JACET

J. ADV COMP ENG TECHNOL, 5(1) Winter 2019

Quadrotor Control Using Fractional-Order $PI^{\lambda}D^{\mu}$ Control

Reham H. Mohammed
Electrical Engineering Department, University of Suez Canal, Ismailia, Egypt.
(riry4mody@yahoo.com)

Received (2018-10-20) Accepted (2018-12-23)

Abstract: *Quadrotor control has been noted for its trouble as the consequence of the high maneuverability, system nonlinearity and strongly coupled multivariable. This paper deals with the simulation depend on proposed controller of a quadrotor that can overcome this trouble. The mathematical model of quadrotor is determined using a Newton-Euler formulation. Fractional Order Proportional Integral Derivative (FOPID) controller tuned by genetic algorithm (GA) is investigated to control and stabilization the position and attitude of quadrotor using feedback linearization. This controller is used as a reference to compare its results with Proportional Integral Derivative (PID) controller tuned by GA. The control structure performance is evaluated through the response and minimizing the error of the position and attitude. Simulation results, demonstrates that position and attitude control using FOPID has fast response, better steady state error and RMS error than PID. By simulation the two controllers are tested under the same conditions using SIMULINK under MATLAB2015a.*

Keywords: *Quadrotor, Proportional Integral Derivative (PID) controller, Fractional Order Proportional Integral Derivative (FOPID).*

How to cite this article:
Reham H. Mohammed. Quadrotor Control Using Fractional-Order $PI^{\lambda}D^{\mu}$ Control. *J. ADV COMP ENG TECHNOL, 5(1) Winter2019 : 1-10*

I. INTRODUCTION

Despite the advancement in the control principle area, PID-type controllers are still the majority regularly used control algorithm in industry. That is because of its design and implementation simplicity [1]-[2]. There are four Shortcomings in classical PID control modelling process which are: oversimplification, error computation, and noise degradation in the derivative control and performance loss in the form of a linear weighted sum in the control law, also difficulties that can be done by the integral control [3]. A FOPID controller is a generalization of the PID controllers has suggested by Podlubny in order to upgrade performance and robustness of PID control systems [4].

Fractional calculus is an area of mathematics that deals with derivatives and integrals using non-integer orders. Fractional order derivatives and integrals have been used in industrial applications and different fields. In FOPID controller modelling process, the 5 parameters (k_p, k_d, μ, k_i and λ) require to be chosen depend upon some design specifications, In this way there is a require to an effective global methodology to optimize these parameters naturally. GA is one of evolutionary optimization strategies used to optimize the 5 parameters of the FOPID controller [5].

This work is licensed under the Creative Commons Attribution 4.0 International Licence.

To view a copy o this licence, visit https://creativecommons.org/licenses/by/4.0/

Quadrotors are motivating platform for Aerial Robotics research. The thriving interest of aerial robots in military, farming, mining, firefighting and remote sensing and so on has given great impetus to controller research and improvement in this field [6]. The research in controller design of quadrotor is as yet having troubles because of high maneuverability, system nonlinearity, and strongly coupled multivariable and under-actuated condition [7]. There are a several literature reviews of quadrotor control for upgraded performance such as classical linear methodologies for example, PID [8] [9] [10] [11], Linear quadratic regulator [10][11] [12] for perfect control, which at lower speeds give good results, but this strategies gave a poor performance at higher speeds as a result of large vibrations of motor controller. In Additionally, various advanced control approaches are likewise utilized, for example, nonlinear feedback linearization [13] [14], H-infinity control design, adaptive approach [15], sliding mode control [16] but noticed amount of chattering, Backstepping [16] [11] [13][17]. Most works have utilized Euler angles for modelling. Additionally it considers the dynamic models of rotors, gears and motors. But most of literature reviews didn't give the acceptable results compared to the required position and attitude where, the target of all controllers' techniques is to stabilize attitude and position of quadrotor with better response.

The main target of this paper is to present FOPID to position and attitude control and stabilization of quadrotor and compared their results with PID tuned by GA. The controller output is straightforwardly fed into the dynamic model without making any mapping in the actuator space. In the simulations presented here, the thrust input cannot be more than double the weight of the matrix; similarly a suitable threshold is additionally put in the torque input. These thresholds have been put to make the control laws as practical as possible.

The organization of this paper: The quadrotor configuration introduced in Section 2. Section 3 introduces the quadrotor modelling. Control strategy is introduced in Sections 4. Simulation results for both developed controllers (PID and FOPID tuned using GA) are illustrated in Section 5, followed by the concluding remarks in Section 6.

II. QUADROTOR CONFIGURATION

Quadrotor is an Unmanned Aerial Vehicle (UAV) with four rotors. As presented in Figure 1, the nearby rotors have inverse sense of rotation. This is done to adapt the total angular momentum of the craft; otherwise the UAV will begin rotating around itself. The Quadrotor has 6 DOF but only four actuators (Rotors). Hence, Quadrotors are under actuated. The Rotors produce thrust, torque and drag force and the control input to the system is the angular velocity of the motors. A low level controller balances out the rotational speed of each blade. The Quadrotor can perform Vertical Take Off and Landing (VTOL), hover and make slow precise movements. The four rotors give a higher payload capacity. Quadrotors are moderately less difficult because they don't bring convoluted swash plates and linkages [18].

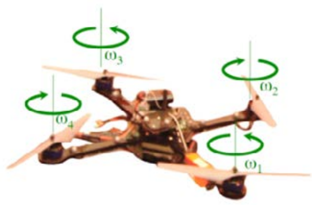

Fig. 1. Quadrotor UAV

There are some states that we require in UAV recorded as: Estimation State, calculate position and velocity of quadrotor. Control, drive motors and delivers desired actions in order to navigate to the desired state. Mapping, the quadrotor must have basic ability to map its environment. Planning: Finally, the quadrotor must be able to track the trajectory planning [18].

III. QUADROTOR MODELING

1. Basic Mechanics

The mechanics of the quadrotor is investigated in this section. The model has been gotten from [18]. The forces following up on the framework are the thrusts F_i from each of the rotors and the force of gravity $-mga_3$. The moments following up on the framework are the moments due to each of the thrust and the drag moment *Mi* which is created because of the propeller rotation [18].

As appeared in Fig. 2, the Thrust Fi, Drag Moment Mi and Motor Torque Speed Characteristics τ vs. ω. [18].

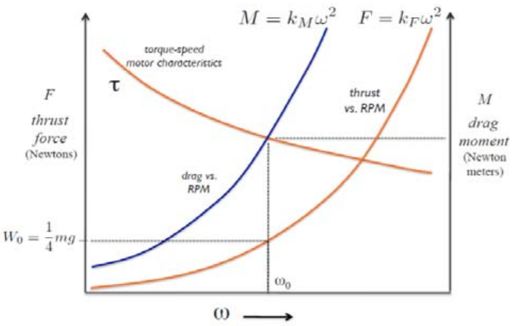

Fig. 2. Thrust Fi, Drag Moment Mi and Motor Torque Speed Chacteristics τ vs ω.

- Speed of the Motor at Hover Configuration:

$$K_f\,\omega_0^2 = \frac{mg}{4} \qquad (1)$$

Motor Torques and Drag Torque (They have same magnitude however inverse signs):

$$M_i = \tau_i = K_M\omega_i^2 \qquad (2)$$

Thrust

$$F_i = K_F\,\omega_i^2 \qquad (3)$$

Resultant Force:

$$F = F_1 + F_2 + F_3 + F_4 - mga_3 \qquad (4)$$

Resultant Moment:

$$M = r_1 * F_1 + r_2 * F_2 + r_3 * F_3 + r_4 * F_4 + M_1 + M_2 + M_3 + M_4 \qquad (5)$$

2. Quadrotor Dynamics

The dynamics of a quadrotor by using the Newton-Euler formalism presentenced in this section. The inspiration is gotten from Mellinger work [19].

- Newton Euler equation

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} mI_3 & O_3 \\ O_3 & I_3 \end{bmatrix}\begin{bmatrix} a \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ \omega * I_3\omega \end{bmatrix} \qquad (6)$$

where, τ is the net torque, F is the net force acting on the quadrotor, a is the linear acceleration of the center of mass, I3 is a 3 × 3 identity matrix called the moment of inertia, ω is quadrotor velocity angle, v is the linear velocity, m is the mass and α is the acceleration angle.

Rotation Matrix:

$$R_B^W = \begin{bmatrix} C\psi C\theta - S\varphi S\psi S\theta & -C\varphi S\psi & C\psi S\theta + C\theta S\varphi S\psi \\ C\theta S\psi + S\varphi C\psi S\theta & C\varphi C\psi & S\psi S\theta - C\theta S\varphi C\psi \\ -C\varphi S\theta & S\varphi & C\varphi C\theta \end{bmatrix} \qquad (7)$$

φ = Roll, θ = Pitch and ψ=Yaw

The equation of Linear Motion is:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_B^W \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \qquad (8)$$

The equation of angular Motion is:

$$I\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}=\begin{bmatrix}L(F_2-F_4)\\ L(F_3-F_1)\\ M_2+M_4-M_1-M_3\end{bmatrix}-\begin{bmatrix}p\\ q\\ r\end{bmatrix}*I\begin{bmatrix}p\\ q\\ r\end{bmatrix} \quad (9)$$

Where, I = Moment of inertia, r = [x y z]^T, [p q r] the body angular Velocities, m = the system Mass. The relationship between the rate of change of (φ, θ and ψ) and body angular velocities is:

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix}=\begin{bmatrix}c_\theta & 0 & -c_\varphi s_\theta\\ 0 & 1 & s\varphi\\ s_\theta & 0 & c\varphi c\theta\end{bmatrix}\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \quad (10)$$

Thrust Input: $u1 = F1 + F2 + F3 + F4$

$$u_2=\begin{bmatrix}L(F_2-F_4)\\ L(F_3-F_1)\\ M_2+M_4-M_1-M_3\end{bmatrix} \quad (11)$$

IV. CONTROL STRATEGY

In this paper, controller techniques of quadrotor have been studied and implemented in MATLAB2015a/Simulink. The quadrotor block diagram Control using feedback linearization is appeared in Fig.3. As appeared in the block attitude controller is inner loop while position controller is the outer loop. It is sensible to perceive that the dynamics of the inner loop must be quicker than the dynamics of the outer loop. In hover arrangements the dynamics of attitude do not matter much in general, however in situations where the robot needs to make maneuvers, it is essential to have a quicker attitude controller.

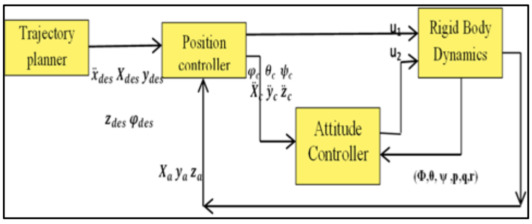

Fig. 3. Control Block Diagram.

In the following sections, PID and FOPID Controller are discussed and the results are presented to control the outer loop.

1. Quadrotor control using PID tuned using GA

The point of PID is to design a position and attitude controller of a quadrotor by choice of a PID parameters gains (kp, kd and ki) utilizing GA, where GA is an optimization method rely upon the mechanisms of regular selection[20].

This control law works well under hover conditions. Linearizing the dynamic model at the hover configuration, where the system model is reduced to:

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}(c\psi s\theta+c\theta s\varphi s\psi)u_1\\ (s\psi s\theta-c\theta s\varphi c\psi)u_1\\ -mg+c\varphi c\theta u_1\end{bmatrix} \quad (12)$$

$$I\begin{bmatrix}\dot{P}\\ \dot{q}\\ \dot{r}\end{bmatrix}=u_2 \quad (13)$$

The reference trajectory is:
$r_{ref} = [X_{des}\, y_{des}\, z_{des}\, \varphi_{des}]^T$

The commanded linear accelerations can be calculated as:

$$\ddot{x}_c = \ddot{x}_{des} + k_{dx}(\dot{x}_{des} - \dot{x}_a) + k_{px}(x_{des} - x_a) + k_{ix}\int(x_{des} - x_a) \quad (14)$$
$$\ddot{y}_c = \ddot{y}_{des} + k_{dy}(\dot{y}_{des} - \dot{y}_a) + k_{py}(y_{des} - y_a) + k_{iy}\int(y_{des} - y_a) \quad (15)$$
$$\ddot{z}_c = \ddot{z}_{des} + k_{dz}(\dot{z}_{des} - \dot{z}_a) + k_{pz}(z_{des} - z_a) + k_{iz}\int(z_{des} - z_a) \quad (16)$$

The commanded roll, pitch and yaw are:

$$\varphi_c = \frac{1}{g}(\ddot{x}_c \, sin(\psi_{des}) - \ddot{y}_c \, cos(\psi_{des})) \quad (17)$$
$$\theta_c = \frac{1}{g}(\ddot{x}_c \, cos(\psi_{des}) + \ddot{y}_c \, sin(\psi_{des})) \quad (18)$$
$$\psi_c = \psi_{des} \quad (19)$$

Using equations 14 to 19

$$u_1 = m(g + \ddot{z}_c) \quad (20)$$

$$u_2 = I\begin{bmatrix} K_{P\varphi}(\varphi_C - \varphi_a) + K_{d\varphi}(p_c - p) \\ K_{P\theta}(\theta_C - \theta_a) + K_{d\theta}(q_c - q) \\ K_{P\psi}(\psi_C - \psi_a) + K_{d\psi}(r_c - r) \end{bmatrix} \quad (21)$$

Using equation 10 one can get $[pc; qc; rc]^T$.

GA applied for tuning PID gains k_p, k_d and k_i for the three position (x, y and z) utilizing Integral Square-Error (ISE) to guarantee ideal control performance at nominal operating conditions. The Three gains of PID after tuning for X (k_{p1}=45.75and k_{d1}=12, k_i=33.5), for Y (k_{p2}=51.5 , k_{d2}=56.599, k_i=24.5) and for Z (k_{p3}=130.962, k_{d3}=55.25, k_i=58.526) at that point alter this error signal to give control input for system. The control input then forces the system to deliver output as close as possible to the desire position.

The solution proposed in this work is to use FOPID controller.

2. Quadrotor control using FOPID

2.1. Principles of FOPID

Fractional Order Calculus (FOC) is a generalization of the conventional integration and differentiation that include non-integer orders. Fundamental operator representing the fractional-order differential and integration is presented in (22) where α is a real number [21].

$$_aD_t^a = \begin{cases} \frac{d^a}{dt^a} & ,a > 0 \\ 1 & ,a = 0 \\ \int_x^t (d\tau)^a & ,a < 0 \end{cases} \quad (22)$$

Linear operator D was translated as integrator when a is negative and differentiator when a is positive. Something else, D is a unity when a is zero. The most widely recognized form of a FOPID controller is the PI$^\lambda$D$^\mu$ controller. Including an integrator of order λ and a differentiator of order μ where λ and μ can be any real numbers. The transfer function of such a controller is appeared in equation 23:

$$G_c(s) = \frac{U(s)}{E(s)} = k_p + k_I\frac{1}{s^\lambda} + k_D s^\mu, (\lambda\,, \mu > 0) \quad (23)$$

Where *Gc(s)* is the transfer function of the controller, *E(s)* is the error, and *U(s)* is controller's output. The control signal *u(t)* can then be expressed in the time domain as:

$$u(t) = k_p e(t) + k_I D_t^{-\lambda} e(t) + k_D D_t^{\mu}\, e(t) \quad (24)$$

Fig. 3 is a block-diagram of FOPID. Where, choosing λ=1 and μ=1, a traditional PID controller can be recovered. The choosing of λ=1, μ=0, and λ=0, μ=1 respectively corresponds traditional PI & PD controllers. All these traditional cases of PID controllers are the special cases of the fractional PI$^\lambda$D$^\mu$ controller given by:

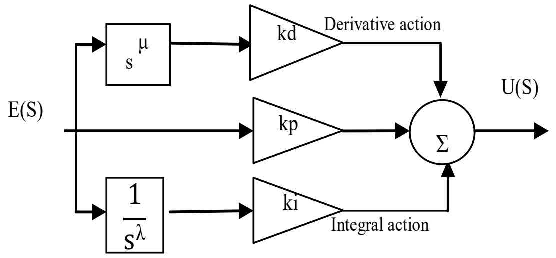

Fig. 4. Block-diagram of FOPID

It can be expected that the $PI^{\lambda}D^{\mu}$ controller may upgrade the systems efficiency. Control of industrial systems is one of the most important features of the $PI^{\lambda}D^{\mu}$ controller. Another feature lies in the fact that $PI^{\lambda}D^{\mu}$ controllers are low sensitive to the parameters changes of the controlled system also FOPID provide more flexibility in the controller design compared with the PID [22].

2.2. Structure of Quadrotor Based on FOPID

A block diagram of the quadrotor controlled using the FOPID controllers is presented in Fig.5. FOPID optimized by GA using Integral Square Error (ISE) cost function to ensure ideal control efficiency at nominal operating conditions. Where, each FOPID controller has 5 parameters, there are totally 15 parameters to be optimized by GA. All of the parameters are updated at every simulation time, where GA parameters $[kp_1\ ki_1\ kd_1\ \lambda_1\ \mu_1\ kp_2\ ki_2\ kd_2\ \lambda_2\ \mu_2\ kp_3\ ki_3\ kd_3\ \lambda_3\ \mu_3]$ with lower bounds = [0 0 0 0.01 0.01 0 0 0 0.01 0.01 0 0 00.01 0.01] and upper bounds= [200 200 200 1 1 200 200 200 1 1 200 200 200 1 1].

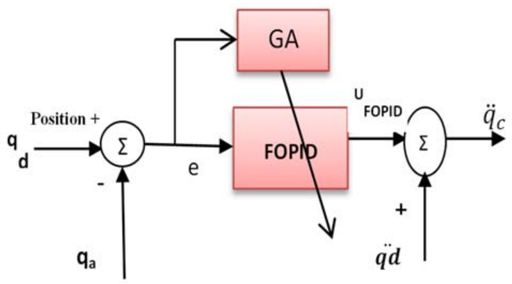

Fig.5. The block diagram of the proposed FOPID controller

The 5 gains of FOPID after tuning for X (k_{p1}=0.35 , k_{d1}=8.24 , k_{i1}=13.2 , λ_1=0.372 and μ_1=0.93), for Y are (k_{p2}=36.37 , k_{d2}=17.13 , k_{i2}=58.6 , λ_2=0.96 and μ_2=0.96) and for Z are (k_{p3}=99.37 , k_{d3}=6.08 , k_{i3}=24.53 , λ_3=0.98 and μ_3=0.94) .

V. SIMULATION RESULTS

The simulation has been performed for position (X, Y, Z) and attitude (Φ , θ, ψ) control of quadrotor using SIMULINK-MATLAB 2015a by considering the dynamic of the quadrotor from [18] for demonstrating the effectiveness of the suggested FOPID position controller than PID tuned by GA where, the controllers tried to track the path of a helical trajectory. Starting from random initialized parameters, GA progressively minimizes various integral performance indices iteratively while finding optimal set of parameters for the FOPID and PID controller. The algorithm calculation ends if the estimation value of the objective function does not change obviously over some progressive iteration. The values of the 9 PID parameters obtain by GA with fitness value 0.025411after 260 epochs. Where, the required position (X, Y, and Z) presented in fig. 6.

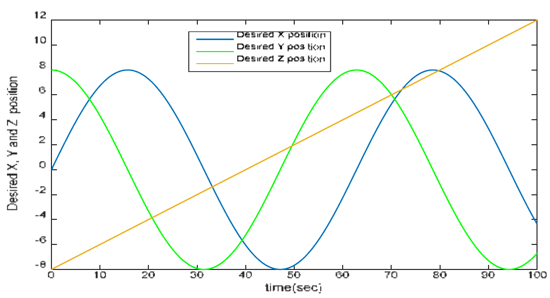

Fig.6. presents the required position(X, Y and Z).

As shown in fig.7 the particular helical trajectory screens shot is taken when the FOPID Control was utilized for X, Y and Z position with the existence of wind.

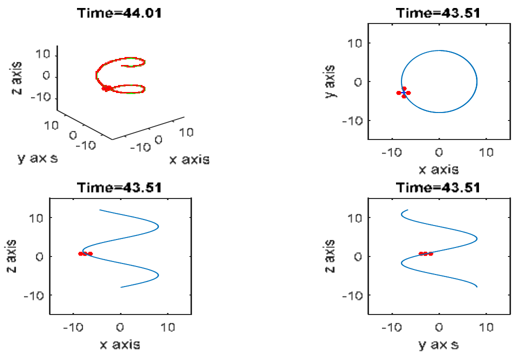

Fig. 7. The Helical Trajectory after using FOPID controller.

FOPID control provides the quadrotor with minimum error between desired and actual position for (X, Y and Z) respectively compared with PID controller as presented in Figs 8, 9 and10. Where, GA reaches to the values of the 15 FOPID parameters after 46 epochs with fitness 0.404491.

Table1 show a comparison between RMS error, steady state error for X, Y and Z for the two types of controllers PID tuned by GA and FOPID implemented to control the position of quad rotor

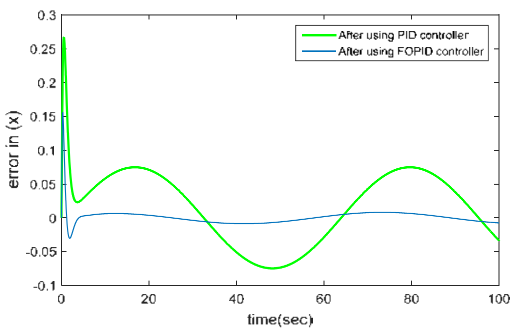

Fig. 8. Error in X-Position after using the two controllers.

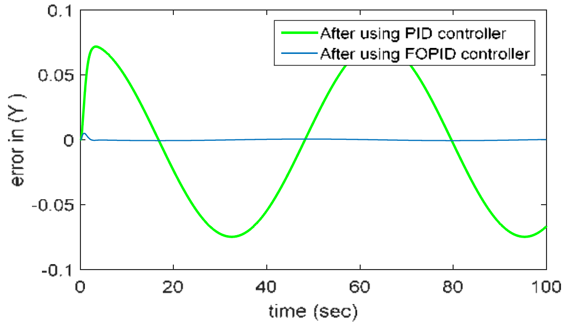

Fig. 9. Error in Y-Position after using the two controllers.

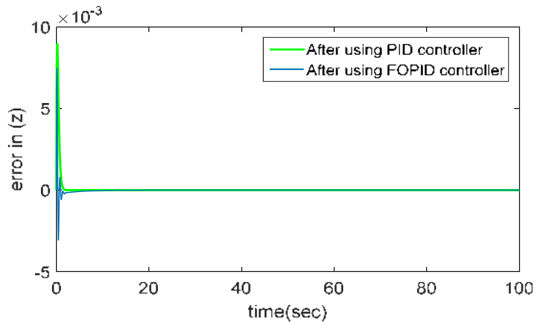

Fig.10. Error in Z-Position after using the two controllers.

TABLE1: THE COMPARISON RESULTS OF PID AND FOPID CONTROLLERS.

Controller type	RMS error	S.S. error for X	S.S. error for Y	S.S. error for Z
PID tuned using GA	0.00695	-0.03367	-0.06726	$6.217*10^{-15}$
FOPID	0.00012	-0.001838	0.0002049	$-2.66*10^{-15}$

From Table1 position control using FOPID has better steady state error and RMS error than controlled based on PID tuned using GA. By comparing steady state and RMS error in a system it was found that the FOPID's errors (Steady State error for X position=-0.001838, Y=0.0002049, Z=-$2.66*10^{-15}$ and RMS error=0.00012) less than PID's errors (Steady State error for X=-0.03367, Y=-0.06726, Z=$6.217*10^{-15}$ and RMS error=0.00695). FOPID controller has fast response and small errors for the required position of quad rotor.

Figures7, 8 and 9 give complete comparisons between the three controllers for X, Y and Z errors respectively.

Also attitude angles (roll(Φ) pitch(θ) yaw(ψ) angle) after using FOPID has fast response and small errors for the required orientation than controlled based on PID tuned by GA as shown in Figs 11, 12 and 13.

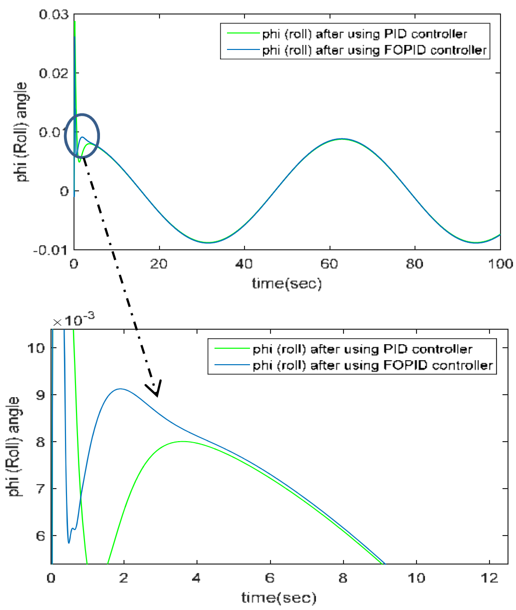

Fig. 11. Roll (Φ) angle after using the three controllers.

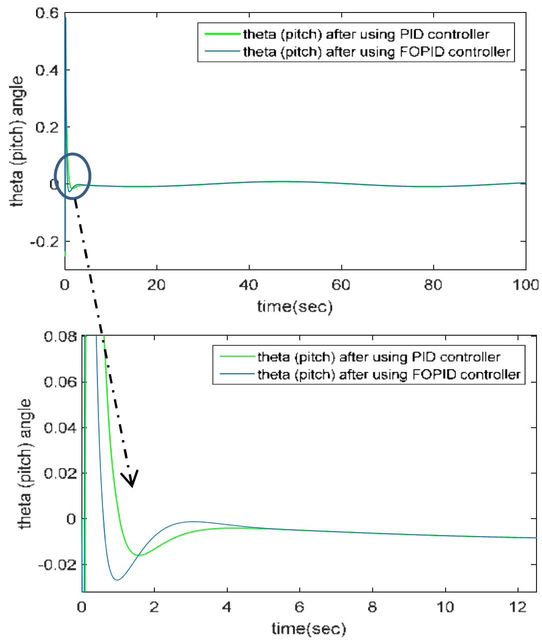

Fig. 12. Pitch (θ) angle after using the three controllers.

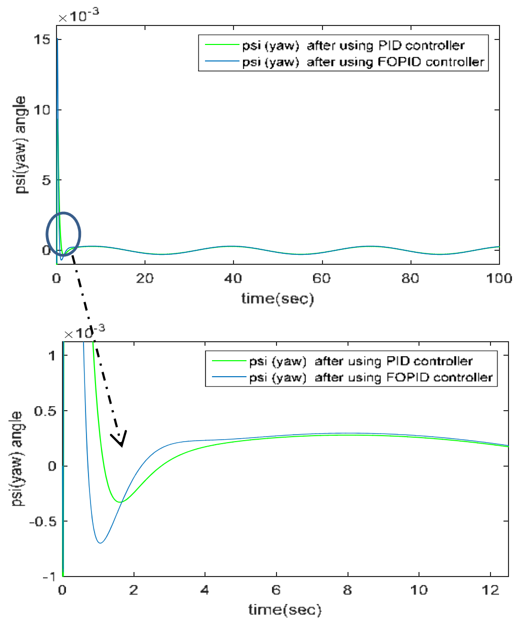

Fig. 13. Yaw (ψ) angle after using the three controllers.

VI. CONCLUSION

In this work, FOPID controllers have been used to position and attitude control of quadrotor to achieve the required position with fast response and minimum error. As appeared in results FOPID technique compared with PID tuned using GA, so from the simulation results it was concluded that:

- By comparing steady state and RMS error the position control of the X, Y and Z controlled using FOPID has better performance, steady state error and RMS error than controlled using PID.
- The attitude angle responses had showed to us that the system designed based on FOPID controller has much faster response than using the PID controller.

In future work It is needed to focus on further reducing the position and attitude errors of quadrotor using a new optimization algorithm and proved its superiority and robustness such as Fuzzy Ant Colony Optimization (FACO) or Fuzzy Bee Colony Optimization (FBCO) algorithm where, they are recommended to use ACO or BCO specifically for tuning membership functions of the fuzzy controller for more stability

REFERENCES

1. S. E. Hamamci, "An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers," IEEE Trans. Autom. Control, vol. 52, no. 10, pp. 1964–1969, Oct. 2007.
2. M. Ö. Efe, "Neural network assisted computationally simple PIλDμ control of a quadrotor UAV," IEEE Trans. Ind. Informat., vol. 7, no. 2, pp. 354–361,May 2011.
3. J. Han, "From PID to active disturbance rejection control," IEEE Trans. Ind. Electron., vol. 56, no. 3, pp. 900–906, Mar. 2009.
4. Podlubny, "Fractional-order systems and PIλDμ controller," IEEE Trans. Automatic Control, vol. 44, no. 1, pp. 208–214, 1999.
5. Z. Bingul and O. Karahan," Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control", Turk J Elec Eng & Comp Sci, Vol.20, No.Sup.1, pp. 1123-1136, 2012.
6. Kendoul, Farid. "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems." Journal of Field Robotics 29.2 (2012): 315-378.
7. Hou, Hongning, et al. "A simple controller of minisize quad-rotor vehicle." Mechatronics and Automation (ICMA), 2010 International Conference on. IEEE, 2010.
8. Reizenstein, Axel. "Position and Trajectory Control of a Quadcopter Using PID and LQ Controllers."(2017).
9. Bresciani, T. "Modeling, identification and control of a quadrotor helicopter. Master's thesis, Department of Automatic control, Lund University, October 2008, p. 170." (2008).
10. De Lellis Costa de Oliveira, Marcelo. "Modeling, Identification and Control of a Quadrotor Aircraft." (2011).
11. Al-Younes, Younes M., Mohammed A. Al-Jarrah, and Ali A. Jhemi. "Linear vs. nonlinear control techniques for a quadrotor vehicle." Mechatronics and its Applications (ISMA), 2010 7th International Symposium on. IEEE, 2010.
12. Minh, Ly Dat, and Cheolkeun Ha. "Modeling and control of quadrotor MAV using vision-based measurement." Strategic Technology (IFOST), 2010 International Forum on. IEEE, 2010.
13. Altug, Erdinc, James P. Ostrowski, and Robert Mahony. "Control of a quadrotor helicopter using visual feedback." Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 1. IEEE, 2002.
14. Mokhtari, Abdellah, et al. "Feedback linearization and linear observer for a quadrotor unmanned aerial vehicle." Advanced Robotics 20.1 (2006): 71-91.
15. Mellinger, Daniel, et al. "Design, modeling, estimation and control for aerial grasping and manipulation." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011.
16. Subodh Mishra ,"Robust Control of Quadrotors." ,Master thesis , Department of Automatic Control and Robotics,(2017).

17. Madani, Tarek, and Abdelaziz Benallegue. "Control of a quadrotor mini-helicopter via full state backstepping technique." Decision and Control, 2006 45th IEEE Conference on. IEEE, 2006.

18. Mahony, Robert, Vijay Kumar, and Peter Corke. "Multirotor aerial vehicles." IEEE Robotics and Automation magazine 20.32 (2012).

19. Mellinger, Daniel Warren. "Trajectory generation and control for quadrotors." (2012).

20. Youns, Majed D., Salih M. Attya, and Abdulla I. Abdulla. "Position Control Of Robot Arm Using Genetic Algorithm Based PID Controller." AL Rafdain Engineering Journal 21.6 (2013): 19-30.

21. D. Xue and Y. Chen, "A Comparative Introduction of Four Fractional Order Controllers," in Proceedings of the 4th World Congress on Intelligent Control and Automation, pp. 3228–3235, 2002.

22. M. R. Faieghi and A. Nemati , "On Fractional-Order PID Design", Applications of MATLAB in Science and Engineering, ISBN: 978-953-307-708-6, InTech, 2011. Available from: http://www.intechopen.com/books/applications-of-matlab-in-science-and engineering/on fractional-order-pid-design.