Altitude Control of Quadrotor using Fuzzy Self Tuning PID Controller

Fahmizal[1], Atikah Surriani[2], Ma'un Budiyanto[3], Muhammad Arrofiq[4]
Department of Electrical and Informatics Engineering,
Vocational College, Universitas Gadjah Mada
Yogyakarta, Indonesia
e-mail: 1)fahmizal@ugm.ac.id 2)atikah.surriani.sie13@ugm.ac.id 3)m.budiyanto@ugm.ac.id 4)rofiq@ugm.ac.id

***Abstract* - This paper presents fuzzy self-tuning PID controller for controlling the altitude of the quadrotor. Although the PID controllers have many advantages, it has a limitation that the main term gains called proportional gain, integration gain, and derivative gain have to be tuned manually. One of a solution to manage its limitation is adding the special feature called self-tuning. So, a fuzzy self-tuning PID controller is proposed to adjust the gain parameter of PID controller. As a result, two control techniques were then developed and synthesized for comparing; a linear PID controller only and fuzzy self-tuning PID controller. A complete simulation was then implemented on MATLAB/Simulink relying on the derived mathematical model of the quadrotor.**

Keywords—Altitude, Quadrotor, Fuzzy Logic, PID Controller.

I. INTRODUCTION

Controlling of unmanned aerial vehicles (UAVs) is a very challenging field of research especially for vertical take-off and landing (VTOL) vehicles such as Quadrotor. Quadrotor is a typical design for small UAV because of its simple structure. Quadrotor is an unmanned aerial VTOL vehicle that is classified among aerial vehicles with rotary wings. This aerial vehicle has four motors whose propulsion thrust is generated by transmission of power to propellers. This vehicle can be controlled and stabilized by altering engine rpm. Quadrotors are used in surveillance [1], search and rescue [2], construction inspections [3], military [4] and several other applications.

State of the art in control quadrotor has changed drastically in the last several years. The number of projects tackling this issue has enough and suddenly increased. The basic dynamical model of the quadrotor is the starting point for all of the studies but more complex aerodynamic properties has been introduced as well [5]-[6]. Different control methods has been researched, including PID controllers [7]-[8]-[9], LQR controllers [10] and nonlinear controllers with nested saturations [11]-[12].

PID controllers have been standard tools in industry due to their practicality. The availability of well-established rules for tuning the parameters of the controller is another reason making them preferable in real-time applications. It has main-term gains that have primary function to take over to the responses of system, especially quadrotor system. PID controllers give the efficient and simplest solution to control the system of quadrotor [13].

Although the PID controllers have many advantages, it has a limitation that the main-term gains called proportional gain, integration gain and derivative gain have to be tuned manually. One of solution to manage its limitation is adding the special feature called self-tuning [14]. Fuzzy logic is the chosen method of self-tuning for the PID controller. M. Santos et al. [15] proposed an intelligent system based on fuzzy logic to control a quadrotor. C. Coza et al. [16] used an adaptive fuzzy control algorithm to control a quadrotor in the presence of sinusoidal wind disturbance. The author addressed a robust method to prevent drift in the fuzzy membership centers.

The purpose of this paper is to present the basics of quadrotor modeling and control as to form a basis for further research and development in the area. This is pursued with two aims. The first aim is to study the mathematical model of the quadrotor dynamics. The second aim is to develop proper methods for altitude control of the quadrotor. This paper is aimed at using Fuzzy Self Tuning PID Controller for controlling the altitude of quadrotor. Two control techniques were then developed and synthesized; a linear PID controller and fuzzy self-tuning PID controller. A complete simulation was then implemented on MATLAB/Simulink relying on the derived mathematical model of the quadrotor. The challenge in controlling a quadrotor is that the quadcopter has six degrees of freedom but there are only four control inputs.

The remainder of this paper is organized as follows. In Section II, the dynamic model and state space model of a quadrotor are proposed. Section III elaborates the control system architectures especially on fuzzy system for tuning PID gains. Section IV presents several simulation results which show the effectiveness and merit of the proposed methods. Section V concludes the paper with remarks and suggestions for future works.

II. MODELING OF QUADROTOR

A. Dynamics Model of Quadrotor

The motion of the quadrotor can be divided into two subsystems; rotational subsystem (roll, pitch and yaw) and translational subsystem (altitude and x and y position). The rotational subsystem is fully actuated while the translational subsystem is under actuated [17]. The rotational equations of motion are derived in the body frame using the Newton- Euler method with the following general formalism as in (1).

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

Where: J is quadrotor's diagonal inertia matrix, ω is angular body rates, M_G is gyroscopic moments due to rotors' inertia, M_B is moments acting on the quadrotor in the body frame. The gyroscopic moments are defined to be $\omega\times\begin{bmatrix}0 & 0 & J_r\Omega_r\end{bmatrix}^T$, thus the rotational equation of the quadrotor's motion can be written as in (2).

$$J\dot{\omega}+\omega\times J\omega+\omega\times\begin{bmatrix}0 & 0 & J_r\Omega_r\end{bmatrix}^T=M_B \tag{2}$$

Where: J_r is rotors' inertia and Ω_r is rotors' relative speed ($\Omega_r=-\Omega_1+\Omega_2-\Omega_3+\Omega_4$). The inertia matrix for the quadrotor as shown in (3) is a diagonal matrix, the off-diagonal elements, which are the product of inertia, are zero due to the symmetry of the quadrotor.

$$J=\begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix} \tag{3}$$

Where I_{xx}, I_{yy} and I_{zz} are the area moments of inertia about the principle axes in the body frame. Since for the case of quadrotors, the maximum altitude is usually limited, thus the air density can be considered constant. So, Equations (4) and (5) show the aerodynamic force F_i and moment M_i produced by the i^{th} rotor [18].

$$F_i=K_f\Omega_i^2 \tag{4}$$

$$M_i=K_M\Omega_i^2 \tag{5}$$

Where K_f and K_M are the aerodynamic force and moment constants respectively and Ω_i is the angular velocity of rotor *i*. The aerodynamic force and moment constants can be determined experimentally for each propeller type. By identifying the forces and moments generated by the propellers, we can study the moments M_B acting on the quadrotor. Fig. 1 shows the forces and moments acting on the quadrotor. Each rotor causes an upwards thrust force F_i and generates a moment M_i with direction opposite to the direction of rotation of the corresponding rotor *i*. Starting with the moments about the body frame's x-axis, by using the right-hand rule in association with the axes of the body frame, F_2 multiplied by the moment arm L generates a negative moment about the y-axis, while in the same manner, F_4 generates a positive moment. Thus the total moment about the x-axis can be expressed as in (6). For the moments about the body frame's y-axis, also using the right-hand-rule, the thrust of rotor 1 generates a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis. The total moment can be expressed as in (7). For the moments about the body frame's z-axis, the thrust of the rotors does not cause a moment. On the other hand, the moment caused by the rotors' rotation as per Equation (5). By using the right-hand-rule, the moment about the body frame's z-axis can be expressed as in (8).

$$M_x=-F_2L+F_4L \tag{6}$$

$$M_y=F_1L-F_3L \tag{7}$$

$$M_z=M_1-M_2+M_3-M_4 \tag{8}$$

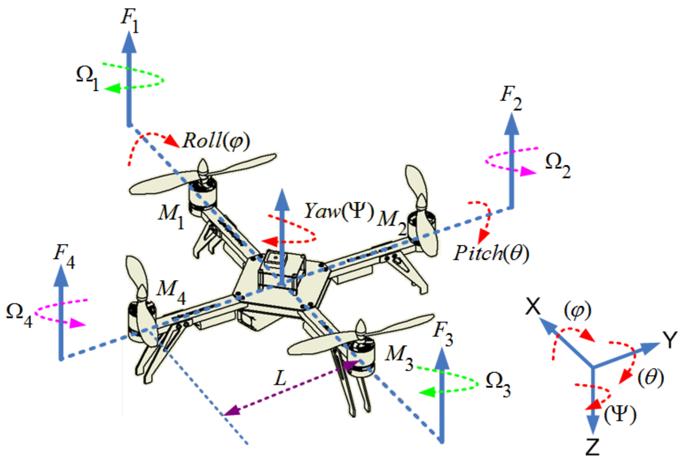

Fig. 1. Forces and moments acting on quadrotor

Combining equations (6), (7), (8) and with substitute from equation (4) – (5) in vector form, we get,

$$M_B=\begin{bmatrix}LK_f(-\Omega_2^2+\Omega_4^2)\\ LK_f(\Omega_1^2-\Omega_3^2)\\ K_M(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2)\end{bmatrix} \tag{9}$$

where L is the moment arm, which is the distance between the axis of rotation of each rotor to the origin of the body reference frame which should coincide with the center of the quadrotor. The translation equations of motion for the quadrotor are based on Newton's second law and they are derived in the Earth inertial frame

$$m\ddot{r}=\begin{bmatrix}0\\ 0\\ mg\end{bmatrix}+RF_B \tag{10}$$

Where $r=\begin{bmatrix}x & y & z\end{bmatrix}^T$ is quadrotor's distance from the inertial frame, m is quadrotor's mass, g is gravitational acceleration ($g=9.81 \quad m/s^2$), R is rotational matrix as shown in (12) and F_B is non-gravitational forces acting on the quadrotor in the body frame. When the quadrotor is in a horizontal orientation (i.e. it is not rolling or pitching), the only non-gravitational forces acting on it is the thrust produced by the rotation of the propellers which is proportional to the square of the angular velocity of the propeller as per Equation (4). Thus, the non-gravitational forces acting on the quadrotor, F_B, can be expressed as,

$$F_B=\begin{bmatrix}0\\ 0\\ -K_f(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2)\end{bmatrix} \tag{11}$$

The first two rows of the force vector are zeros as there is no forces in the X and Y directions, the last row is simply an addition of the thrust forces produced by the four propellers. The negative sign is due to the fact that the thrust is upwards while the positive z-axis in the body framed is pointing downwards.

$$R=\begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\phi\sin\theta & \sin\phi\sin\psi+\cos\phi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \cos\theta\sin\psi+\sin\phi\sin\psi\sin\theta & \cos\phi\sin\psi\sin\theta-\cos\psi\sin\theta \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \quad (12)$$

Define the state vector of the quadrotor is described in (13).

$$\boldsymbol{x}=\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T \quad (13)$$

Where each state space equation (11) can be represented as follows

$$\boldsymbol{x}=\begin{bmatrix} \phi\dot{\phi}\theta\ \dot{\theta}\ \psi\ \dot{\psi}\ x\ \dot{x}\, y\, \dot{y}\, z\, \dot{z} \end{bmatrix}^T \quad (14)$$

The state space equation (14) shows the information about the degree of freedom of quadrotor such as the position, $[x\,y\,z]$, angular angel $[\phi\theta\psi]$, linear velocity $[\dot{x}\ \dot{y}\,\dot{z}]$ and angular velocity $[\dot{\phi}\dot{\theta}\dot{\psi}]$. The input is defined as follows,

$$\boldsymbol{u}=\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T \quad (15)$$

Whereas,

$$\begin{aligned} u_1 &= K_f(\Omega_1+\Omega_2+\Omega_3+\Omega_4) \\ u_2 &= K_f(-\Omega_2+\Omega_4) \\ u_3 &= K_f(\Omega_1-\Omega_3) \\ u_4 &= K_M(\Omega_1-\Omega_2+\Omega_3-\Omega_4) \end{aligned} \quad (16)$$

From (16), we get the input as this following matrix,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}=\begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & K_M & K_M & K_M \end{bmatrix}\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (17)$$

B. Altitude Model of Quadrotor

One of the main objectives with physical modeling is to simplify the physical properties of the system that is to be modeled. The idea being that one obtains a much simpler model which, serves to explain the main characteristics of the real system. Fig. 2 shows the details about how we described altitude model at quadrotor.

From Newton's second law it is known that

$$\begin{aligned} F-mg &= ma \\ \frac{dv}{dt} &= \frac{F-mg}{m} \end{aligned} \quad (18)$$

A Laplace transformation leads to

$$S^2 h=\frac{F-mg}{m} \quad (19)$$

and defining $F-mg=F_m$ gives the final transfer function:

$$h(s)=\frac{1}{mS^2}F_m(s) \quad (20)$$

Defining the following states, inputs and outputs as

$$\begin{aligned} &input: u=F_m \\ &states: \\ &x_1=h(altitude) \\ &x_2=v(velocity) \\ &output: y=x_1 \end{aligned} \quad (21)$$

will allow the transfer function to be expressed in a standard state space model. The definitions described above in (28) leads to

$$\begin{aligned} u &= F_m \\ \dot{x}_1 &= \frac{u}{m} \\ \dot{x}_2 &= x_2 \\ y &= x_1 \end{aligned} \quad (22)$$

This allows one to express the final state space model as:

$$\begin{aligned} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}+\begin{pmatrix} 0 \\ 1 \end{pmatrix}u \\ y &= \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \end{aligned} \quad (23)$$

If we assume for stable hover position should have $\phi=0$ and $\theta=0$, so the equation (24) is described the motion speed of quadrotor at z-axis.

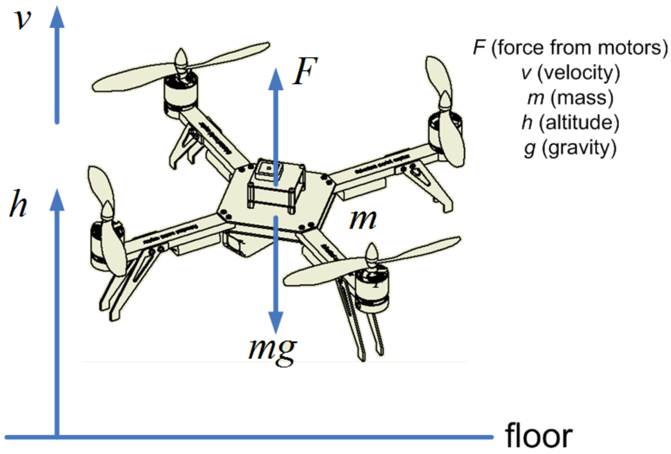

Fig. 2. Altitude model for quadrotor

$$\ddot{z}=g-\frac{-K_f(\Omega_1+\Omega_2+\Omega_3+\Omega_4)}{m} \quad (24)$$

In (20) is described transfer function of z can be defined by the following equation (25).

$$z(s)=\frac{K_f(\Omega_1+\Omega_2+\Omega_3+\Omega_4)-mg}{mS^2} \quad (25)$$

Motion of the quadrotor along z-axis is affected by all of the propellers. Thus, the dynamic model of the height could be written as in (26).

$$m\ddot{z}=4F\cos\phi\cos\theta-mg \quad (26)$$

Where $\boldsymbol{F}$ is thrust which is generated by each propeller, $\boldsymbol{m}$ is total mass of the quadrotor, z is height, ϕ and θ are roll and pitch angles. By integrated two times in (26), we get height model of quadrotor as in (27) - (28).

$$\ddot{z} = \frac{4F}{m}\cos\phi\cos\theta - g \tag{27}$$

$$z = \iint \frac{4F}{m}\cos\phi\cos\theta - g \tag{28}$$

III. Control System Architectures

A. Altitude Control

First, a PID controller is developed to control the altitude of the quadrotor. It generates the control input u_1 as shown in Fig. 3 which is responsible for the altitude for the quadrotor as per Equation (17) and in (28). In this case, we assume the quadrotor is stable in rotational sub-system. So, the value angles of roll and pitch are zero ($\phi = 0$ and $\theta = 0$). The block diagram for a PID controller is shown in Fig. 4 and in (29) as equation of PID controller with independent type structure [19]. Thus, the derived control law for altitude control is described in (30).

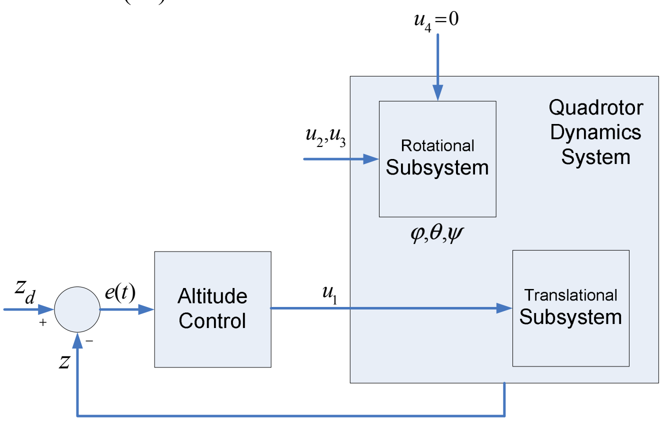

Fig. 3. Block diagram for altitude controller

The altitude controller takes an error signal e as an input which is the difference between the desired altitude z_d and the actual altitude z and produces a control signal u_1.

$$u_{(t)} = K_p e_{(t)} + K_i \int_0^t e_{(t)}\,dt + K_d \frac{de_{(t)}}{dt}$$
$$u_{(s)} = \left[K_p + \frac{K_i}{s} + K_d s\right] e_{(s)} \tag{29}$$

$$u_1 = K_p(z_d - z) + K_d(\dot{z}_d - \dot{z}) + K_i \int (z_d - z)dt \tag{30}$$

Where: K_p is proportional gain, K_d is derivative gain, K_i is integral gain, z_d is desired altitude and $\dot{z}_d$ is desired altitude rate of change.

B. Fuzzy Self Tuning PID Controller

Fuzzy logic has become a mean of collecting human knowledge and experience and dealing with uncertainties in the control system process [20]. A fuzzy self-tuning PID controller is a controller that is based on fuzzy logic with a PID structure [21]. In this sub-section, the proportional, derivative and integral gains of the PID controller are tuned using fuzzy logic as described in Fig. 4.

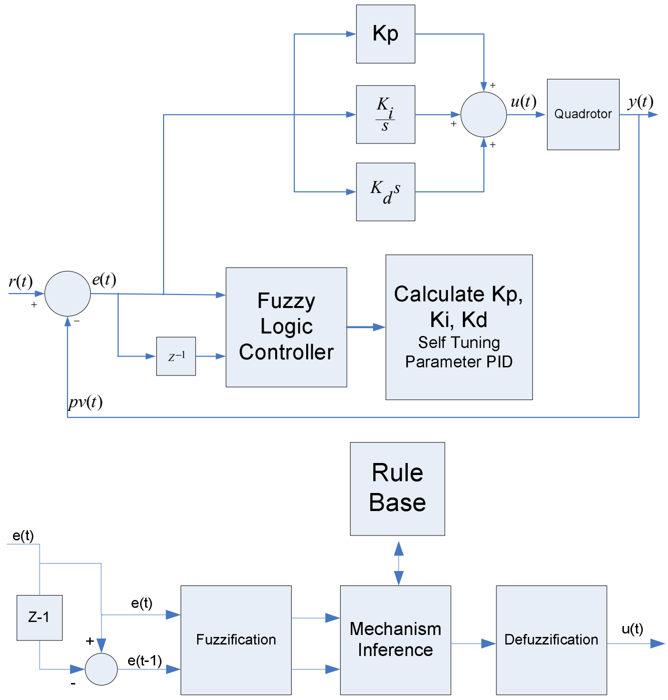

Fig. 4. Block diagram for fuzzy self tuning PID controller (up) and block diagram for fuzzy logic system (down)

IV. Simulation Result And Discussion

By using Simulik-Matlab, the equation in (28) can be modeled as described in Fig. 5.

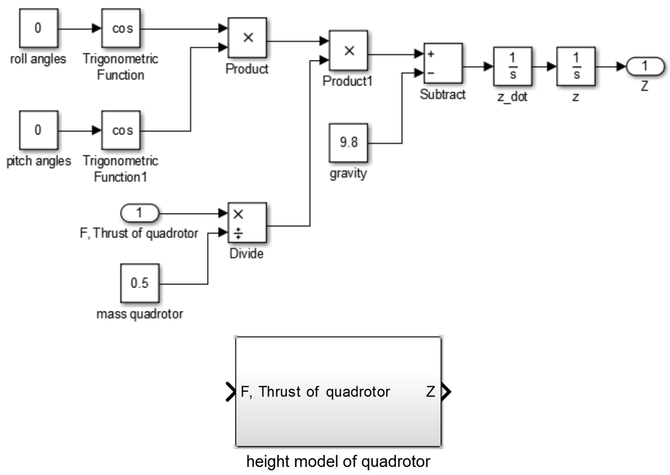

Fig. 5. Simulink diagram for altitude/height mode

To seen the open loop response of the system is shown in Fig. 8. The graphics depicted that the system is unstable. To achieve stable response, controller is needed. Firstly, we tried the PID controller only, we used Ziegler-Nichols Quarter

Decay [22] for tuning the parameter gain of PID controller. Next, we used fuzzy self-tuning PID controller as described in Fig. 6. And the result is better than the only use PID controller. When we apply disturbance such as wind using step functions as shown in Fig. 7, the fuzzy self-tuning PID controller has no overshoot than PID controller only as shown in Fig. 9. We successfully used the fuzzy system to adjust online parameter of PID gains (K_p, K_i, K_d) with better result.

In this method, we adjust PID gain parameters by online tune from evaluating error (e) and rate error $(\dot{e})$ from feedback sensor in this case is altitude feedback using fuzzy system as shown in Fig. 10(a). In fuzzification procedure, the proposed fuzzy logic system in this paper consists of two input variables $(e, \dot{e})$ are shown in Fig. 10(b) and three output variable *alpha, beta, gamma* (α, β, γ) are shown in Fig. 10(c). The various inputs divided into seven fuzzy subsets. In input system, we used triangle membership function and for output (*alpha and beta*) are a trapezional membership function with two variable fuzzy sets. In *gamma,* we used four fuzzy set with triangle membership function. For inference method, we applied 49 fuzzy control rules. To operate the fuzzy combination, Mamdani's method with Max-Min was selected [23]. In defuzzification stage, the last stage of the designed fuzzy logic controller, the centroid algorithm was used to get three output parameter gain to adjust PID parameters and described in (39), (40) and (41) respectively.

$$K_p = \mathrm{Kp}_{\min} + (\mathrm{Kp}_{\max} - \mathrm{Kp}_{\min}) \times \alpha \qquad (39)$$

$$K_d = \mathrm{Kd}_{\min} + (\mathrm{Kd}_{\max} - \mathrm{Kd}_{\min}) \times \beta \qquad (40)$$

$$K_i = \frac{K_p^2}{K_d \times \gamma} \qquad (41)$$

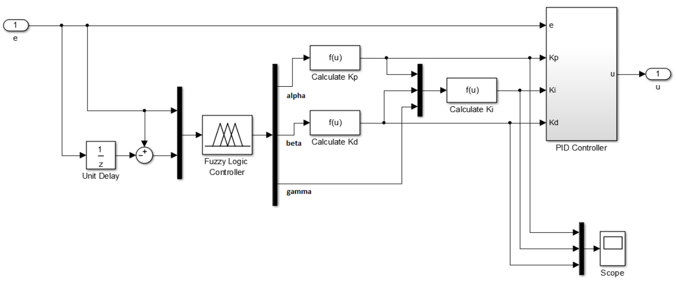

Fig. 6. Simulink diagram for fuzzy self tuning PID controller

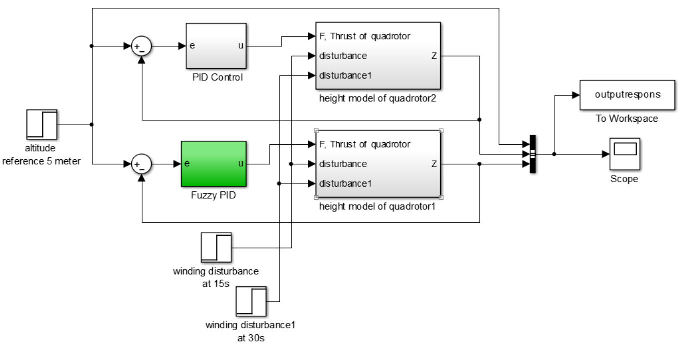

Fig. 7. Simulink diagram for fuzzy self-tuning PID controller and comparing with PID controller only.

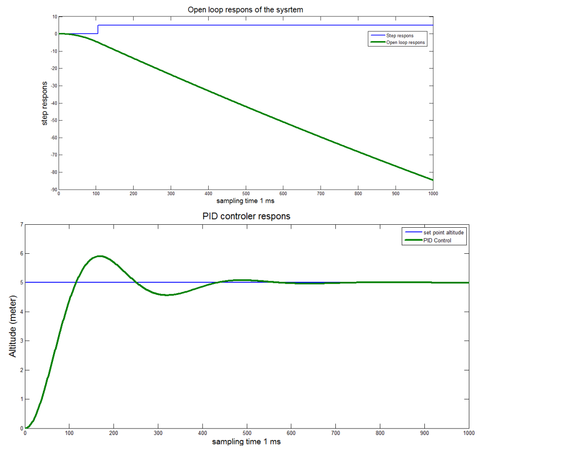

Fig. 8. Open loop respons of the system (up) and PID control respons (down)

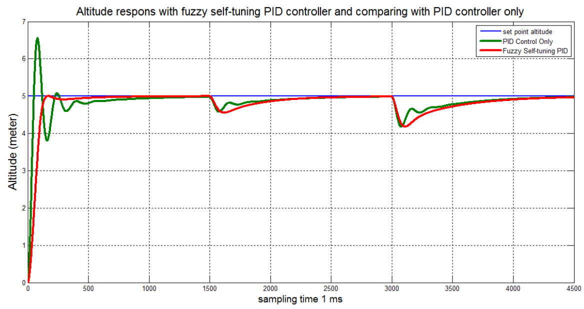

Fig. 9. Comparing result of altitude respons using fuzzy self-tuning PID controller with PID controller only

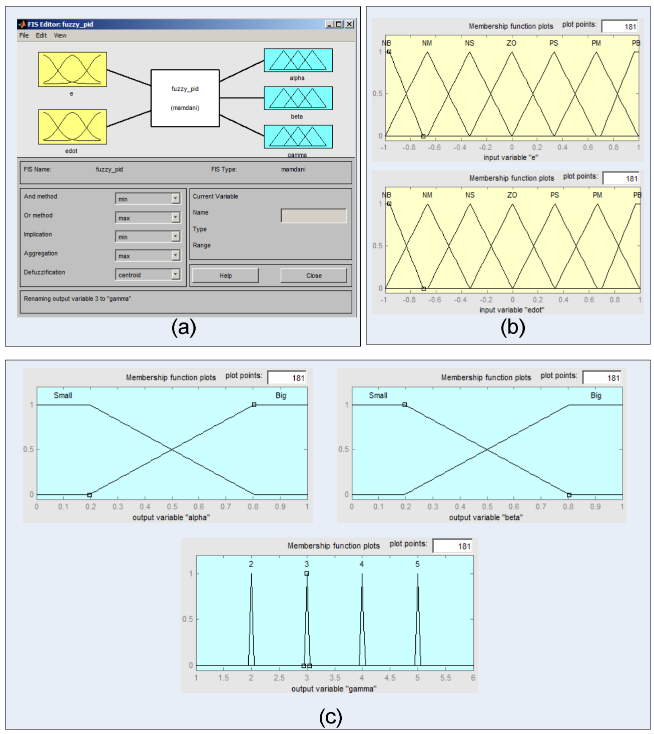

Fig. 10. (a) Fuzzy system; (b) Fuzzification; (c) Defuzzification

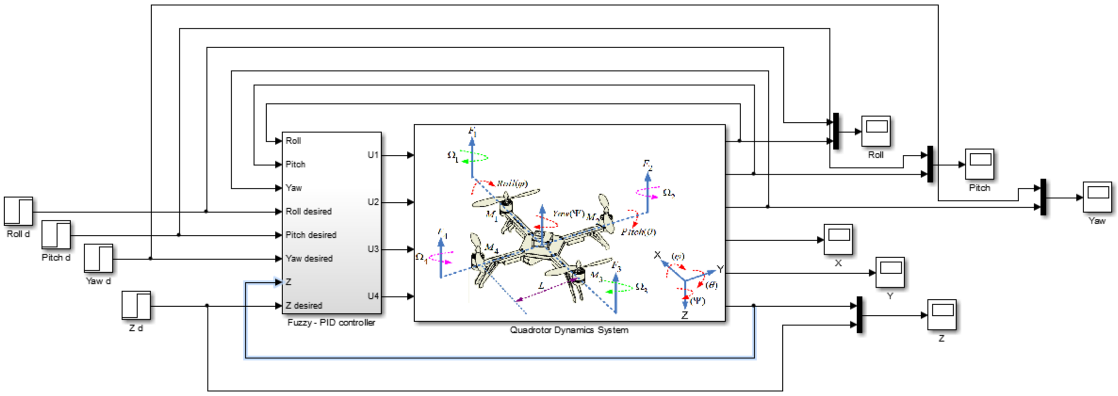

Fig. 11. A complete simulation used MATLAB/Simulink relying on the derived mathematical model of the quadrotor with Fuzzy self- tuning PID controller

V. Conclusion and Futureworks

The purpose of this study was to obtain a mathematical model for the quadrotor used to control the altitude. Two control techniques were then developed and synthesized to compare; linear PID controller and fuzzy self-tuning PID controller. The full simulation is then implemented on MATLAB/Simulink by relying on the derived mathematical model of the quadrotor as shown in Fig. 11. In addition, in our study it is assumed that all model parameters are known accurately without any uncertainty, which is not, thus, developing the algorithm such as an adaptive controls to calculate the uncertainty of the system will improve quadrotor performance when operating on a real environment.

Acknowledgment

We would like to thank to Department of Electrical and Informatics Engineering, Vocational College, Universitas Gadjah Mada for supporting this work.

References

[1] Patel, P. N., Patel, M. A., Faldu, R. M., & Dave, Y. R., "Quadcopter for agricultural surveillance." Advance in Electronic and Electric Engineering, pp. 427-432, 2013.

[2] Murthy, Meta Dev Prasad. "Design of a Quadcopter for Search and Rescue Operation in Natural Calamities." PhD diss., National Institute of Technology, Rourkela, 2015.

[3] Teixeira, J. M., Ferreira, R., Santos, M., & Teichrieb, V. "Teleoperation using google glass and ar, drone for structural inspection," Virtual and Augmented Reality (SVR), pp. 28-36, 2014.

[4] Gadda, Jinay S., and Rajaram D. Patil. "Quadcopter (uavs) for border security with gui system." International Journal of Engineering Research and Technology, pp. 620-624, 2013.

[5] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, 2007.

[6] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," IEEE International Conference on Robotics and Automation, pp. 3277–3282, 2009.

[7] A. Tayebi and S. McGilvray, "Attitude stabilization of a four-rotor aerial robot," 43rd IEEE Conference on Decision and Control, vol. 2, pp. 1216–1221, 2004.

[8] I. C. Dikmen, A. Arısoy, and H. Temelta, "Attitude control of a quadrotor," 4th International Conference on Recent Advances in Space Technologies, pp. 722-727, 2009.

[9] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," IET Control Theory Appl., vol. 4, no. 11, pp. 2343-2355, 2010.

[10] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2451-2456, 2004.

[11] P. Castillo, R. Lozano, and A. Dzul, "Stabilisation of a mini rotorcraft with four rotors," IEEE Control Systems Magazine, pp. 45-55, 2005.

[12] J. Escare˜no, C. Salazar-Cruz, and R. Lozano, "Embedded control of a four-rotor UAV," American Control Conference, vol. 4, no. 11, pp. 3936-3941, 2006.

[13] K. H. Ang, G. Chong, S. Member, and Y. Li, "PID Control System Analysis , Design , and Technology," IEEE transactions on control systems technology vol. 13, no. 4, pp. 559-576, 2005.

[14] P. J. Gawthrop and P. E. Nomikos, "Automatic tuning of commercial PID controllers for single-loop and multiloop applications," IEEE Control Syst. Mag., vol. 10, no. 1, pp. 34-42, 1990.

[15] M. Santos, V. Lopez, F. Morata, "Intelligent fuzzy controller of a quadrotor." International Conference on Intelligent Systems and Knowledge Engineering (ISKE). pp.141-146, 2010.

[16] C. Coza, C. J. B. Macnab, "A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization." Annual Meeting of the North American Fuzzy Information Processing Society. pp.454-458, 2006.

[17] Amr Nagaty, Sajad Saeedi, Carl Thibault, Mae Seto, and Howard Li. "Control and navigation framework for quadrotor helicopters." Journal of Intelligent and Robotic Systems, 70(1-4):12, 2013.

[18] A. Azzam and Xinhua Wang, "Quad rotor arial robot dynamic modeling and configuration stabilization," In Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on, volume 1, pp. 438-444, 2010.

[19] O'Dwyer, Aidan. "Handbook of PI and PID controller tuning rules." World Scientific, 2009.

[20] Fahmizal and C. H. Kuo, "Development of a fuzzy logic wall following controller for steering mobile robots," International Conference on Fuzzy Theory and Its Applications (iFUZZY), pp. 7-12, 2013.

[21] Ross, T., "Fuzzy Logic with Engineering Applications. 3rd ed." West Sussex, UK: Wiley, 2010.

[22] O'Dwyer, Aidan., "Handbook of PI and PID controller tuning rules." World Scientific, 2009.

[23] Fahmizal and C. H. Kuo, "Trajectory and heading tracking of a mecanum wheeled robot using fuzzy logic control." International Conference on Instrumentation, Control and Automation (ICA), pp. 54-59., 2016.