



Optimized PID Control of Quadrotor System Using Extremum Seeking Algorithm

Abdullah N. Muhsen ^{a*}, Safanah M. Raafat ^b

^{a*}, Control and System Eng. Dept., University of Technology, Baghdad, Iraq.
61364@student.uotechnology.edu.iq

^b Control and System Eng. Dept., University of Technology, Baghdad, Iraq,
safanah.m.raafat@uotechnology.edu.iq

*Corresponding author.

Submitted: 07/10/2020

Accepted: 28/02/2021

Published: 25/06/2021

KEY WORDS

Quadrotor Controller, PI-D, Automatic Tuning PID, extremum seeking, ES.

ABSTRACT

This paper presents the development of an automatic tuning of conventional PI-D controllers for Quadrotor system. The most critical problem of tuning the parameters of the PID, is that they can drastically affect the performance of the system. Accordingly, Extremum seeking (ES) algorithm is utilized here to reduce a selected cost function that brings the required performance aspects. The results show that the ES-PID controller provides better stability and performance as compared with only PI-D controller. The parameters of ES-PID controller have been successfully tuned and modified to reach the desired destination with significantly less overshoot, oscillations, and faster convergence.

How to cite this article: A. N. Muhsen and S. M. Raafat, "Optimized PID Control of Quadrotor System Using Extremum Seeking Algorithm," Engineering and Technology Journal, Vol. 39, No. 06, pp. 996-1010 2021.

DOI: <https://doi.org/10.30684/etj.v39i6.1850>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>

1. INTRODUCTION

A quadrotor or drone is a type of helicopter with four rotors. They are well suited to autonomously perform complicated civilian or military operations. Automated drones can complete a wide range of applications [1,2]. Automated drones can be used in oil & gas facilities for security [3], surveillance [4], emergency response [5] and infrastructure inspection [6]. Quadrotor is generally a small aircraft with multi-rotors, with nonlinear, under-actuated systems. When controlling the thrust, and consequently, the velocity of each one of the rotors, a motion in three paths (x, y, and z) and the rotation around three axes (Roll, Pitch, and Yaw) are generated. The quadrotors under-actuated property will cause a strong coupling between rotary and translational motions. As a result, it is a difficult task to control the trajectory tracking or the position of the quadrotor [7]. To overcome this issue, many control techniques have proven to provide positive performances, and to solve the problems of trajectory-tracking.

Recently, many works have been developed to model and control these platforms. An adaptive state feedback control method denoted as Classical Multiple Model Adaptive Control had been implemented by Raafat and Mahmoud (2018) [8], where each Kalman filter had been designed for a specific value of

an equilibrium point with LQ-Servo controller. Suitable performance and tracking design have been achieved. Another study is done by Alawsi et al. (2020) [9], where a multimode control algorithm has been proposed as a solution for GPS signal denied through the autonomous flight, where it is based on GPS and accelerometer modes with Unscented Kalman Filter (UKF). The proposed algorithm can keep the quadrotor under control during GPS signal denial period and continue moving toward the desired path.

Kotarski et al (2016) [10], designed a quadcopter control algorithm to improve the performance of standard PI-D controller. The implemented PI-D controller achieved an improvement in robustness, stability, and error boundedness. The utilization of PI-D Controller stops the control system from getting higher error signal (e) situation considering use of the derivative of the state rather than the error in its equation, the (D) derivative term works on the calculated plant output instead on the error signal. This PI-D controller avoids big control efforts throughout a step-change in the reference signal. Typically, tuning of PI-D parameters is commonly done manually. However, the effectiveness of the tuning is commonly based on the control engineer experiences. Also, in most practical scenarios, the plant model is undiscovered or just approximately known. Hence, a strategy for tuning the PID controller is very appealing in the closed loop setting [11]. Several approaches were developed to achieve this purpose. One of the most well-suited approach for achieving automated PID tuning, which is the Extremum Seeking Control (ESC) [12], where it utilizes a cost function that is based upon the difference between the real response and the preferred response. The output of the cost function will then be fed back to the extremum seeking (ES) loop to be able to tune the three parameters K_p , K_i and K_d of the PID controller [13, 14]. The ESC is a paradigm that aims to find an extremum value of an unknown nonlinear mapping. Therefore, it had been shown that it is useful to apply ESC to tune PID controller.

Killingsworth and Krstic (2006) [15], presented an enhanced PID controller with discrete type of ES. Mainly, ES reduces a cost function that quantifies the accuracy of the PID controller. ES which is a non-model-based technique, iteratively modifies the cost function arguments (PID parameters) to ensure that the cost function output goes to a local minimum. ES technique has an advantage than standard PID model in some applications with actuator saturation. Moreover, the ES cost function may be specified to reflect the preferred performance attributes. Guay and Dochain (2014) [16], proposes an alternative design approach of ESC as a solution of real-time enhancement control issues. The approach considers a proportional integral method that avoids the requirement for a timescale separating in the formulation of the ESC. Raafat et al. (2015) [17], proposed a real-time optimized (PID) controller for a three Degree of Freedom (3DOF) helicopter. Similarly, Raafat and Hussein (2018) [18], developed an ES strategy for tuning PI controller to maximize the power capture in a variable speed variable pitch wind turbine (VSPWT).

In this work, ESC has been developed for tuning PI-D control parameters. Six PID controllers has been used for (x, y, z, roll, pitch, and yaw) in the feedback and six PI controller in the feedforward at the reference signal. The ES algorithm is used in order updates the PI-D controller parameters to minimize the cost function which is computed from a step-response experiment. Then, we compared the results from our proposed model with traditional PID controller for three cases, displacement, when subject to uncertainty and the disturbance rejection performance.

2. MODELLING OF QUADROTOR:

Quadrotors are typically modeled and flown in two configurations: the (cross "x") and the (plus "+") configuration, as shown in Fig.1, where two motors rotate clockwise, and the two other motors rotate counterclockwise. This strategy helps the quadrotor to not rotate on its vertical axis as the rotational inertia is canceled out, discarding the requirement for a tail rotor that is used to balance the conventional quadrotor [7, 19].

The quadrotor has 6 degrees of freedom, three rotational and three translational, as in Figure 1. The location on the x and y axes is named as such, although the position on the z-axis is described as altitude. Also, the orientation angles measured are referred to as Roll (ϕ), pitch (θ), and yaw (ψ), to be the rotation angles over the x, y, and z-axis, respectively. The ψ angle may also be referred to as heading [19].

The motors are numbered from 1 to 4, and each motor has its separate motor signal, denoted U_1, U_2, U_3 , and U_4 for each motor. When every motor rotates at the similar angular velocity, then the torques τ_1, τ_2, τ_3 and τ_4 (these are generally the counter torques given to the aircraft because of the motor's rotation) will terminate each other out, and therefore the quadrotor is not going to spin about its z^b -axis ($\dot{\psi} = 0$). The quadrotor is going to hover in case the angular velocities are in a way that the total thrust ($f_1 + f_2 + f_3 + f_4$) made by the rotors is corresponding to the gravity force. To be able to

identify the quadrotor movement and its attitude, a couple of reference frames are used, which is, the inertial and the body frame (see Fig. 1). The inertial frame is specified by the ground, with gravity pointing in the negative z -direction. The body frame is specified by the quadrotor orientation, with the rotor axes positioned in the positive z^b direction and the arms positioned in the x^b and y^b directions. The quadrotor attitude is concluded by three angles, which are terms, roll- ϕ , pitch- θ , and yaw- ψ [14, 20].

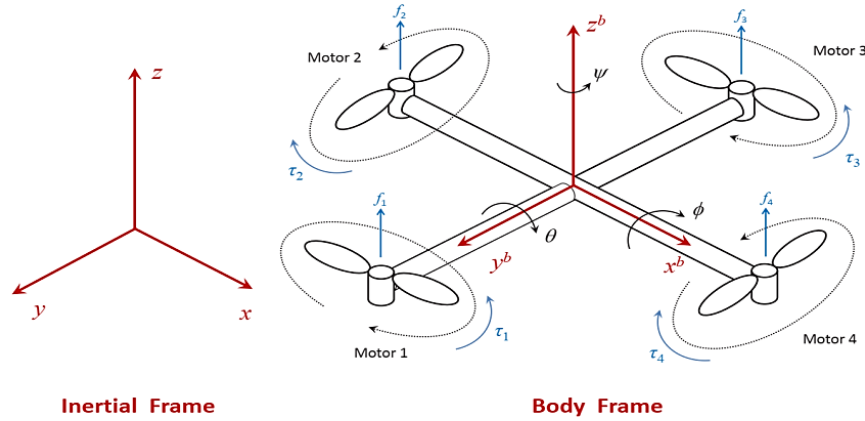


Figure 1: Quadrotor degrees of freedom [19]

The equations of quadrotor linear motion are [21]:

$$\text{For } I_F: m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R(\Theta)T_B + \Gamma_D = \frac{R(\Theta)}{m}T_B - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \Gamma_D \quad (1)$$

where x , y , and z are the coordinates of the position of the quadrotor in the inertial frame, m is the system mass, g is the acceleration related to gravity, Γ_D is the drag force due to air friction, T_B is the thrust vector in the body frame, and $R(\Theta)$ is the overall rotation matrix. The drag Γ_D due to air friction is modeled as a force relative to the linear velocity in every direction:

$$\Gamma_D = -k_d \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (2)$$

where, k_d is the coefficient of air friction.

The thrust f_i generated by the $i = 1$ to n rotor (where $n=4$ rotor for a quadrotor) is given by the following expression:

$$T_i = k \sum_{i=1}^4 \omega_i^2 = k(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (3)$$

where k is the friction constants; ω is the motor angular velocity.

The total thrust T_B produced by the 4 rotors is determined by following equation:

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 f_i^2 \end{bmatrix}, T_B \in \mathbb{R}^3 \quad (4)$$

The overall rotation matrix $R(\Theta) \in \mathbb{R}^{3 \times 3}$ is equal to $R(\phi)R(\theta)R(\psi)$, then it is given by:

$$R(\Theta) = \begin{bmatrix} c_\phi c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \quad (5)$$

where, $c=\cos$, $s=\sin$,

At this point, it is useful to use Euler's equations designed for rigid body dynamics, that can be defined as follows:

$$\tau = I\dot{\omega} + \omega \times (I\omega) \quad (6)$$

where " $\times$ " denotes cross product, $\tau = [\omega_x \omega_y \omega_z]^T$ is the vector of external torques, $I \in \mathbb{R}^{3 \times 3}$ is the inertia matrix, and $\omega = [\omega_x \omega_y \omega_z]^T$ is the angular velocity vector. The Body Frame Dynamics is:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (7)$$

where, I_{xx} , I_{yy} and I_{zz} are the moments of inertia of the quadrotor about the x_B , y_B and z_B axes respectively.

The roll moment τ_ϕ and pitch moment τ_θ torques can be derived from the essential mechanics as follows [22]:

$$\tau_\phi = L(T_1 - T_3) = Lk(\omega_3^2 - \omega_1^2) \quad (8)$$

$$\tau_\theta = L(T_2 - T_4) = Lk(\omega_4^2 - \omega_2^2) \quad (9)$$

where L is the distance between motor and the quadrotor centre (radius). The following equation gives the total torque about the z^b -axis that is the yaw τ_ψ torque:

$$\tau_\psi = b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (10)$$

where, b is the coefficient of propellers drag.

The Roll, Pitch, and yaw rates are relating to the factors of the angular velocity vector utilizing the following expression:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix} \quad (11)$$

where, $t = \tan$.

The quadrotor motors generate Roll, Pitch, and yaw torques in the path corresponding to the main body frame [23]:

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} Lk(\omega_1^2 - \omega_3^2) \\ Lk(\omega_4^2 - \omega_2^2) \\ k(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{bmatrix} \quad (12)$$

where, L is the distance from the quadrotor center to any of the propellers.

To control the angles and throttle, the control signals $U_{throttle}$, U_{roll} , U_{pitch} , and U_{yaw} are used [23].

1- Total Thrust τ_B

$$U_{throttle} = k(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (13)$$

2- pitch moment, τ_θ

$$U_{roll} = Lk(\omega_3^2 - \omega_1^2) \quad (14)$$

3- Roll Moment τ_ϕ

$$U_{pitch} = Lk(\omega_4^2 - \omega_2^2) \quad (15)$$

4- Yaw Moment τ_ψ

$$U_{yaw} = b(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \quad (16)$$

Where L is the distance from the centre of the quadrotor to any of the propellers; U_2 , U_3 and U_4 are input forces to the quadcopter generated by motors.

The overall rotation matrix $R(\Theta)$ is equal to $R(\phi)R(\theta)R(\psi)$, then it is given by:

$$R(\Theta) = \begin{bmatrix} c_\phi c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \quad (17)$$

By substance equations (4) and (15) in (1) we will get

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{\begin{bmatrix} c_\phi c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix}}{m} \begin{bmatrix} 0 \\ 0 \\ T_i \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{T}{m} \begin{bmatrix} c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi s_\theta c_\phi - c_\psi s_\phi \\ c_\phi c_\theta \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (19)$$

Then we get:

$$\ddot{x} = \frac{U_{throttle}}{m} (c_\psi s_\theta c_\phi + s_\psi s_\phi), \quad (20)$$

$$\ddot{y} = \frac{U_{throttle}}{m} (s_\psi s_\theta c_\phi - c_\psi s_\phi) \quad (21)$$

$$\ddot{z} = \frac{U_{throttle}}{m} (c_\phi c_\theta - g) \quad (22)$$

The relations between the angular velocity of motors and generated forces $U_{throttle}$, U_{roll} , U_{pitch} , and U_{yaw} by motors can be presented in the matrix form [13, 14, 22]

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k} & 0 & \frac{-1}{2k} & \frac{-1}{4d} \\ \frac{1}{4k} & \frac{-1}{2k} & 0 & \frac{1}{4d} \\ \frac{1}{4k} & 0 & \frac{1}{2k} & \frac{-1}{4d} \\ \frac{1}{4k} & \frac{1}{2k} & 0 & \frac{1}{4d} \end{bmatrix} \begin{bmatrix} U_{throttle} \\ U_{pitch} \\ U_{roll} \\ U_{yaw} \end{bmatrix} \quad (23)$$

The angular Euler-Lagrange equations are []

$$\tau = \tau_B = J\ddot{\eta} + \frac{d}{dt}(J)\dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T J \dot{\eta}) \quad (24)$$

$$\tau_B = J\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} \quad (25)$$

The matrix $C(\eta, \dot{\eta})$ has the form, as shown in [23],

$$C(\eta, \dot{\eta})\dot{\eta} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{32} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (26)$$

By compensating the value of η and value of matrix $C(\eta, \dot{\eta})$ in equation (26), and the following angular acceleration equations can be found as:

$$\ddot{\phi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{U_{pitch}}{I_x} \quad (27)$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} + \frac{U_{roll}}{I_y} \quad (28)$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z} \dot{\theta} \dot{\phi} + \frac{U_{yaw}}{I_z} \quad (29)$$

Equations (20, 21, 22) and (27, 28 29) represent linear motions and angular motions of the quadrotor, respectively.

3. PI-D CONTROLLER DESIGN:

The main objective of deriving the quadrotor mathematical model is to help in develops of controllers for real quadrotors. The inputs of the proposed system involve the angular velocities of every rotor, as the voltages across the motors are all that can be controlled.

As specified in [23], PI-D control or three-term' control is based on computing the error between measured state and the desired set point along with the change rate of this error and the integration of this error. The controller output is a signal that is relative to the sum of all these errors, each one multiplied by a constant. To control the quadrotor using PI-D or PID -ES control, six single-input and single-output controllers are required to control $(\phi, \theta$ and $\psi)$ (heading), and movement in the i, j, k directions. The 1st step is to balance the $(\phi, \theta,$ and $\psi)$.

The mathematical form of PI-D is given as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} state \quad (30)$$

The errors can be computed from the following equation:

$$e(t) = \text{desired state} - \text{measured state} \quad (31)$$

The required thrust force, $U_{throttle}$, can be calculated by:

$$U_{throttle} = K_{pz} e_z + K_{Iz} \int e_z - K_{Dz} \dot{z}_{meas} + mg \quad (32)$$

where, K_{pz} , K_{Iz} , and K_{dz} are altitude PI-D gains controller; g is the earth's gravity and e_z is the altitude error. To control attitude, three control signals are utilized to obtain the required quadrotor attitude. These signals are utilized to control the Roll (ϕ) , Pitch (θ) , and Yaw (ψ) angles, which describe the warp angles of the quadrotor's body over the x, y and z axis, respectively. To calculate the controlling signals, a number of equations is used for each direction, which can be specified as in follows [24]:

- The torque U_{roll} to rotate the quadrotor over the x-axis can be computed by:

$$U_{roll} = K_{p\phi} e_\phi + K_{I\phi} \int e_\phi - K_{D\phi} \dot{\phi}_{meas} \quad (33)$$

where: $K_{p\phi}$, $K_{I\phi}$ and $K_{D\phi}$ are roll angle PI-D gains controller; e_ϕ is roll angle error.

- The torque U_{pitch} to rotate the quadrotor over the y-axis can be computed by:

$$U_{pitch} = K_{p\theta} e_\theta + K_{I\theta} \int e_\theta - K_{D\theta} \dot{\theta}_{meas} \quad (34)$$

where: $K_{p\theta}$, $K_{I\theta}$ and $K_{D\theta}$ are pitchangle PI-D gains controller; e_θ is pitch angle error.

The torque U_{yaw} to rotate the quadrotor over the z-axis can be computed by:

$$U_{yaw} = K_{p\psi} e_\psi + K_{I\psi} \int e_\psi - K_{D\psi} \dot{\psi}_{meas} \quad (35)$$

where: $K_{p\psi}$, $K_{I\psi}$ and $K_{D\psi}$ are yaw angle PI-D gains controller; e_ψ is pitch angle error.

The quadcopter has 6DOF and just Four actuators (motors) therefore it is impossible to control each one of these DOF straight. Thus, the control equations (21) to (24) are utilized to control Four DOF directly (z, ϕ , Θ , ψ). The roll and pitch angles allow to move quadcopter in the direction of the desired x and y direction. Hence, the desired x and y can be utilized first to determine the desired roll and pitch angles to make it possible for controlling x and y positions indirectly. Thus, to calculate the desired roll and pitch angles, several equations have been used which are generally written in terms of required x and y accelerations. The equations that used to determine desired acceleration through desired x, y and z are [7]:

$$\ddot{x}_{des} = \frac{K_{px} e_x + K_{Ix} \int e_x - K_{Dx} \dot{x}_{meas}}{m} \quad (36)$$

$$\ddot{y}_{des} = \frac{K_{py} e_y + K_{Iy} \int e_y - K_{Dy} \dot{y}_{meas}}{m} \quad (37)$$

The implementation of time-domain of the discrete-time extreme seeking algorithm in Figure 2 can be described as in follows:

$$\zeta(k) = -h\zeta(k-1) + J(\vec{\theta}(k-1)) \quad (45)$$

$$\hat{\theta}_i(k+1) = \hat{\theta}_i(k) - \gamma_i \alpha_i \cos(\omega_i k) [J(\vec{\theta}(k)) - (1+h)\zeta(k)] \quad (46)$$

where: $\zeta(k)$ It is a scalar, i is the subscript signifies the i_{th} entry of a vector. α_i is the perturbation amplitude and γ_i is the adaptation gain.

Convergence and stability are affected by the values of α , and the cost function $J(\vec{\theta})$ shape is near the minimizer. The modulation frequency ω_i is selected in a way that $\omega_i = a^i \pi$, where a satisfies $0 < a < 1$. In addition, the high-pass filter $(z-1)/(z+h)$ is designed with $0 < h < 1$ and a cut-off frequency which is under the modulation frequency ω_i .

From Figure 3, The ES algorithm updates the PI-D controller parameters $\theta(k)$ to minimize the cost function $J(\vec{\theta})$, which is computed from a step-response experiment carried out within the dashed box. However, the dashed box in figure 3 represents the step-response experiment, which is run iteratively. The cost $J(\vec{\theta}(k))$ is calculated after the step-response experiment. The extreme seeking algorithm utilized the cost function value $J(\vec{\theta}(k))$ in order to calculate new controller parameters $\vec{\theta}(k)$. An additional step function experiment will then be implemented with the new controller parameters, and then the process proceeds iteratively.

The proposed control structure consists of six PID controllers (for x, y, z, roll, pitch and yaw) in the feedback and six PI controller in the feedforward at the reference signal, so $\vec{\theta}$ contains 30 elements to be tuned. They include $\Theta 1$ for position parameters and $\Theta 2$ for acceleration parameters. The overall ES-PID tuning scheme is illustrated in Figure 3.

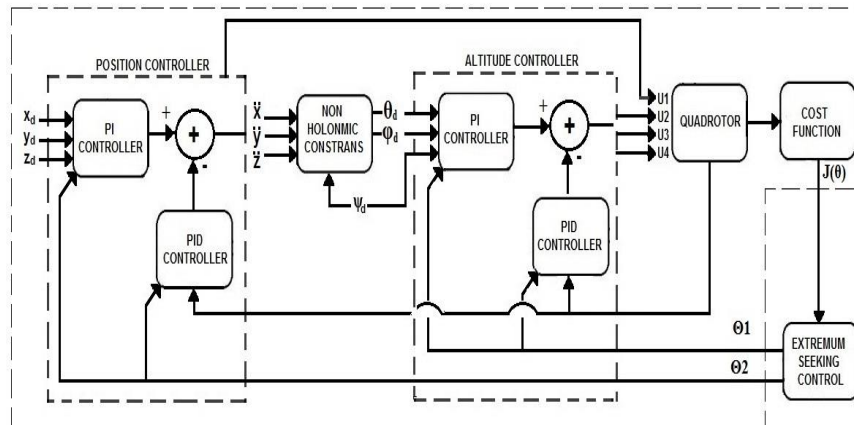


Figure 3: Scheme of overall ES-PID tuning of quadrotor.

5. SIMULATION RESULTS:

The simulation and visualization of quadrotor have been designed by MATLAB software based on the system's model, as described in Section 2. For our test, we tested the proposed auto-tuning PI-D controller (ES-PID) controllers and compared them with PI-D of best gain parameters used for the same model. The Quadrotor parameters, PI-D and ES-PID control parameters used in our simulation are given in Table I, Table II and Table III.

TABLE I: The quadcopter Simulation parameters

Parameters	Values and Units
Simulation Time	20 sec
Mass (m)	0.8 kg
Gravity (g)	9.81 (m/s ²)
Length of the rods (L)	0.25m
Propellers Drag Coefficient (b)	3.13×10^{-5}

TABLE II: The gain parameters for PID

Parameters	PID	Parameters	PID
K_{Px}	2	$K_{P\phi}$	1.5
K_{Ix}	0.008775	$K_{I\phi}$	0.8775
K_{Dx}	2.8	$K_{D\phi}$	0.5
K_{Py}	2	$K_{P\theta}$	1.5
K_{Iy}	0.008775	$K_{I\theta}$	0.8775
K_{Dy}	2.8	$K_{D\theta}$	0.5
K_{Pz}	10	$K_{P\psi}$	0.21
K_{Iz}	2	$K_{I\psi}$	0.01225
K_{Dz}	7	$K_{D\psi}$	0.7997

TABLE III: The ES-PID control parameters

Parameters	Values and Units
Gamma	[2 1 2]
Alpha	[.4 .001 .58]

The desired position in x, y, z coordinates of (10; 13; 6) have been used in all simulations. The previously described PI-D controller has been used to drive the quadrotor to the desired position, in which we compared both the controller in both driving error and response time. Also, we applied two turbulence factors that can be affected by the quadrotor, which are disturbance uncertainty.

I. Displacement Results

In this part, we simulate the quadrotor displacement and compare results for PI-D and ES-PID controller. The results are shown in Figures 4, 5 and 6.

From figure 4 (a), Figure 5 (a) and Figure 6 (a), both controllers start to guide quadrotor from zero position of x, y, z= [0, 0, 0] to the desired position of [10, 13, 6]. However, we can notice that ES-PID control is more accurate than PI-D control on both of stability and response time. By checked results, the ES-PID control is uniformly raised until reach to close from desired distance which is about 10.05m (i.e., error about 0.05m) and in about 9 sec while the results from PI-D control is non-uniform which is raised and down until reach stability at a distance near desired distance of about 10.08m (i.e., error about 0.08m) and in about 12 sec. From figure 5 (a), the ES-PID control is uniformly raised until reach to close from desired distance which is about 13.05m (i.e., error about 0.05m) and in about 9 sec while the results from PI-D control is non-uniform which is raised and down until reach stability at a distance near desired distance of about 13.1m (i.e., error about 0.1m) and in about 11 sec. From figure 6 (a), the ES-PID control is uniformly raised until reach to close from desired distance which is about 10.02m (i.e., error about 0.02m) and in about 10 sec while the results from PI-D control is nonuniform which is raised to 6.55m and down until reach stability at a distance near desired distance of about 10.02m (i.e., error about 0.02m) and in a relatively long time of about 16 sec. From figure 6 (c), we can notice that ES is tuning the parameters (K_p , K_i , K_d) of the PID controllers and modifies them until reaching the desired displacement, and as compared with another algorithm [18], where PSO algorithm had been used, ES algorithm achieves better tracking performance.

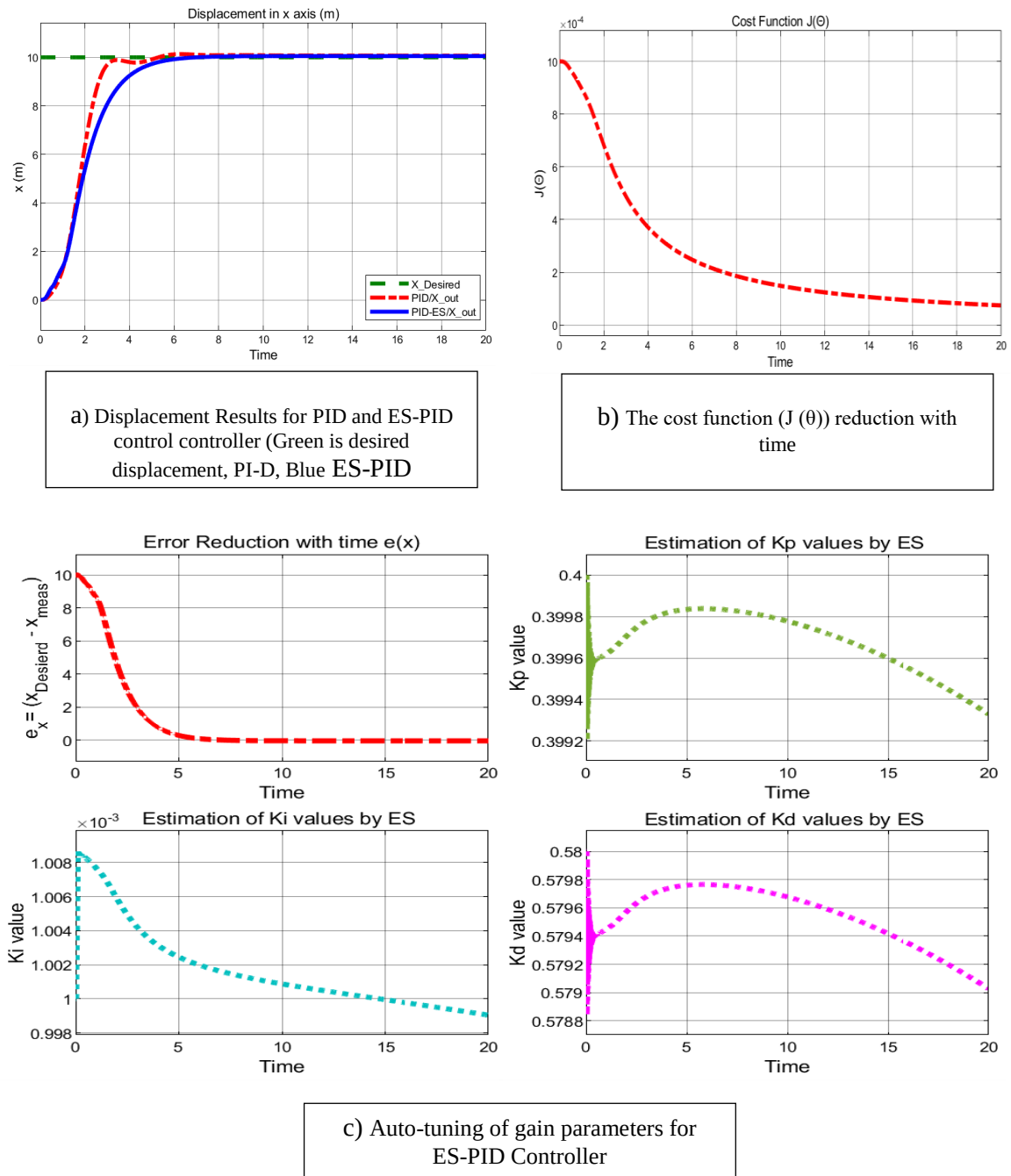


Figure 4: Simulation results for quadrotor displacement in the x-axis for PI-D and ES-PID control.

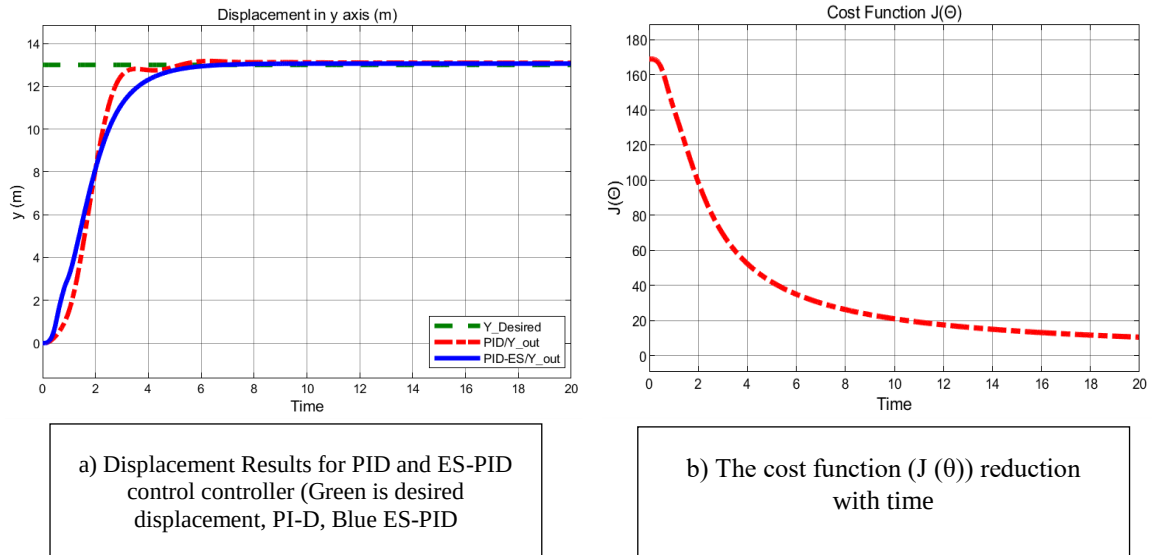


Figure 5: Simulation results for quadrotor displacement in the y-axis for PI-D and ES-PID control.

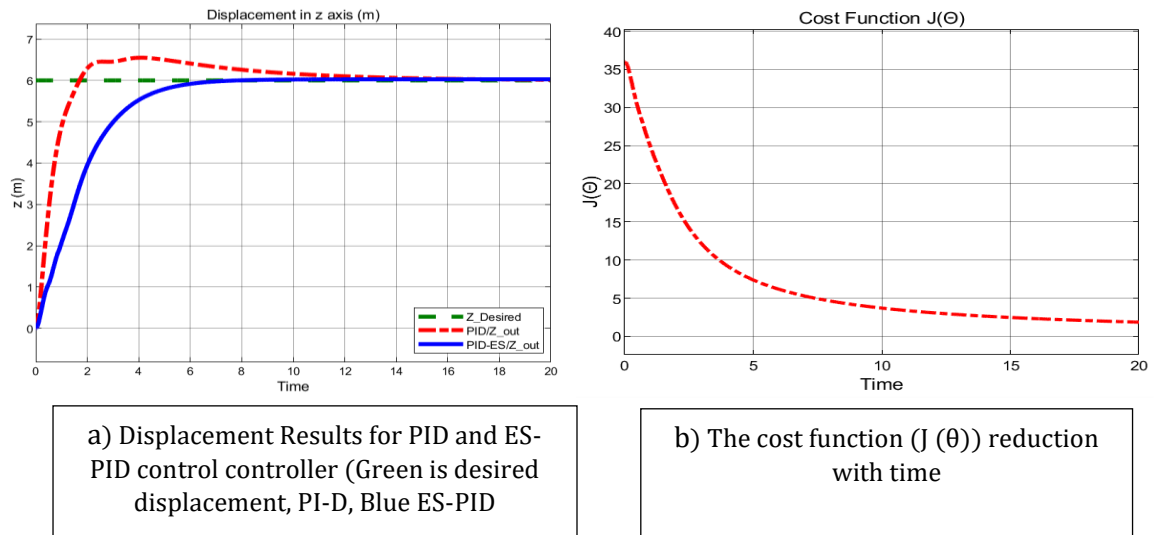


Figure 6: Simulation results for quadrotor displacement in the z-axis for PI-D and ES-PID control.

II. Disturbance Rejection Performance of the Controllers:

In this part, we test the disturbance rejection performance of both controllers. In this test, we applied disturbance at 10 seconds. The results are shown in Figure 7.

Simulation results for disturbance effect on quadrotor displacement for From Figure7 (a), (b), and (c) as previously, both controllers start to guide quadrotor from zero position to the desired position. From the results, we can see the effect of disturbance started from the tenth second. Both controllers start to overcome disturbance and driven quadrotor to the desired value. However, ES-PID is starting to reduce disturbance faster than PI-D. From Figure 7 (a), the ES-PID starts to reduce the effect from about 10.4 seconds reach the desired value in the fifteenth second, while PI-D is starting at 11 second and reach the desired value in the fifteenth second. From figure7 (b), the ES-PID starts to reduce the effect from about 10.4 sec and reach the desired value in the 17th second while PI-D is starting at 10.8 seconds and reach to the desired value in 17th second. From Figure 7 (c), the ES-PID is starting to reduce instantly from about 10th sec and reach the desired value in the 17th second; however, the PI-D starting also reduce instantly from about 10th second and reach to the desired value in 17th second. From these results, the ES-PID achieves better disturbance rejection performance than PI-D only.

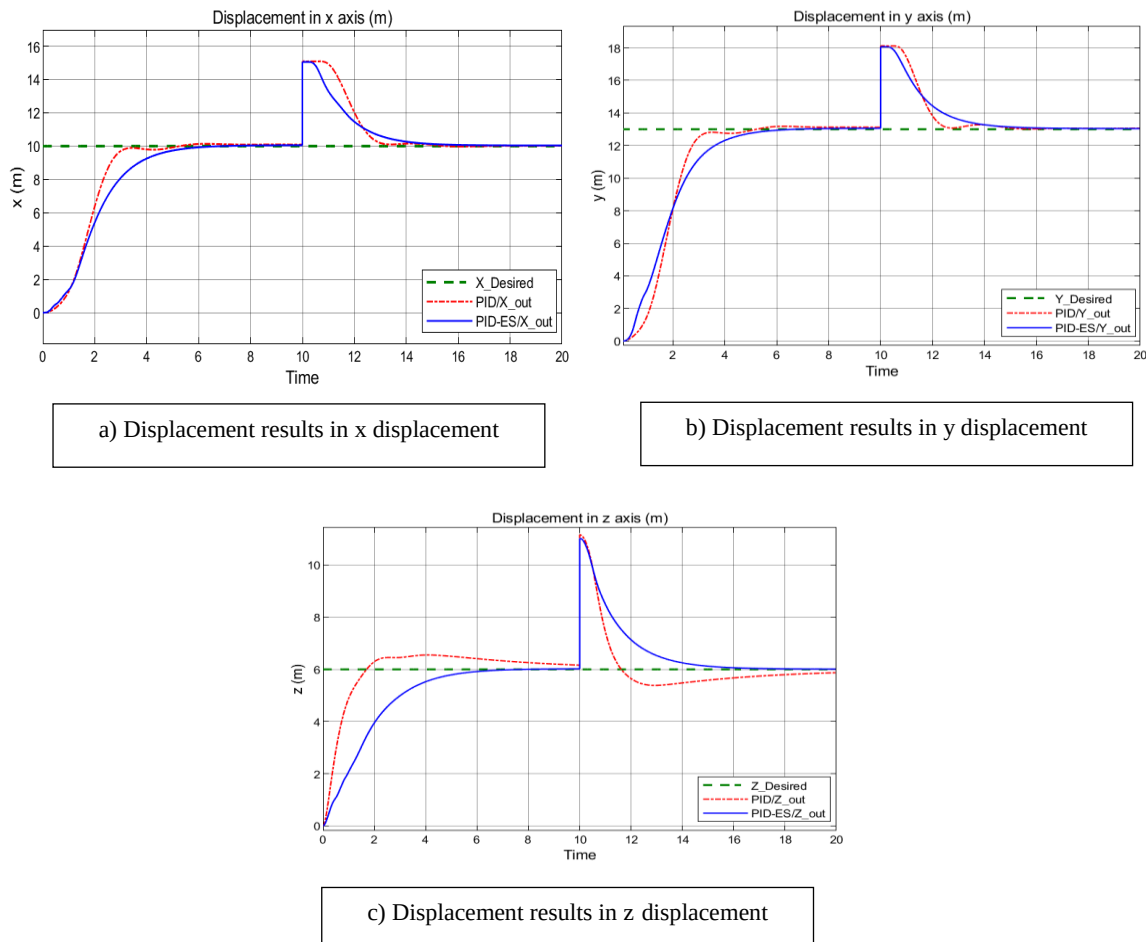


Figure 7:

Figure 8: PI-D and ES-PID control (Green is desired displacement, Red PI-D, Blue ES-PID).

III. Effect of the system Uncertainty

The nominal value of mass (m) in our application is 0.8kg, when we use a PI-D controller and ES-PID controller. In this part, we test the quadrotor controls for controlling the uncertain nonlinear to minimize the effect of the system uncertainties due to the uncertainty in the mass. For this purpose, we made a high variation of mass value in every second, these mass values are varied from (0.5 to 5) kg. The results are shown in Figure 8 (a), (b), and (c).

For Figure 8, both controllers start to guide quadrotor from zero position to the desired position. From results, we can see the effect of uncertainty started from the beginning, and there are acute effects when there is a high variation. However, both controllers start to overcome uncertainty and driven quadrotor to the desired value. From results, PI-D is suffering from instability when compared with the ES-PID that shows uniform change and stability. From figure 8 (a), the ES-PID is start driven quadrotor to desired x position. It not affected by uncertainty. At the same time, there are acuate effects on PI-D from starting to 10th second and reach near desired value in the 11th second. From figure 8 (b), the ES-PID is start driven quadrotor to desired y position. It is also not affected by uncertainty, while there are acuate effects on PI-D from starting to 10th second. From figure8 (c), the uncertainty is more effected in z position than x, y position; however, ES-PID has successfully driven quadrotor to desired z position at 12th second smoothly, while there are acuate effects on PI-D from starting to 18th second.

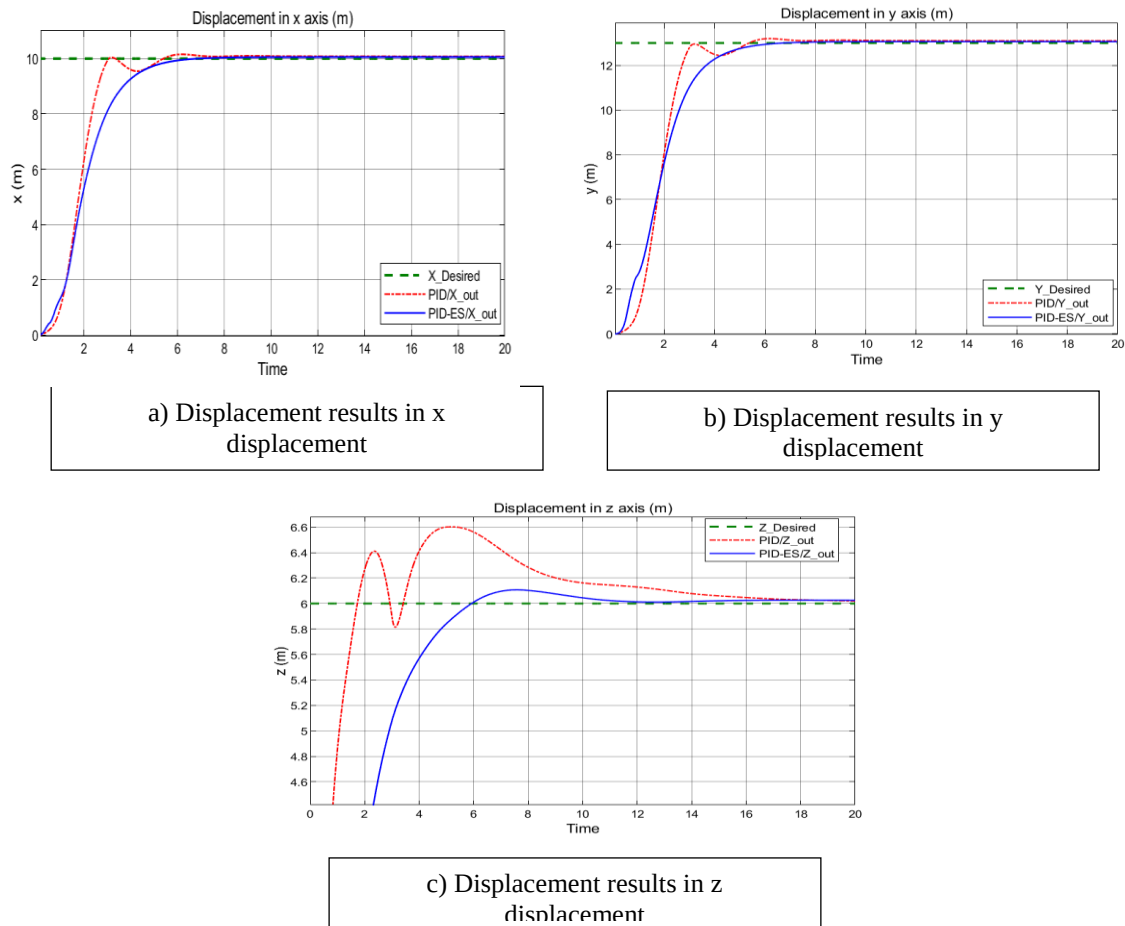


Figure 9: Simulation results for an uncertain parameter on quadrotor displacement for PI-D and ES-PID control (Green is desired displacement, Red PI-D, Blue ES-PID (a))

TABLE IV: The Comparison the slandered results and disturbance rejection results between the proposed model and traditional PID

		PID			ES-PID		
		Overshooting Value (m)	SteadyState Error(m)	Setting Time (sec)	Overshooting Value (m)	SteadyState Error(m)	Setting Time (sec)
Error reduction Normal	x	0	0.08	12	0	0.05	9
	y	0.2	0.1	11	0	0.05	9
	z	0.7	0.02	16	0	0.02	10
In presence of Uncertainty	x	0.2	0.06	11	0	0.02	7.6
	y	0.2	0.06	10	0	0.02	7.6
	z	0.6	0.02	18	0	0.02	12

In Table IV and table V we described the main defenses between the proposed model and traditional PI-D controller in term of overshooting, SteadyState error and setting time.

TABLE V: Comparison the uncertainty results between proposed model and traditional PID

		PID			ES-PID		
		Uncertainty Effect Time (sec)	SteadyState Error(m)	Setting Time (sec)	Uncertainty Effect Time (sec)	SteadyState Error(m)	Setting Time (sec)
x	15	1	0.06	8	0.5	0.02	7
y	18	1	0.02	8	0.5	0.02	7
z	11	0.1	0.2	10	0.1	0.02	7

6. CONCLUSION

In this study, the advantage of using ES optimization algorithm to tune PI-D controller for quadrotors is investigated. Two controllers have been compared, which are: PI-D and ES-PID that are presented with the same disruptions and by using the exact derivative and quadrotor parameters. The test results show that the ES-PID controller is better than the PI-D controller at avoiding steady-state error. The results also show that the ES-PID controller tuning is successfully tuned gain parameters (K_p , K_i , K_d) and modified them until they reach the desired displacement. The automatic selection of PID parameters is considerably better in overall. The ES tuned PID controller noticeably have less overshoot, a little swing in value, and faster converge, as we have seen in all results of uncertainty and disturbance the preference of the proposed method for all parameters of the model the obvious improvement. This is a result of the fact that the cost function of ES is emphasizes squared error, and therefore offers priority to reducing total error magnitude instead of long-term convergence.

References:

- [1] Y. Naidoo, R. Stopforth., G. Bright, Quadrotor unmanned aerial vehicle helicopter modelling & control, Int. J. Adv. Robot. Syst. 8 (2011) 139-149.
- [2] A. Zulu., S. John, A review of control algorithms for autonomous quadrotors. arXiv preprint arXiv:1602.02622. (2016). <https://doi.org/10.48550/arXiv.1602.02622>
- [3] S. O. Johnsen., T. Bakken., A. A. Transeth, S. Holmstrøm., M. Merz., E. I. Grøtli, Safety and security of drones in the oil and gas industry. Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference, ESREL2020-PSAM15 Organizers, Singapore, 2020.
- [4] A. Vacca, H. Onishi, Drones: military weapons, surveillance, or mapping tools for environmental monitoring? The need for legal framework is required, Transp. Res. Procedia, 25 (2017) 51-62. <https://doi.org/10.1016/j.trpro.2017.05.209>
- [5] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S. Nurre, L. Bian, Drones for disaster response and relief operations: A continuous approximation model, Int. J. prod. Econ., 188 (2017) 167-84. <https://doi.org/10.1016/j.ijpe.2017.03.024>
- [6] D. Carramiñana, I. Campaña, L. Bergesio, A. M. Bernardos, J. A. Besada, Sensors and Communication Simulation for Unmanned Traffic Management, Sensors, 21 (2021) 927. <https://doi.org/10.3390/s21030927>
- [7] I. Siti, M. Mjahed, H. Ayad, A. El Kari, New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods, Appl. Sci., 9 (2019) 1780. <https://doi.org/10.3390/app9091780>
- [8] S. M. Raafat, Z. Sh. Mahmoud, Robust Multiple Model Adaptive Control for Dynamic Positioning of Quadrotor Helicopter System, Eng. Technol. J., 36 (2018) 1249-1259. <https://doi.org/10.30684/etj.36.12A.6>
- [9] A. A. Alawsi, B. H. Jasim, S. M. Raafat, Improved Positioning of Delivery Quad copter Using GPS-Accelerometer Multimode Controller and Unscented Kalman Filter, IOP Conference Series: Mater. Sci. Eng., 745 (2020) 1-18. <https://doi.org/10.1088/1757-899X/745/1/012018>
- [10] D. Kotarski, Z. BeniĆ, M. Krznar, Control design for unmanned aerial vehicles with four rotors, Interdiscip. Descry. Complex Syst., 14 (2016) 236-245. <https://doi.org/10.7906/indec.14.2.12>
- [11] C. Zhang, R. Ordóñez, Extremum-seeking control, and applications: a numerical optimization-based approach, Springer Sci. Bus. Media, 2012. <https://doi.org/10.1007/978-1-4471-2224-1>
- [12] M. Krstić, Performance improvement and limitations in extremum seeking control, Sys. Control Lett., 39 (2000) 313-326.
- [13] J. Sobota, M. Schlegel, Iterative Feedback Tuning of PID Controller, Pro. Control 2004, (2004) 1-16.
- [14] T. R. Oliveira, L. R. Costa, A. V. Pino, P. Paz, Extremum Seeking-based Adaptive PID Control applied to Neuromuscular Electrical Stimulation. Anais da Academia Brasileira de Ciências. 91 (2019). <https://doi.org/10.1590/0001-3765201820180544>
- [15] N. J. Killingsworth, M. Krstic, PID tuning using extremum seeking online, model-free performance optimization, IEEE control systems magazine, 26 (2006) 70-79. <https://doi.org/10.1109/MCS.2006.1580155>
- [16] M. Guay, D. Dochain, A proportional integral extremum-seeking control approach, IFAC Proc. Vols., 47 (2014) 377-382. <https://doi.org/10.3182/20140824-6-ZA-1003.02492>

- [17] S. M. Raafat, Sh. S. Ali, R. Akmeliawati, Real-time optimization, and control of 3DOF helicopter system via extremum seeking algorithm, 2015 10th Asian Control Conference, Kota Kinabalu, Malaysia, (2015) 1-6. <https://doi.org/10.1109/ASCC.2015.7244822>
- [18] S. M. Raafat, R. Hussein, Multivariable Extremum Seeking Control for Power Maximization and PI Tuning of Wind Turbine System, 2018 Third Scientific Conference of Electrical Engineering , Baghdad, Iraq, (2018) 128-133. <https://doi.org/10.1109/SCEE.2018.8684099>
- [19] L. Ding, Z. Wang, Robust Control for an Aerial Robot Quadrotor under Wind Gusts, J. Rob., Hindawi, 2018. <https://doi.org/10.1155/2018/5607362>
- [20] L. Dewasme, A. V. Wouwer, Model-Free Extremum Seeking Control of Bioprocesses: A Review with a Worked Example, Processes, 8 (2020) 1209. <https://doi.org/10.3390/pr8101209>
- [21] T. Luukkonen, Modelling and control of quadcopter, Independent res. Proj. appl. Math., Espoo. 22 (2011).
- [22] K. Thu, G. A. Igorevich, Analysis, design and implementation of quadcopter control system, 2015 the 5th International Workshop on Computer Science and Engineering: Information Processing and Control Engineering, (2015) 277-283.
- [23] V. Katsikis, editor. MATLAB: A fundamental tool for scientific computing and engineering applications, BoD–Books on Demand; 3 (2012).
- [24] O. M. Agudelo, B. D. Moor, Computergestuurde Regeltechniek exercise session Case study: Quadcopter, 2019.