in Fig. 10.

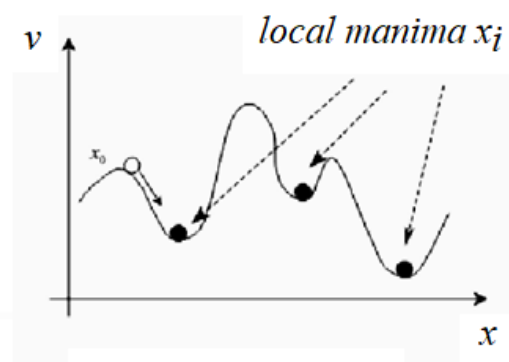


Figure 10 Multiple local minima

Each particle has multidimensional search space to take where the particle should adjust and updating its position depends on its own experience and the sharing information

with the neighbor particles [22,23,25]. The position of each particle is updated according to eq. 38.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \text{ with } x_i^0 \sim U(x_{min}, x_{max}) \quad (38)$$

Where

$v_i^t$  is the velocity of the  $i^{th}$  particle,

$x_i^t$  denote to the position of  $i^{th}$  particle,

$U(x_{min}, x_{max})$  it is uniform distribution for minimum and maximum respectively.

The strategy of the PSO is start with a big number of particles moving in multidimensional search space. The update equations of PSO parameters can be defined by the eqs:-

$$v_{iD}(t+1) = \omega \times v_{iD}(t) + l_1 \times rand_1(.) \times (p_{iD}(t) - x_{iD}(t)) + l_2 \times rand_2(.) \times (p_{gD}(t) - x_{iD}(t)) \quad (39)$$

$$x_{iD}(t+1) = x_{iD}(t) + v_{iD}(t+1) \quad (40)$$

Equation (39) is used for the update velocity while the equation (40) is used for updating the position of the particles [22,23]. In the PSO all particles are initialized randomly then its fitness functions are calculated to find the personal best (best value of each particle) after that the global best (best value of the particle in all the swarm) is found. Then the loop continues to find the optimum solution. [24,25]. When the loop start, the particle should always update its velocity by the personal and global best, and then every particle will update its position by the updated velocity, the loop will be ended when the specified value of error is reached, or the maximum number of iteration is reached [26,27]. The flow chart of the PSO algorithm is shown in Fig. 11.

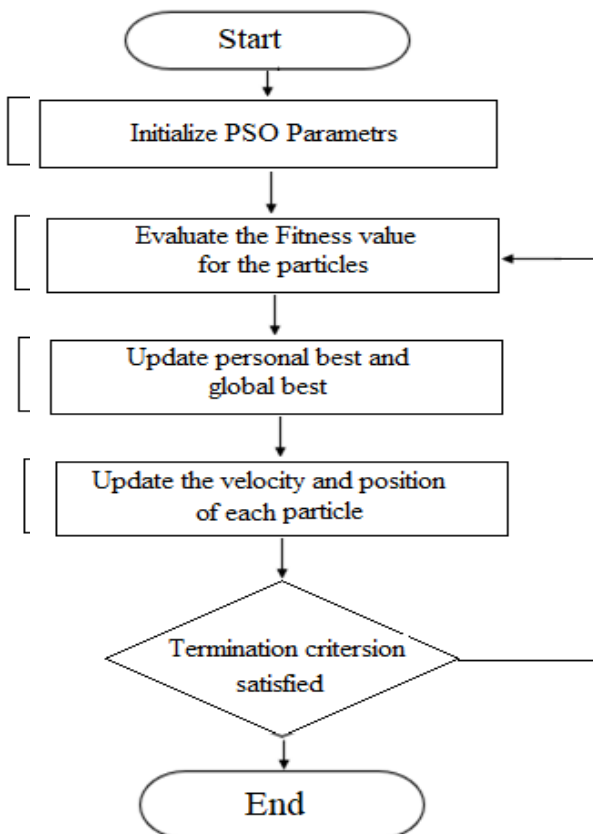


Figure 11 Flowchart of PSO

In the reference [28], the authors described the application of Particle Swarm Optimization to the problem of human body pose estimation from multiple view video sequences. The authors used a subdivision body model with an underlying skeleton layer to estimate and illustrate the body pose. The optimization looks for the best match between the silhouettes extracted from the original video sequence and the silhouettes generated by the projection of the model in a pose suggested by the PSO.

In the reference [29], the authors designed a new method for determining the optimal proportional-integral-derivative (PID) controller parameters of an UAV system based on the particle swarm optimization (PSO) algorithm. They demonstrated in detail how to employ the PSO method to search efficiently the optimal PID controller parameters of an AVR system. In reference [30], the authors presented a new designing technique of PID controller, which can effectively control the nonlinear optimization problems. The technique which has been introduced in this research is particle swarm optimization. It is population based stochastic optimization algorithm derived from human behavior and animal behavior as well.

Conventional gain tuning schemes such as the Ziegler Nichols method usually produces big overshoot therefore modern approach has been used in this paper to tune the parameters of PID controller. For designing of PID controller different performance indices have been used here for different plant transfer functions. In reference [30], the authors studied and designed PSO-PID controller, which is used in a high performance drilling system for controlling the obtained output and hence to minimize the integral of absolute errors (IAE). The main objective is to obtain a stable, robust and controlled system by tuning the PID controller using Particle Swarm Optimization algorithm. Authors in reference [31] studied Generic Particle Swarm Optimization Matlab Function on two examples to show the efficiency of using PSO.

Reference [32], the authors designed and optimized of a optimal controller for an autonomous quadrotor. Firstly, the dynamic model of the aerial vehicle is mathematically formulated. Then, an optimal backstepping controller (OBC) is proposed. Conventionally, control parameters of a backstepping controller (BC) are often chosen arbitrarily. To this end, it is necessary to invoke a well-established optimization algorithm in order to find the best parameters. Here, the particle swarm optimization is utilized as a new key idea to determine the optimal values of the BC parameters. In the reference [33], authors designed and applied the PSO for the UAV .The goal is to find optimal positions at each instant of each UAV to guarantee the best performance of a given task by minimizing a predefined objective function .

## VI. PSO-PID CONTROLLER

As mentioned earlier, PSO will be applied to tune the parameters of PID controller ( $K_p$ ,  $K_i$  and  $K_d$ ), to optimize the performance of the quadrotor. At the start, the swarm particles are initialized in the search space, each particle represents the possible solution for the PID gains, after that, each particle updates its velocity and then uses it to update its position and calculate its fitness function to choose the  $P_{best}$  and  $g_{best}$ , this

will continue until the specified value of error is reached or the maximum number of iterations is reached. At the end, the obtained values of PID controller are applied to the plant to give the optimum performance with minimum error.

To use the PSO, the number of swarms, the maximum number of iterations and the absolute error permitted should be defined. The well choosing of those parameters may make the PSO reaches the convergence faster, of course that will come by experience of the behavior of systems. Figure 12 shows the, main structure of PSO-PID controller.

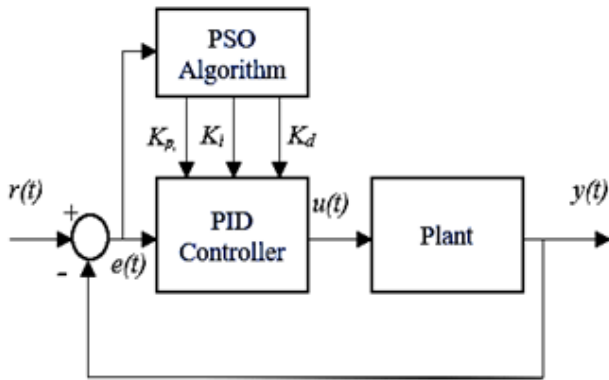


Figure 12 Main structure of POS-PID Controller

VII. SIMULATION RESULTS

1- PID controller results

The aim of this work is to propose a robust controller for the UAV to control its movement during the serious maneuvers. The UAV is a nonlinear inherently system and it has uncertainty parameters. Therefore, the design of robust controller is a serious challenge. In this work, the proposed controller consists of three stages (position controller, orientation controller and motors controller) to ensure that the proposed controller with the system is able to do its verity missions. First PID controller proposed and used to control all the three stages. The gains ( $K_p, K_i$  and  $K_d$ ) should be chosen to minimize the tracking error as low as possible to provide a good tracking for the quadrotor. The PID parameters are found by using the conventional Ziegler-Nichols. The simulation results are performed by using Matlab 2018b to show the comparison between the position of the UAV ( $x, y$  and  $z$ ) and the desired position ( $x_d, y_d$  and  $z_d$ ). The tracking error is measured to compare with the tracking error when the PSO-PID controller is used in the next part. Figures 13,14 and 15 show the position of ( $x, y$  and  $z$ ) and desired position ( $x_d, y_d$  and  $z_d$ ) respectively for the quadrotor using PID controllers.

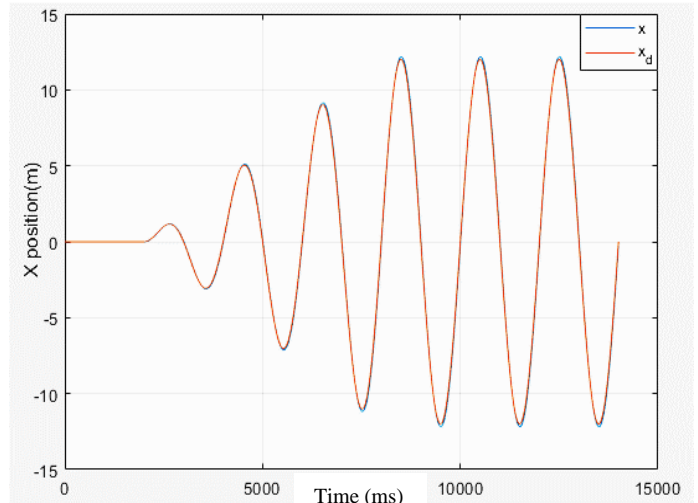


Figure 13 X position of the quadrotor

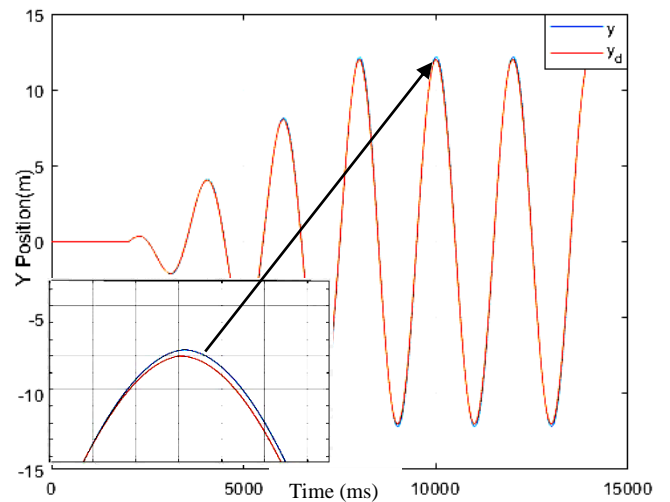


Figure 14 Y position of the quadrotor

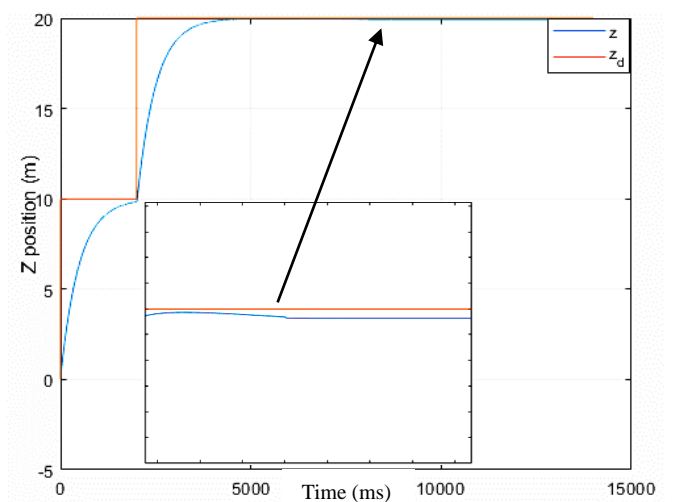


Figure 15 Z position of the quadrotor

Table (1) shows the value of the PID control parameters are used to control the quadrotor when conventional Ziegler-Nichols is used.



Table (1) PID parameters

PID Parameters	Position	Angular	Motor
<b>K<sub>p</sub></b>	3	10	1
<b>K<sub>i</sub></b>	10	20	0
<b>K<sub>d</sub></b>	0	0	10

2- PSO-PID controller results

Now, the simulation results of the PSO-PID controller will be illustrated to show the robustness of the proposed controller with and without applying sudden disturbance to the controlled system during maneuvers.

PID controller is easy and simple with good performance, but it should be choosing its parameters in a way of minimizing error as low as possible and this is the weakness spot of using PID controller. Many methods were studied and used to tune the PID gains like Ziglar-Nicolas. Unfortunately, these methods will not provide the perfect performance.

Optimization method has been designed and used for tuning the PID controllers. The PSO algorithm is an optimization method has advantages of using like fast convergence, saving time effort, sharing information between particles or swarm to reach the solution in shortest time, also it can be applied for continuous and discrete or hybrid systems. For those advantages of PSO algorithm, in this work, PSO algorithm is used for finding the parameters of PID controller

In the first scenario, the performances of the proposed controller are studied when no disturbances are applied on the quadrotor.

Figures 16,17and 18 show the position ( x,y and z ) and desired position (x\_d, y\_d and z\_d ) when PSO-PID cntrollers are used .

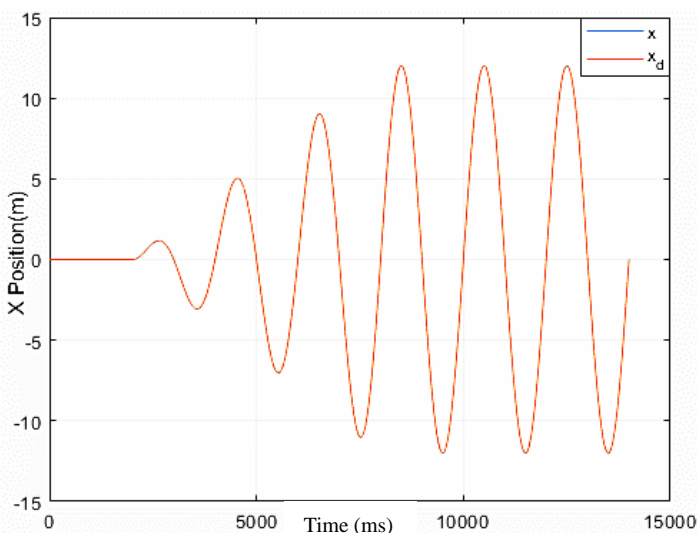


Figure 16 X position of the quadrotor with PSO

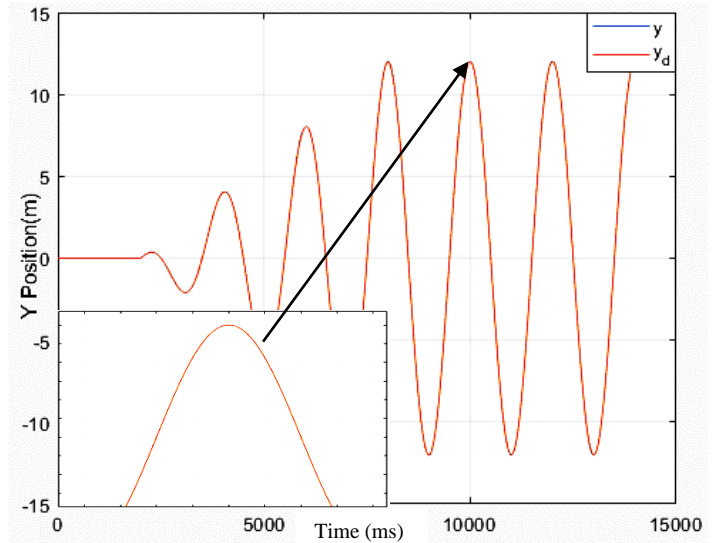


Figure 17 Y position of the quadrotor with PSO

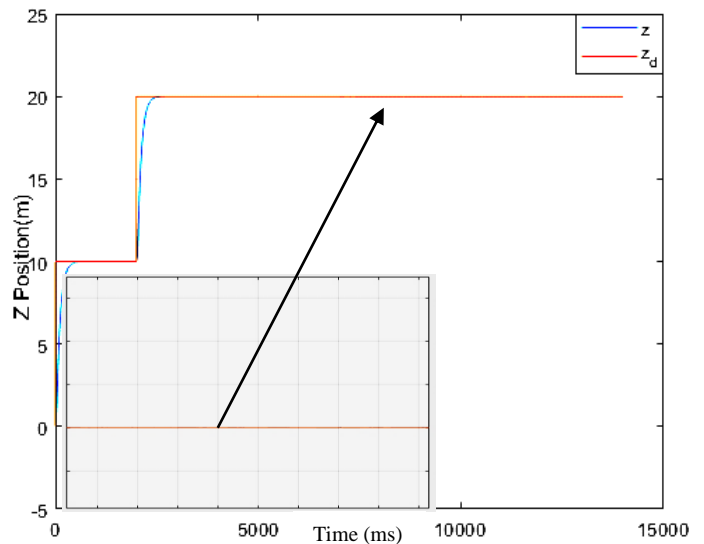


Figure 18 Z position of the quadrotor with PSO

Table (2) shows the PID parameters which are found by using PSO .

Table (2) PID parameters using PSO

PID Parameters	Position	Angular	Motor
<b>K<sub>p</sub></b>	17.032	10.0112	12.32
<b>K<sub>i</sub></b>	18	18.0102	0
<b>K<sub>d</sub></b>	0	0	2.212

3- PSO-PID controller with disturbance

Sometimes the conditions or working environments could be changed like heat, weight, voltage or current so the controller should be still able to do its job by keeping the controlled system stable during sudden disturbances with minimum error. The PSO-PID will be test under the existence of disturbance. To study the effect of disturbances on the performance of the proposed controller, the mass of the quadrotor is increased by 50% of its weight.

Simulation results under disturbance shows that the PSO-PID controller still robust and able to keep the quadrotor stable.

Figures 19, 20 and 21 show the position control ( x,y and z) and desired position ( x\_d, y\_d\_ and z\_d) of the quadrotor under disturbance .

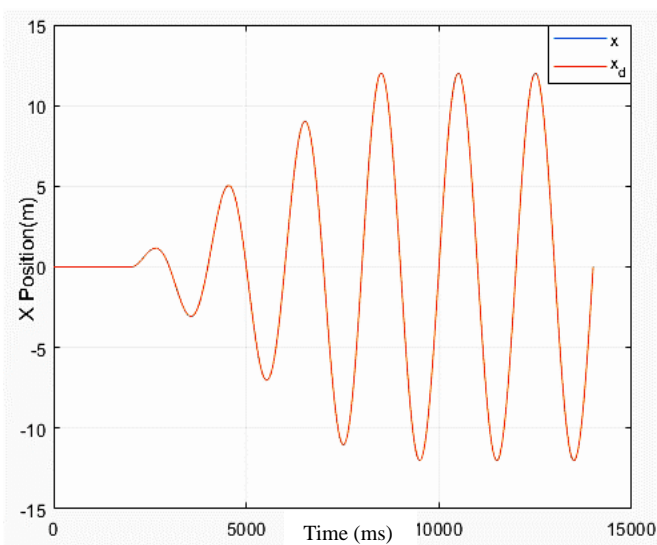


Figure 19 X position using PSO under

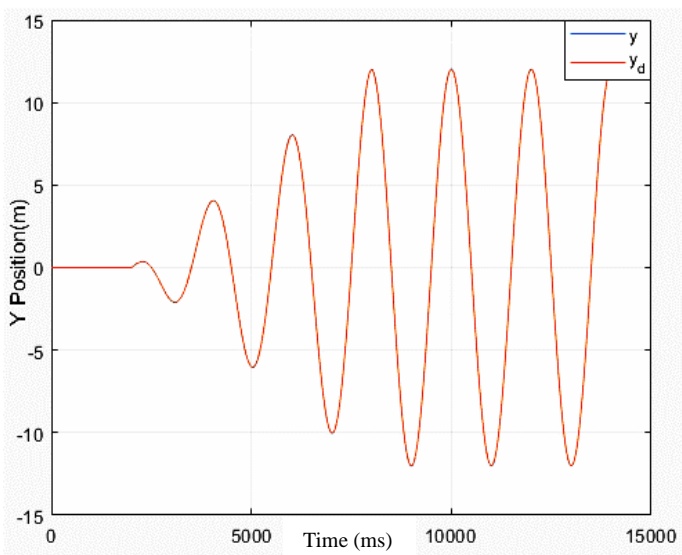


Figure 20 Y position using PSO under disturbance

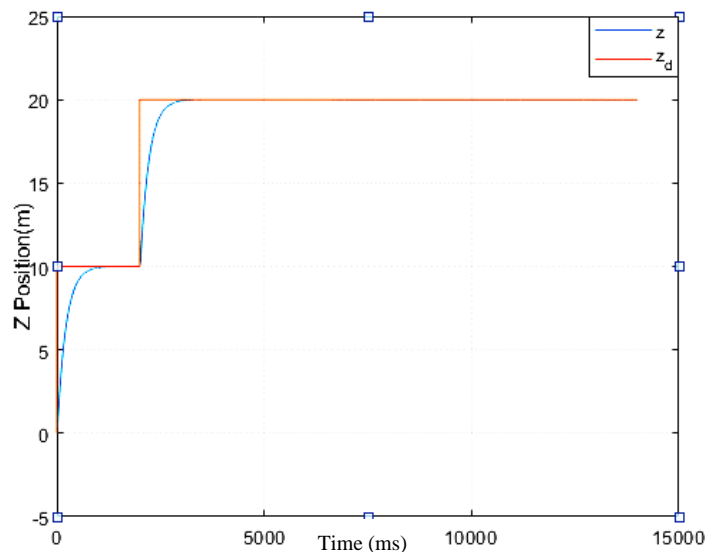


Figure 21 Z position using PSO under disturbance

Table (3) shows the Different controllers applied to the system with tracking position error.

Controller	Error of x position	Error of y position	Error of z position
<b>PID</b>	<b>0.2036</b>	<b>0.1903</b>	<b>0.0492</b>
<b>PSO-PID</b>	<b>0.00117</b>	<b>0.001491</b>	<b>0.001033</b>
<b>PSO-PID with disturbance</b>	<b>0.008772</b>	<b>0.005385</b>	<b>0.008666</b>

Table (3) Controllers with tracking error

VIII. CONCLUSION

Quadrotor is a nonlinear system with six-degree of freedom it needs a powerful, robust and adaptive controller to make the system stable under different flying condition. The simulation results for the different controllers with or without a disturbance show that the PSO-PID is good choice to that system. Without a doubt, by using PSO for tuning the PID's parameters, the error is minimized to a very small value, which means that the controller improves the system performance compare with the performance of PID controller. To check how that controller is robust, a disturbance included by increasing the quadrotor mass (50%) to see if the controller still be able to stabilize the quadrotor under that mass variation. The results show that, the system controlled by PSO-PID even with that disturbance still made the system stable with error much less than the classic PID and for sure using this kind of controllers will ensure the best performance of the system when working conditions are changed. PSO parameters play an important role to find the best solution or the optimized path during the running programing. Also, for fast convergence, if the designer has an idea about the range for the PID parameters it will be useful to use the lower and

upper bond to use in the PSO because it will reach the convergence faster. Not always increasing swarm size or number of iterations lead to optimal solution especially in complex systems because it will need a long time for processing and it might be not getting the right solution. In PSO-PID tuning, it used 30 max-iteration, 50 swarm size and  $1e-6$  for function tolerance.

## REFERENCES

- [1] S. Norouzi Ghazbi , Y. Aghli “ Quadrotors Unmanned Aerial Vehicles: A Review “ *International Journal on Smart Sensing and Intelligent Systems* Vol. 9, No. 1, March 2016.
- [2] B. Kada, Y. Ghazzawi “ Robust PID Controller Design for an UAV Flight Control System “ *Proceedings of the World Congress on Engineering and Computer Science 2011 Vol II WCECS 2011*, October 19-21, 2011, San Francisco, USA.
- [3] Roman Czyba1, Grzegorz Szafranski” Control Structure Impact on the Flying Performance of the Multi-Rotor VTOL Platform - Design, Analysis and Experimental Validation “ *International Journal of Advanced Robotic Systems* .2013
- [4] Hossein Bolandi1, Mohammad Rezaei “ Attitude Control of a Quadrotor with Optimized PID Controller “ *Intelligent Control and Automation*, 2013, 4, 335-342 .
- [5] Riaan Stopforth, Glen Bright “ Quad-Rotor Unmanned Aerial Vehicle Helicopter Modelling & Control “ *School of Mechanical Engineering, University of KwaZulu Natal* , Received 11 February 2011; Accepted 25 Aug 2011.
- [6] Shahida Khatoon, Mohammad Shahid “ Quadrotor Control Using PID Controller “ *Emerging Trends in Electrical And Electronics Engineering (ETEEE-2015)* Jamia Millia Islamia, New Delhi .
- [7] Mohammed Jasim “ Modelling, Control and Implementation of Quadrotor “ *A Thesis Submitted To The College Of Engineering University Of Basrah* April, 2015.
- [8] M. Eatemadi “ Mathematical Dynamics, Kinematics Modeling and PID Equation Controller of QuadCopter “ *International Journal of Applied Operational Research* Vol. 7, No. 1, pp. 77-85, Winter 2017
- [9] Oguz Kose, Tugrul Oktay1 “ Dynamic Modeling and Simulation of Quadrotor for Different Flight Conditions “ *European Journal of Science and Technology* No. 15, pp. 132-142, March 2019.
- [10] Heba talla Mohamed “ Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches “ *School of Sciences and Engineering* , Spring 2014.
- [11] Teppo Luukkonen “ Modelling and control of quadcopter “*School of Science Mat-2.4108 Independent research project in applied mathematics Espoo*, August 22, 2011 .
- [12] Amr Nagaty·Sajad Saedi·Carl Thibault “ Control and Navigation Framework for Quadrotor Helicopters “ *J Intell Robot Syst*”.
- [13] Nizar Hadi Abbas , Ahmed Ramz Sami “Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm – Particle Swarm Optimization “*International Journal of Computer Applications* (0975 – 8887) Volume 172 – No.4, August 2017.
- [14] Jacob M. Goodman\*a, Jinho Kima “ System and Mathematical Modeling of Quadrotor Dynamics “ a *University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA 21250*; b *U.S. Army Research Lab, Aberdeen, MD, USA 21001*.
- [15] Tommaso Bresciani “ Modelling, Identification and Control of a Quadrotor Helicopter “ *Department of Automatic Control Lund University* October 2008.
- [16] FRANCESCO SABATINO “ Quadrotor control: modeling, nonlinear control design, and simulation “*Master’s Degree Project Stockholm, Sweden* June 2015
- [17] Aminurrashid Noordin#1, Mohd Ariffanan Mohd Basri “ Modelling and PSO Fine-tuned PID Control of Quadrotor UAV “ *international journal Advanced science engineering information technology* , Vol.7 (2017) No. 4 ISSN: 2088-5334 .
- [18] Yong Taoa, Guang Xieb, Youdong Chena “ A PID and fuzzy logic based method for Quadrotor aircraft control motion “ *Journal of Intelligent & Fuzzy Systems* 31 (2016) 2975–2983 DOI:10.3233/JIFS-169182 IOS Press .
- [19] Iman Salih KareemAL-Ghabban “ Automatic Tuning of (PID) Controller Using Particle Swarm Optimization (PSO) Algorithm for Steam Engine Speed Control (SESC “ *Journal of Engineering and Development*, Vol. 19, No.1, January 2015, ISSN 1813- 7822 .
- [20] Sarah Pontes Madruga\*, Gabriel Fernando Basso “ Apso-Basedtuningalgorithmforquadcoptercontroller” *Universidade Federal da Paraíba Rua dos Escoteiros, s/n, Mangabeira João Pessoa, Paraíba, Brasil* .
- [21] Andries P. Engelbrecht “*Computational Intelligence An Introduction Second Edition* “ *John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England* 2007 .
- [22] Huu-Khoa Tran 1,2 and Juing-Shian Chiou “ PSO-Based Algorithm Applied to Quadcopter Micro Air Vehicle Controller Design “ *Micromachines* 2016, 7, 168.
- [23] Xingyang Lu1, Xiangyin Zhang, Songmin Jia1, Jichao Shan “ Design of Quadrotor Hovering Controller Based on Improved Particle Swarm Optimization “ *2017 10th International Symposium on Computational Intelligence and Design* .

[24] Mohammed Abdallah Khodja<sup>1,2,\*</sup>, Mohamed Tadjine<sup>1</sup>, Mohamed Seghir Boucherit<sup>1</sup>, and Moussa Benzaoui<sup>2</sup> “ Tuning PID attitude stabilization of a quadrotor using particle swarm optimization (experimental) “ *Int. J. Simul. Multisci. Des. Optim.* 2017, 8, A8 M. Khodja et al., Published by EDP Sciences, 2017 .

[25] Yul Y. Nazaruddin, Angela Dian Andrini and Bobby Anditio “ PSO Based PID Controller for Quadrotor with Virtual Sensor “ *Preprints of the 3rd IFAC Conference on Advances in ProportionalIntegral-Derivative Control*, Ghent, Belgium, May 9-11, 2018 .

[26] Thi Thoa Mac<sup>1,2</sup>, Cosmin Copot<sup>1,3</sup>, Trung Tran Duc<sup>2</sup> “ AR.Drone UAV control parameters tuning based on particle swarm optimization algorithm “ 1 Department of Dynamical Systems and Control (DySC), Ghent University, Belgium.

[27] Satyobroto Talukder “ Mathematical Modelling and Applications of Particle Swarm Optimization “ Master’s Thesis Mathematical Modelling and Simulation Thesis no: 2011.

[28] Spela Ivekovic and Emanuele Trucco “ Human Body Pose Estimation with PSO “ 2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada .

[29] Zwe-Lee Gaing “ A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System “*IEEE TRANSACTIONS ON ENERGY CONVERSION*, VOL. 19, NO. 2, JUNE 2004 .

[30] S.M.GirirajKumar , Deepak Jayaraj “ PSO based Tuning of a PID Controller for a High Performance Drilling Machine 2010 *International Journal of Computer Applications* (0975 - 8887) Volume 1 – No. 19 .

[31] Soren Ebbesen, Pascal Kiwitez and Lino Guzzella “ A Generic Particle Swarm Optimization Matlab Function “ 2012 *American Control Conference Fairmont Queen Elizabeth*.

[32] Mohd Ariffanan Mohd Basri Kumeresan A. Danapalasingam Abdul Rashid Husain “ Design And Optimization Of Backstepping Controller For An Underactuated Autonomous Quadrotor Unmanned Aerial Vehicle “ *Transactions Of Famena Xxxviii-3* (2014) .

[33] A. Belkadi, L. Ciarletta, and D. Theillio “ Particle swarm optimization method for the control of a fleet of Unmanned Aerial Vehicles “ 12th *European Workshop on Advanced Control and Diagnosis (ACD 2015)* .