Praise Worthy Prize
International Review of Aerospace Engineering (I.RE.AS.E), Vol. 12, N. 6
ISSN 1973-7459 December 2019

Quadrotor Tracking Control Based on Optimized Fuzzy Logic Controller

Iswanto, Faaris Mujaahid, Rohmansyah, Toha Ardi Nugraha, Vineet Shekher

Abstract – *A quadrotor is required by the military to conduct surveillance and monitor in the border and coastal areas of Indonesia. To do so, it requires a tracking path. The widely used tracking control to follow the path is fuzzy logic controller algorithm. This algorithm is applied to control the quadrotor when it follows the trajectory. The problem is that it takes a long rise time; moreover, there is an error on the steady time which generates overshoot. The error member set of the fuzzy logic controller was optimised to obtain more optimum results. The proposed algorithm showed that the quadrotor moved on a predefined trajectory in 1.5 seconds rise time, and 1.9 seconds steady-state time, and there was no overshoot. The problem is that it takes long rise time, generates Error on the steady-state time and there is overshoot. The research aims to accelerate the quadrotor, remove the Error on the steady-state time and reduce overshoot. It optimises the Error member set of the fuzzy logic controller to obtain a more optimum result. The research result showed that the quadrotor moved on a predefined trajectory in 1.5 seconds rise time. The steady-state time of the quadrotor was reached at the 1.9th seconds. The proposed algorithm enables the quadrotor to perform faster rise time, remove Error on the steady time and generate minute overshoot that is 0.1.*

Keywords: *Tracking Control, Quadrotor, Fuzzy Logic Controller*

Nomenclature

T	Quadrotor thrust
τ_x	Thrust to the x-axis
τ_y	Thrust to the y-axis
τ_z	z-axis rotation force
K_f	Aerodynamic forces constant
K_M	Moment constant
Ω	Rotor angular velocity
M_x	Moment of the x-axis
M_{1-4}	Moment of motor one until four
${}^W R_B$	Rotational matrix
$C2$	Cos roll $(\cos\varphi)$
$C1$	Cos Pitch $(\cos\theta)$
$C3$	Cos Yaw $(\cos\psi)$
$S1$	Sin roll $(\sin\varphi)$
$S2$	Sin Pitch $(\sin\theta)$
$S3$	Sin Yaw $(\sin\psi)$
F	Force
F_g	Gravity force
F_{thrust}	Thrust Force
$\dot{\varphi}$	Roll angular velocity
$\dot{\theta}$	Pitch angular velocity
$\dot{\psi}$	Yaw Angular velocity
$\ddot{\varphi}$	Roll acceleration
$\ddot{\theta}$	Pitch Acceleration
$\ddot{\psi}$	Yaw Acceleration

I. Introduction

Tracking control is a method used to direct the robot to follow the existing path [1]. Several previous researchers have investigated tracking control to follow the solar path using fuzzy logic controller algorithm such as conducted by El-Khateb [2]. By following the path, solar panels can capture solar thermal energy more optimally. The research on wind trajectory was conducted by Mahdavian et al. using Fuzzy logic controller [3]. The optimum electrical energy was yielded by controlling the direction to the wind source powered by the wind power generator, The research on the fuzzy control was conducted by Karray and Feki [4] to control the robot to move on a predetermined path. Research on controlling welding robots to follow the path was carried out by Chen et al. [5]. PD control and vision navigation sensors were used to manage the welding robot to follow the specified path. The research on controlling moving robots to follow the dynamic pathway was studied by Mehrjerdi and Saad [6] by using Full exponential sliding mode. The study of controlling position tracking in the electro-hydraulic servo system was investigated by

 https://doi.org/10.15866/irease.v12i6.16666

Ghazali et al. [7] by using Full sliding mode. Song et al. [8] investigated underwater robot controlled by comparing PD feedback controller to the PID feedback controller. Tracking control is necessary for rotary-wing type of UAVs such as quadrotor to monitor the borders and conduct surveillance in a disaster area. Several earlier researchers have conducted the study about quadrotor tracking control.

Mu Huang et al. [9] studied the adaptive tracking control with backstepping control. Xie Heng et al. [10] investigated quadrotor full tracking using LQR algorithm to follow the predetermined path. A nonlinear tracking control applied to the quadrotor was investigated by Cunha et al. [11]. By using a nonlinear adaptive state feedback control, the quadrotor could follow the predetermined path. The underactuated control H∞ used for tracking control was investigated by Raffo et al. [12] to track the predetermined path. Geometry control researched by Lee et al. aimed for quadrotor tracking control [13]. With this control, quadrotor was able to follow the predetermined path. A nonlinear control was studied by Rodriguez-Rodriguez and Rodriguez-Cortes to control the quadrotor to follow the predetermined path [14]. Based on the previously mentioned researches, it can be concluded that tracking control has been applied to the quadrotor. The proposed tracking control described in this paper optimises the Error member set of the fuzzy logic control algorithm to obtain faster rise time, steady time, and minimise overshoot. The purpose of the research is to enable the quadrotor to perform faster rise time, remove the Error in the steady-time, and minimise the overshoot by optimising the Error member set of the algorithm.

Section I presents the research that has been conducted by previous researchers on quadrotor control.

Based on the previously mentioned researches, tracking control has been applied to the quadrotor. The widely used tracking control to follow the path is fuzzy logic controller algorithm.

Section II describes quadrotor modelling to determine the altitude and attitude control of the quadrotor to control the quadrotor in the trajectory.

Section III explains the proposed quadrotor tracking control. The proposed tracking control described in this section optimises the Error member set of the fuzzy logic control algorithm to obtain faster rise time, steady time, and minimise overshoot. Section IV explains about fuzzy control optimisation by changing the values of the member set to obtain a more optimum result. Section V analyses the experiments in implementing the fuzzy logic controller by comparing it to the optimised fuzzy logic controller, and Section VI presents the conclusion of the research.

II. Quadrator Model Mathematics

Like a helicopter, quadrotor is one of the UAV types with rotary-wing. The difference between quadrotor and helicopter is in the control system. The control system of the helicopter controls one motor, while in the quadrotor four motors.

The model of the quadrotor based on Corke model [15] is shown in Fig. 1. It shows that G is the global frame with (x_G, y_G, z_G) coordinates used as a non-moving reference frame of the system. Meanwhile, B is the quadrotor frame with (x, y, z) coordinates which has one rotor at each end of the two crossed arms. Each arm consists of a pair of rotors rotating in the same direction.

As the frame of the quadrotor B rotates around x_G, y_G, and z_G axes, there are three Euler angular transformations.

The B rotating around z_G the axis (yaw) has a Euler angular rotation transformation $R_{z_G}(\psi)$, the B rotating around x_G axis (pitch) has a Euler angular rotation transformation $R_{x_G}(\theta)$, and the B rotating around y_G axis (roll) has a Euler angular rotation transformation $R_{y_G}(\varphi)$. The XYZ rotation matrix is defined as:

$$R = \begin{bmatrix} C1C3 & S2S1C3 - C2S3 & C2S1C3 + S2S3 \\ C1S3 & S2S1S3 + C2C3 & C2S1S3 - S2C3 \\ -S1 & S2C1 & C2C1 \end{bmatrix} \quad (1)$$

and the translational matrix obtained is:

$$W = \begin{pmatrix} 1 & 0 & S1 \\ 0 & C2 & -S2C1 \\ 0 & S2 & C1C1 \end{pmatrix} \quad (2)$$

with $C1 = \cos\theta$, $C3 = \cos\psi$, $C2 = \cos\varphi$, $S3 = \sin\psi$, $S1 = \sin\theta$, and $S2 = \sin\varphi$. According to Newton's second law of translational motion by Bouabdallah et al. [16] the equation obtained is:

$$F = m\dot{v} + (\omega \times mv) \quad (3)$$

where $\omega = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$, and $v = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$.

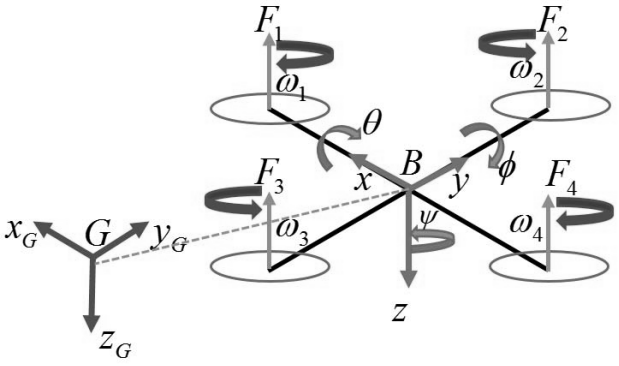

Fig. 1. Quadrotor Modeling

The force acting on the Quadrotor obtained from Fig. 1 is written in the following equation:

$$F = F_g - F_{thrust}$$
$$F = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (4\text{-a})$$

By substituting Newton's second law in equation (3) with force acting on the quadrotor in equation (4-a), the following equation is obtained:

$$m\dot{v} + (\omega \times mv) = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (4\text{-b})$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \left(\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times m \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (5)$$

where T is a Quadrotor vertical thrust against gravity that has the following equation:

$$T = b\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \quad (6)$$

Therefore equation (5) can be expressed as follows:

$$\dot{v} = \begin{bmatrix} mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - R \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} x \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (7)$$

where m is the Quadrotor mass and R is the rotational matrix of the body frame with an inertial frame. The following equation is obtained by substituting XZY rotation matrix of equation (1) into equation (7):

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -T\left(C2S1C3 + S2S3\right) \\ -T\left(C2S1S3 + S2C3\right) \\ mg - T\left(C2C1\right) \end{bmatrix} - \begin{bmatrix} \left(\dot{\theta}\dot{z} - \dot{\psi}\dot{y}\right) \\ \left(\dot{\theta}\dot{x} - \dot{\varphi}\dot{z}\right) \\ \left(\dot{\varphi}\dot{y} - \dot{\theta}\dot{x}\right) \end{bmatrix} \quad (8)$$

Therefore, the acceleration of x, y and z-axes are formulated as follows:

$$\ddot{x} = -\frac{1}{m}T\left(C2S1C3 + S1S3\right) - \left(\dot{\theta}\dot{z} - \dot{\psi}\dot{y}\right) \quad (9)$$

$$\ddot{y} = -\frac{1}{m}T\left(C2S1S3 - S2C3\right) - \left(\dot{\theta}\dot{x} - \dot{\varphi}\dot{z}\right) \quad (10)$$

$$\ddot{z} = mg - T\left(C2C1\right) - \left(\dot{\varphi}\dot{y} - \dot{\theta}\dot{x}\right) \quad (11)$$

The following equation is obtained by using rigid-body rotation law:

$$\Lambda = I\dot{\omega} + \left(\omega \times I\omega\right) \quad (12)$$

where, I is the moment inertia of the Quadrotor:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (13)$$

and:

$$\Lambda = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \end{bmatrix} \quad (14)$$

where:

$$\tau_x = db\left(\omega_4^2 - \omega_2^2\right) \quad (15)$$

$$\tau_y = db\left(\omega_1^2 - \omega_3^2\right) \quad (16)$$

$$\tau_z = k\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \quad (17)$$

Then, equation (12) can be written as:

$$I\dot{\omega} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \left[\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_x\dot{\varphi} \\ I_y\dot{\theta} \\ I_z\dot{\psi} \end{bmatrix} \right] \quad (18)$$

Thus, the equations for the angular acceleration of the Quadrotor obtained are:

$$\ddot{\varphi} = \frac{db}{I_x}\left(\omega_4^2 - \omega_2^2\right) - \frac{I_z - I_y}{I_x}\dot{\theta}\dot{\psi} \quad (19)$$

$$\ddot{\theta} = \frac{db}{I_y}\left(\omega_1^2 - \omega_3^2\right) - \frac{I_x - I_z}{I_y}\dot{\varphi}\dot{\psi} \quad (20)$$

$$\ddot{\psi} = \frac{k}{I_x}\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) - \frac{I_y - I_x}{I_z}\dot{\varphi}\dot{\theta} \quad (21)$$

The roll, pitch and yaw (RPY) of equation (2) are the functions of the angular velocity obtained by using Wronskian matrix inverse and denoted as:

$$W^{-1} = \frac{1}{C1}\begin{bmatrix} C1 & S2S1 & C2S1 \\ 0 & C2C1 & -S2C1 \\ 0 & S2 & C1 \end{bmatrix} \quad (22)$$

The relationship between RPY acceleration and angular velocity can be seen in the following equation:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W^{-1} \begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{23}$$

Therefore, the speeds of roll, pitch and yaw are:

$$\dot{\varphi} = \ddot{\varphi} + S2C1\ddot{\phi} + C2S1\ddot{\psi} \tag{24}$$

$$\dot{\theta} = C2\ddot{\theta} + S2\ddot{\psi} \tag{25}$$

$$\dot{\psi} = \frac{S2}{C2}\ddot{\theta} + \frac{C1}{C2}\ddot{\psi} \tag{26}$$

The whole nonlinear model then can be written as follows [17]-[20], [26], [27]:

$$\dot{x}_1 = \dot{x} = x_7 \tag{27}$$

$$\dot{x}_2 = \dot{y} = x_8 \tag{28}$$

$$\dot{x}_3 = \dot{z} = x_9 \tag{29}$$

$$\begin{aligned} \dot{x}_4 = \dot{\varphi} = x_{10} + \left(\sin x_4\right)\left(\tan x_5\right)x_{11} + \\ + \left(\cos x_4\right)\left(\tan x_5\right)x_{12} \end{aligned} \tag{30}$$

$$\dot{x}_5 = \dot{\theta} = \cos x_4 x_{11} + \sin x_4 x_{12} \tag{31}$$

$$\dot{x}_6 = \dot{\psi} = \frac{\cos x_4}{\sin x_5}x_{11} + \frac{\cos x_4}{\cos x_5}x_{12} \tag{32}$$

$$\dot{x}_7 = \ddot{x} = -\frac{1}{m}T\begin{pmatrix} \cos x_4 \sin x_5 \cos x_6 + \\ + \sin x_4 \sin x_6 \end{pmatrix} \tag{33}$$

$$\dot{x}_8 = \ddot{y} = -\frac{1}{m}T\begin{pmatrix} \cos x_4 \sin x_5 \cos x_6 + \\ - \sin x_4 \sin x_6 \end{pmatrix} \tag{34}$$

$$\dot{x}_9 = \ddot{z} = g - \frac{1}{m}T\left(\cos x_4 \cos x_5\right) \tag{35}$$

$$\dot{x}_{10} = \ddot{\varphi} = \frac{db}{I_x}\left(u_4 - u_2\right) - \frac{I_z - I_y}{I_x}x_5 x_6 \tag{36}$$

$$\dot{x}_{11} = \ddot{\theta} = \frac{db}{I_y}\left(u_1 - u_3\right) - \frac{I_x - I_z}{I_y}x_4 x_6 \tag{37}$$

$$\dot{x}_{12} = \ddot{\psi} = \frac{k}{I_x}\left(u_1 - u_2 + u_3 - u_4\right) - \frac{I_y - I_x}{I_z}x_4 x_5 \tag{38}$$

III. Proposed Control

Fig. 2 [15] shows that the construction of the control system designed based on the state-space model of P. Corke consists of an outer loop and inner loop. The input of the outer loop is the cross position on the x-axis and y-axis of the quadrotor. The outputs of outer loop are roll and pitch angles that serve to drive the quadrotor to the x and y-axes, whereas the yaw angle serves to direct the quadrotor to the x and y-axes' trajectory.

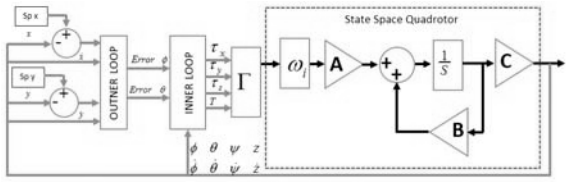

Fig. 2. Inner and outer loop control block diagram on quadrotor

Fig. 2 shows the torque relationship of $\begin{bmatrix} \tau_x & \tau_y & \tau_z & T \end{bmatrix}^T$ which is the output of the inner loop controller converted by using Γ matrix to obtain the rotor speed values of $\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$ on the quadrotor system as shown in the following equation:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{b}{m} & -\frac{b}{m} & -\frac{b}{m} & -\frac{b}{m} \\ 0 & -\frac{db}{I_x} & 0 & \frac{db}{I_x} \\ \frac{db}{I_y} & 0 & -\frac{db}{I_y} & 0 \\ \frac{k}{I_z} & -\frac{k}{I_z} & \frac{k}{I_z} & -\frac{k}{I_z} \end{bmatrix}}_{\Gamma} \cdot \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{39}$$

The control system designed in Fig. 2 aims at moving the quadrotor from the starting point to the goal point.

The system is designed by separating the non-linear quadrotor model into the following four subsystems:

$$SUBSYSTEM\_1 \begin{cases} \dot{x} = x_7 \\ \ddot{x} = -\frac{1}{m}T \cdot \\ \cdot \begin{pmatrix} \cos x_4 \sin x_5 \cos x_6 + \\ + \sin x_4 \sin x_6 \end{pmatrix} \\ \dot{\varphi} = x_{10} + \\ + \left(\sin x_4\right)\left(\tan x_5\right)x_{11} + \\ + \left(\cos x_4\right)\left(\tan x_5\right)x_{12} \\ \ddot{\varphi} = \frac{db}{I_x}\left(u_4 - u_2\right) + \\ - \frac{I_z - I_y}{I_x}x_5 x_6 \end{cases} \tag{40}$$

$$SUBSYSTEM\_2\begin{cases}\dot{y}=x_8\\ \ddot{y}=-\dfrac{1}{m}T\cdot\\ \cdot\begin{pmatrix}\cos x_4\sin x_5\cos x_6+\\ -\sin x_4\sin x_6\end{pmatrix}\\ \dot{\theta}=\cos_{x_4}x_{11}+\sin_{x_4}x_{12}\\ \ddot{\theta}=\dfrac{db}{I_y}(u_1-u_3)+\\ -\dfrac{I_x-I_z}{I_y}x_4x_6\end{cases}\tag{41}$$

$$SUBSYSTEM\_3\begin{cases}\dot{z}=x_9\\ \ddot{z}=g-\dfrac{1}{m}T\left(\cos x_4\cos x_5\right)\end{cases}\tag{42}$$

$$SUBSYSTEM\_4\begin{cases}\dot{\psi}=\dfrac{\cos x_4}{\sin x_5}x_{11}+\dfrac{\cos x_4}{\cos x_5}x_{12}\\ \ddot{\psi}=\dfrac{k}{I_x}\begin{pmatrix}u_1-u_2+\\ +u_3-u_4\end{pmatrix}+\\ -\dfrac{I_y-I_x}{I_z}x_4x_5\end{cases}\tag{43}$$

The stability of the control technique used is based on the linear approach of the subsystem dynamics. The Taylor series approach to the first subsystem of the fourth-order system can be written as:

$$\begin{aligned}\dot{x}&=x_7\\ \ddot{x}&=-\frac{1}{m}T\varphi\\ \dot{\varphi}&=\dot{q}\\ \ddot{q}&=\frac{db\left(u_4-u_2\right)}{I_x}\end{aligned}\tag{44}$$

The Taylor series approach to the second subsystem of the fourth-order system can be written as:

$$\begin{aligned}\dot{y}&=v_y\\ \ddot{y}&=-\frac{1}{m}T\theta\\ \dot{\theta}&=\dot{p}\\ \ddot{p}&=\frac{db\left(u_1-u_3\right)}{I_y}\end{aligned}\tag{45}$$

Taylor's approach to the third subsystem of the fourth-order system can be written as:

$$\begin{aligned}\dot{z}&=v_z\\ \dot{v}_z&=g-\frac{1}{m}T\psi\end{aligned}\tag{46}$$

and Taylor's approach to the fourth subsystem of the fourth-order system can be written as:

$$\begin{aligned}\dot{\psi}&=r\\ \dot{r}&=\frac{k\left(u_1-u_2+u_3-u_4\right)}{I_z}\end{aligned}\tag{47}$$

Equations (44)-(47) of the dynamics system are defined as:

$$\begin{aligned}\dot{x}_1&=A_1x_1+B_1U_1,\dot{x}_2=A_2x_2+B_2U_2\\ \dot{x}_3&=A_3x_3+B_3U_3,\dot{x}_4=A_4x_4+B_4U_4\end{aligned}\tag{48}$$

where the $x_1\cdots x_4$ matrixes are defined as:

$$\begin{aligned}x_1&=\begin{bmatrix}x-x_{ref}\\ v_x\\ \varphi\\ q\end{bmatrix},x_2=\begin{bmatrix}y-y_{ref}\\ v_y\\ \theta\\ p\end{bmatrix}\\ x_3&=\begin{bmatrix}z-z_{ref}\\ v_z\end{bmatrix},x_4=\begin{bmatrix}\psi-\psi_{ref}\\ v_r\end{bmatrix}\end{aligned}\tag{49}$$

where the $A_1\cdots A_4$ matrixes are defined as:

$$\begin{aligned}A_1&=\begin{bmatrix}0&1&0&0\\ 0&0&-\frac{T}{m}&0\\ 0&0&0&1\\ 0&0&0&0\end{bmatrix},A_2=\begin{bmatrix}0&1&0&0\\ 0&0&-\frac{T}{m}&0\\ 0&0&0&1\\ 0&0&0&0\end{bmatrix}\\ A_3&=\begin{bmatrix}1&0\\ 0&g-\frac{T}{m}\end{bmatrix},A_4=\begin{bmatrix}1&0\\ 0&1\end{bmatrix}\end{aligned}\tag{50}$$

where the $u_1\cdots u_4$ matrixes are defined as:

$$u_1=\tau_x,u_2=\tau_y,u_3=\tau_z,u_4=T\tag{51}$$

where the $B_1\cdots B_4$ matrixes are defined as:

$$B_1=\begin{bmatrix}0\\0\\0\\ \frac{1}{I_y}\end{bmatrix},B_2=\begin{bmatrix}0\\0\\0\\ \frac{1}{I_x}\end{bmatrix},B_3=\begin{bmatrix}0\\ \frac{1}{I_z}\end{bmatrix},B_4=\begin{bmatrix}0\\1\end{bmatrix}\tag{52}$$

The followings are the U_1-U_4 Full-state feedback control:

$$U_1=-kx_1,U_2=-kx_2,U_3=-kx_3,U_4=-kx_4\tag{53}$$

The following is the Full-state feedback control for the roll subsystem:

$$\tau_x = \left(k_1\left(x_{ref} - x\right) - k_2 v_x\right) + \left(k_3\theta - k_4 q\right) \quad (54)$$

The following is the Full-state feedback control for the pitch subsystem:

$$\tau_\varphi = \left(k_5\left(y_{ref} - y\right) - k_6 v_y\right) - \left(k_7\varphi + k_8 p\right) \quad (55)$$

The following is the Full-state feedback control for the yaw subsystem:

$$\tau_\psi = k_9\left(\psi_{ref} - \psi\right) - k_{10} r \quad (56)$$

and the following is the Full-state feedback control for the thrust subsystem:

$$T = k_{11}\left(z_{ref} - z\right) - k_{12} v_z \quad (57)$$

Based on the equation (53), it is seen that the input is obtained from the sum of the linear velocity and angular velocity, while the outputs are the roll, pitch, yaw, and thrust forces [21]-[25], [28], [29].

The block diagram of Fuzzy Algorithm Control can be designed in the basis of those equations as illustrated in Fig. 3.

The figure shows that there are four Fuzzy Logic Algorithm controls namely Fuzzy Logic Controller roll (FLCR), Fuzzy Logic Controller pitch (FLCP), Fuzzy Logic Controller yaw (FLCY) and Fuzzy Logic Controller Thrust (FLCT). The block diagram of Fig. 3 shows that the Fuzzy Logic Controller has two inputs, namely the angular velocity and linear velocity and four outputs namely roll, pitch, yaw, and thrust forces.

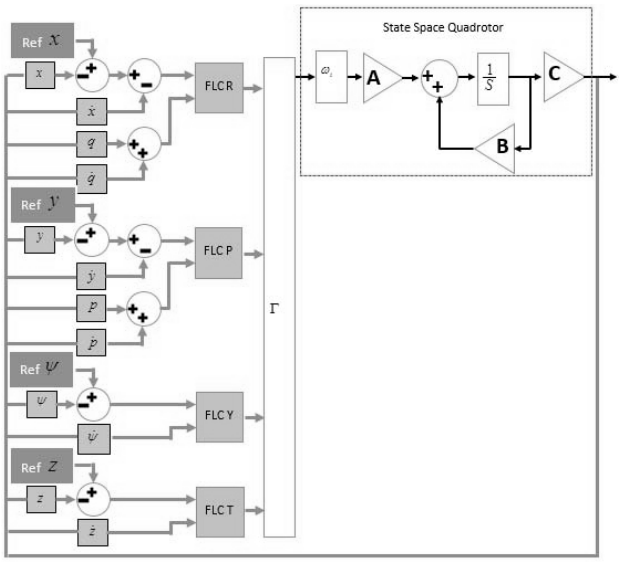

Fig. 3. Block diagram of the inner control and outer loop on the quadrotor

The full control proposed in this research uses Fuzzy Logic Controller Algorithm with rule base 5×5. The design of this fuzzy rule base uses a lockup table using 25 rules to determine the decision. AND operator links all the rules shown in Fig. 4. The input and output of this rule base consist of three membership functions of NS (Negative Small), Z (zero), and PS (Positive Small) as shown in Fig. 5 with the minimum and maximum values of the universe of discourse of roll, pitch, and yaw errors [-0.3 0.3]. Fig. 5 shows that variable Error is divided into three fuzzy set members namely of NS (Negative Small), Z (zero), and PS (Positive Small) the domain of fuzzy set NS is [-0.6 -0.3997 -0.2000 0], Z is [-0.2000, 0, 0.2000], and PS is [0 0.2000 0.3997 0.6]. The minimum value of the universe of discourse of change rate of roll, pitch, yaw, and altitude errors and the maximum value of the universe of discourse of roll, pitch, yaw, and altitude errors are [-1 1] as shown in Fig. 5. The figure shows that there are three membership functions in the error rate of the universe of discourse namely NM (Negative Medium), of NS (Negative Small), Z (zero), and PS (Positive Small), and PM (Positive Medium). Fig. 6 shows that variable Dyaw is divided into three fuzzy set members, namely of NS (Negative Small), Z (zero), and PS (Positive Small). The domain of fuzzy set NS is [[-1.4 -0.9337 -0.4665 0]], Z is [[-0.4665 0 0.4666]], and PS is [[0 0.4666 0.9337 1.4]].

		Error				
		NB	NS	Z	PS	PB
dz	PB	PB	PB	PB	PS	O
	PS	PB	PB	PS	O	NS
	Z	PB	PS	O	NS	NB
	NS	PS	O	NS	NB	NB
	NB	O	NS	NB	NB	NB

Fig. 4. Rule base of attitude and altitude control

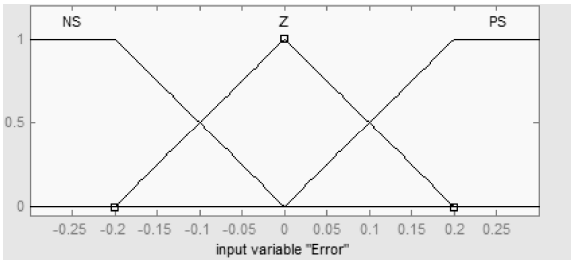

Fig. 5. Input variable 'Linear' design

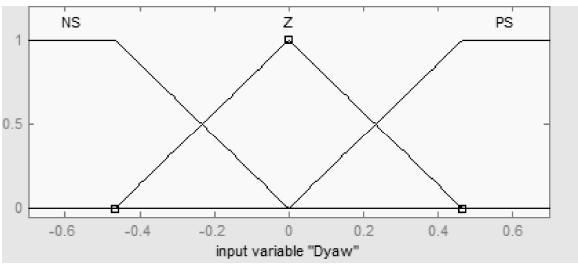

Fig. 6. Input variable 'Angular' design

The design of the output variables of the fuzzy set is shown in Fig. 7 with the range of the universe of discourse of roll, pitch, yaw, and altitude outputs is [-1000 1000] and there are three membership functions namely NS of NS (Negative Small), Z (Zero), and PS (Positive Small). Fig. 7 shows that variable Error is divided into three fuzzy set members, namely NS of NS (Negative Small), Z (zero), and PS (Positive Small). The domain of fuzzy set NS is [-2.727 -1.818 -0.9091 0], Z is [-0.9091 0 0.9091], and PS is [0 0.9091 1.818 2.727]. The universe of discourse of altitude Error has the minimum and maximum values of [-0.3 0.3] and three membership functions consisting of NS (Negative Small), Z (zero), and PS (Positive Small) as shown in Fig. 8. Fig. 8 shows that variable error is divided into three fuzzy set members namely NS (negative small), Z (zero), and PS (positive small) with the domain of NS is [-0.6 -0.3997 -0.2000 0], Z is [-0.2000 0 0.2000], and PS is [0 0.2000 0.3997 0.6]. The minimum value of the universe of discourse of change rate of error of roll, pitch, yaw, and altitude and the maximum value of the universe of discourse of Error of roll, pitch, yaw, and altitude is [-1 1] as shown in Fig. 9. The figure shows that there are three membership functions in the error rate of universe of discourse namely of NS (Negative Small), Z (zero), and PS (Positive Small).

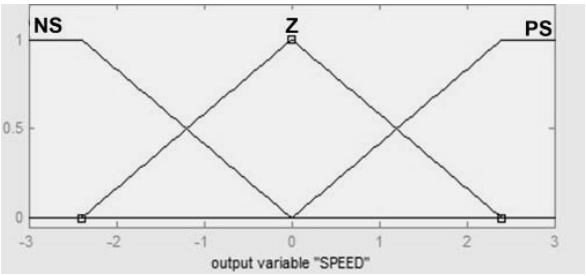

Fig. 7. Output variable 'Speed' design

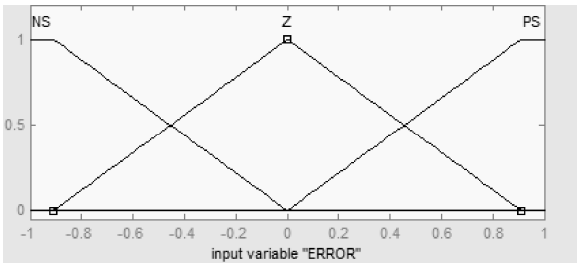

Fig. 8. Input variable 'Error' design

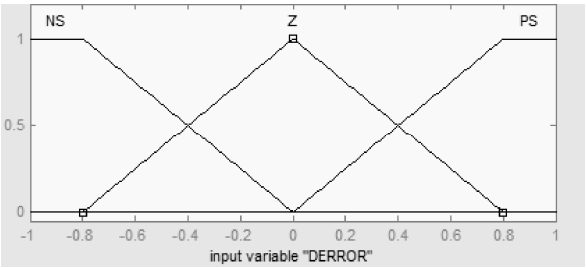

Fig. 9. Input variable 'DError' design

Fig. 9 shows that variable Draw is divided into three fuzzy set members, namely NS (negative small), Z (zero), and PS (positive small). The domain of fuzzy set], NS is [-2.4 -1.6 -0.8 0], Z is [-0.8 0 0.8], PS is [0 0.8 1.6 2.4]. The design of the output variable for the fuzzy set is shown in Fig. 10 with the range of the universe of discourse of roll, pitch, yaw, and altitude outputs of [-1000 1000] and there are three membership functions namely NM NS of NS (Negative Small), Z (zero), and PS (Positive Small). Fig. 10 shows that variable error is divided into three fuzzy set members, namely NS (Negative Small), Z (Zero), and PS (Positive Small). The domain of fuzzy set NS is [[-4000 -2000 -1000 0], Z is [-1000 0 1000], and PS is [0 1000 2000 3000].

IV. Optimize Fuzzy Control

The control systems of the roll, pitch, yaw, and altitude proposed in this paper are optimized by changing the Error set value of the membership range of the Fuzzy Logic Controller algorithm. The first optimisation is to change the altitude control of the membership of the fuzzy error set as shown in Fig. 11. It shows that the fuzzy error set consists of NS (Negative Small), Z (Zero), and PS (Positive Small). The domain value of the Z variable is changed from [-1.5 1.5] to [-0.3 0 0.3]. The change of domain values is intended to obtain a fast steady-state. The values of the PS and NS variables are changed from [-3 -1.5 0] to [-2,727 -1,818 -0.1 0] and [0 1.5 3] to [0 0.1 1,818 2,727] respectively. The second optimisation is to change the attitude, pitch and yaw control memberships of the fuzzy linear set as shown in Fig. 12. It shows that the fuzzy error set has NS (Negative Small), Z (Zero), and PS (Positive Small) variables.

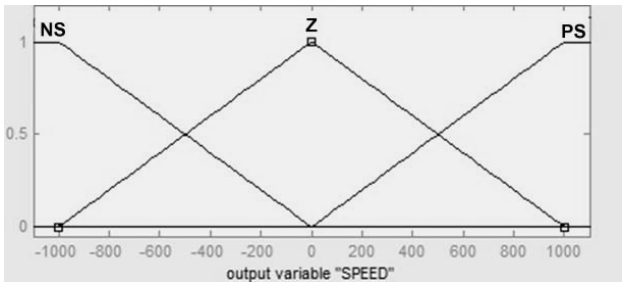

Fig. 10. Output variable 'Speed' design

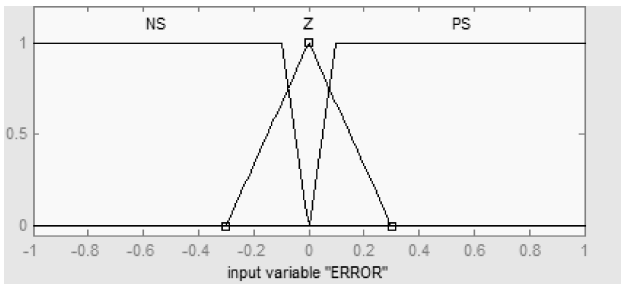

Fig. 11. Fuzzy altitude set membership optimisation

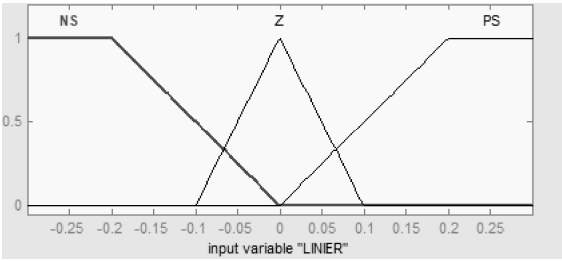

Fig. 12. Fuzzy attitude set membership Optimization

The domain of the Z variable is changed from [-1.5 1.5] to [-0.1 0 0.1]. The change is intended to obtain a fast steady-state. The values of the PS and NS variables are changed from [-3 -1.5 0] to [-3 -0.5 0] and [0 1.5 3] to [0 0.5 3] respectively.

V. Results and Discussion

The Fuzzy Logic Controller Algorithm presented in this paper was simulated by using the Corke (2011) simulator used for the Quadrotor tracking. The specification is shown in Table I. It displays that the weight of the Quadrotor is 4 kg, the distance between the motors with COG is 0.165 m, the thrust constant is $1{,}2953\times10^{-5}$ kg m, the drag force constant is 1.0368×10^{-7} kg m, and the tested gravity acceleration is 9.8 m/s^2. To simulate the attitude and altitude controls, the parameters in the simulator were set, including the initial position of x and y of the Quadrotor. The initial position was set on (-1.0), the desired height was 1 m, and the x and y hover positions of the Quadrotor was (-1.0).

The experiment was carried out without wind disturbance simulation. In the first experiment, the quadrotor was placed on the z-axis at 0.15 m, and the x and y-axes were placed at the starting position of (0,0). The quadrotor was set to reach the hover position at 1 m on the z-axis, and the position of the x and y-axes of (-2, 1) as shown in Fig. 13.

The figure shows that six graphs are consisting of 3 graphs of the x, y, and z-axes for Fuzzy algorithm and three graphs of the x, y, and z-axes for optimised fuzzy algorithms. The rise time and steady time needed are 2 seconds and 1.5 seconds. The settling time is reached at the 3rd second. By using the optimised fuzzy logic controller algorithm, the quadrotor was able to take off quickly that required 1.5 seconds for rising time and 2 seconds for steady state and settling time at the 2nd seconds.

TABLE I
QUADROTOR MODEL PARAMETERS

Parameter	Value	Unit	Remark
G	9,81	m/s^2	Gravitational speed
M	4,34	kg	A.R. Drone 2 Mass
D	0,165	m	Distance from the rotor to COG
B	$1{,}2953\times10^{-5}$	kg m	Thrust constant
K	$1{,}0368\times10^{-7}$	kg m	Friction Constant
Ix	0,082	kg m^2	Moment Inertia of x-axis
Iy	0,0845	kg m^2	Moment Inertia of y-axis
Iz	0,1377	kg m^2	Moment Inertia of z-axis

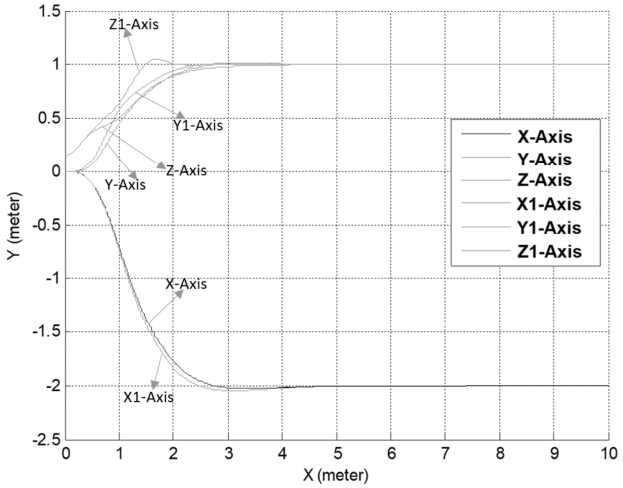

Fig. 13. Comparison of Hover Positions

There was no oscillation when the quadrotor was taking off and hovering, but it was slightly overshot at 0.1. In the second experiment, the quadrotor was placed on the z-axis at 0.15 m, the x and y-axes were at the starting position of (0,0). The quadrotor was set to reach the hover position on the z-axis at 1 m height and on the x and y-axes of (-2, 0) as shown in Fig. 14. This figure shows that the quadrotor moves from point (-2, 0) to (-2, 2). Six graphs are consisting of 3 graphs for the x, y, and z-axes for fuzzy algorithms and three graphs for the x, y, and z-axes for optimised fuzzy algorithms. At the 4th second, the quadrotor moves to the position (-2, 2). The quadrotor was able to take off quickly, and it required 1.5 seconds rise time and 2 seconds steady state at the position of (-2, 0). The rise time and steady time needed are 1.5 seconds and 2 seconds respectively. The settling time is reached at the 3rd second. By using the optimised Fuzzy Logic Controller Algorithm, there was no oscillation when the quadrotor took off and hovered, but it was slightly overshot at 0.1. Moreover, the quadrotor was able to quickly move by requiring 2 seconds rise time, 2.5 seconds steady state, and settling time at the 3rd second. There was no oscillation when the quadrotor took off and hovered. In the third experiment, the quadrotor was placed on z-axis at 0.15 m, the x and y axes were at the starting position of (0, 0). The quadrotor was set to reach the hover position on the z-axis at 2 m height, and at the position of the x and y-axes of (-2, 1) as shown in Fig. 15.

The figure shows that quadrotor is landing at the 5th second. There are six graphs, consisting of 3 graphs of the x, y, and z-axes for fuzzy algorithm and three graphs of the x, y, and z-axes for optimized fuzzy algorithm. The landing position of the quadrotor is (-2, 1). The quadrotor was able to perform a quick landing requiring 3 seconds of landing time, 3.5 seconds steady state. The settling time was at the 4th second at the position of (-2, 0).

By using the optimised Fuzzy Logic Controller Algorithm, there was no oscillation when the quadrotor took off and landed, but it was slightly overshot at 0.1.

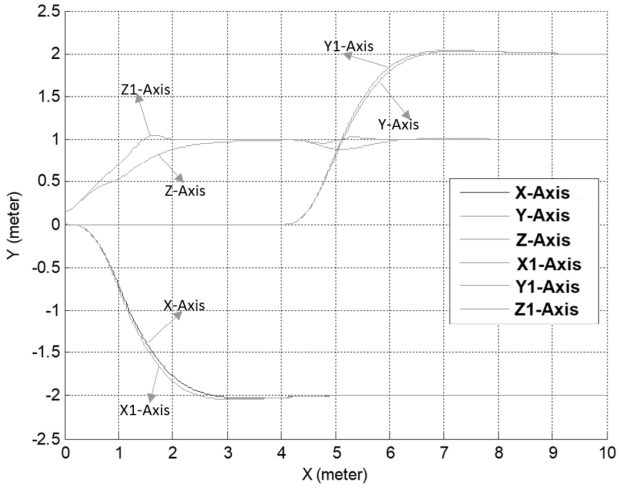

Fig. 14. Comparison of Tracking Positions

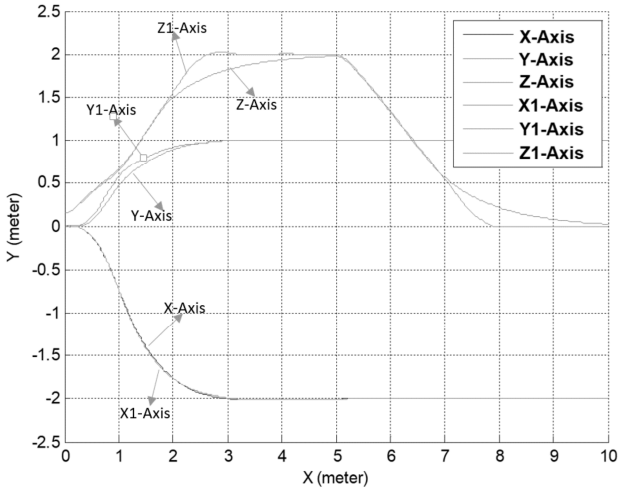

Fig. 15. Comparison Hover and Take Off Positions

VI. Conclusion

The latest development of the quadrotor control using fuzzy logic algorithm is presented. Control is necessary for path planning. Hence, this paper proposes the quadrotor control development using optimised fuzzy logic algorithm by optimising the Error member set of the algorithm to obtain fast rise time, steady time, and remove overshoot.

Three experiments have been conducted to control the quadrotor weighted of 4.34 kilograms at 1 m height. The initial position of the quadrotor is (-1.0), the desired height is 1 m, and the x and y hover position of the quadrotor is (-1.0). The experiments are conducted without wind disturbance. The first experiment is on the quadrotor hover performance, while the second is on the quadrotor hover position and movement. The third experiment is on the quadrotor hover performance, movement, and landing performance. The experiments show that the rise time, steady time, and settling time are faster after applying the optimised fuzzy algorithm. The proposed algorithm shows that the quadrotor moves on a predefined trajectory in 1.5 seconds rise time and 2 seconds steady state time, and there is a slight overshoot at the 0.1st second. For future improvement, it needs the addition of disturbance rejection control to stabilise the quadrotor when there is a wind disturbance to return to its position quickly.

Acknowledgements

This research was supported by Research Project Grant from Institute for Research, Publication and Community Service, Universitas Muhammadiyah Yogyakarta with the contract number: 196/SK-LP3M/I/2019, awarded to A/P Iswanto, S.T., M.Eng.

References

[1] A. El Khateb, N. A. Rahim, J. Selvaraj, and M. N. Uddin, Fuzzy-Logic-Controller-Based SEPIC Converter for Maximum Power Point Tracking, *IEEE Trans. Ind. Appl.*, vol. 50, no. 4, pp. 2349–2358, Jul. 2014.

[2] J. Aymen, Z. Ons, M. M. Nejib, and A. Craciunescu, Maximum Power Point Tracking of Photovoltaic Modules: Comparison of Fuzzy Logic and Artificial Network Controllers' Performances, in 2016 *Third International Conference on Mathematics and Computers in Sciences and Industry (MCSI)*, 2016, pp. 89–93.

[3] M. Mahdavian, N. Wattanapongsakorn, G. Shahgholian, S. H. Mozafarpoor, M. Janghorbani, and S. M. Shariatmadar, Maximum power point tracking in wind energy conversion systems using tracking control system based on fuzzy controller, in 2014 *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1–5.

[4] A. Karray and M. Feki, Tracking Control of a mobile manipulator with fuzzy PD controller, in 2015 *World Congress on Information Technology and Computer Applications (WCITCA)*, 2015, pp. 1–5.

[5] H. Chen, J. Li, G. Xing, J. Xing, and H. Sun, Trajectory tracking control of a Macro-Micro welding robot based on the vision navigation, in 2010 *IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 667–672.

[6] H. Mehrjerdi and M. Saad, Dynamic tracking control of mobile robot using exponential sliding mode, in *IECON* 2010 - *36th Annual Conference on IEEE Industrial Electronics Society*, 2010, pp. 1517–1521.

[7] R. Ghazali, Y. M. Sam, M. F. Rahmat, A. W. I. M. Hashim, and Zulfatman, Position tracking control of an electro-hydraulic servo system using sliding mode control, in 2010 *IEEE Student Conference on Research and Development (SCOReD)*, 2010, no. SCOReD, pp. 240–245.

[8] S. Song, Y. Shan, K. J. Kim, and K. K. Leang, Tracking control of oscillatory motion in IPMC actuators for underwater applications, in 2010 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2010, pp. 169–174.

[9] Mu Huang, Bin Xian, Chen Diao, Kaiyan Yang, and Yu Feng, Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping, in *Proceedings of the 2010 American Control Conference*, 2010, pp. 2076–2081.

[10] Xie Heng, D. Cabecinhas, R. Cunha, C. Silvestre and Xu Qingsong, A trajectory tracking LQR controller for a quadrotor: Design and experimental evaluation, *TENCON* 2015-2015 *IEEE Region 10 Conference*, Macao, 2015, pp. 1-7. doi: 10.1109/TENCON.2015.7372729

[11] R. Cunha, D. Cabecinhas, and C. Silvestre, Nonlinear trajectory tracking control of a quadrotor vehicle, in 2009 *European Control Conference (ECC)*, 2009, pp. 2763–2768.

[12] G. V. Raffo, M. G. Ortega, and F. R. Rubio, An underactuated H∞ control strategy for a quadrotor helicopter, in 2009 *European Control Conference (ECC)*., 2009, pp. 3845–3850.

[13] T. Lee, M. Leoky, and N. H. McClamroch, Geometric tracking

control of a quadrotor UAV on SE(3), in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
[14] M. J. Rodriguez-Rodriguez and H. Rodriguez-Cortes, Nonlinear control for trajectory tracking of a quadrotor unmanned vehicle, in 2011 *8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2011, no. 1, pp. 1–6.
[15] R. Mahony, V. Kumar, and P. Corke, Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor, *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.
[16] S. Bouabdallah, P. Murrieri, and R. Siegwart, Design and control of an indoor micro quadrotor, in *IEEE International Conference on Robotics and Automation*, 2004. Proceedings. ICRA '04. 2004, 2004, pp. 4393-4398 Vol.5.
[17] Iswanto, I., Ar-Drone Navigation Based on Laser Sensor and Potential Field Algorithm, (2018) *International Review of Aerospace Engineering (IREASE)*, 11 (6), pp. 260-267. doi: https://doi.org/10.15866/irease.v11i6.14614
[18] Iswanto, I., Avoiding Local Minima for Path Planning Quadrotor Based on Modified Potential Field, (2018) *International Review of Aerospace Engineering (IREASE)*, 11 (4), pp. 146-154. doi: https://doi.org/10.15866/irease.v11i4.14438
[19] I. Iswanto, O. Wahyunggoro, and A. I. Cahyadi, Formation Pattern Based on Modified Cell Decomposition Algorithm, *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 3, p. 829, Jun. 2017.
[20] Iswanto, I., Ataka, A., Inovan, R., Wahyunggoro, O., Imam Cahyadi, A., Disturbance Rejection for Quadrotor Attitude Control Based on PD and Fuzzy Logic Algorithm, (2016) *International Review of Automatic Control (IREACO)*, 9 (6), pp. 405-412. doi: https://doi.org/10.15866/ireaco.v9i6.9930
[21] Bunjaku, D., Nadzinski, G., Stankovski, M., Stefanovski, J., Dynamic Modeling and Flight Control Design for Multicopter, (2018) *International Review of Aerospace Engineering (IREASE)*, 11 (5), pp. 224-235. doi: https://doi.org/10.15866/irease.v11i5.15512
[22] Krafes, S., Chalh, Z., Saka, A., Visual Servoing of a Spherical Inverted Pendulum on a Quadrotor Using Backstepping Controller, (2018) *International Review of Aerospace Engineering (IREASE)*, 11 (1), pp. 6-14. doi: https://doi.org/10.15866/irease.v11i1.13460
[23] Grillo, C., Montano, F., An EKF Based Method for Path Following in Turbulent Air, (2017) *International Review of Aerospace Engineering (IREASE)*, 10 (1), pp. 1-6. doi: https://doi.org/10.15866/irease.v10i1.10501
[24] Bouallegue, S., Khoud, K., Integral Backstepping Control Prototyping for a Quad Tilt Wing Unmanned Aerial Vehicle, (2016) *International Review of Aerospace Engineering (IREASE)*, 9 (5), pp. 152-161. doi: https://doi.org/10.15866/irease.v9i5.10476
[25] Shouman, M., El Bayoumi, G., Adaptive Robust Control of Satellite Attitude System, (2015) *International Review of Aerospace Engineering (IREASE)*, 8 (1), pp. 35-42. doi: https://doi.org/10.15866/irease.v8i1.5322
[26] Agustinah, T., Isdaryani, F., Nuh, M., Tracking Control of Quadrotor Using Static Output Feedback with Modified Command-Generator Tracker, (2016) *International Review of Automatic Control (IREACO)*, 9 (4), pp. 242-251. doi: https://doi.org/10.15866/ireaco.v9i4.9431
[27] Belhadri, K., Kouadri, B., Zegai, M., Adaptive Neural Control Algorithm Design for Attitude Stabilization of Quadrotor UAV, (2016) *International Review of Automatic Control (IREACO)*, 9 (6), pp. 390-396. doi: https://doi.org/10.15866/ireaco.v9i6.9919
[28] Omar, H., A Geno-Fuzzy Fast Attitude Controller for Satellites Stabilized by Reaction WheelsA Geno-Fuzzy Fast Attitude Controller for Satellites Stabilized by Reaction Wheels, (2018) *International Journal on Engineering Applications (IREA)*, 6 (5), pp. 150-155. doi: https://doi.org/10.15866/irea.v6i5.16628
[29] Grillo, C., Montano, F., Automatic EKF Tuning for UAS Path Following in Turbulent Air, (2018) *International Review of Aerospace Engineering (IREASE)*, 11 (6), pp. 241-246. doi: https://doi.org/10.15866/irease.v11i6.15122

Authors' information

Iswanto was born in Sleman, Yogyakarta, Indonesia, in 1981. He received the Bachelor degree, Master Degree and Doctoral degree of Engineering from Universitas Gadjah Mada, Yogyakarta, Indonesia in 2007, 2009 and 2018 respectively. He has been a Lecturer and Researcher in the Electrical Engineering Department at Universitas Muhammadiyah Yogyakarta since 2010. He has been an Associate professor in the Engineering Department at Universitas Muhammadiyah Yogyakarta. His current research is focused on formation control, path planning and Control UAV.

Faaris Mujaahid was born in Bandung, Indonesia, in 1987. He received his Bachelor degree from Saxion University of Applied Sciences and Master degree from the University of Southampton in 2010 and 2015 respectively. He has been a Lecturer and Researcher in the Electrical Engineering Department at Universitas Muhammadiyah Yogyakarta since 2017. His current research is focused on Solar cell fabrication and electronics automation...

Rohmansyah was born in Sukabumi Jawa Barat, Indonesia. He received the S2 degree from Universitas Islam Negeri Sunan Kalijaga Yogyakarta, Yogyakarta, Indonesia in 2014 and 2009. Now, he is on S3 degree Program at Universitas Islam Negeri Sunan Kalijaga Yogyakarta. He has been a Lecturer Al Islam dan Kemuhammadiyahan (AIK) and Researcher in the Electrical Engineering Department at Universitas Muhammadiyah Yogyakarta since 2014. His current research is focused on Hadis.

Toha Ardi Nugraha received the B.Sc. degree in Telecommunication Eng. from Telkom University, Indonesia, in 2011 and the M.Eng degree in IT Convergence Eng. from Kumoh National Institute of Tech., South Korea, in 2014. He joined at Department of Electrical Eng., Universitas Muhammadiyah Yogyakarta, Indonesia, as a lecturer in 2016. He has worked at R&D Center PT. Telkom Indonesia for three years from 2009 to 2012, as a research assistant. He is currently pursuing his PhD degree at the Department of Telecommunication Eng., Czech Technical University in Prague, Czech Republic. His current research includes wireless and mobile networks, D2D, Li-Fi, and IoT.

Dr. **Vineet Shekher** was born in Chapra, Bihar, in 1978. He received his bachelor degree from North Maharashtra University, Master degree from Delhi College of Engineering and PhD from BIT Sindri, Dhanbad, Jharkhand, India in 1999, 2009 and 2016 respectively. He has been an Associate professor in the Electrical and Electronics Department at Noida Institute of Engineering and Technology, Greater Noida, Uttar Pradesh since 2017. His current research is focused on Controller Design based in Artificial Intelligent Technique.