Sriwijaya Electrical and Computer Engineering Journal (SELCO)
Vol. 1, No. 1, February 2024, pp. 20 – 29
DOI: 10.62420/selco.v1i1.2

Simulation of robot arm system control using fuzzy logic

Syarifa Fitria[1], Sariman[1] , Zainuddin Nawawi[1], Rizda Fitri Kurnia[1], Dwirina Yuniarti[1], Tresna Dewi[2]
[1]Department of Electrical Engineering, Sriwijaya University, Jl. Raya Palembang – Prabumulih Km 32, Indralaya 30662, Indonesia
[2]Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Jalan Srijaya Negara Palembang, 30139, Indonesia

ARTICLE INFO

Article history:

Received November 20, 2023
Revised January 12, 2024
Published February 1, 2024

Keywords:

Arduino;
Sensor;
Controller;
Actuator;
Robot;
Power

ABSTRACT

This study explains a robot arm system control simulation using a fuzzy logic controller. The robot arm is a nonlinear system and is not easily controlled using a conventional control system since the mathematical model of the system is not easy to obtain. Considering these problems, this final project presents a simulation of a fuzzy logic controller on a robot arm system. The aim of this control is to move the robot arm at an angular speed thus the robot gets the desired torque or the right position. Determining *fuzzy rules* greatly influences system stability. This control network produces output in the form of torque while the robot arm system calculates the next torque from the robot. The best output results were found on the triangular curve at response 2 because of the difference in values 0.01 on the final angle obtained.

Corresponding Author:

Coresponding Author : Syarifa Fitria, Sriwijaya University, Jl. Raya Palembang-Prabumulih, Indralaya 30662, Indonesia
Email: syarifafitria@ft.unsri.ac.id

1. INTRODUCTION

The robot arm is the prevalent choice in industrial settings, often employed as a straightforward and easily adaptable solution to replace human labor in manufacturing lines. This type of robot can be tailored to replicate the movements of the human arm and fingers (end-effector), enabling it to perform tasks like pick-and-place operations and welding [1]–[4].

One of the important parts of designing a robot arm is knowing the mathematical model of the robot arm system. Mathematical models for linear systems can still be obtained, but for non-linear systems, it is not easy to derive a mathematical model from the system. Fuzzy logic control provides another alternative in control systems[5]. Fuzzy logic control works based on rules that are extracted according to human thinking and knowledge, either as operators or experts. One method used to design a robot arm is with a fuzzy logic controller. Fuzzy logic is a methodology used to express the operational laws of a system with linguistic expressions, not with mathematical equations.

This controlling algorithm is derived from verbal knowledge, experience, and intuition. Thus the performance depends on a person's expertise and experience to determine the rules and functions of its membership. For this reason, in this final project, a robot arm control simulation was created using the visual studio C# language based on fuzzy logic.

2. METHODS

2.1 Mathematical Robot Arm Design

In this robot arm theory, a voltage signal is given to the system, which is called the input variable, which is transferred to the DC motor, gears, encoder, and finally the robot arm. The encoder signal is called the output sensor and the angle of the robot arm is called the controlled variable (Figs. 1 and 2)[6]. The system uses the following mathematical model:

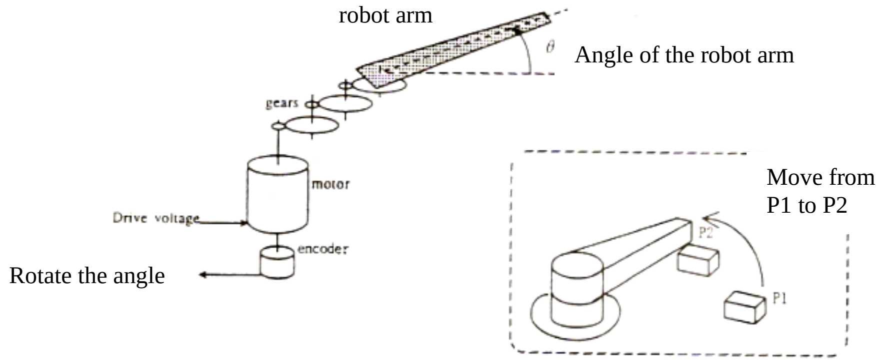

Fig. 1 Robot arm control [6]

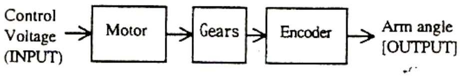

Fig. 2 Robot arm block diagram [6]

Mathematical models consist of symbols and mathematical equations to describe the system (Eq. (1) – (10)) . To obtain a mathematical model of the robot arm system, Newton's equation is used as follows:

$$\sum F = ma \tag{1}$$

because it uses rotational dynamics thus the equation becomes:

$$\sum \tau = I.\alpha \tag{2}$$

where:

$$\tau = c.v \tag{3}$$

$$v = \frac{d\theta}{dt} \tag{4}$$

$$I = J \tag{5}$$

$$\alpha = \frac{dv}{dt} = \frac{d^2\theta}{dt^2} \tag{6}$$

$\sum \tau$ is the amount of torque = resulting torque – friction torque (7)

If we plug (5), (6) and (7) into (2), we get:

$$\sum \tau = I.\alpha$$

$$k.u(t) - c\frac{d\theta}{dt} = J\frac{d^2\theta}{dt^2} \tag{8}$$

$$J\frac{d^2\theta}{dt^2}y(t) + c\frac{d}{dt}y(t) = k.u(t) \tag{9}$$

where :

$\sum \tau$ = total torque $= k.u(t) - c\dfrac{d\theta}{dt}$

k : motor torque constant (NM/Volt)
u : motor voltage
c : friction torque constant
k. u(t) : torque produced by the motor
$c\dfrac{d\theta}{dt}$: friction torque

From the explanation above, it can be concluded that the robot arm uses rotational dynamics equations using Newton's equations for rotational dynamics.

$$J\frac{d^2\theta}{dt^2}y(t) + c\frac{d}{dt}y(t) = k.u(t) \tag{10}$$

where J, c, and k are parameters, because these parameters depend on the weight of the robot arm, motor strength and gear ratio.

2.2 Robot arm system description

$$J\frac{d^2}{dt^2}y(t) + c\frac{d}{dt}y(t) = k.u(t) \quad (11)$$

Where :

$$\text{Angle} \quad : x_1 = y(t) = \theta(t) \quad (12)$$

District Angle : $x_2 = y(t) = \theta(t)$ Hence $x_1 = x_2$

$$J.x_2 + c.x_2 = k.u \quad (13)$$

$$x_2 = -\frac{c}{j}x_2 + ku$$

Matrix:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{c}{j} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k}{j} \end{bmatrix} u \quad (14)$$

Or

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} u \quad (15)$$

Equations (11) to (15) are called the state equation.
The output of the system is y(t), which is the angle of the robot arm. Output y(t) is shown by the equation:

$$y(t) = x_1 \quad (16)$$

or

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (17)$$

input vector : u(t) = $[u]$, output vector : y(t) = $[y]$, so:

$$\dot{x} = Ax + Bu \quad (18)$$

$$y = Cx + Du \quad (19)$$

All parameters used for the robot arm system are shown in the Table 1 [6]

Table 1. System Parameters Robotic arm

Parameter	Information	Measurement Value
J	Moment of inertia	0.000471 kgm^2
c	Biscous friction coefficient	0.000586 kgm^2/s
k	Torque gain	0.982 N/V

2.3 Fuzzy Logic controller theory

Membership function is a curve that shows the mapping of data input points into their membership values (often also called membership degrees) which have an interval between 0 and 1. One way that can be used to get membership values is by going through membership function approach [7]. Membership function is a curve that shows the mapping of input data points into their membership values (degree of membership) which has an interval between 0 and 1.

a) Triangular Curve Representation [7].
A triangular curve is basically a combination of 2 lines (linear) (Fig. 3).

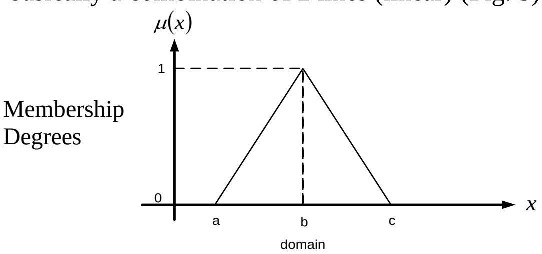

Fig. 3. Representation of a triangular curve

Membership function:

$$\mu(x)=\begin{cases} 0 & untuk \quad x<a \\ (x-a)/(b-a) & untuk \quad a\le x\le b \\ (c-x)/(c-b) & untuk \quad b\le x\le c \\ 0 & untuk \quad x>c \end{cases}$$

b) Trapezoidal Curve Representation [7]

A trapezoidal curve is basically like a triangle shape, only there are several points that have a membership value of 1 (Fig. 4).

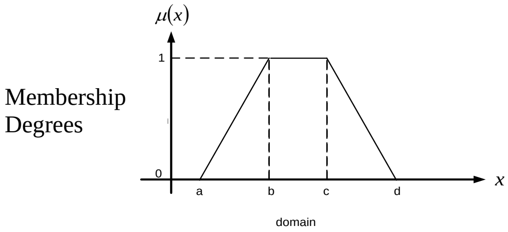

Fig. 4. Representation of a trapezoidal curve

Membership function:

$$\mu(x)=\begin{cases} 0 & untuk \quad x\le a \quad atau \quad x\ge d \\ (x-a)/(b-a) & untuk \quad a\le x\le b \\ 1 & untuk \quad b\le x\le c \\ (d-x)/(d-c) & untuk \quad c\le x\le d \end{cases}$$

c) Bell Shape Curve Representation (*Bell Curve*) [8]

To represent *fuzzy logic numbers* , a bell-shaped curve is usually used . The difference between the three curves lies in their gradients (Fig. 5).

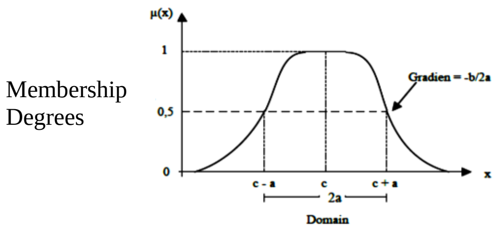

Fig. 5. Representation of the bell shape curve

Membership function:

$$gbellmf(x;a,b,c)=\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{20}$$

where the shape of the curve is determined by parameter a which shows the width of the function, parameter b which shows the slope of the curve and finally parameter c which shows the midpoint. The b parameter is usually positive, if the b value is negative then this membership function will take the form of an inverted bell as shown in (20).

d) Gaussian Curve Representation [8].

To represent *fuzzy logic numbers* , you can also use a Gaussian curve (Fig.6).

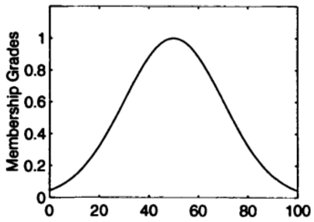

Fig. 6. Representation of Gaussian curve

Membership functions:

$$gaussian(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (21)$$

The Gaussian membership function is completely described by c and σ ; c explains the center of the membership function and σ explains the width of the membership function as in (21).

2.4 Determining the input and output of the Robot arm

Fuzzy Logic control system the robot arm uses 2 inputs, namely:

1. Corner (θ)
   Corner represents the angle between the x-axis and the robot's axis.
2. The angular velocity($\dot{\theta}$)
   Angular velocity states the angular velocity at the initial time when input is given which will determine the movement of the robot arm.

Meanwhile, this *fuzzy* logic control system only has 1 output, namely torque which then determines the direction of rotation. This torque is a signal of the direction angle, namely the angle between the axis and the direction of the robot arm. The purpose of this control system is how to move the robot arm to get the desired position of the robot arm.

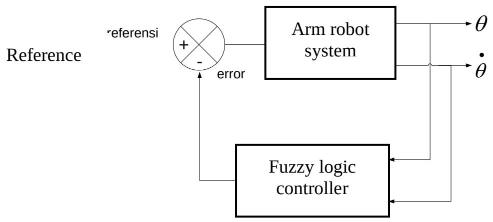

Fig. 7. Fuzzy logic control block diagram

Fuzzy logic control is used to control the torque of the robot arm so that it is in the desired position (according to the reference angle). The input for this fuzzy logic control is the arm angle (θ) and the arm angular velocity ($\dot{\theta}$) (Fig. 7) This arm angle is the angle used as input when the angle is given, the angle to the robot arm which is expressed in degrees (deg). Angular speed is the angular speed of the robot arm to arrive at the desired corner point, angular speed is the speed of change in the angle per unit time, angular speed is expressed in degrees/seconds (deg/s). The output of this fuzzy logic control is the torque that will move the robot arm according to the input given.

2.5 Runge-Kutta Method

The method used to simulate the robot arm system in this final project is the fourth order Runge-Kutta method because the Runge-Kutta method is the most popular and most frequently used for simulating differential equations. The following is a form of the classical fourth order Runge-Kutta method:

$$y_{i+1} = y_i + \left[\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right]h \quad (22)$$

with :

$$k_1 = f(x_i, y_i) \quad (23)$$
$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right) \quad (24)$$
$$k_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2\right) \quad (25)$$
$$k_4 = f(x_i + h, y_i + hk_3) \quad (26)$$

$\dot{\theta}$ from (12) and (13) for rotational motion is required which will be entered into equation (8), provided that only first order variables appear on the right side of the equation θ
From (8), (12), and (13), we get:

$$J\ddot{\theta} + c\dot{\theta} = k.u \quad (27)$$

Sriwijaya Electrical and Computer Engineering Journal (SELCO)
Vol. 1, No. 1, February 2024, pp. 20 – 29
DOI: 10.62420/selco.v1i1.2

By changing the angle to a change in angle and the angular velocity to a change in angular velocity as below:

$$\dot{\theta} = v \quad (28)$$
$$\ddot{\theta} = v \quad (29)$$

From (27) above, it cannot be used to use the Runge-kutta method, since there is a 2nd derivative in the equation. Therefore, the equation will be changed to change in angular velocity. By substituting (29) into (27), we get:

$$\omega = \frac{k.u - c\theta}{J} \quad (30)$$

From

$$J\ddot{\theta} + c\dot{\theta} = k.u \quad (31)$$

$$\ddot{\theta} = \frac{k.u - c\dot{\theta}}{J} \quad (32)$$

Equations. (31) to (32), there is still a derivative form, for that it must be changed to (33) and (34), thus the equation is obtained:

$$x[0] = \theta \quad (33)$$

$$x[1] = \dot{\theta} \quad (34)$$

$$\dot{\omega} = \frac{k.u - c.\theta}{J} \quad (35)$$

$$\dot{\omega} = \frac{k.u - c.x[1]}{J} \quad (35)$$

3. RESULTS AND DISCUSSION

In this section, the results of the research are explained of the project uses a robot arm simulation application with a fuzzy logic controller . This simulation uses different inputs for each membership function used since the membership function that has been created through MATLAB which is used to determine the fuzzy set variables has the same value. This simulation was carried out using 3 test responses.

3.1. Response 1

This first robot arm system response calculates the input input with an initial angle of 0 degrees (0^0) and an initial angular velocity of 2 degrees/second (2^0/s), while the desired angle is 4 degrees (4^0).

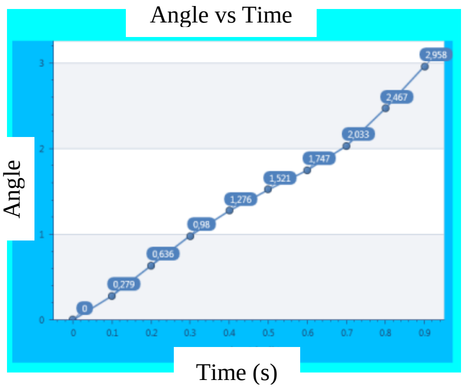

Fig. 8. Response simulation 1 for angular input

From Figs. 8 and 9, it can be seen that the final angle achieved is 3.4 degrees (3.4^0), and the final angular velocity that can be achieved is 4.914 degrees/second (4.914^0/s) since the initial angle is made at 0 degrees therefore this does not happen error in the robot arm system.

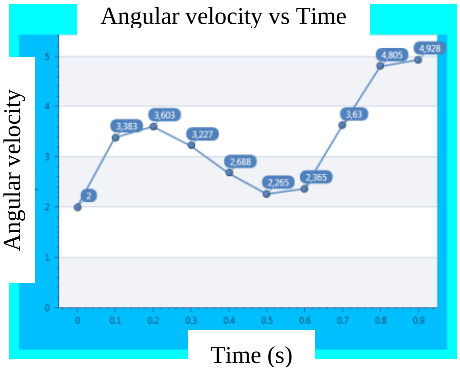

Fig. 9. Response simulation 1 for angular velocity input

3.2. Response 2

Simulation with triangular membership functions

With an input angle of 4 degrees (4^0) and an angular velocity of 2 degrees/second (2^0/s) and the desired angle of 8 degrees (8^0). Simulation results:

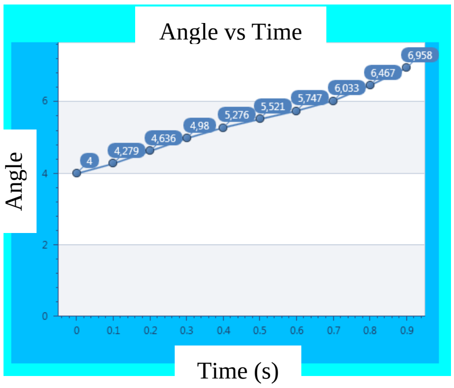

Fig. 10. Simulation with triangular membership function for angle input

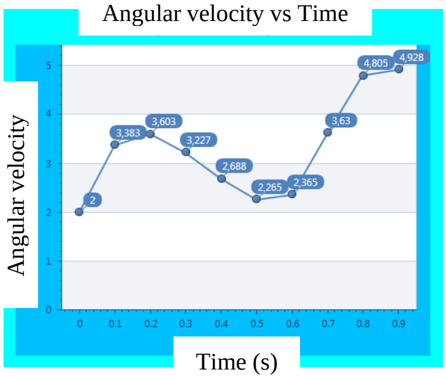

Fig. 11. Simulation with triangular membership function for angular velocity input

Based on Fig. 10, the desired angle value is 8 degrees (8^0) which produces a final angle value of 7.4 degrees (7.4^0) after 1 second (1s). Meanwhile, from Fig. 11, the initial angular velocity value of 2 degrees/second (2^0/s) produces a final angular velocity value of 4.914 degrees/second (4.914^0/s) after 1 second (1s). The resulting torque is 0.380 Nm from both inputs after 1 second (1s).

3.3. Response 3

Simulation with trapezoidal membership function

With an input angle of 2 degrees (2^0) and an angular velocity of 2 degrees/second (2^0/s) and the desired angle of 6 degrees (6^0). Simulation results:

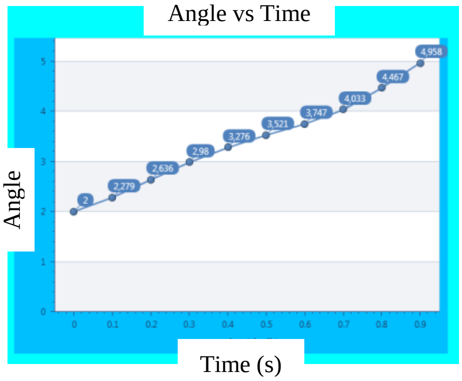

Fig. 12. Simulation with trapezoidal membership function for angle input

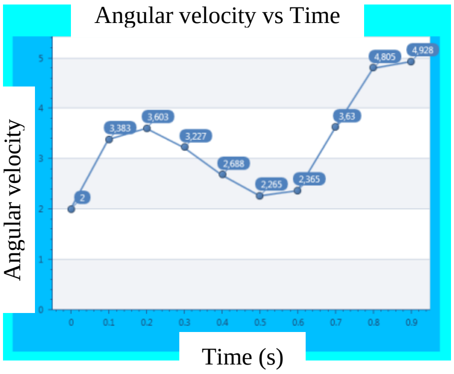

Fig. 13. Simulation with trapezoidal membership function for angular velocity input

Based on Fig. 12, the initial angle input value of 2 degrees (2^0) produces a final angle value of 5.4 degrees (5.4^0) after 1 second (1s). Meanwhile, based on Fig. 13, the initial angular velocity value of 2 degrees/second (2^0/s) produces a final angular velocity value of 4.914 degrees/second (4.914^0/s) after 1 second (1s). The resulting torque is 0.380 Nm from both inputs after 1 second (1s).

3.4. Response 4

In response 4, the desired response results do not meet the desired angle thus the system simulation is unstable. The initial angle given in this system is 1 degree (1^0) and the angular velocity is 2 degrees/second (2^0/s) for the desired angle is 4 degrees (4^0), but in the simulation results it is found that the final angle obtained is - 2.545 degrees (-2.545) and the final angular speed is -11.908 degrees/second (-11.908^0/s) with a torque of 0.505Nm (Fig. 14). The simulation results obtained:

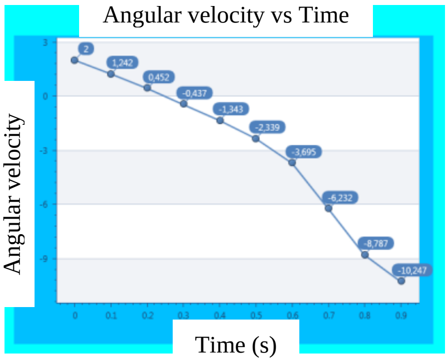

Fig. 14. Response simulation 3 for angular input

In this simulation, there is also an excess of values *(overflow)* in the angle and angular velocity if the given input values are not in accordance with the system and *fuzzy control* thus the error that occurs exceeds the limits permitted for *fuzzy simulation*, as can be seen in the Fig. 15.

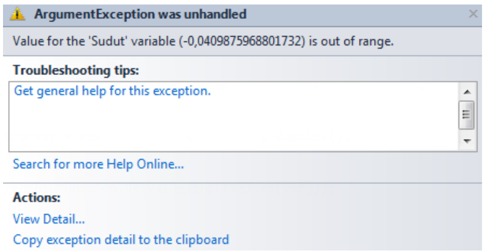

Fig. 15. Response error display 3

Based on the simulation results data, it is known that fuzzy logic is used using 3 robot arm system responses. The response uses 4 membership functions. The results obtained from the variation graph. Response 1 The graphic results obtained are quite stable since there are no errors. Response 2 for the 4 membership functions varies according to the input given, the input given also varies for each membership function used. This simulation is carried out 100 times in repetition where the robot arm will be moved by torque according to the last output for 1 second. Based on the results, the robot arm system that uses fuzzy logic obtains system output results that are quite good/stable and some are not yet stable. The torque which is a fairly good/stable output is the torque which has a value of 0.380 Nm.

The results are quite stable, namely for the triangular membership function and trapezoidal membership function since the input data entered is good enough for this system. Meanwhile, the results that are not yet stable are for generalized bell and Gaussian membership functions since the input data entered does not meet the desired system value. For the last response, namely response 4, is a display of simulation results that occur that exceed the system limit (overflow) thus the *system* cannot to process the output. Based on the results of testing the robot arm system with fuzzy logic control , it produces quite stable values since it depends on the input value when entered into the fuzzy logic and robot arm system.

4. CONCLUSION

In this paper, determining fuzzy rules greatly influences system stability. The simulation investigation was performed by used runge-kutta method. The best output results found on the triangular curve at response 2 since of the difference in values 0.01 on the final angle obtained. From the test results there are errors if the value entered is out from the universe of conversation. fuzzy logic control will only work on systems that are used in accordance with the boundaries that have been determined in the fuzzy set.

REFERENCES

[1] D. Karastoyanov and S. Karastanev, "Reuse of Industrial Robots," *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 44–47, 2018, doi: 10.1016/j.ifacol.2018.11.243.

[2] J. Brüning, B. Denkena, M. A. Dittrich, and H.-S. Park, "Simulation Based Planning of Machining Processes with Industrial Robots," *Procedia Manuf.*, vol. 6, pp. 17–24, 2016, doi: 10.1016/j.promfg.2016.11.003.

[3] R. Shah and A. B. Pandey, "Concept for Automated Sorting Robotic Arm," *Procedia Manuf.*, vol. 20, no. 2017, pp. 400–405, 2018, doi: 10.1016/j.promfg.2018.02.058.

[4] Y. Oktarina, F. Septiarini, T. Dewi, P. Risma, and M. Nawawi, "Fuzzy-PID Controller Design of 4 DOF Industrial Arm Robot Manipulator," *Comput. Eng. Appl. J.*, vol. 8, no. 2, pp. 123–136, 2019, doi: 10.18495/comengapp.v8i2.300.

[5] A. T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, "Fuzzy control systems: Past, present and future," *IEEE Comput. Intell. Mag.*, vol. 14, no. 1, pp. 56–68, 2019, doi: 10.1109/MCI.2018.2881644.

[6] Y. Ohyama, *Feedback control*. Tokyo, Japan, 1998. doi: 10.1007/978-981-13-5770-1_7.

[7] S. Kusumadewi, *Artificial Intelligence (Techniques and Applications)*. Yogyakarta: ANDI, 2003.

[8] J.-S. R. Jang, *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*. London: Prentice Hall, 1997.

BIOGRAPHY OF AUTHORS

Syarifa Fitria was received the B.S. degree in electrical engineering from the Sriwijaya University, Palembang in 2010 and Doctoral degree in environmental science from the University of Sriwijaya, Palembang in 2020. She is currently lecturer in electrical engineering of University of Sriwijaya. Her research interests include ozone production, high voltage, plasma discharge and environmental health.
E-mail: syarifafitria@ft.unsri.ac.id
Orcid ID : 0000-0003-4329-3889

Sariman was received the B.S. degree in electrical engineering from the Sriwijaya University and Master Degree from Institute of Technology Bandung. He is currently lecturer in electrical engineering of University of Sriwijaya. Her research interests include analysis of electric machine, electric machine, advanced control system and optimal control system.
E-mail: sariman@ft.unsri.ac.id

Zainuddin Nawawi was born in Lubuk Linggau, South Sumatera, Indonesia in 1959. He received the B.S. degree in electrical engineering from University of Sriwijaya, Indonesia in 1984 and Ph.D in Electrical Engineering from the Universiti Teknologi Malaysia (UTM), Malaysia in 2011. He is currently Professor in Electrical Department of University of Sriwijaya. His research interests i.e. (1) High voltage phenomenon and materials insulation, and (2) High voltage and protection systems.
E-mail: nawawi_z@yahoo.com

Rizda fitri kurnia was born in South Sumatera Province, Indonesia in 1987.She received the B.S. degree in electrical engineering from the University of Sriwijaya of Indonesia, Palembang in 2008 and Master degree in Electrical Engineering from the Universiti Teknologi Malaysia (UTM), Johor Bahru in 2010. She is currently lecturer in Electrical Engineering of University of Sriwijaya. Her research interests include high voltage engineering, partial discharge and nanodielectrics.
E-mail: rizdafitrikurnia@gmail.com

Tresna Dewi has been a lecturer in Electrical Engineering Department of Politeknik Negeri Sriwijaya, Palembang, Indonesia since 2000. She received Ph.D. Degree from Toyohashi University of Technology, Japan in 2016. Her research interests include robotics, control theory, vision-based control system and its application, and solar energy related research.
E-mail: tresna_dewi@polsri.ac.id
ORCID: https://orcid.org/0000-0003-0239-1538