

Fuzzy-proportional-integral-derivative-based controller for stable control of unmanned aerial vehicles with external payloads

Do Khac Tiep, Nguyen Van Tien

Faculty of Electrical and Electronic Engineering, Vietnam Maritime University, Hai Phong, Vietnam

Article Info

Article history:

Received Dec 18, 2023

Revised Jun 20, 2024

Accepted Jul 2, 2024

Keywords:

Attitude control

Fuzzy control

MATLAB/Simulink

Position control

Proportional derivative controller

Quadcopter

ABSTRACT

In the paper, a proportional derivative (PD) controller and a fuzzy system tuning gains from proportional integral derivative controller are applied to stabilize an unmanned aerial vehicle (UAV), to control the attitude. Inputs of fuzzy logical controller consist of the speed required for the distance between the current position of quadcopter and the defined reference point and differences between orientation angles and variance in differences. Outputs of fuzzy logical controller consist of the proportional integral derivative coefficients which make pitch, roll, yaw and height values. The fuzzy-PD control algorithm is real-time applied to the quadcopter in MATLAB/Simulink environment. Based on data from experimental studies, although both classical proportional integral derivative controller and fuzzy-PD controller have accomplished to track a defined trajectory with the quadcopter.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Do Khac Tiep

Faculty of Electrical and Electronic Engineering, Vietnam Maritime University

484 Lach Tray, Le Chan, Hai Phong, Vietnam

Email: dokhactiep@vimaru.edu.vn

1. INTRODUCTION

A robust control approach is to design a feedback controller that treats the nonlinear and time-varying properties as uncertainties. A controller is proposed that can guarantee the stability of a closed-loop system within a range of these uncertainties. A disadvantage of this method is that the bound of uncertainty is required as prior information when designing the controller. Previously, many control methods were used for the stabilization of unmanned aerial vehicles (UAVs), including the conventional proportional integral derivative (PID) controller, fuzzy controller, adaptive controller, and so on [1]. Sliding mode control is one of the most popular robust control approaches, with implementations applied to UAVs in various settings [2], [3]. Approaches based on the Lyapunov theory by the use of a backstepping based direct adaptive control for a quadrotor UAV [4], Alternatively the approach used in [5], where the authors described an algorithm for altitude control for a DJI-F450 drone fitted with a laser range sensor. Reinforcement learning techniques and algorithms have been employed in more modern intelligent computation systems for unmanned aerial vehicles [6]. However, there are two drawbacks to such approaches, particularly when used for live and onboard estimate. First, tracking control is not asymptotically accomplished; instead, residual error is present due to the discrepancy between the approximation and real values of the nonlinear term containing the unknown parameter. The second problem is that the intelligent computing approach requires a processing capability not often associated with integrated UAV flight control systems due to its high algorithmic complexity. In studies [7], [8] trajectory tracking of a quadrotor UAV is obtained by controlling attitude and position of the quadrotor simultaneously. Two independent control methods are used to track desired

trajectories accurately. Moreover, there are some intelligent controllers such as neural-network-based control [9], fuzzy logic control [10]–[14]. These methods can control fast the difficult object Scheduling the PID gain can improve the UAV in-flight performance and change the flight characteristics according to the performed task, as shown in [15]–[21]. In [22] presents a fuzzy logic based PID (FPID) control algorithm for attitude stabilization of quadrotor UAV. Usually, UAV systems are unstable and attitude stabilization control plays a very important role in it. The proposed algorithm for attitude controlling of UAV uses fuzzy controller to online update the parameters of PID controller.

To take advantage of different controllers such as highly reliable PID controllers and highly adaptable fuzzy logic controllers, we propose to combine PID controllers and fuzzy controllers. Using this controller, the UAV will fly more stably and reach a state of balance with various loads that was difficult to deal with previous controllers. In the paper, the control of such UAVs requires the UAV stability and constant performance with the external payload during the flight. Fuzzy-PD controller applied to stabilize the UAV, to control the attitude and to track the trajectory. The fuzzy-PD control algorithm is real-time applied to the UAV in MATLAB/Simulink environment. Based on data from experimental studies, although both the classical PD controller and the fuzzy-PD controller have accomplished stable flight control of the UAV with various external loads.

2. AR.DRONE QUADCOPTER

2.1. Platform of the AR.Drone quadrotor

The AR.Drone 2.0 which is manufactured by Parrot is used in this study. The AR.Drone is a low-cost quadrotor. Open application programming interface and Wi-Fi control features make the aircraft used in the study very suitable for allowing quick development of algorithms. The main reasons for using the Parrot AR.Drone 2.0 in this study include outstanding maneuverability, an environment to develop open-source software, low-cost and availability of support for spare parts.

Figure 1 depicts an AR.Drone with an outdoor hull in Figure 1(a) and an indoor hull in Figure 1(b). The AR.Drone 2.0 is powered by an 11.1 V, 1500 mAh Li-Po battery with a discharge rate of 100 C. The aircraft is activated by four three-phase brushless DC motors with 14.5 W and 28,500 r/min. An 8-MIPS AVR processor is available for each driver. This aircraft is suitable for academic studies due to its electronic and mechanical features. The mainboard of AR.Drone 2.0 contains a 1 GHz 32-bit ARM Cortex A8 processor and TMS320DMC64X digital signal processor of 800 MHz with unshared video support. The system also has a 200 MHz, 1 GB DDR2 RAM memory. With such features, it is able to run a 32-bit operating system. The mainboard contains modules to communicate with the outer world. These modules particularly include IEEE 802.11 wireless communication protocol allowing transmission control protocol/internet protocol (TCP/IP) communication. This protocol uses the same 2.4 GHz frequency as wireless local area network that is widely used today. Communication between the aircraft and the controller (tele-operator or the master computer) is achieved by employing three different user datagram protocol (UDP) communication channels through Wi-Fi connection in a coverage of 100 m.



Figure 1. The AR.Drone 2.0: (a) outdoor hull and (b) indoor hull

2.2. The dynamics of the AR.Drone quadcopter

The dynamics of the AR.Drone quadcopter are described using the Newton-Euler equations, with several assumptions that need to be satisfied. Firstly, we assume that the body and the blades are rigid. Additionally, we consider the aerodynamic drag of the fuselage to be negligible, as the drone is restricted to low-speed flight. The gyroscopic torque of the spinning rotors is also assumed to be insignificant under the condition that the aircraft rotates slowly. Moreover, we assume there is no significant ground effect since the quadcopter flies sufficiently high above the ground level. Finally, we assume that the system is symmetrical across its lateral and longitudinal axes. The equations for the translational dynamics of the rigid body of the AR.Drone can be described as shown in (1) and (2):

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \sum k_F \cdot \omega_i^2 (\cos \psi \cdot \sin \theta + \cos \theta \cdot \sin \phi \cdot \sin \psi) \\ \sum k_F \cdot \omega_i^2 (\sin \psi \cdot \sin \theta - \cos \psi \cdot \cos \theta \cdot \sin \phi) \\ \sum k_F \cdot \omega_i^2 (\cos \phi \cdot \cos \theta) - mg \end{bmatrix} \quad (1)$$

Figure 2 describes the structure of AR.Drone and its frames. The mass of the system, denoted by "m", and the second derivatives of the (x, y, z) position, represented by \ddot{x} , \ddot{y} , \ddot{z} are essential for understanding the system's dynamics. The attitude of the system, indicated by the $[\phi, \theta, \psi]^T$, captures the roll, pitch, and yaw. Additionally, the motor and force constant, denoted as k_F , are assumed to be identical. The angular velocity of each motor is represented by ω_i , and the constant force of gravity is depicted by the g . Furthermore, its rotational dynamics can be presented as (2).

$$\begin{bmatrix} I_{xx} \dot{p} \\ I_{yy} \dot{q} \\ I_{zz} \dot{r} \end{bmatrix} = \begin{bmatrix} l \cdot k_F (\omega_2^2 - \omega_4^2) \\ l \cdot k_F (\omega_3^2 - \omega_1^2) \\ k_M (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (2)$$

where I_{xx} , I_{yy} , and I_{zz} denotes second moment, which represents the rotational inertia across (x, y, z) - axes, k_M represents the moment constant, and l represents the distance between the motor to the center of the quadcopter. Throttle can be used to control the vertical velocity V_z of the drone. However, applying θ and ϕ will lead to some undesirable effects, known as the secondary effects, which can cause a significant altitude drop. The six degree-of-freedom of the system can be represented using twelve states as follows: $[p_x \ p_y \ p_z, \ \dot{p}_x \ \dot{p}_y \ \dot{p}_z, \ \phi \ \theta \ \psi, \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$, where $p = [p_x \ p_y \ p_z]^T$ indicates the coordinate position of the quadcopter, $v = [\dot{p}_x \ \dot{p}_y \ \dot{p}_z]^T$ denotes its linear velocities. Mean while $[\phi \ \theta \ \psi]^T$ represents a vector of angular positions, where its angular velocities are given by $\omega = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$.

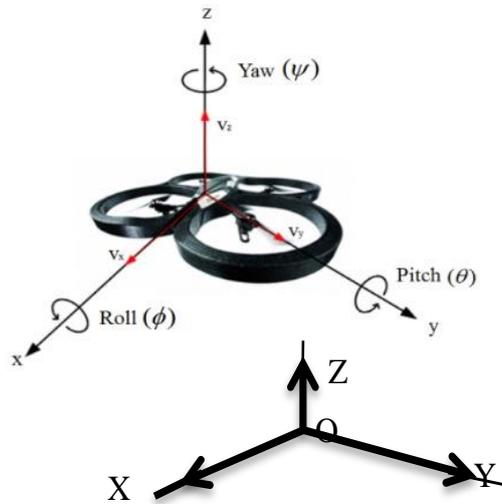


Figure 2. The structure of AR.Drone and its frames

2.3. Real-time Wi-Fi control of the parrot AR.Drone 2.0

Open application programming interface and Wi-Fi control features make the aircraft used in the study very suitable for allowing quick development of algorithms. A MATLAB/Simulink interface is developed to transmit data to the AR.Drone and to read data from the AR.Drone. The developed Wi-Fi interface allows quick performance of system identification, control and guidance tests. Figure 3 shows a block representation of the developed system.

The communication is established through UDP protocol using UDP block in Simulink. Once the communication is established, control signals are transmitted to the aircraft. At the same time, the aircraft starts transmitting data to the computer. This data exchange between the aircraft and the MATLAB/Simulink model in the computer is executed at 16 Hz frequency. The data received by MATLAB are read and recorded. Thus, the aircraft is directly controlled wirelessly via Simulink using real-time Windows target for real-time experiments. Wireless connectivity which is established in this way enables rapid guidance and control.

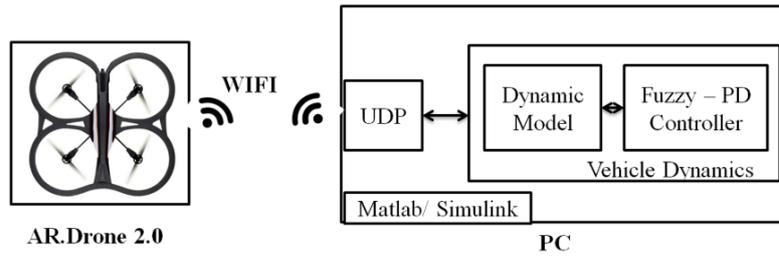


Figure 3. Block representation of the developed system UDP

The AR.Drone can be controlled using four control signals, with values ranging between -1 and 1. The pitch control uses positive values for backward movement and negative values for forward movement. The roll control uses positive values for moving to the right and negative values for moving to the left. *Yaw_rate* utilizes positive values for clockwise pivoting and negative values for counterclockwise pivoting, while the vertical rate employs positive values for vertical ascent and negative values for vertical descent. Additionally, there are three toggle switches for fly, hover, and emergency commands. Accessing navigation data from the AR.Drone provides information such as pitch, forward speed, roll, sideward speed, yaw, vertical speed, altitude, and time stamp, which can be used to estimate the x, y, z position of the drone. The inputs and outputs of the AR.Drone system are depicted in Figure 4.

MATLAB/Simulink provides powerful tools for developing and testing real-time control systems. Real-time testing allows you to simulate the behavior of your control system on real hardware, such as an embedded controller or a Windows computer. This helps you verify the performance of your system in a real-world environment before deploying it. Figure 5 shows a testbed for a real-time Windows target using MATLAB/Simulink.

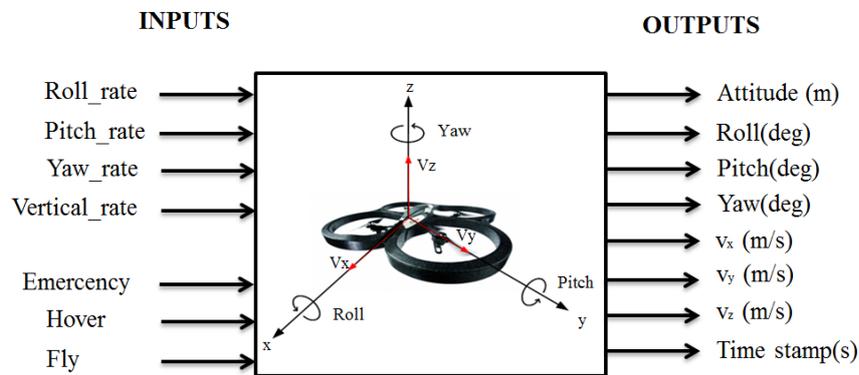


Figure 4. AR.Drone inputs-outputs signal



Figure 5. Testbed for real-time Windows target using MATLAB/Simulink

3. DESIGN CONTROLLER

Figure 6 depicts a fuzzy-PD based controller designed for achieving stable control of UAVs carrying external payloads. This specific controller type addresses the challenges associated with the additional complexity introduced by external loads. Traditional control methods might struggle in these situations. The fuzzy-PID controller leverages fuzzy logic to account for uncertainties and nonlinearities that arise due to the payload's weight and aerodynamic properties. This allows the controller to adapt to changing conditions and maintain stability.

The fuzzy gain scheduled PD controller utilizes error " e " and rate of change-in-error " ce " as input to modify PD gains online [16]–[19]. Self-tuning the PD controller involves establishing the fuzzy relationship between the two gains of PD controller and " e ", " ce ". By applying the principle of fuzzy control, the two gains are adjusted to meet varying requirements for control gains when " e " and " ce " differ, thereby ensuring optimal dynamic and static performance of the control object.

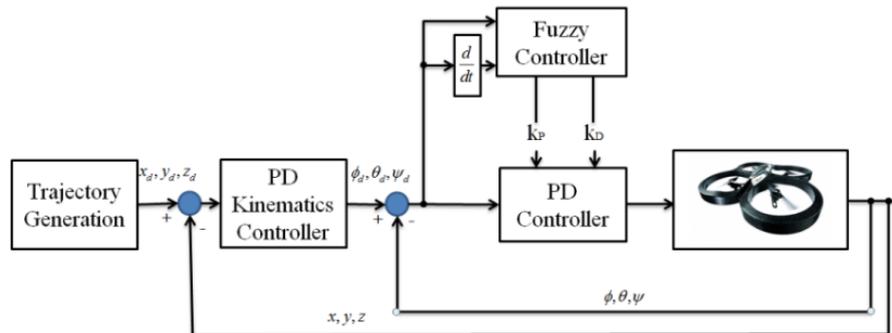


Figure 6. Block diagram of fuzzy PD controller

3.1. PID controller

Because of their simple structure and robust performance, PID controllers are the widely used in control systems. The PID control scheme derives its name from its three correcting terms, which together form the manipulated variable (MV). By summing the proportional, integral, and derivative terms, the output of the PID controller is calculated. If we define $u(t)$ as the controller output, the final form of the PID algorithm can be represented as (3).

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{d}{dt} e(t) \quad (3)$$

where K_p , K_I , and K_D are called the propositional, integral and derivative gains, respectively. For adjusting the controller parameters the Ziegler-Nichols method will be used and response of the system [13].

The success of PID controller depends on proper selection of gain parameters. The PID controller in the literature can be divided mainly into two categories. In the first category, the controller's parameters are unchanged during control after they have been tuned or chosen in certain optimal ways.

3.2. Fuzzy adaptive control

3.2.1. Generalized fuzzy control scheme

The fuzzy control technique is a relatively new approach where input variables are transformed into membership functions through a process called fuzzification. The membership values then quantify the degree of implication of a rule from the rule base. Based on the rule base, an inference mechanism is used to draw a conclusion, and finally, defuzzification is applied to convert the membership values into a crisp output. Figure 7 illustrates the generalized fuzzy control system.

The fuzzy controller consists of four key components. Firstly, the fuzzification interface is responsible for adjusting the inputs to ensure they can be properly understood and compared with the rules in the rule-base. Secondly, the inference mechanism determines which control rules are applicable at the present moment and then makes decisions regarding the input to the plant. Thirdly, the rule-base contains the knowledge, in the form of a set of rules, necessary for optimal system control. Finally, the defuzzification interface transforms the conclusions made by the inference mechanism into the inputs required for the plant.

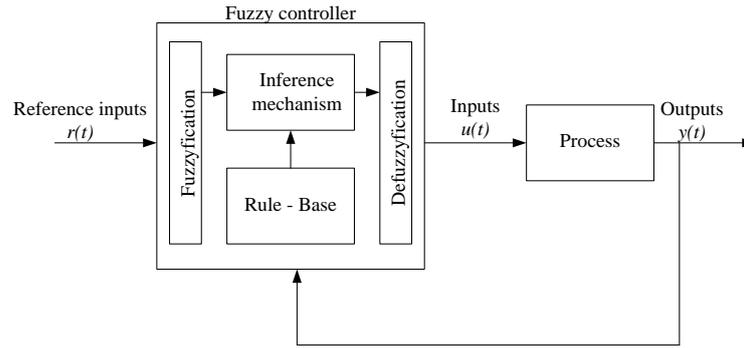


Figure 7. Generic fuzzy control scheme

3.2.2. Fuzzy gain scheduled pd control

The transfer function of a conventional PD controller is:

$$G(s) = K_p + \frac{K_d}{s} \quad (4)$$

where K_p and K_d are the proportional and derivative gains. A Takagi-Sugeno (TS) type fuzzy logic controller (FLC) tunes the PD gains online where the tracking error “e” and the change of the tracking error “edot” are used to determine control parameters. Linear transformation gives the modified controller gains as follows [11], [15], [18] as (5):

$$\begin{aligned} K_d &= (K_{dmax} - K_{dmin})K'_d + K_{dmin} \\ K_p &= (K_{pmax} - K_{pmin})K'_p + K_{pmin} \end{aligned} \quad (5)$$

with $[K_{pmin}; K_{pmax}]$ and $[K_{dmin}; K_{dmax}]$ are predetermined ranges of K_p , K_d respectively.

The type of fuzzy controllers used in this research is based on Takagi-Sugeno (TS) type fuzzy inference mechanism. A typical rule in a Sugeno fuzzy model has the form:

$$\text{If Input 1} = x \text{ and Input 2} = y, \text{ then Output is } z = ax + by + c$$

For selecting the language small variables of “e” and “ce”, is choose seven fuzzy values (*NB*, *NM*, *NS*, *Z*, *PS*, *PM*, *PB*) which *NB* denotes negative big, *NM* denotes negative medium, negative small (*NS*), zero (*Z*), positive small (*PS*), positive medium (*PM*) and *PB* denotes positive big, and for the outputs we are choose seven fuzzy values (*B*, *M*, *S*, *Z*) which big (*B*), medium (*M*), small (*S*), zero (*Z*). Fuzzy PD rules for K_p and K_d depicted in the Table 1.

In this study, the fuzzy- PD controller includes two input membership functions ($e(t)$, $ce(t)$) and two output membership functions (K_p , K_d) that apply to determine a parameter of PD controller has a triangle form. The ranges of these membership functions are determined by experience. We choose the value range for error ($e(t)$) input as from -1 to +1, and the value range for change in error ($ce(t)$) input as from -10 to +10. If the value of the input is outside the specified range, then a saturation is applied to ensure the input values are always in this range. By the same way, we also choose the value range for K_p output as from 0 to +1, and the value range for K_d output as from 0 to +0.5. The basic part of fuzzy logic control includes the set of linguistic rules. These rules hold the knowledge which is converted from expert opinions. In the form of a set of rules define an action of the system will create a number of rules respectively.

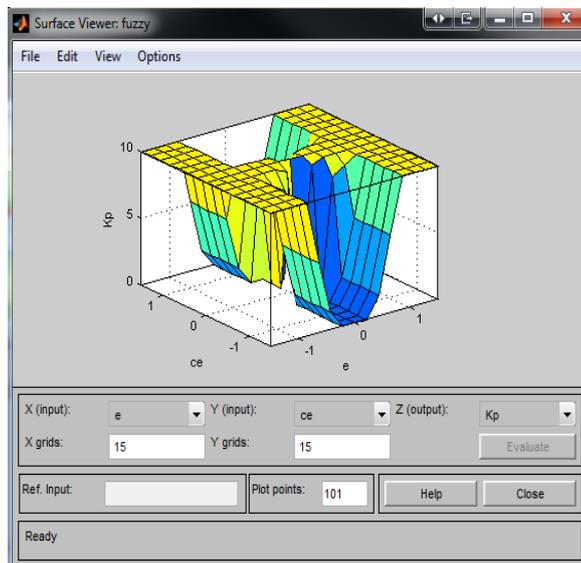
Using the MATLAB fuzzy logic toolbox, we can set rules for fuzzy inference system editor. From the expert opinions and experience, we can establish the relationship between the variables as Table 1. Data from this table allows us to create 49 rules for two fuzzy logic controllers. For example, the rule in Table 1 can explain following example: if the input $e(t)$ (error) is *NB* and the input $ce(t)$ (change in error) is *NB*, then the output K_p is *B* and the output K_d is *B*. (where: *NB* is negative big, *B*: big).

With surface viewer as Figure 8 with control surface for K_p in a Figure 8(a) and control surface for K_d in Figure 8(b), the relationship between each input and output can be seen clearly. Surface viewer is a good interpretation of fuzzy inference system (FIS) when it deals with 2 inputs and 2 outputs model. It gives a 3D interpretation of the input–output relations in the problem domain. From surface viewer, this fuzzy system is evaluated relatively good.

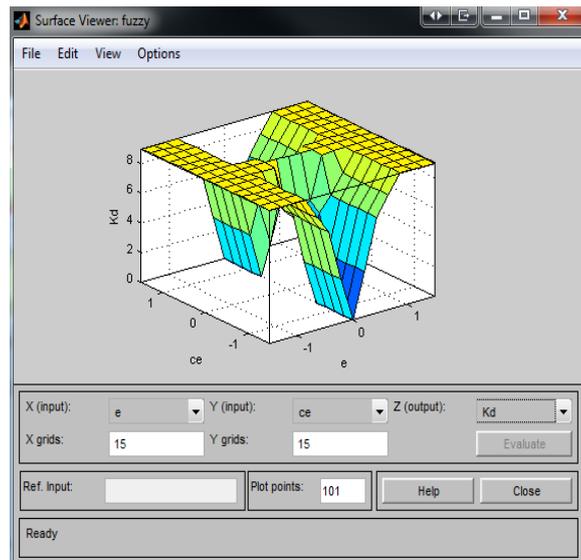
Figure 9 shows a fuzzy–PD controller for Euler’s angle with fuzzy adaptive control strategy for roll angle in Figure 9(a), for pitch angle in Figure 9(b) and for yaw angle in Figure 9(c). The fuzzy–PD controller for roll angle includes two inputs: lateral velocity (v), lateral velocity reference (v_{ref}) and one output: roll angle reference. For pitch angle includes two inputs: forward velocity (u), forward velocity reference (u_{ref}) and one output: pitch angle reference. For yaw angle includes two inputs: yaw angle, yaw angle reference and one output: yaw rate reference.

Table 1. Fuzzy PD rules for K_p/K_d

e/ce	NB	NM	NS	Z	PS	PM	PB
NB	B/B						
NM	M/S	M/B	B/B	B/B	B/B	M/B	M/S
NS	S/S	M/S	M/B	B/B	M/B	M/B	S/S
Z	Z/S	S/S	M/B	B/B	M/B	S/S	Z/S
PS	S/S	M/S	M/B	B/B	M/B	M/S	S/S
PM	M/S	M/B	B/B	B/B	B/B	M/B	M/S
PB	B/B						



(a)



(b)

Figure 8. Control surfaces for (a) K_p and (b) K_d

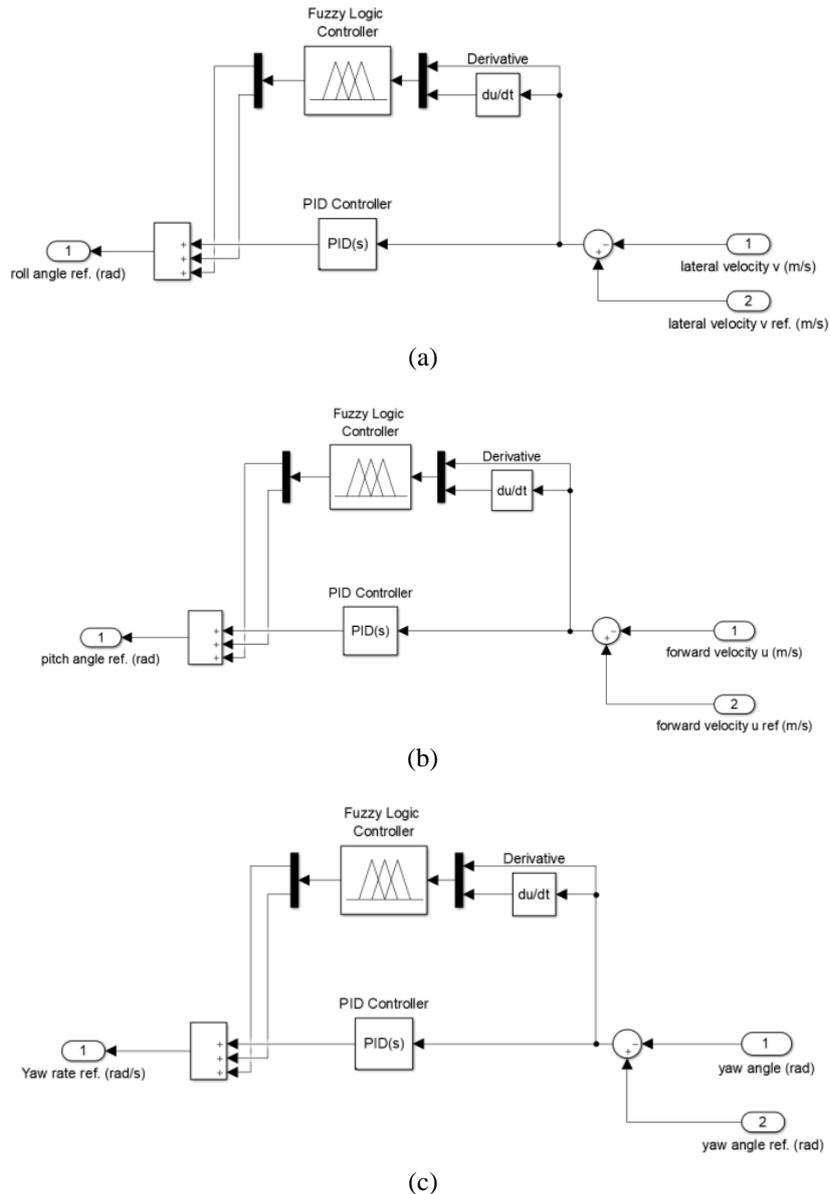


Figure 9. Fuzzy adaptive control strategy for: (a) Roll angle, (b) Pitch angle, and (c) Yaw angle

4. EXPERIMENT RESULTS

Real-time control applications are performed on the AR.Drone 2.0 quadrotor with the dynamic model and controllers created in MATLAB/Simulink environment [20]–[22]. The main objective of this study is to enable the quadcopter to follow a defined trajectory with algorithm of an improved controller.

The relative position between vehicle and target is calculated in MATLAB/Simulink [23]. The quadcopter's position and velocity are measured by its inertial sensors. The performance of classical PD controller of balance ability is compared with the performance of Fuzzy-PD controller [24], [25].

In this section to perform the stability performance experiment tests, quadcopter will be fixed on the frame with the rotary joint (prevents further upward) and then we are hanging the heavy thing on the wing to create an unbalanced state for quadcopter as presented in Figure 10 and Figure 11. From navigation data received from AR.Drone, we can get the position and vertical velocity of the AR.Drone. The model of the AR.Drone in MATLAB/Simulink will calculate the position of target and vehicle.

Applying a fuzzy-PD controller PD controller test to stabilize the UAV with the external payload during the flight. Experiments are performed indoors to avoid impact of environmental factors. During the experiments the AR.Drone will be hung the external payload ($m = 50$ g) and be attached to the frame with a height of 0.5m. The results will be compared with each other.

Figures 10 and 11 show the stability of an AR.Drone quadcopter. By comparing the figures we can see, a fuzzy-PD controller has better stability than a PD controller. Figure 11 highlights the fuzzy-PD controller's effectiveness. The AR.Drone quadcopter maintains balance with significantly greater stability compared to PD controller. Figures 12 and 13 show Euler's angle of AR.Drone with a fuzzy-PD controller and a PD controller. Overshoot in three Euler angles with a fuzzy-PD controller is less than a PD controller. As expected, the same scaling is ensured for the same type of results with different control schemes where possible.

From Figures 14 and 15 we can see the attitude error of AR.Drone using a fuzzy-PD controller and a PD controller. Overshoot in attitude errors with the fuzzy-PD controller is less than classical the PD controller. At the time six seconds, deviation of attitude error of a AR.Drone using a fuzzy-PD controller equals 7.452 (mm) while deviation of attitude of a AR.Drone using a PD controller equals 86.09 (mm).



Figure 10. An unbalanced state for quadcopter with PD controller



Figure 11. An unbalanced state for quadcopter with fuzzy-PD controller

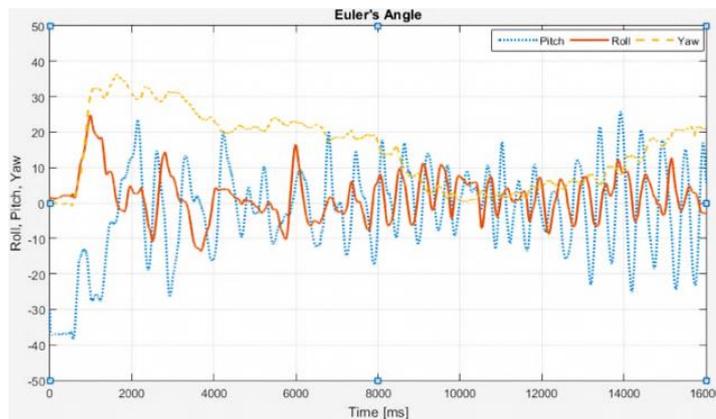


Figure 12. Euler's angles with PD controller

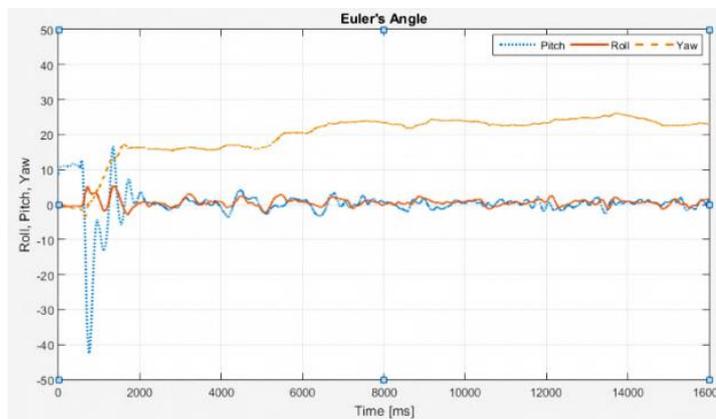


Figure 13. Euler's angles with fuzzy-PD controller

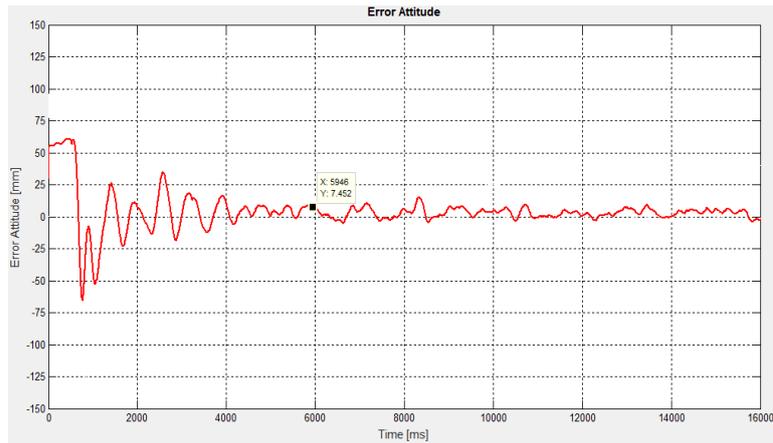


Figure 14. Attitude error of quadcopter using a fuzzy–PD controller

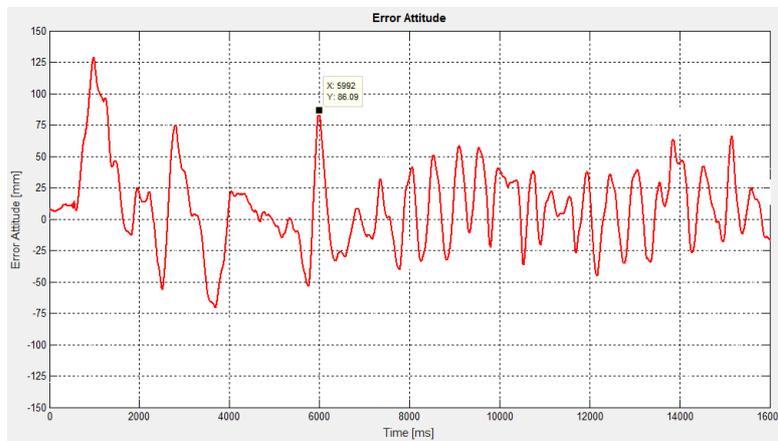


Figure 15. Attitude error of quadcopter using a PD controller

Figures 16 to 19 showcase the results of two distinct control methods for propeller direction and motor commands: a fuzzy-PD controller and a PD controller. A significant difference in performance between these two approaches is readily apparent. This difference might manifest in various ways, such as the level of precision achieved in propeller direction control or the responsiveness of the motor commands, as demonstrated by the experimental results obtained.

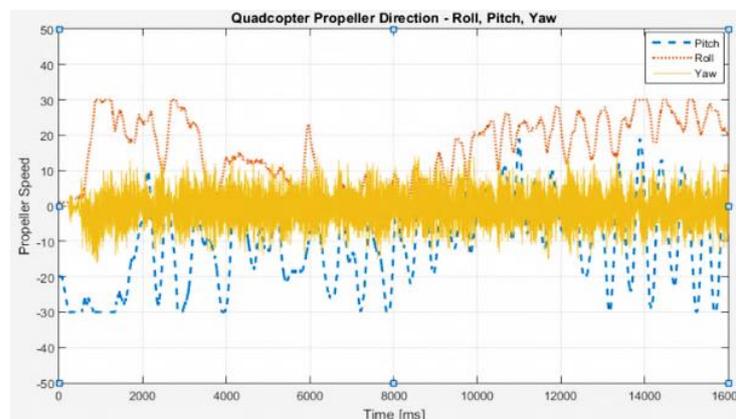


Figure 16. Quadcopter propeller direction with PD controller

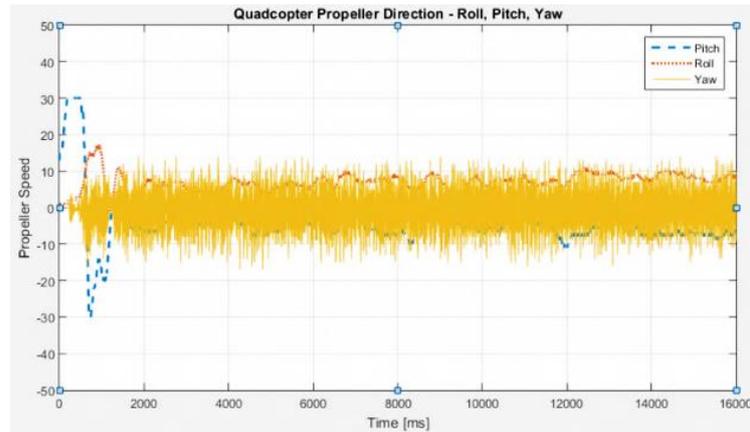


Figure 17. Quadcopter propeller direction with fuzzy-PD controller

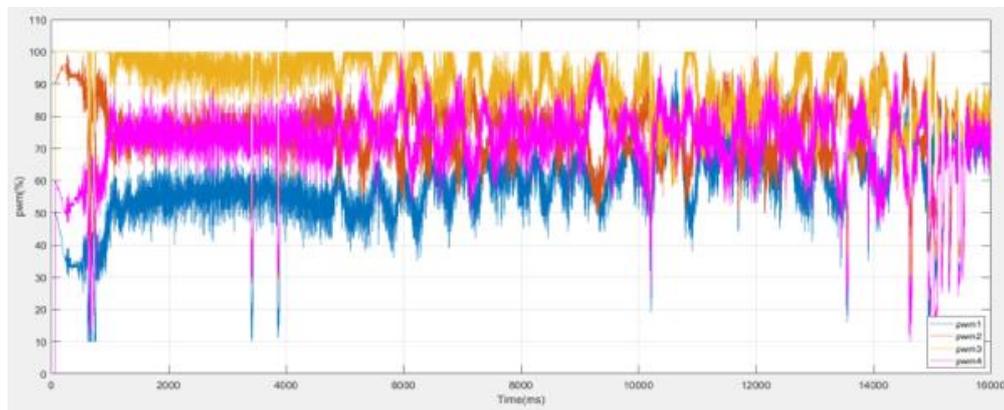


Figure 18. Pulse width modulation (PWM) command with PD controller

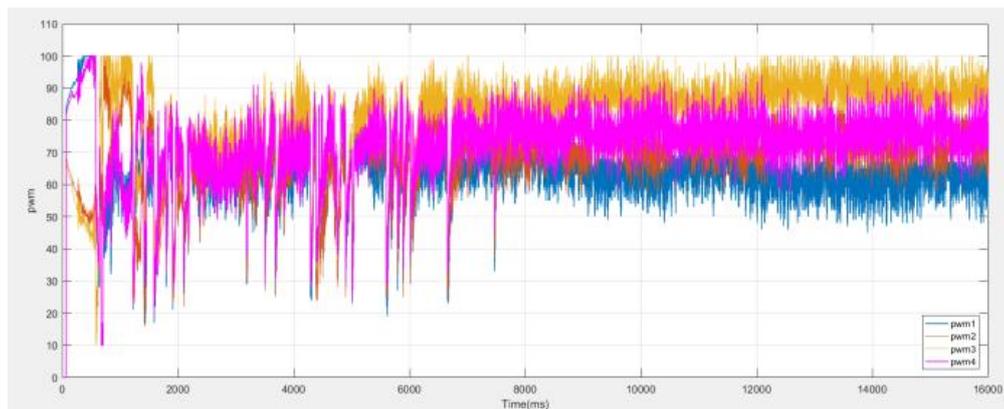


Figure 19. Pulse width modulation (PWM) command with fuzzy-PD controller

5. CONCLUSION

In this paper, we have introduced an algorithm for designing trajectory tracking for a quadcopter. Our main goal was to establish an analytical relationship between attitude and linear acceleration while also eliminating the two-time scale separation assumption between attitude and linear dynamics. We conclude by conducting and analyzing numerical simulations for a quadcopter under various conditions. The results illustrate the effectiveness of the controller in tracking trajectory and its robustness in handling modeling

errors and external disturbances in comparison to typical PD with fuzzy-PD controller.

MATLAB/Simulink-based dynamic model of quadrotor and real-time trajectory tracking control applications were successfully performed using classical PD controller and fuzzy-PD controller and are presented in this paper. Although the literature includes many studies on self-tuning fuzzy PD control of dynamic systems, this method has only recently been used to control quadcopters. We can classify that the studies used this control method as performed in simulation environments and real-time applications. In the presented study, the novelty is using this method for real-time trajectory tracking.

The fuzzy controller relies much less on the quadcopter model compared to traditional controllers. It is important to note that the fuzzy controller is non-linear. Therefore, it is more suitable for non-linear systems such as a quadcopter. Experimental results indicate that the self-tuning fuzzy-PD controller has an outstanding performance compared to the conventional controllers in terms of the amount of error.

REFERENCES

- [1] H. Desa, S. Faiz Ahmed, and A. Zul Azfar, "Adaptive hybrid control algorithm design for attitude stabilization of quadrotor (UAV)," *Archives Des Sciences*, vol. 66, no. 2, pp. 51-64, 2013.
- [2] H. Nematy and A. Montazeri, "Analysis and design of a multi-channel time-varying sliding mode controller and its application in unmanned aerial vehicles," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 244-249, 2018, doi: 10.1016/j.ifacol.2018.11.549.
- [3] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum, and S. El-Ferik, "An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter UAV," *IEEE Access*, vol. 8, pp. 205968-205979, 2020, doi: 10.1109/access.2020.3037557.
- [4] N. Koksai, H. An, and B. Fidan, "Backstepping-based adaptive control of a quadrotor UAV with guaranteed tracking performance," *ISA Transactions*, vol. 105, pp. 98-110, Oct. 2020, doi: 10.1016/j.isatra.2020.06.006.
- [5] N. Xuan-Mung and S.-K. Hong, "Improved altitude control algorithm for quadcopter unmanned aerial vehicles," *Applied Sciences*, vol. 9, no. 10, p. 2122, May 2019, doi: 10.3390/app9102122.
- [6] O. Elhaki and K. Shojaei, "A novel model-free robust saturated reinforcement learning-based controller for quadrotors guaranteeing prescribed transient and steady state performance," *Aerospace Science and Technology*, vol. 119, p. 107128, Dec. 2021, doi: 10.1016/j.ast.2021.107128.
- [7] Z. Xie, Y. Xia, and M. Fu, "Robust trajectory-tracking method for UAV using nonlinear dynamic inversion," *2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)*. IEEE, Sep. 2011. doi: 10.1109/iccis.2011.6070308.
- [8] A. P. Schoellig, "Improving Tracking Performance by Learning from Past Data," Institute for Dynamic Systems and Control ETH Zurich Switzerland, 2013. doi: 10.3929/ethz-a-009758916.
- [9] M. R. Khosravani, "Application of neural network on flight control," *International Journal of Machine Learning and Computing*, pp. 882-885, 2012, doi: 10.7763/ijmlc.2012.v2.258.
- [10] H. Chen, X. Wang, and T. Lei, "Fuzzy logic controller optimization design for tiltrotor aircraft landing conversion," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 3482-3486.
- [11] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-418, 1990, doi: 10.1109/21.52551.
- [12] L. Reznik, "Fuzzy controller implementation," in *Fuzzy Controllers Handbook*, Elsevier, 1997, pp. 201-216. doi: 10.1016/b978-075063429-8/50012-9.
- [13] E. H.-K. Fung, Y.-K. Wong, Y. Ma, C.-W. M. Yuen, and W.-K. Wong, "Smart hanger dynamic modeling and fuzzy controller design," *International Journal of Control, Automation and Systems*, vol. 9, no. 4, pp. 691-700, Aug. 2011, doi: 10.1007/s12555-011-0410-1.
- [14] C. Cavalcante, J. Cardoso, J. J. G. Ramos, and O. R. Neves, "Design and tuning of a helicopter fuzzy controller," *Proceedings of 1995 IEEE International Conference on Fuzzy Systems. The International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*. IEEE. doi: 10.1109/fuzzy.1995.409884.
- [15] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Fuzzy logic-based self-tuning autopilots for trajectory tracking of a low-cost quadcopter: A comparative study," *2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. IEEE, Oct. 2015. doi: 10.1109/icamimia.2015.7508004.
- [16] M. Rabah, A. Rohan, S. A. S. Mohamed, and S.-H. Kim, "Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-PI," *IEEE Access*, vol. 7, pp. 38407-38419, 2019, doi: 10.1109/access.2019.2906345.
- [17] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, and I. Nilkhamhang, "Path tracking of UAV using self-tuning PID controller based on fuzzy logic," in *Proceedings of SICE Annual Conference 2010*, 2010, pp. 1265-1269.
- [18] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 5, pp. 1392-1398, 1993, doi: 10.1109/21.260670.
- [19] D. K. Tiep and Y.-J. Ryoo, "An autonomous control of fuzzy-PD controller for quadcopter," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 2, pp. 107-113, Jun. 2017, doi: 10.5391/ijfis.2017.17.2.107.
- [20] J. R. García-Martínez, E. E. Cruz-Miguel, R. V Carrillo-Serrano, F. Mendoza-Mondragón, M. Toledano-Ayala, and J. Rodríguez-Reséndiz, "A PID-type fuzzy logic controller-based approach for motion control applications," *Sensors*, vol. 20, no. 18, p. 5323, Sep. 2020, doi: 10.3390/s20185323.
- [21] S. Zeghlache, A. Djerioui, L. Benyettou, T. Benslimane, H. Mekki, and A. Bouguerra, "Fault tolerant control for modified quadrotor via adaptive type-2 fuzzy backstepping subject to actuator faults," *ISA Transactions*, vol. 95, pp. 330-345, Dec. 2019, doi: 10.1016/j.isatra.2019.04.034.
- [22] W. Si, H. She, and Z. Wang "Fuzzy PID controller for UAV tracking moving target" *29th chinese control and decision conference (CCDC)*, 2017, doi: 10.1109/CCDC40495.2017.
- [23] L. V. Santana, A. S. Brandao, M. Sarcinelli-Filho, and R. Carelli, "A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor," *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, May 2014. doi: 10.1109/icuas.2014.6842321.
- [24] A. Hernandez, C. Copot, R. De Keyser, T. Vlas, and I. Nascu, "Identification and path following control of an AR.Drone quadrotor," *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, Oct. 2013. doi:

10.1109/icstcc.2013.6689022.

- [25] P. Vadakkepat, T. C. Chong, W. A. Arokiasami, and X. Weinan, "Fuzzy logic controllers for navigation and control of AR. Drone using Microsoft Kinect," *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, Jul. 2016. doi: 10.1109/fuzz-ieee.2016.7737778.

BIOGRAPHIES OF AUTHORS



Do Khac Tiep    received his BS from Electrical and Electronics Department of Vietnam Maritime University, Haiphong, Vietnam in 2009. He received MS degree of Vietnam Maritime University in 2012. From 2009 to 2015, he was a teacher at Vietnam Maritime University. He received his Ph.D. degree in the Department of Control Engineering and Robotics, Mokpo National University, South Korea, in 2019. He is currently a lecturer at the Department of Electric and Electronics Engineering of Vietnam Maritime University. His research interests are control of automation control theory, robot control and power plant on ship. He can be contacted at email: dokhactiep@vimaru.edu.vn.



Nguyen Van Tien    received his B.S and M.S degrees in Electrical and Electronics Department and Automation from the Vietnam Maritime University in 2009 and 2012, respectively. He received his Ph.D. degree in electric and control engineering from the Mokpo National Maritime University, Korea, in 2018. He is currently a lecturer at the Department of electric and electronics engineering of Vietnam Maritime University. His research interests are control of industrial machines, robot control, LabVIEW software, and recycling energy. He can be contacted at email: nguyenvantien@vimaru.edu.vn.