Design and optimization of backstepping controller applied to autonomous quadrotor

Mina Qays Kadhim¹² and Mohammed Y Hassan¹

¹Control and Systems Engineering Department, University of Technology-Iraq
²Corresponding author’s e-mail: 60024@uotechnology.edu.iq

Abstract. In this paper a Quadrotor dynamics is exploited. This system dynamics is nonlinear, multivariable, coupled and unstable, and suffers from parameter uncertainties and external disturbances. Hence, controlling of Quadrotor is on demand to meet the stability, robustness, and desired dynamic properties, furthermore, to overcome the hindrance of nonlinearity and to have a system that is pliant to changing parameters and environmental disturbances. Three PID position controllers are used in the outer feedback loop to track the reference trajectory, while the angular rotations are controlled through the inner feedback backstepping control. The control law is derived based on Lyapunov stability theorem to render strong closed-loop stability. The tuning of the gains for both controllers is not convenient with this kind of system model due to high non-linearity and instability. Thereafter, the gains and parameters referred to both controllers are optimized using Particle Swarm Optimization algorithm (PSO) to find the best navigation routes and ensure compensation of nonlinearities and disturbances. This is performed by minimizing the 3 Dimensional position errors and 3 Angular rotation errors using ITAE as a performance index. Simulation results presented using different types of trajectories have proved the enhancement in motion as compared with previous published papers.

1. Introduction

Unmanned aerial vehicles (UAV) became considerably prevailing due to the highly utilization opportunities that it could be engaged in like; exploring, navigation, as well as mapping. Today they are being used for surveillance and inspection missions. This kind of systems is a tough one that suffers from a lot of challenges to deal with, such as nonlinearity, disturbances and under-actuating problem presented by: having six elements to control although, there are only four control action to follow the required trajectories. Bouabdallah and Siegwart presented the control of an autonomous take-off, hover, landing, and collision avoidance vehicle in 2007 [1]. The major goal for Becker et al. in 2012 [2], was to avoid collision in OS4 systems through tracking a desired route as well as minimizing it’s operating time and the total weight by not depending on the grounded sensors. Hua et al. in 2013 applied feedback control design for a family of robotic aerial vehicles with vertical take-off and landing capabilities, such as; Quadrotors, and helicopters. It showed that with optimization, this design is useful to choose the best parameters to maintain the vehicles stability [3]. Basri et al. in 2014 introduced Backstepping controller to control a Quadrotor UAV after optimizing the rotatinal control parameters using Particle Swarm Optimization (PSO). It uses the integral absolute error (IAE) as a performance index [4]. Two control schemes were applied in 2014 by Swarup and Sudhir to obtain the desired trajectory planning of a Quadrotor as a comparative study, in which it showed that controlling the OS4 was quite smoother and faster using sliding mode controller [5]. In 2015, Galvez et al. used Neural Network as a computer algorithm to mimic the biological structure of neurons (nerve cell) to
obtain an accurate tracking route for a Quadrotor. Introducing the effect of feedback control to easily manipulate the control of takeoff and landing for robotic aerial vehicles [6]. Lin and Xu presented in 2016 the 3D dynamics model of Quadrotor robots using Newton’s second law and Euler equations [7]. Lopes et al. applied in 2018 reinforcement learning methods to optimize a Stochastic control policy (during training), in order to perform the position control of the ”model-free” Quadrotor [8].

The Quadrotor system dynamics is nonlinear, multivariable, coupled and unstable, suffers from parameter uncertainties and external disturbances. Hence, controlling of Quadrotor is on demand to meet the stability, robustness, and desired dynamic properties. Furthermore, to overcome the hindrance of nonlinearity and to have a system that is pliant to changing parameters and environmental disturbances.

In this work, the suggested control system is composed of two control structures: the first controller is located in the inner loop for angular position control using the Backstepping technique. The second controller located in the outer loop is used to control the Quadrotor trajectories. It is composite of 3 PID controllers. Both controllers’ gains and parameters are optimized using PSO algorithm. PSO is used off-line to optimally adjust the gains of the PID controllers and the parameters of backstepping controller for Quadrotor system. The novelty of this work involved in the performance index to deal with the errors in X,Y, Z positions and the errors in angular positions, this technique has not been developed with the backstepping method in aviation applications. Computer simulation is guiding the trajectory tracking performance of the 3D translation and yaw routes. Using backstepping instead of other control methodology is due to its design flexibility and the recursive use of Lyapunov functions [4]. Therefore; the stability of the recursive control strategy is guaranteed.

2. Quadrotor system modeling

Figure 1 shows the simplified Quadrotor configuration frame. The circular arrows indicate the direction of rotation of each rotor. \(\Omega_1, \Omega_2, \Omega_3,\) and \(\Omega_4\) represent the speed of four propellers, \(M\) is the mass of the Quadrotor, and \(g\) is the acceleration constant.

The following assumptions are presented to conciliate the Quadrotor model [9]:
I. The Quadrotor is considered as a rigid body and it can be derived using Newton- Euler formula.
II. Ignoring aerodynamic effects at low speed for simplicity.
III. The rotor dynamics is of a high speed thus it considered negligible.
IV. The center of Mass for the Quadrotor is on the body fixed frame origin as a coincidence.

![Figure 1. Simplified Quadrotor configuration frame.](image)

The Quadrotor dynamics is derived from the Newton-Euler approach. The translational dynamic equation and the rotational ones can be derived depending on the earth fixed frame and Body fixed frame respectively. Taking into consideration that: increasing or decreasing all the propeller speed at once will produce a throttle command, depending on the first assumption. In order to produce a pitch command; the rear propeller speed should be increasing or decreasing at the same time, that the speed
of the front ones should do the opposite. Regarding the roll movement, this opposite reaction should be done between the two pairs of the left and right propellers. Finally, the yaw movement is executed by increasing or decreasing the front-rear propeller’s speed and the action should be the antipode regarding the left-right propeller’s speed [10]. The dynamic model is derived using Euler-Lagrange Affirmation and the system model in terms of position (X, Y, Z) and rotation (\(\Theta, \Phi, \Psi\)) is shown below [10-11]:

\[
\begin{align*}
\ddot{\Phi} &= \left(\frac{IYY-IZZ}{IXX-IZX}\right) \dot{\Theta} + \frac{JTP}{IXX} \dot{\Theta} + \frac{L}{IXX} u_2 \\
\dot{\Theta} &= \left(\frac{IXX-IZX}{IYY}\right) \dot{\Phi} + \frac{JTP}{IYY} \dot{\Phi} + \frac{L}{IYY} u_3 \\
\dot{\Psi} &= \left(\frac{IZX-IXX}{IZZ}\right) \dot{\Phi} + \frac{JTP}{IZZ} \dot{\Phi} + \frac{L}{IZZ} u_4 \\
\ddot{Z} &= -g + (\cos \Phi \cdot \cos \Theta). \frac{1}{M} u_1
\end{align*}
\]

where \((\Phi, \Theta, \Psi)\) are the roll, pitch and yaw angles respectively w.r.t earth fixed frame. These angles are the gyroscopic influence resulting from the rotation of the rigid body. Moreover, \((\dot{\Phi}, \dot{\Theta}, \dot{\Psi})\) are the angular velocities, \(I_{XX}\) is inertia on X-axis, \(I_{YY}\) is the inertia on Y-axis, \(I_{ZZ}\) is the inertia on Z-axis, \(b\) is the thrust coefficient, \(d\) is the drag coefficient, \(JTP\) is the rotor Inertia, and \(L\) is the arm length.

The input force and torques to the system, \((u_1, u_2, u_3, u_4)\), are defined as [4]:

\[
\begin{align*}
u_1 &= b \left(\Omega_2 + \Omega_3 + \Omega_4^2 + \Omega_5^2\right) \\
u_2 &= b \left(\Omega_4^2 - \Omega_5^2\right) \\
u_3 &= b \left(\Omega_3^2 - \Omega_4^2\right) \\
u_4 &= d \left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right)
\end{align*}
\]

\(W\) is the disturbance and defined as [4]:

\[
W = \Omega_2 + \Omega_3 - \Omega_4 - \Omega_5
\]

(3)

where \(\Omega_1, \Omega_2, \Omega_3, \text{ and } \Omega_4\) are calculated using equation (2).

3. Proposed Control System Design

To control the translational part, X, Y, and Z axes, three PID controllers have been proposed as follows [10]:

\[
\begin{align*}
U_X &= K_{P_X} e_X(t) + K_{I_X} \int e_X(t) \, dt + K_{D_X} \frac{de_X(t)}{dt} \\
U_Y &= K_{P_Y} e_Y(t) + K_{I_Y} \int e_Y(t) \, dt + K_{D_Y} \frac{de_Y(t)}{dt} \\
U_Z &= K_{P_Z} e_Z(t) + K_{I_Z} \int e_Z(t) \, dt + K_{D_Z} \frac{de_Z(t)}{dt}
\end{align*}
\]

(4)

(5)

(6)

where \(U_X, U_Y, U_Z, e_X, e_Y, \text{ and } e_Z\) are the respective controlled signals and tracking error signals for the controlled coordinates X, Y, and Z.

Back stepping approach will emphasis the need to force the system to follow the reference trajectory with error free actions as will be shown in the results, refer to references 5, 11, 12 for more details. For the above mentioned system, rewriting equation (1) in a form of a state space strict feedback form [11]:
\[ \dot{X} = F(X, u) = \begin{pmatrix} X_2 \\ X_4X_6c_1 + X_4c_2W + b_1u_2 \\ X_4 \\ X_2X_6c_3 + X_2c_4W + b_2u_3 \\ X_6 \\ X_4X_6c_5 + b_3u_4 \\ X_8 \\ -g + (\cos X_1\cos X_3)M^{-1}u_1 \\ X_{10} \\ u_6M^{-1}u_1 \\ X_{12} \\ u_7M^{-1}u_1 \end{pmatrix} \] 

(7)

where \( X^T = [\dot{\theta} \dot{\varphi} \dot{\psi} Z \dot{X} \dot{Y}] \) assuming \( X_1 = \dot{\theta}, X_2 = \ddot{\dot{\dot{\theta}}}, X_3 = \theta, X_4 = \dot{\theta}, X_5 = \varphi, X_6 = \dot{X}_5 = \dot{\phi} X_7 = \dot{Z}, X_8 = \dot{X}_7 = \dot{Z}, X_9 = X, X_{10} = \dot{X}, X_{11} = Y, X_{12} = X_{11} = \dot{Y} \).

From equation (1) and equation (2), we get [5]:

(8)

, where \( c_1 \frac{\ddot{X}Y - \ddot{X}Z}{\dot{X}X}, c_2 = \frac{\ddot{X}P}{\dot{X}X}, c_3 = \frac{i\ddot{X}Z - i\ddot{X}X}{i\dot{X}Y}, c_4 = -\frac{\ddot{X}P}{i\dot{X}Y}, c_5 = \frac{i\ddot{X}X - i\ddot{X}Y}{i\ddot{X}Z} \), and \( b_1 = \frac{L}{\dot{X}X}, b_2 = \frac{L}{i\dot{Y}Y} \),

(9)

, and \( b_3 = \frac{L}{i\ddot{X}Z} \), \( u_x = \cos X_1, \sin X_3, \cos X_5 + \sin X_4, \sin X_5 \) and \( u_y = \cos X_1 \sin X_3 \sin X_5 - \sin X_4 \cos X_5 \).

Then, define the tracking error [10]:

\[ e_1 = X_{1d} - X_1 \]

(10)

, where \( X_{1d} \) is the desired value of state \( X_1 \). Applying Lyapunov terms of stability will produce the following:

\[ V(e_1) = \frac{1}{2}e_1^2 \]

(11)

\[ \dot{V}(e_1) = e_1 \dot{e}_1 = e_1(X_{1d} - X_2) \]

(12)

, where \( X_{1d} \) is the derivative of state \( X_1 \). Introducing a virtual control input \( X_2 \):

\[ X_2 = X_{1d} + k_1 e_1 \quad k_1 > 0 \quad \text{(Constant)} \]

(13)

Then equation (10) will become equation (12) for the sake of keeping the derivative of \( V \) as negative semi definite:

\[ \dot{V}(e_1) = -k_1 e_1^2 \]

To proceed:

\[ e_2 = X_2 - X_{1d} - k_1 e_1 \]

(14)

The same procedure will be repeated as follows:

\[ V(e_1, e_2) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \]

(15)

, and the Lyapunov approved time derivative will be as follows:

\[ \dot{V}(e_1, e_2) = -k_1 e_1^2 + e_2 \left( -e_1 + X_2 - X_{1d} + k_1 (e_2 + k_1 e_1) \right) \]

(16)

Since \( X_2 = X_4 X_6 c_1 + X_4 c_2 W + b_1 u_2 \) and from equation (15), then:

\[ \dot{V}(e_1, e_2) = -k_1 e_1^2 + e_2 \left( -e_1 + X_4 X_6 c_1 + X_4 c_2 W + b_1 u_2 - X_{1d} + k_1 (e_2 + k_1 e_1) \right) \]

(17)

Then satisfying Lyapunov stability theorem: \( \dot{V}(e_1, e_2) < 0 \) then:

\[ u_2 = \frac{1}{b_1} (-k_1 e_2 + e_1 - X_4 X_6 c_1 - X_4 c_2 W - k_1 e_2 - k_1^2 e_1) \]

(18)

, hence:

\[ \dot{V}(e_1, e_2) = -k_1 e_1^2 - k_2 e_2 \]

(19)

, and the derivation of \( u_3 \) and \( u_4 \) are concluded by the same criteria:

\[ u_3 = \frac{1}{b_2} (e_3 - c X_6 X_6 - c_4 X_2 W - k_3 (e_4 + k_3 e_3) - k_4 e_4) \]

(20)

\[ u_4 = \frac{1}{b_3} (e_5 - c_5 x_2 X_4 - k_5 (e_6 + k_5 e_5) - k_6 e_6) \]
Taking into consideration that the trajectory movement on the X, Y axes is depending on \( u_1 \), then the roll and pitch angles in equation (7) can be excluded to determine \( u_x \) and \( u_y \):

\[
\begin{align*}
    u_x &= \frac{M}{\cos \alpha \cos \gamma} (e_7 + g - k_7 (e_8 + k_7 e_7) - k_6 e_8) \\
    u_y &= \frac{M}{u_1} (e_9 - k_9(e_{10} + k_9 e_9) - k_{10} e_{10}) \\
\end{align*}
\]

where:

\[ e_3 = X_{3d} - X_3 \]
\[ e_4 = X_4 - X_3d - k_3 e_3 \]
\[ e_5 = X_5d - X_5 \]
\[ e_6 = X_6 - X_5d - k_5 e_5 \]
\[ e_7 = X_{7d} - X_7 \]
\[ e_8 = X_8 - X_{7d} - k_7 e_7 \]
\[ e_9 = X_9d - X_9 \]
\[ e_{10} = X_{10} - X_9d - k_9 e_9 \]
\[ e_{11} = X_{11d} - X_{11} \]
\[ e_{12} = X_{12} - X_{11d} - k_{11} e_{11} \]

The reference attitude is necessary as an input variable to the backstepping control equations. The pitch attitude is calculated as [9]:

\[ \theta = \arctan \left( \frac{u_1 \sin \phi - u_2 \cos \phi}{u_3 + \theta} \right) \]

and the roll attitude is calculated as [9]

\[ \varphi = \arcsin \left( \frac{u_1 \cos \varphi + u_2 \sin \varphi}{\sqrt{u_1^2 + u_2^2 + (u_3 + \varphi)^2}} \right) \]

4. Particle Swarm Optimization (PSO) Technique

Bird flocks were the key idea for the PSO to take place. In PSO, instead of involving more parameters each particle is dealing with and adapting its own velocity and position and changing it according to its own trial and its neighbor trial. Two equations are updating whereas these equations are the position and velocity. Their modifications are important in every iteration to reach the optimum goal [4]. The updated positions and velocities are in a vector form and shown below [4]:

\[
v_{i}^{k+1} = Q, v_{i}^{k} + C_1, \text{rand.}(p_{\text{best}} \cdot X_{i}^{k}) + C_2, \text{rand.}(g_{\text{best}} - X_{i}^{k}) \]
\]

where \( \text{rand} \) is a distributed random variable that can take any value between 0 and 1. \( v_{i}^{k} \) is the current velocity at iteration \( k \), \( v_{i}^{k+1} \) is the future velocity of the particle at iteration \( k \), \( C_1 \) and \( C_2 \) are the cognition parameter and the social parameter respectively, \( p_{\text{best}} \) is the best population that the particle is a part of, and \( g_{\text{best}} \) is the global best particle \( X_p \) is the position of the particle \( i \). This initialization allows the particles to be in random places all over the swarm space.

The velocity vector is utilized to update the position of each particle as shown in equation (25) [4]:

\[ X_{i}^{k+1} = X_{i}^{k} + v_{i}^{k+1} \]

In order to provide the required equilibrium between the reconnaissance and the utilization process, the inertia weight \( Q \) determines the rate between the current and previous particle’s velocity [14]:

\[ Q = 0.5 + \frac{\text{rand}(t)}{2} \]

In this paper, an Integral Time Absolute Error (ITAE) is involved to monitor the performance of the proposed control system (performance index), to minimize the occurred errors of the translational and rotational parameters. The ITAE is expressed as follows [6]:

\[ \text{ITAE}_i = \int_0^T t |e_i(t)| \, dt \]
Where \( i \) is the index for \( X, Y, Z, \phi, \theta, \) and \( \varphi \). Thus, the total performance index is:

\[
ITAE_{\text{total}} = ITAE_X + ITAE_Y + ITAE_Z + ITAE_\phi + ITAE_\theta + ITAE_\varphi
\]  

\((29)\)

5. Simulation Results

The proposed control system behavior is examined and implemented using Matlab/Simulink environment and files, see figure 2. A mini helicopter consists of four rotors in cross configuration known as OS4. The design methodology was developed at Autonomous Systems Laboratory (ASL) [2].

![Figure 2. Proposed Quadrotor controlled system.](image)

The following constructive parameters are used for the OS4 Quadrotor and listed in table 1.

### Table 1. Constructive parameters of OS4 Quadrotor [2].

| Parameter                  | Value         |
|----------------------------|---------------|
| Mass (M)                   | 0.65 (kg)     |
| Inertia on X-axis (IXX)    | 7.5*10^{-3} (kg m^2) |
| Inertia on Y-axis (IYY)    | 7.5*10^{-3} (kg m^2) |
| Inertia on Z-axis (IZZ)    | 1.3*10^{-2} (kg m^2) |
| Thrust coefficient(b)      | 3.13*10^{-5} (N sec^2) |
| Drag coefficient(d)        | 7.5*10^{-5} (N m^2) |
| Rotor Inertia(JTP)         | 6*10^{-5} (kg m^2) |
| Arm length (L)             | 0.23 (m)      |
| Acceleration constant (g)  | 9.81 (m sec^{-2}) |

The training using PSO is done using different trajectories with total time of simulation 25 seconds and sampling time of 1 msec. The parameters of the Weighted PSO are listed in table 2.
The MATLAB/Simulink of the complete controlled system for Quadrotor.

**Table 2. Parameters of Weighted PSO algorithm.**

| Parameter                  | Value |
|----------------------------|-------|
| Swarm Size                 | 40    |
| No. of Particles           | 21    |
| Maximum No. of Iterations  | 50    |
| C1, C2                     | 2     |

The ITAE reaches the minimum value of 124.6 after 48 iterations. Figure 4 shows the performance of the ITAE for PSO Algorithm.

![Performance of ITAE of PSO algorithm](image)

The optimal parameters of both controllers are presented in table 3. Initial conditions of position (X Y Z)=(0 0 0). The solid line represents the actual response while the dash line represents the reference response.
Table 3. Optimal parameters for both controllers.

| $K_{px}$ | $K_{iy}$ | $K_{dz}$ | $K_{pw}$ | $K_{iw}$ | $K_{pw}$ | $K_{pz}$ |
|--------|--------|--------|--------|--------|--------|--------|
| 0.00001 | 0.00001 | 0.00001 | 0.0243 | 0.3463 | 0.00001 | 1 |
| $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $k_6$ | $k_7$ |
| 1 | 1 | 0.00001 | 0.7067 | 0.00001 | 1 | 0.0288 |
| $k_8$ | $k_9$ | $k_{10}$ | $k_{11}$ | $k_{12}$ |
| 0.9102 | 0.0121 | 0.3757 | 1 | 0.9632 | 1 | 0.9237 |

The following trajectories are applied:

5.1 Reaching a 3D fixed position

The first simulation test is by reaching the 3D position of $(X, Y, Z)=(2,2,2)$ m with Yaw angle of 0° [10]. The initial position of the Quadrotor is $(X, Y, Z)=(0,0,0)$ with initial condition of the three angles $(\Phi, \theta, \varphi)=(\pi/4, \pi/4, \pi/4)$ rad. The simulation results are shown in figures 5-8.

![Figure 5. Trajectory of Quadrotor to reach 3D position of (2, 2, 2) m.](image)

![Figure 6. Responses of position and yaw angle to reach 3D position of (2, 2, 2) m.](image)

![Figure 7. Tracking error in position and yaw angle.](image)

![Figure 8. Responses of roll and pitch angles attitude.](image)
From the result the errors in X, Y and Z positions are zero while the error in Yaw angle is 0.00007 rad.

5.2 Maneuvering in closed circle trajectory

The trajectory considered in this test is shown in figure 9. The Quadrotor flights in straight forward up to a height of 5 m starting from the initial position and yaw angle (X,Y,Z, $\phi$) = (0,0,0, 0). Afterward, the quadrotor moves in a circle counterclockwise with a radius of 5 m. the detailed equations of the reference trajectory shown in reference 7. The simulation results are depicted in figure 9-12.

**Figure 9.** Closed circle trajectory of a Quadrotor.

**Figure 10.** Responses of position and yaw angle.  
**Figure 11.** Tracking Error in position and yaw angle.

**Figure 12.** Responses of roll and pitch angles.
From the result the error in X is -0.00001 m, in Y is 0.0000014 m, in Z is zero, and in Yaw angle is 0.00014 rad.

5.3 Test of controller robustness
Robustness by changing the parameters of mass (M), thrust coefficient (b), and drag coefficient (d) by 50%. The simulation test in section 5.2 is repeated again and the results are shown below:

![Figure 13. Closed circle trajectory of a Quadrotor.](image1)

![Figure 14. Responses of position and yaw angle.](image2)

It can be noticed from the results that the both controllers keep the Quadrotor follow the desired trajectory with small overshoot in Z-direction. The error in X=-0.00001 m, while the error in Y= 0.000001m and the error in Z= 0. Error in Yaw angle is -0.00006 rad.

6. Discussion
By simulating the proposed closed loop controlled systems using two case studies; reaching a 3D fixed position and maneuvering in closed circle trajectory, the Quadrotor reaches the required 3D position and yaw angle or follows the desired trajectory with minimum overshoot and approximately zero error in position, see figures 7 and 11. The yaw, pitch, and row angles moved fast to align the Quadrotor with the reference trajectory, which conclude the consumption of a minimum energy, see figures 8 and 12. This increases the flight time and efficiency of batteries. The robustness of the optimal controller was proved in section 5.3. Simulation results showed the controller efficiency of tracking to reach a reference point or trajectory although the changing rates in model parameters and external disturbance effects were up to 50%, see figure 13. An overshoot exists. This exists due to the slight change in reference trajectory.

7. Conclusion
A Quadrotor system which is known for its complexity and nonlinearity has been controlled through its two parts of translation and rotation. Using PID as position control and backstepping as rotation control respectively, both controllers’ gains and parameters have been optimized using PSO algorithm. Backstepping scheme stabilized the attitude of the rotation part as it depends on Lyapunov theory. The performance was proved to track the required reference trajectories with minimum error in position as the performance index (ITAE) plays an effective role in choosing the exacts value that lead the entire system to work in the right path.

From the simulation results, it can be concluded that the system succeeded to follow the required trajectories with high-precision in transient and tracking responses, as compared with results in references 5 and 11. This is due to the optimal selection of PID gains and the parameters of Backstepping controller using PSO algorithm. One more advantage accommodated with such controllers is the preserving of the Quadrotor rotors consumption energy. Simulation results for testing robustness elucidate the controller efficiency of tracking to reach a point and trajectory robustly,
despite the changes in model parameters and external disturbance effects. As a future work, PID controllers can be replaced with intelligent ones to reduce the effects of model uncertainties.

References:
[1] Bouabdallah S and Siegwart R Oct. 29 – Nov. 2 2007 Full control of a quadrotor Proc. of the 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (CA: USA) pp 153-8
[2] Becker M, Coronel R, Sampao B, Bouabdallah S, de Perrot V and Siegwart R Sept. 2012 “In-flight collision avoidance controller based only on OS4 embedded sensors J. of the Braz. Soc.of Mech. Sci. and Eng. pp 294-307
[3] Hua M, Hamel T, Morin P and Claude S. Feb. 2013 Introduction to feedback control of underactuated VTOL vehicles IEEE Control Systems Mag. pp 61-75
[4] Ariffanan M, Basri M, Danapalasingam K A and Husain A R 2014 Design and optimization of backstepping controller for an underactuated autonomous quadrotor unmanned aerial vehicle Trans. of Famena XXXVIII-3 27 pp 27-44
[5] Swarup A and Sudhir 22-24 Feb. 2014 Comparison of quadrotor performance using backstepping and sliding mode control Proc. of the 2014 Int. Conf. on Circuits, Syst. and Control (Switzerland) pp 79-82
[6] Galvez R L, Dadios E P and Bandala A A 9-12 Dec. 2015 Predicting the motion of quadrotor using neural network 8th IEEE Int. Conf. Humanoid, Nanotechnology, Inf. Technology Communication and Control, Environment and Management (HNICEM) (Philippines)
[7] Lin L C and Xu W C Modeling and adaptive backstepping control for quadrotor robots with blade flapping 2016 Int. J. Mech. Syst. Eng. 2 117
[8] Cano Lopes G C, Ferreira M, SimOes A S and Colombini E L. 2018 Intelligent control of a quadrotor with proximal policy optimization reinforcement learning 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE) (Brazil)
[9] Zuo Z. 2010 Trajectory tracking control design with command-filtered compensation for a quadrotor IET Control Theory Appl. 4 11 pp 2343–55
[10] Bresciani T Oct. 2008 Modelling, Identification and Control of a Quadrotor Helicopter (Master thesis: Department of Automatic Control Lund University, Sweden)
[11] Bouabdallah S and Siegwart R April 2005 Backstepping and sliding-mode techniques applied to an indoor micro quadrotor Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation (Barcelona: Spain) pp 2259-64
[12] Freeman R A and Kokotović P V 1992 Backstepping design of robust controllers for a class of nonlinear systems IFAC Nonlinear Control Systems Design (France) pp 431-6
[13] Bansal J C, Singh P K, Saraswat M, Verma A, Jadon S S and Abraham A Oct. 19-21 2011 Inertia weight strategies in particle swarm optimization 2011 Third World Cong. on Nature and Biologically Inspired Computing (Salamanca: Spain) pp 640-47