

Optimizing the Tuning of Fuzzy-PID Controllers for Motion Control of Friction Stir Welding Robots

Eka Marlina¹, Arif Wahjudi², Latifah Nurahmi³, I Made Londen Batan^{4*}, Guowu Wei⁵

^{1,2,3,4} Department of Mechanical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

¹ Department of Mechanical Engineering, Universitas 17 Agustus 1945 Surabaya, Surabaya, Indonesia

⁵ School of Science, Engineering and Environment, University of Salford, Greater Manchester, United Kingdom

Email: ¹ 7007211010@student.its.ac.id, ¹ ekamarliana@untag-sby.ac.id, ² arif_w@me.its.ac.id,

³ latifah.nurahmi@me.its.ac.id, ⁴ londbatan@me.its.ac.id, ⁵ g.wei@salford.ac.uk

*Corresponding Author

Abstract—Friction stir welding (FSW) is defined as a solid-state welding method that is required to be accurate, especially for its motion. This requirement can be satisfied by implementing an accurate controller. The aim of this research was to develop an accurate control system based on a fuzzy-proportional integral derivative (PID) controller for parallel manipulator FSW robots. In order to achieve a higher accuracy in motion control, the tuning optimisation process for a fuzzy-PID controller was conducted using a genetic algorithm (GA) and particle swarm optimisation (PSO). The optimisation algorithms were applied to simultaneously tune the fuzzy rules and output of the membership function from the fuzzy inference system (FIS). The PID controller was designed and tuned using a MATLAB® PID Tuner to obtain the desired response. It was then developed into a fuzzy-PID controller with Sugeno type-1 FIS with 2 inputs and 1 output. The tuning optimisation of the fuzzy-PID controller using GA and PSO was performed to achieve the global minimum integral absolute error (IAE) of the angular velocity. MATLAB® Simulink® was employed to test and simulate the controllers for three motors in the FSW robot model. The IAE values of the PID controller implemented for each motor were 0.03644, 0.04893, and 0.04893. The IAEs of the implemented fuzzy-PID-GA (output and rules) controller were 2.061, 2.048, and 2.048; of the implemented fuzzy-PID-GA (output) controller were 0.03768, 0.05059, and 0.05059; of the fuzzy-PID-PSO (output and rules) controller were 0.01886, 0.0253, and 0.02533; and of the fuzzy-PID-PSO (output) controller were 0.03767, 0.05059, and 0.05059. Therefore, the fuzzy-PID-PSO (output and rules) controller gave the most accurate results and outperformed the others.

Keywords—Angular Velocity; Control System; Friction Stir Welding; Fuzzy-Pid, Genetic Algorithm; Motion; Motor; Parallel Manipulator; Particle Swarm Optimisation.

I. INTRODUCTION

Friction stir welding (FSW) is defined as a solid-state welding method that has to be accurate, precise, and rigid. FSW has rapidly advanced, and numerous studies have demonstrated its practical use for joining various materials used in the automotive, maritime, railway, and construction industries [1]–[3]. However, robot stiffness, rigidity, and motion control system accuracy are commonly expressed concerns since they can

contribute to welding defects. One of the well-known issues in the realm of robotics is the motion control of robot manipulators, particularly for systems with unknown disturbances [4]–[11]. The identical issue occurred during the development of a parallel robot manipulator for FSW. In this work, the FSW robot refers to a prior work [12], where the motion of the FSW robot was generated by an actuator, specifically a DC motor at each actuated limb. The motion generated by the DC motor directly defined the pose and position of the robot's end-effector, as described in the three-prismatic-universal-universal (3-PUU)-based FSW robot system subsection. The position of the robot's end-effector is important because the FSW tool is located in that exact position. Thus, motion control is important for robot manipulators, specifically for FSW robots.

Numerous strategies have been developed for the control of robot manipulators, including the adaptive proportional integral derivative (PID) approach [4], [13]–[18], hybrid position/force [1], deflection/temperature control [19], artificial neural network (ANN)-based [20]–[26], sliding mode control [27]–[30], and radial basis function neural network (RBFNN) [31]. However, despite the achieved accuracy, these aforementioned methods are complex and expensive in terms of computational cost. On the other hand, in recent years, a traditional controller, such as the PID controller, has been developed to the next level to enable it to achieve a higher level of accuracy with a less complex and less expensive computational cost. A PID controller is the most basic and exact approach for real-world robotic control problems compared to other ways [4], [32], [33]. Due to the high accuracy required by FSW robots, it is important to improve the existing accuracy level of conventional PID controllers.

One promising solution is to combine the PID controller with fuzzy logic, to be later indicated as a fuzzy-PID controller. In other previous research, in order to combine fuzzy logic and PID, the fuzzy logic receives inputs and error changes in the fuzzification process and outputs, (K_P , K_I), and (K_D).



Furthermore, the membership function parameters and fuzzy logic rule base are identified, and therefore, the error input and output from the fuzzy system are compatible with the entire system [53]–[55]. Additionally, the Mamdani fuzzy logic with triangular membership function type is utilised [52]. In this work, in order to compensate for the lack of expert knowledge in generating a fuzzy rule base, a different approach was implemented. Here, fuzzy logic was employed as the control input regulator. Therefore, for different input combinations of error ((E)) and error change ($\Delta(E)$) different outputs were produced appropriately.

Furthermore, prior research was done by [34], [35] and [36]–[39], [41]–[44] to explore the application of a PID controller with fuzzy logic for motion control. Previously, the research compared the performance of a PID controller and a combination of fuzzy logic with a traditional PID controller in regulating the speed of a DC motor [45]–[47]. It was found that the fuzzy-PID controller was better able to control the speed of the DC motor compared to systems with conventional PID controllers. In fuzzy logic, the fuzzy inference system (FIS) parameters need to be tuned to get a response with the minimum feasible error, and thus, higher accuracy. There are several options for tuning optimisation methods, such as genetic algorithm (GA) and particle swarm optimisation (PSO).

Previously, Demidova et al. conducted research on a fuzzy PID-based adaptive control system with GA auto-tuning [48]–[51]. In those aforementioned studies, the results showed an improvement in the accuracy of the control system. The GA method itself offers several advantages. Firstly, it excels in global optimisation. Secondly, it is inherently robust, less sensitive to noise and uncertainties, and adapts well to environmental changes, making it suitable for real-world applications [49], such as for FSW robots. It has been shown that the fuzzy-PID-GA is appropriate for solving control problems in a dynamic motion system [48].

The other promising tuning optimisation method is PSO, which can provide solutions of the same high quality but with less effort. PSO itself is a basic method that can be utilised to solve optimisation issues across a wide range of functions [56]. Furthermore, a study with regard to fuzzy-PID based on PSO was carried out by Wang et al. Based on this research on real value functions, simulated results were achieved where the fuzzy-PID-PSO was superior to the basic PSO (SAPSO), chaotic PSO (CLSPSO), and hybrid PSO (BreedPSO) regarding convergence speed and search capabilities. So, it can be inferred that compared to the SAPSO, CLSPSO, and BreedPSO, the fuzzy-PID-PSO offers benefits in terms of search time and accuracy [57]. Another research by Liu et al. revealed that an improved controller using fuzzy-PID-PSO considerably increases the response speed, tracking accuracy, and follower characteristics of a system. It can be concluded that the fuzzy-PID-PSO controller gives a better performance compared to the

fuzzy-PID controller without tuning optimisation [58].

To the best of the authors' knowledge, fuzzy-PID-based controllers with GA and PSO tuning optimisations for FSW robots have yet to be synthesised. Thus, in this work, tuning optimisations were performed on two different configurations: (1) the output of the FIS, and (2) the output and rule base of the FIS. Each configuration was tuned using GA and PSO optimisation methods by minimizing the integral absolute error (IAE) of the desired angular velocity. Hence, there were four novel controllers. The design, tuning, testing, and simulation processes were done with MATLAB® Simulink®. The results were compared with those of a traditional PID controller to obtain the best results between those controllers. The development of a fuzzy-PID control system was proposed to acquire a highly accurate control system for a 3-PUU-based FSW robot. Thus, this study dealt with the fuzzy rules based on the output and error. Moreover, the GA and PSO were applied for the optimisation procedure to produce the best suitable fuzzy rules.

The several contributions of this presented work include:

- (1) The adoption of the GA tuning optimisation on the output of the FIS in the fuzzy-PID controller for the FSW robot.
- (2) The adoption of the GA tuning optimisation on the output and rules of the FIS in the fuzzy-PID controller for the FSW robot.
- (3) The adoption of the PSO tuning optimisation on the output of the FIS in the fuzzy-PID controller for the FSW robot.
- (4) The adoption of the PSO tuning optimisation on the output and rules of the FIS in the fuzzy-PID controller for the FSW robot.
- (5) The evaluation of the performance of the controller and its comparison with existing controllers based on the IAE.

II. METHOD

The overall method for this work is described in the flowchart shown in Fig. 1. Based on the flowchart, the overall work process is described step-by-step as follows:

- (1) Modelling the mechanical system of the FSW robot based on a 3-PUU parallel manipulator to derive the initial high order transfer function (HOTF) of that system.
- (2) Generating a first-order plus dead time (FOPDT) transfer function based on the initial HOTF.
- (3) Modelling the system with MATLAB® Simulink® and implementing the PID controller. The initial values are set accordingly for K_P , K_I , and K_D .
- (4) Tuning the PID using a MATLAB® PID Tuner. If the IAE ≤ 0.1 rad/s, then continue the process with the tuned K_P , K_I , and K_D .
- (5) Designing the Fuzzy-PID controller.

(6) a. Optimising the tuning of the fuzzy-PID using GA for the FIS output. b. Optimising the tuning of the fuzzy-PID using GA for the FIS output and rules.

(7) a. Optimising the tuning of the Fuzzy-PID using PSO for the FIS output. b. Optimising the tuning of the Fuzzy-PID using PSO for the FIS output and rules.

(8) Comparing the results and choosing the best.

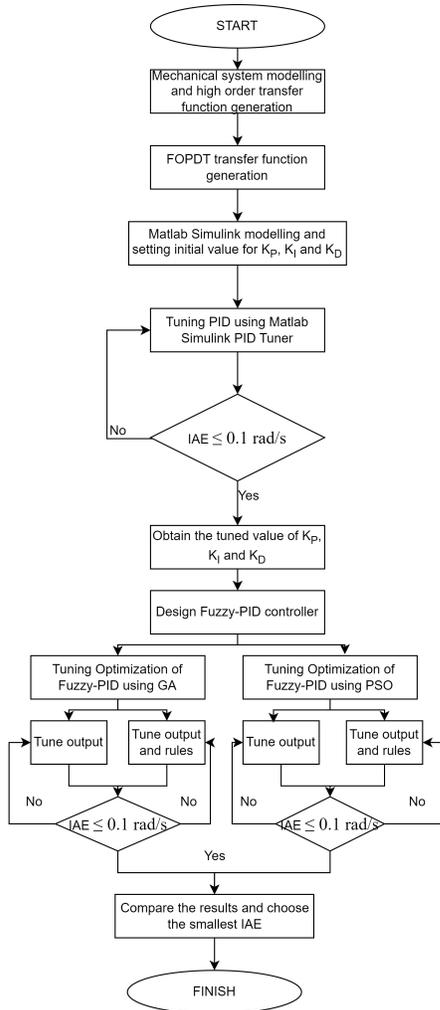


Fig. 1. Flowchart of the overall research process

The detailed method will be explained in the following subsections. Additionally, to avoid the research topic from becoming too broad, the research limitations were as follows: (a) the pitch, yaw, and roll angle of the FSW robot moving platform was equal to zero, (b) the welding path was a linear horizontal line, (c) the research was only in the form of MATLAB® Simulink® simulations, and (d) the tuning configuration of the FIS utilised the default setting of MATLAB® Simulink®.

A. 3-PUU Based FSW Robot System

The 3-PUU parallel manipulator in this paper had to perform a translation motion with 3 degrees of freedom (DoF) and carry a heavy motor load of the FSW tool. Therefore, the structure of the manipulator had to have good controllability features. The 3-PUU parallel manipulator under study was composed of three identical limbs consisting of a prismatic joint (P) and two universal joints (U) in Fig. 2. Each limb was constructed of an actuated prismatic joint (P) that moved along the X-axis of the displacement, q_i , and two universal joints (U). The first universal joint (U) was attached to the prismatic joint (P) denoted by point A_i , and the second universal joint (U) was mounted on the moving platform denoted by point B_i .

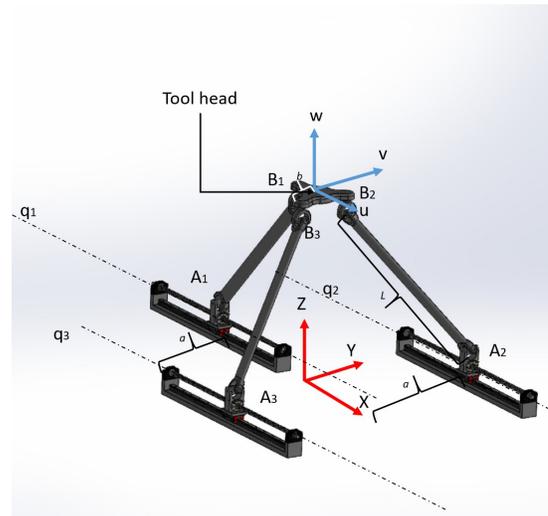


Fig. 2. FSW robot based on 3-PUU parallel manipulator

The moving platform was an equilateral triangle of circum-radius b . The distance between the prismatic joint of the second and third limbs to the X-axis was denoted by a . The link length between points A_i and B_i was defined by l .

The position vectors in this paper were described by a homogeneous coordinate system. The detailed representations of vector points A_i and B_i with respect to the fixed and moving coordinates were respectively written as follows:

$$\begin{aligned} A_1 &= [q_1, 0, 0, 1]^T \\ A_2 &= [q_2, a, 0, 1]^T \\ A_3 &= [q_3, -a, 0, 1]^T \end{aligned} \tag{1}$$

$$\begin{aligned} b_1 &= [-b, 0, 0, 1]^T \\ b_2 &= [\frac{1}{2}b, \frac{1}{2}b\sqrt{3}, 0, 1]^T \\ b_3 &= [\frac{1}{2}b, -\frac{1}{2}b\sqrt{3}, 0, 1]^T \end{aligned} \tag{2}$$

In this paper, the transformation of the global coordinates was carried out utilizing a four-by-four homogeneous transformation

matrix, \mathbf{T} . Moreover, those values were employed for the further computation and analysis of the kinematics. The homogeneous matrix \mathbf{T} was written as follows:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3)$$

where,

$$\mathbf{R} = \begin{bmatrix} C\varphi C\theta & C\varphi S\theta S\psi - S\theta C\psi & C\varphi S\theta C\psi + S\varphi C\psi \\ S\varphi C\theta & S\varphi S\theta S\psi + C\varphi C\psi & S\varphi S\theta C\psi - C\theta S\psi \\ -S\theta & C\theta S\psi & C\theta C\psi \end{bmatrix} \quad (4)$$

$$\mathbf{d} = [X, Y, Z]^T \quad (5)$$

$$\varphi = \theta = \psi = 0 \quad (6)$$

Cosine and sine were respectively denoted by C and S . The homogeneous transformation that was performed to point b_i yielded B_i as follows:

$$B_i = \mathbf{T} b_i, \quad i = 1, 2, 3 \quad (7)$$

The moving platform was unable to rotate as the rotational angles were maintained as $\varphi = \theta = \psi = 0$ as shown in Eq. (6). Hence, the moving platform only achieved translations with 3-DOF. The translation displacement of the middle point of the moving platform was denoted by \mathbf{d} .

B. Hardware and Instrumentation

The schematic representation of the FSW robot testing system is shown in Fig. 3.

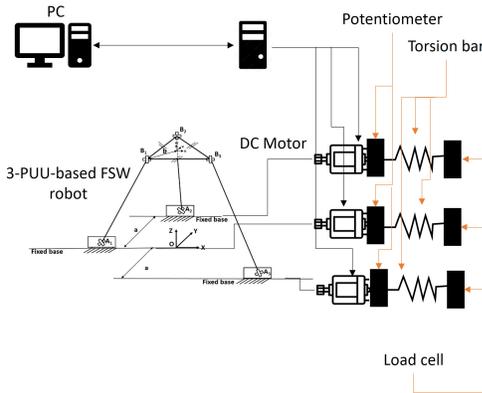


Fig. 3. Schematic of FSW robot testing system

The actuator assembly consisted of a Maxon® DCX 22L precious metal brushes DC motor with a diameter of 22 mm. To enable the translation movement of the limb, a set of ball screws was utilised for each limb. The ball screw employed in this work was a rolled ball screw block nut with a lead screw of 5 mm. There were a total of three DC motors, each of which controlled the rotation of the ball screws. Thus, a translation motion could be achieved in each prismatic joint (P).

A potentiometer was used to regulate the resistance, voltage, and electric current. Furthermore, a torsion bar and load cell were used to measure the force caused by the motion of the overall system. The detailed specifications of the DC motor and ball screws are shown in Table I and II.

TABLE I. SPECIFICATIONS OF MAXON DCX 22L

| | Value | Value |
|-----------------------------|---------|----------|
| Torque constant | 9.73 | mNm/A |
| Stator inductance | 0.035 | mH |
| Stator resistance | 0.343 | Ω |
| Rotor inertia | 9.06 | $g.cm^2$ |
| Back electromotive constant | 0.00102 | V/RPM |

TABLE II. SPECIFICATIONS OF MISUMI ROLLED BALL SCREWS BLOCK NUT

| | Value | Unit |
|--------------------------|----------|------|
| Stroke length | 150-2000 | mm |
| Friction factor (f) | 0.15 | |
| Lead screw (l) | 5 | mm |
| Diameter screw (d_m) | 15 | mm |

C. Modelling the System

1) *Initial transfer function:* The transfer function plant was utilised to model the entire system. It was known that this transfer function plant would be modelled in the FOPDT form [59], [60]. To model a system in this form, it was necessary to know the gain (C), time constant (T), and time delay (L) of the entire system. This subsection explores how the values for these parameters were obtained. Before modelling the transfer function in the FOPDT form, it was necessary to know the initial transfer function model in the initial higher-order form. Previously, it was known that the form of the transfer function plant could be described in Eqs. (8) and (10) as follows:

$$C(s) = \frac{G(s)}{1 - G(s)H(s)} \quad (8)$$

$$G(s) = \frac{K_m}{(Ls + R)(Js + K_f)} \quad (9)$$

$$J = J_r + J_l \quad (10)$$

where $C(s)$ is the closed-loop initial HOTF, $G(s)$ denotes the open loop initial HOTF, and $H(s)$ is the back electromotive force of the motor. Additionally, K_m , L , R , J and K_f denote the torque constant, stator inductance, stator resistance, total inertia, and viscous damping of the motor, respectively. Furthermore, J_r denotes the inertia of the rotor, and J_l the load applied in the motor. Those parameters were obtained from the specifications of the screw and motor shown in Tables I and II.

Based on the desired trajectory and the previous computation [12], the average angular velocity was 0.2667 rad/s. Thus, the torque on the ball screw was obtained by employing Eq.(11).

$$F_e = \frac{2\tau_b}{d_m} \left(\frac{\pi d_m + fl}{l - \pi f d_m} \right) \quad (11)$$

The Eq. (11) was employed to obtain the torque by taking into account the leverage of the ball screw mechanism, where F_e is the average translation force, τ_b the average torque on the ball screw, d_m the screw diameter, l denotes the lead screw, ω the angular velocity, and f the friction factor of the mechanism [58]. The equation shows that the average torque for generating the average translational force was $\tau_b = 0.406$. K_f in the system was computed using Eq. (12) follows:

$$\frac{\text{mean}(\tau_{bs})}{\text{mean}(\omega)} = \frac{0.4060}{0.2667} = 2.3882 \quad (12)$$

The total inertia of the system consisted of the inertia of the limbs and moving platform. The inertia matrix is described in Eqs. (13) and (14):

$$I_p = \begin{bmatrix} 0.041 & 0 & 0 \\ 0 & 0.014 & 0 \\ 0 & 0 & 0.055 \end{bmatrix} \quad (13)$$

$$I_l = \begin{bmatrix} 0.132 & 0 & 0 \\ 0 & 0.132 & 0 \\ 0 & 0 & 0.006 \end{bmatrix} \quad (14)$$

Based on Eqs. (8) - (14), the closed-loop initial HOTF of the system was obtained using Eq. (15) The response based on that transfer function is presented in Fig. 4

$$C(s) = \frac{0.00973}{0.000002142s^2 + 0.02108s + 0.8192} \quad (15)$$

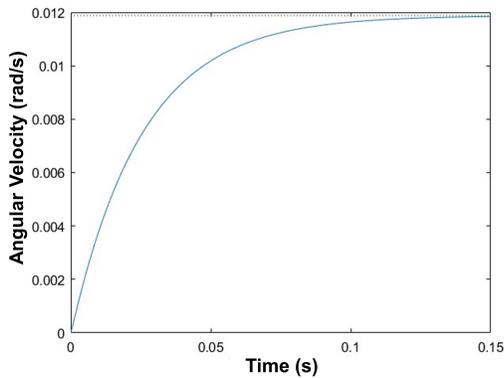


Fig. 4. Step response of the close loop initial higher-order transfer function without tuned controller

2) *First-order plus dead time (FOPDT) transfer function:* Once the closed loop of the initial HOTF is known, the FOPDT transfer function modelling can be carried out. This modelling method was chosen due to the accuracy of the model in mimicking the initial transfer function response, despite its lower-order transfer function [60]. Moreover, a lower-order transfer function made it easier to design a control system with higher accuracy. Thus, this method was implemented in this work. First-order plus dead time (FOPDT) modelling is known to be carried out using the Karim and Riggs model, where the FOPDT equation is [61].

$$C'(s) = \frac{C_e e^{-Ls}}{Ts + 1} \quad (16)$$

$$C_e = \frac{1}{r(t_f) - y(t_f)} \quad (17)$$

$$C_d = \min(T, \frac{L}{2} \times C_e) \quad (18)$$

$$C_0 = \frac{1}{CC_e(\tau_c + \frac{L}{2})} \quad (19)$$

$$C_1 = \max(T, \frac{L}{2}) \times C_0 \quad (20)$$

where, $C'(s)$ and (C) denote the FOPDT transfer function and the system gain, respectively; C_e and C_d denote the scaling factors to normalise the (E) and ΔE ; C_0 and C_1 denote the scaling factors to map the fuzzy logic controller output into the system's input; C , L , T , and τ_c are the system gain, dead time or time delay, time constant, and closed loop time constant, respectively; and $r(t_f)$ and $y(t_f)$ denote the reference and system output, respectively at time t_f .

Based on the response in Fig. 4, manual calculations could be carried out when it was known that without a controller, the system would move constantly at an angular speed of 0.01188 rad/s. Then, the time required for the system to fulfil a steady-state response of 0.02826 s was 0.01050 s. With this data, the C , L , and T values were obtained as 0.001188, 0.000352, and 0.02537, respectively. Furthermore, the FOPDT transfer function modelling was carried out using Eq. (21), Eq. (16)–(20). The responses based on that transfer function are presented in Fig. 5.

$$C'(s) = \frac{0.01188}{0.02537s + 1} e^{-0.000352s} \quad (21)$$

As seen in Fig. 4 and Fig. 5, the two responses were similar, even with different order transfer functions. In Fig. 5, the red line describes the behaviour of the FOPDT system, while the blue one describes the behaviour of the initial high-order system. Thus, the FOPDT transfer function was able to accurately depict the behaviour of the system.

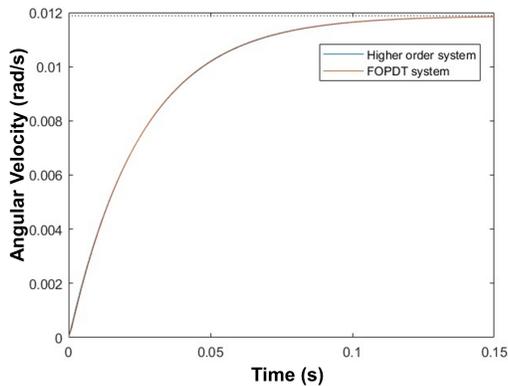


Fig. 5. Step response of the close loop FOPDT versus initial higher-order transfer function without tuned controller

D. Designing the Proportional Integral Derivative (PID) Controller

The PID controller was tuned using the PID Tuner add-on in MATLAB®. The PID Tuner performs tuning automatically by linearizing the plant. Fig. 6 shows the response from the system after the implementation of a PID controller that had been tuned accordingly, thus minimising the IAE. The red line represents the PID-tuned response and the blue one represents the initial step response without the PID tuning. The level of robustness and the response time were set at the intermediate level. Based on the tuning process using the PID parameters via the PID Tuner toolbox, the PID control parameters obtained were $K_P=110.483$, $K_I=8573.405$, $K_D=0.15003$, and $N=736.3952$.

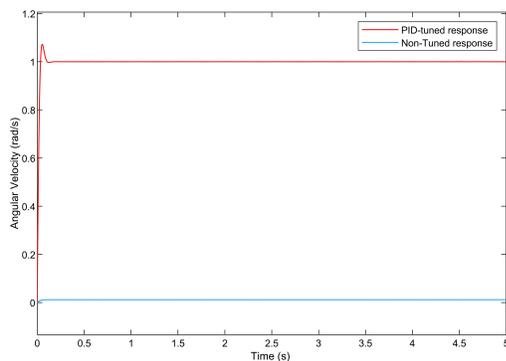


Fig. 6. Comparison between the step system response before and after the PID controller tuned

E. Designing the Fuzzy-Proportional Integral Derivative (PID)-Genetic Algorithm (Fuzzy-PID-GA) Controller

The fuzzy-PID controller was tuned based on the output, and the rule plus the output. The three DC motors (actuators 1, 2, and 3) used in the system were tuned so that they could respond

to the input pulses with minimal IAE. The use of pulses as the input itself was based on the expected behaviour of the system, where the system could respond to transient changes in the input with minimal error [62].

Before tuning the FIS, the gain scaling factor, with which to normalise the PI and PD control phases, had to be calculated. This was accomplished using Eq. (21). As such, the gain scaling factors were $C_e=1$, $C_d=0.000176$, $C_0=8271.90$, and $C_1=209.86$.

Next, an initial system was needed to start the tuning process on the FIS. The initial FIS used was the FIS Sugeno Type-1. It consisted of two inputs in the form of the error (E) and delta error (ΔE), with each having three membership functions in the form of negative (N), zero (Z), and positive (P), and one output in the form of U , which symbolised the angular speed with five member functions in the form of negative big (NB), negative medium (NM), zero (Z), positive medium (PM), and positive big (PB), and consisting of nine rules (Table III).

The initial FIS was the FIS Sugeno type-1 with default settings. Moreover, the triangular membership function, default configuration of membership functions, and rule base were according to the reference [59]. These configurations gave good results for the tuning optimisation process. The initial FIS was tuned to meet the needs of the system using the fuzzy logic designer feature in MATLAB®. The tuning process was conducted twice. The first tuning process was conducted to determine the membership function output parameters and rules, while the second tuning process was conducted to only determine the membership function output parameters. As previously mentioned, the fuzzy-PID controller was tuned to the input pulse using a custom cost function. The custom cost function used was the IAE, with an initial value of 0.1 rad/s. For its implementation on MATLAB® Simulink®, the custom cost function was configured, as shown in Fig. 7, from which was determined the cost of a system with untuned fuzzy-PID after running the input pulse for a certain period. The tuning process moved from this number, where the GA tried to obtain a cost value that was smaller than the value above by changing the working parameters of the initial fuzzy-PID according to the prediction.

Fig. 8 illustrates the GA tuning process. Genetic algorithm (GA) was chosen for this tuning optimisation process as it strives to obtain the global optimum result from all of the generations. They are inspired by the process of natural selection and genetics and mimic the concept of evolution by iteratively evolving a population of potential solutions through selection, crossover, and mutation operations, later known as genetic operators. Genetic algorithms (GAs) employ a population-based search strategy; thus, they maintain a population of potential solutions and iteratively evolve them over generations. The selection, crossover, and mutation operations drive the exploration and exploitation of the search spaces, with exploration and exploitation being emphasised simultaneously.

TABLE III. FIS RULE BASE

| E/ ΔE | N | Z | P |
|---------------|-------------------------|-------------------------|------------------------|
| N | $R_Z^1 : U = NB = -1$ | $R_Z^2 : U = NM = -0.5$ | $R_Z^3 : U = Z = 0$ |
| Z | $R_Z^4 : U = NM = -0.5$ | $R_Z^5 : U = Z = 0$ | $R_Z^6 : U = PM = 0.5$ |
| P | $R_Z^7 : U = Z = 0$ | $R_Z^8 : U = PM = 0.5$ | $R_Z^9 : U = PB = 1$ |

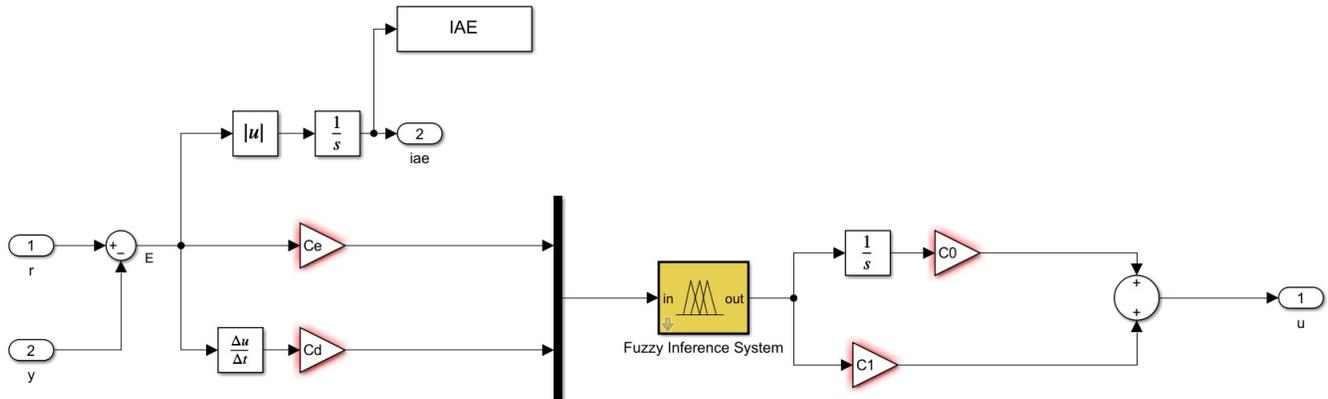


Fig. 7. Custom Cost Function IAE implemented in the Matlab Simulink

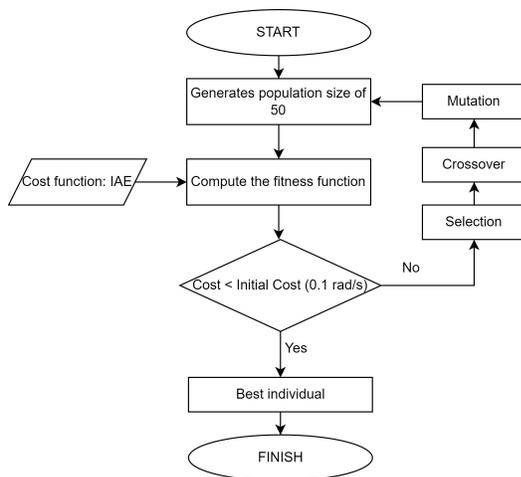


Fig. 8. Flowchart of the GA optimization process with IAE as the cost function

The genetic operators enable exploration by introducing new genetic material, while selection favours exploitation by propagating promising solutions to the next generation. To initialise the tuning process, the "Mersenne twister generator with seed zero for reproducible sequences" was utilised as a random seed generator to tune the FIS process using GA. When the seed is set to zero, the random number generator will start from the same initial state every time to necessarily produce the same sequence of random numbers since consistent and reproducible random sequences are required.

The first tuning process was carried out for the output parameters and rules for the fuzzy-PID controller. Fig. 9 illustrates the result of the process of searching for the smallest cost solution by GA for the FIS with the output and rule base as the tuneable

parameters. The initial value of the IAE was 0.1 rad/s, and thus, the IAE obtained from the tuning process had to be equal to or smaller than this value. The overall tuning was completed after 68 generations when the average change in the fitness value was smaller than the function tolerance set as the default. From the search that was carried out, the FIS was obtained with a new parameter set with a cost of 0.03131 when undergoing the configured input pulse. The new parameter set for the FIS consisted of $NB=-1$, $NM=-0.99751$, $Z=0$, $PM=1$, and $PB=1$.

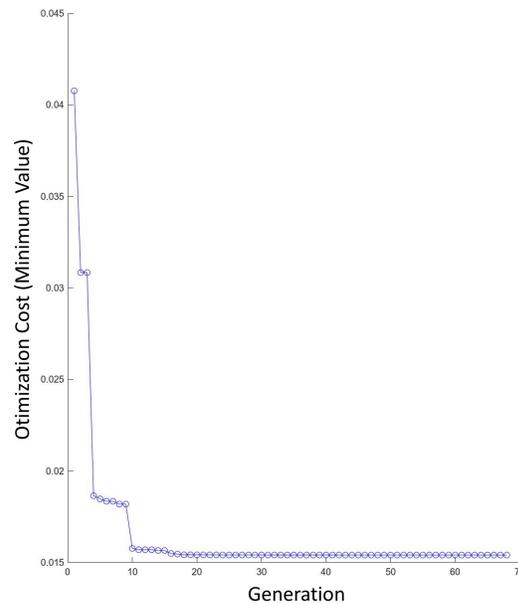


Fig. 9. Total generation of training convergence for output and rule base tuning in GA optimization

In the second tuning process, the results obtained from the tuning process with tuneable parameters were only the membership functions of the output represented in the image. Tuning was completed after 88 generations when the average change in fitness value was smaller than the function tolerance set as the default. From the search that was carried out, the FIS was obtained with a new parameter set with a cost of 0.04071 when undergoing the configured input pulse. The new parameter set for the FIS consisted of $NB = -1$, $NM = -0.99966$, $Z = 0$, $PM = 1$ and $PB = 1$. The training of the tuning process is shown in Fig. 10.

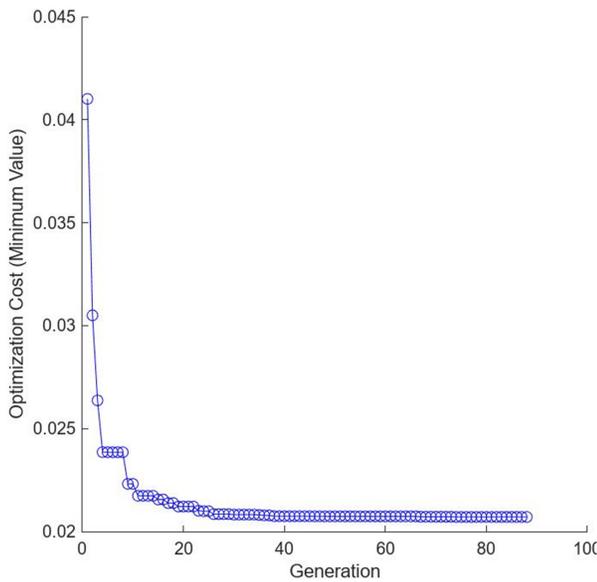


Fig. 10. Total generation of training convergence for output in GA optimization

F. Designing the Fuzzy-Proportional Integral Derivative (PID)-Particle Swarm Optimisation (PSO) Controller

The PSO tuning process is illustrated in Fig. 11. The reason for choosing PSO for this tuning optimisation process was similar to the reason for choosing GA, namely its behaviour, which strives to obtain the global optimum result from all of the generations. PSO is inspired by the collective behaviour of flocks of birds or schools of fish.

It simulates a swarm of particles moving through a search space, with each particle adjusting its position based on the best solution found by itself and its neighbours. PSO utilises a swarm-based search strategy. Each particle adjusts its position and velocity based on its own experience and the best solution found by its neighbours. The particles collectively explore the search space while gradually converging towards promising regions. PSO tends to emphasise exploitation more than exploration. The particles are attracted towards the best solution found so far, which leads to exploitation. Thus, in this

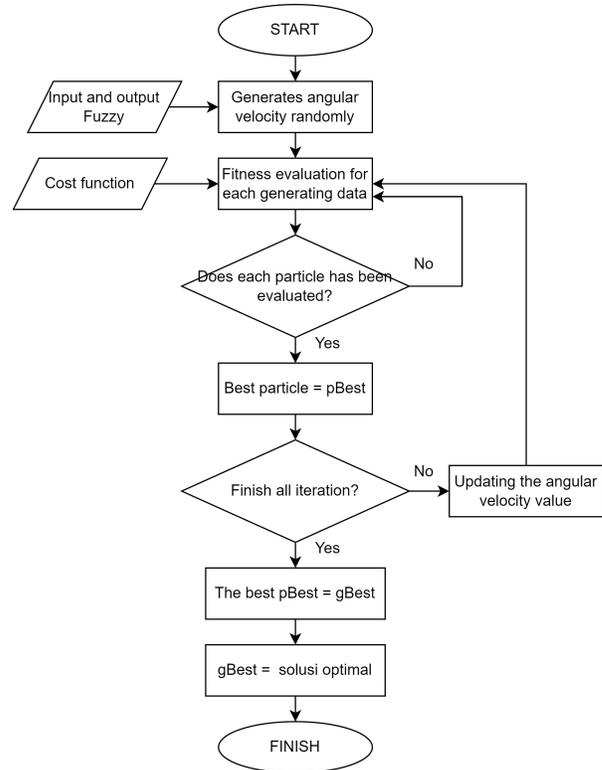


Fig. 11. Flowchart of the PSO optimization process with IAE as the cost function

work, the results from those two methods were compared. Here, the initial IAE value was 0.1 rad/s. Furthermore, in the tuning process using the PSO method, the smallest possible IAE value was obtained by changing the working parameters of the initial FIS.

Fig. 12 depicts the process carried out by the PSO method to find the smallest cost solution for the initial FIS with tuneable parameters in the form of output. The tuning process was completed after 61 iterations when the average change in the fitness value was smaller than the function tolerance set as the default. The new parameter set for the FIS tuning based on the output only consisted of $NB=-1$, $NM=-1$, $Z=1.2086e-10$, $PM=1$, and $PB=0.99993$.

Based on convergence data from the tuning process that was carried out, the FIS was obtained with a new set of parameters with a cost of 0.0207 in running with the input in the form of a pulse with a stop time of 6 s. Fig. 13 depicts a graph of the process of finding the smallest cost solution carried out by the PSO method for the initial FIS with tuneable parameters in the form of rules and output. The tuning process was completed after 110 iterations. The tuning process was automatically completed because the average change in the fitness value was smaller than the function tolerance set as the default. Based on the convergence data from the tuning process that was carried out, the FIS was obtained with a new set of

parameters with a cost of 0.0154 when running with the input in the form of a pulse with a stop time of 6 s. The new parameter set for the FIS tuning based on the output and rules consisted of $NB=-1$, $NM=-1$, $Z=3.9925e-10$, $PM=0.3115$, and $PB=1$.

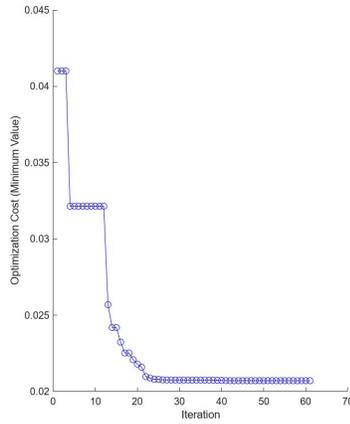


Fig. 12. Total generation of training convergence for output in PSO optimization

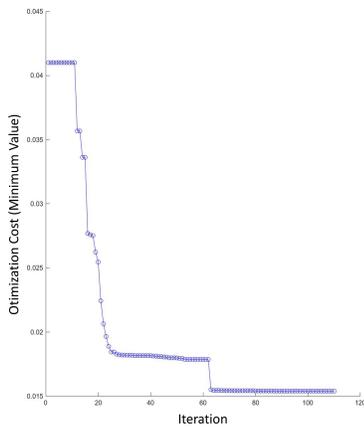


Fig. 13. Total generation of training convergence for output and rules in PSO optimization

III. RESULTS AND DISCUSSION

A. Simulation Results of the Fuzzy-Proportional Integral Derivative (PID)-Genetic Algorithm (Fuzzy-PID-GA) Controller

1) *Simulation results with steps as inputs:* The simulation results of the suggested fuzzy-PID control systems optimised by GA are presented in this section. In this work, the simulations were performed under two different inputs. The first simulation was performed using the step input so that the differences in response could be easily spotted. The results of this simulation are shown in Fig. 14.

The performance matrices utilised in this work were the rise time, settling time, overshoot, and IAE. The rise time signified

the time taken by the system’s output to transition from a specified low value to a specified high value. The settling time was typically measured from the instant the system’s response first entered the tolerance limit until it remained within that limit. The overshoot was expressed as a percentage of the amount by which the system’s response exceeded its final steady-state value. The IAE signified the cumulative absolute error between the system’s output and the desired response over a specified period. In this simulation, the input, illustrated as a green line, was the angular velocity in the step function. Based on Fig. 14, the fastest rise time and settling time were obtained by applying the values of 0.008932 s and 0.014 s, respectively to the fuzzy-PID-GA (rule & output) controller. The smallest overshoot was obtained by applying to the fuzzy-PID-GA (output) controller the value of -0.52 %, with the minus value indicating there was an undershoot instead of an overshoot. The least IAE was obtained by applying the value of 0.009782 rad/s to the fuzzy-PID-GA (output) controller. The detailed results are shown in Table IV. The comparison

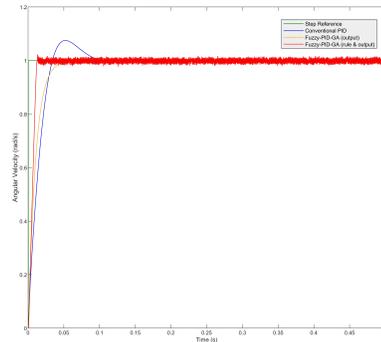


Fig. 14. Step responses comparison with implementing Fuzzy-PID-GA

between the previous design and the proposed controllers is shown in Table V. In the previous work by Singh [63], the rise time, settling time and overshoot are 0.1 s, 0.121 s, and 5.8466%, respectively. In another article by Chao [17], a better performance was obtained with a rise time of 0.044 s, a settling time of 0.054 s, and an overshoot of 2.23%. Additionally, the performance of the PID controller was also given in this article, with a rise time of 0.024914 s, a settling time of 0.125 s, and an overshoot of 6.989%. Moreover, the proposed fuzzy-PID-GA (rule & output) controller gave a slightly better performance with a rise time of 0.022446 s, a settling time of 0.064 s, and an overshoot of -0.52%. The other proposed controller based on GA optimisation, which was the fuzzy-PID-GA (output) controller, also gave a relatively good response, where the rise time was 0.008932 s, settling time was 0.014 s, and the overshoot was 1.83%. Thus, the best rise time and settling time were achieved by implementing the fuzzy-PID-GA (output) controller. Furthermore, the best overshoot could be obtained by applying the fuzzy-PID-GA (rule & output) controller.

According to the simulation results, the implementation of

TABLE IV. IAE COMPARISON WITH IMPLEMENTING FUZZY-PID-GA CONTROLLER

| Parameter | PID | Fuzzy-PID-GA (rule & output) | Fuzzy-PID-GA (output) |
|-------------|----------|------------------------------|-----------------------|
| IAE (rad/s) | 0.020078 | 0.01035 | 0.009782 |

TABLE V. PERFORMANCES COMPARISON WITH IMPLEMENTING FUZZY-PID-GA CONTROLLER

| Parameter | Singh's Fuzzy-GA [63] | Chao's Fuzzy-PID-GA [17] | PID | Fuzzy-PID-GA (output) | Fuzzy-PID-GA (rule & output) |
|-------------------|-----------------------|--------------------------|----------|-----------------------|------------------------------|
| Rise time (s) | 0.1 | 0.044 | 0.024914 | 0.022446 | 0.008932 |
| Settling time (s) | 0.121 | 0.054 | 0.125 | 0.064 | 0.014 |
| Overshoot (%) | 5.8466 | 2.23 | 6.989 | -0.52 | 1.83 |

the fuzzy-PID-GA (output) controller gave a smooth response, with a relatively small rise time and settling time, compared to the conventional PID controller response. Due to the smooth system response, there would not be any higher accumulated IAE. Additionally, the fuzzy-PID-GA (rule & output) controller gave a high oscillation response, and therefore, the accumulated IAE could be larger over time.

Despite the oscillation response, the overshoot was still lower than that of the conventional PID controller, which was less than 0.02 rad/s. The oscillatory behaviour of this controller was caused by the chosen value of the rules in the GA optimisation process. The detailed and enlarged version of the comparison of the controllers can be seen in Fig. 15. It can be said that both the fuzzy-PID-GA controllers gave better results than the conventional PID controller. The rise time and settling time of the fuzzy-PID-GA (rule & output) controller were the fastest compared to the conventional PID and fuzzy-PID-GA (output) controllers. The FSW robot had to be precise and rigid in its application. Thus, it was important for the system response to be always stable and smooth, with less overshoot. A high overshoot and high oscillation response can lead to the formation of welding defects, such as flash. Moreover, it will cause the seam welding result to be less accurate. Hence, for the FSW application, a smooth and faster response from the fuzzy-PID-GA (output) controller was preferable, even though its settling time and rise time were slightly slower than those of the fuzzy-PID-GA (rule & output) controller.

Genetic algorithm (GA) optimisation tends to be robust in finding good solutions, even in the presence of noise or uncertainties in the problem. The population-based nature of the GA helps to maintain diversity and avoid getting trapped in local optima. Unfortunately, it may converge prematurely, meaning it may get stuck in local optima instead of finding the global optimum. This can happen if the population size is too small or if the genetic operators are not appropriately tuned. Since the GA is required to evaluate the fitness of each individual in the population, thus, it is computationally expensive for problems with complex fitness functions or simulations. Hence, for a system with a slightly slower response with the least IAE and smooth response, the fuzzy-PID-GA (output) controller is the best choice. If the system's response needs to be faster and the maximum oscillation is $\leq 1.85\%$, then the fuzzy-PID-GA (rule & output) controller is the best option. Moreover, the

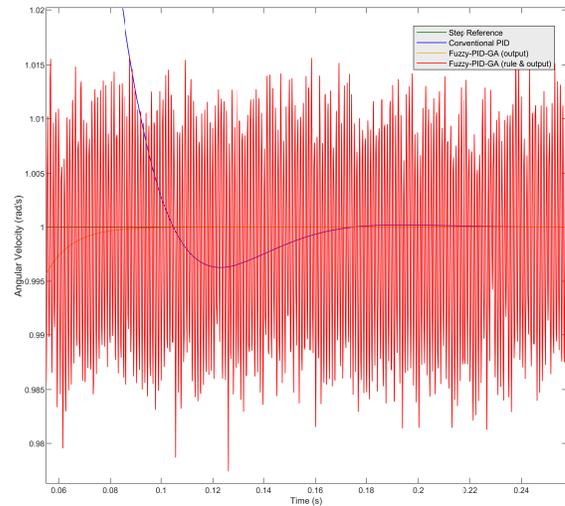


Fig. 15. Enlarge figure of the step responses comparison with implementing Fuzzy-PID-GA

implementation of the fuzzy controller with GA optimisation in the tuning process led to an improvement in the overall performance of the system's response. In future work, the implementation of the proposed controller has to be done with a real FSW robot system. Therefore, the real disturbance and noise can be considered, and hence, it can directly impact the system's response.

2) *Simulation results with trapezoidal trajectories as inputs:* The next simulation was performed using the trapezoidal input, as stated in the previous work [12]. Here, the controller was implemented for each DC motor separately, and the performance of the controller was measured using only the IAE value. For Motor #1, the smallest error obtained by implementing the PID was 0.03664.

For Motor #2, the smallest error obtained by implementing the PID was 0.04893. However, this result was not so different from the implementation of the fuzzy-PID-GA (output) controller. Lastly, for Motor #3, the result was the same as for Motor #2 since the trajectory was similar. The details of the IAE values for each motor are depicted in Table VI. Here, the optimisation process using GA was implemented in the fuzzification process, thus enabling the most suitable rules with

the least error to be chosen. As seen in Fig. 14, the fuzzy-PID-GA (rule & output) controller responded by oscillating, and thus, the IAE was higher than for the other controllers, but the settling time and rise time were faster. It is important to understand the main goal of the desired system, whether it is to be the fastest one or the one with the least IAE. The fuzzy-PID-GA (rule & output) controller is the controller of choice if the aim is to obtain the fastest response. Moreover, if the goal is to obtain the least error or IAE, then the PID or fuzzy-PID-GA (output) controller would be the best choice. The data in Table V shows that the high IAE value of the fuzzy-PID-GA (rule & output) controller can be affected by the accumulation of errors due to the oscillation in the longer term since this trajectory was simulated for 250 s.

TABLE VI. IAE COMPARISON WITH IMPLEMENTING FUZZY-PID-GA CONTROLLER FOR 3 DC MOTORS OF 3-PUU

| IAE | Motor 1 | Motor 2 | Motor 3 |
|---|---------|---------|---------|
| PID | 0.03644 | 0.04893 | 0.04893 |
| Fuzzy-PID-GA (rule & output) | 2.061 | 2.048 | 2.048 |
| Fuzzy-PID-GA (output) | 0.03768 | 0.05059 | 0.05059 |

B. Simulation Results of the Fuzzy-Proportional Integral Derivative (PID)-Particle Swarm Optimisation (PSO) Controller

1) *Simulation results with steps as input:* The simulation results of the suggested control systems with fuzzy-PID optimised by PSO are presented in this section. In this work, the simulations were performed under two different inputs. The first simulation was performed using the step input so that the differences in the responses could be easily spotted. The results of this simulation are shown in Fig. 16. The performance matrices utilised in this work were the rise time, settling time, overshoot, and IAE. The rise time signified the time taken by the system's output to transition from a specified low value to a specified high value. The settling time was typically measured from the instant the system's response first entered the tolerance limit until it remained within that limit. The overshoot was expressed as a percentage of the amount by which the system's response exceeded its final steady-state value. The IAE signified the cumulative absolute error between the system's output and the desired response over a specified period. Based on Fig. 16, the fastest rise time and settling time obtained by applying the fuzzy-PID-PSO (rule & output) controller were 0.01427 s and 0.04 s, respectively. The smallest overshoot obtained by applying the fuzzy-PID- PSO (rule & output) controller was -0.518%, the minus value indicating there was an undershoot instead of an overshoot. The least IAE of 0.009 rad/s was obtained by the fuzzy-PID-PSO (rule & output) controller. The detailed results are shown in Table VIII.

The comparison between the previous design and the proposed controllers is shown in Table VII. In the previous work by Liu [58], the rise time, settling time, and overshoot were 0.5

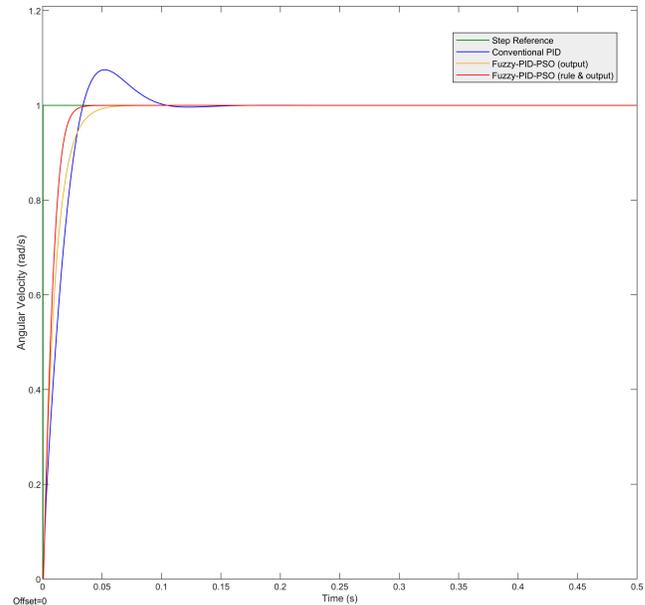


Fig. 16. Step responses comparison with implementing Fuzzy-PID-PSO

s, 1.4 s, and 0%, respectively. Additionally, in this article, the performance of the PID controller was also determined with a rise time of 0.024914 s, a settling time of 0.125 s, and an overshoot of 6.989%. Moreover, the proposed fuzzy-PID-PSO (rule & output) controller gave a slightly better performance with a rise time of 0.01427 s, a settling time of 0.04 s, and an overshoot of -0.518%. The other proposed controller based on PSO optimisation, which was the fuzzy-PID-PSO (output), also gave a relatively good response with a rise time of 0.022416 s, a settling time of 0.071 s, and an overshoot of -0.52%. Thus, the best rise time, settling time, and overshoot were achieved by implementing the fuzzy-PID-PSO (rule & output) controller.

According to the simulation results, the implementation of both controllers gave a smooth response, with a relatively smaller rise time and settling time compared to the response of the conventional PID controller. The rise time and settling time of the fuzzy-PID-PSO (rule & output) controller were the fastest compared to the conventional PID and fuzzy-PID-PSO (output) controllers. The FSW robot needs to be precise and rigid in its application. The high overshoot and high oscillated response could lead to the formation of welding defects such as flash. Moreover, it would cause the seam welding result to be less accurate. Therefore, for the FSW application, a smooth and faster response from the fuzzy- PID-PSO (rule & output) controller was preferable.

Hence, the fuzzy-PID-PSO (rule & output) controller is the best choice for a system that needs a faster response, the least IAE, and a smooth response. Moreover, the implementation of the fuzzy controller with PSO optimisation in the tuning process led to an overall improvement in the performance of

TABLE VII. PERFORMANCE COMPARISON WITH IMPLEMENTING FUZZY-PID OPTIMIZED BY PSO CONTROLLER

| Parameter | Liu's Fuzzy-PID-PSO [58] | PID | Fuzzy-PID-PSO (output) | Fuzzy-PID-PSO (rule & output) |
|-------------------|--------------------------|----------|------------------------|-------------------------------|
| Rise time (s) | 0.5 | 0.024914 | 0.022416 | 0.01427 |
| Settling time (s) | 1.4 | 0.125 | 0.071 | 0.04 |
| Overshoot (%) | 0 | 6.989 | -0.52 | -0.518 |

TABLE VIII. IAE COMPARISON WITH IMPLEMENTING FUZZY-PID-PSO CONTROLLER

| | PID | Fuzzy-PID-PSO (rule & output) | Fuzzy-PID-PSO (output) |
|-------------|----------|-------------------------------|------------------------|
| IAE (rad/s) | 0.020078 | 0.009 | 0.01397 |

the system's response. In future work, the implementation of the proposed controller has to be done with an actual FSW robot system so that the real disturbance and noise can be considered. Hence, this can directly impact the system's response.

2) *Simulation results with trapezoidal trajectories as inputs:* The next simulation was performed using a trapezoidal input, as stated in [12]. In this simulation, the controller was implemented for each DC motor separately. Here, the performance of the controller was measured using the value of the Integral of Absolute Error (IAE). For Motor #1, the smallest error obtained by implementing the fuzzy-PID-PSO (rule & output) controller was 0.01866. For Motor #2, the smallest error obtained by implementing the fuzzy-PID-PSO (rule & output) controller was 0.0253. Lastly, for Motor #3, the smallest error obtained was the same as Motor #2. The detailed IAE value for each motor is depicted in Table IX.

TABLE IX. IAE COMPARISON WITH IMPLEMENTING FUZZY-PID-PSO CONTROLLER FOR 3 DC MOTORS OF 3-PUU

| IAE | Motor 1 | Motor 2 | Motor 3 |
|-------------------------------|---------|---------|---------|
| PID | 0.03644 | 0.04893 | 0.04893 |
| Fuzzy-PID-PSO (rule & output) | 0.01886 | 0.0253 | 0.02533 |
| Fuzzy-PID-PSO (output) | 0.03767 | 0.05059 | 0.05059 |

In this work, the optimisation process using PSO was implemented in the fuzzification process so that the most suitable rules with the least error could be chosen. It is important to understand what is the main goal of the desired system. Overall, the fuzzy-PID-PSO (rule & output) controller gave the fastest response with the lowest IAE and percentage of overshoot.

C. Sensitivity Analysis

The sensitivity analysis of the proposed control system is an important step in assessing the system's robustness and performance. It helps to determine the sensitivity changes and characteristics of the control system, and thus, the potential areas of improvement can be identified. In this work, the Bode diagram, which consists of a phase and magnitude plot, was utilised as the analysis tool. To perform the sensitivity analysis using those tools, several aspects had to be evaluated, namely, the phase sensitivity, and gain sensitivity.

A phase-frequency diagram provides insights into the phase response of a control system at different frequencies. It illus-

trates how the system introduces phase shifts as the input varies. Hence it shows the critical frequency ranges and regions where the phase shift is sensitive to variations in the parameters. This tool evaluates how the phase response of a control system is affected by changes to the parameters. Large shift phases lead to instability. Therefore, it is important to assess the phase sensitivity and ensure that the system remains within the acceptable range of limits. The phase-frequency diagram for all the controllers is shown in Fig. 17.

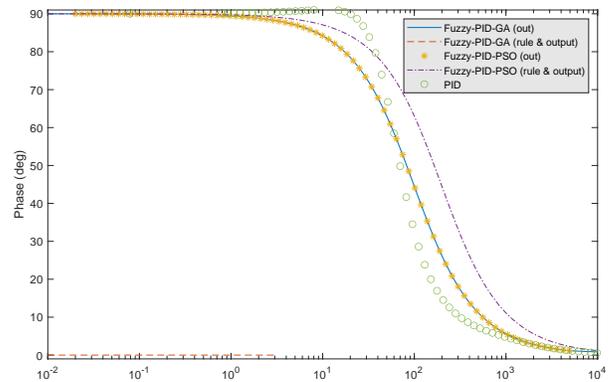


Fig. 17. Phase vs frequency diagram for all controller

Based on that figure, the phase of the fuzzy-PID-GA (output), fuzzy-PID-PSO (output), and fuzzy-PID-PSO (rule & output) controllers shifted smoothly between 90-0°. For comparison, the phase in the PID controller tended to increase in frequency at 80 rad/s and exceeded the phase of 90°, before shifting towards 0°. Additionally, there was no phase change with the fuzzy-PID-GA (rule & output) controller. This means that changes in the system parameters would not affect the phase shift introduced by the system, which could be beneficial for system stability and performance.

The sensitivity gain is directly related to the response of the control system to changes in the amplification factor or gain. The system's change in gain with various frequencies was examined using a magnitude plot. The significant fluctuations and deviations in the gain at certain frequencies signified that the system was more sensitive to variations in the gain at those specific frequencies. A higher sensitivity to gain variations may

lead to a degraded performance. The magnitude diagram for the proposed controllers is illustrated in Fig. 18. Based on that figure, the magnitude shifted smoothly from -60 to 0 dB for the fuzzy-PID-GA (output), fuzzy-PID-PSO (output), and fuzzy-PID-PSO (rule & output) controllers. As a comparison, the magnitude of the PID controller converged smoothly from -60 to 0 dB, but at a frequency of 80 rad/s, the magnitude exceeded 0 dB and indicated a fluctuation. Moreover, for the fuzzy-PID-GA (rule & output) controller, the magnitude remained constant at 0 dB.

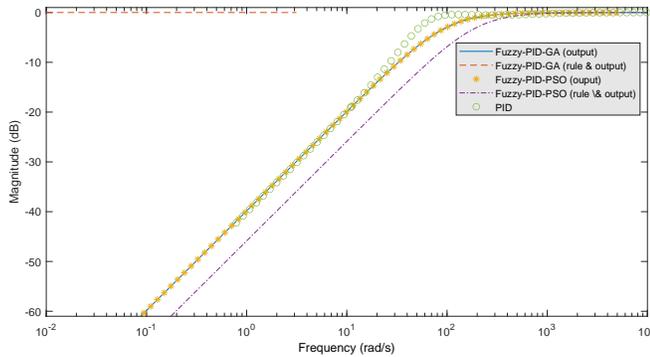


Fig. 18. Magnitude vs frequency diagram for all controller

The above results were concise with the step responses in Fig. 14 and 16. Based on those figures, the responses of the fuzzy-PID-GA (output), fuzzy-PID-PSO (output), and fuzzy-PID-PSO (rule & output) controllers were smooth and stable. These results were achieved due to the small shift phases and lowered sensitivity to gain variations. On the other hand, the fuzzy-PID-GA (rule & output) controller responded by oscillating constantly over time and was on the verge of instability due to its constant shift phase and gain in magnitude. A white noise disturbance was implemented in each of them to compare the robustness of the proposed controller. The computed IAE of each controller is illustrated in Table X.

TABLE X. IAE COMPARISON WITH APPLIED WHITE NOISE

| Controller | IAE |
|-------------------------------|----------|
| PID | 0.020078 |
| Fuzzy-PID-GA (output) | 0.01398 |
| Fuzzy-PID-GA (rule & output) | 0.0882 |
| Fuzzy-PID-PSO (output) | 0.01397 |
| Fuzzy-PID-PSO (rule & output) | 0.009502 |

Based on this data, the lowest IAE (0.009502) was achieved by the fuzzy-PID-PSO (rule & output) controller. Hence, by considering the IAE and other performance matrices from the previous section, the best controller proposed was the fuzzy-PID-PSO (rule & output) controller. For future work, the proposed controller should be applied to a real system in the experimental phase to take into consideration the actual disturbance.

IV. CONCLUSION

The current study sought to obtain a robust control system for an FSW robot based on the 3-PUU parallel robot mechanism. Here, the controlled parameter was the angular velocity of the motor. The system was simplified using FOPDT. The controllers were designed by combining a fuzzy PID with optimisation algorithms (GA and PSO) for the FIS tuning process. The performance matrices utilised in this work were the rise time, settling time, overshoot, and IAE. The following aspects were derived from the present paper:

- A kinematic model of the FSW robot was formulated based on the 3-PUU parallel manipulator.
- The transfer function of the FSW system was derived by utilising the Maxon® DCX 22L and Misumi rolled ball screw block nut.
- The simplification of the transfer function based on FOPDT was performed by introducing several parameters, namely, T , L and C .
- A PID controller was derived based on the FOPDT transfer function of the 3-PUU FSW robot, and thus the gain in the K_P , K_I , and K_D could be obtained.
- Controllers based on the fuzzy PID and tuning optimisation algorithms (GA and PSO) were designed by utilising the gain from the former PID controller.
- Four proposed controllers were derived in this work; (a) fuzzy-PID-GA (output), (b) fuzzy-PID-GA (rule & output), (c) fuzzy-PID-PSO (rule), and (d) fuzzy-PID-PSO (rule & output) controllers.
- To evaluate the proposed technique, a comparative simulation was carried out between the PID, fuzzy-PID-GA, and fuzzy-PID-PSO controllers based on various conditions, with and without white noise disturbance.
- From the simulation without the introduction of white noise disturbance, the least rise time and settling time were achieved by implementing the fuzzy-PID-GA (rule & output) controller, while the least overshoot and IAE were obtained by implementing the fuzzy-PID-PSO (rule & output) controller. The proposed controllers clearly showed a superior performance compared to the conventional PID controller.
- The simulation with disturbances was performed to evaluate the robustness of the proposed controllers. From the simulation with the step function as the trajectory and with the introduction of white noise disturbance, the least IAE was achieved by implementing the fuzzy-PID-PSO (rule & output) controller, which was the most robust among the controllers.
- The FSW robot has to be precise and rigid in its application. The high overshoot and high oscillation response can lead to the formation of welding defects and lower

accuracy. Therefore, for the FSW application, a smooth and moderately fast response from the fuzzy-PID-PSO (rule & output) controller was preferable.

- The sensitivity analysis that was performed showed that the responses of the fuzzy-PID-GA (output), fuzzy-PID-PSO (output), and fuzzy-PID-PSO (rule & output) controllers were smooth and stable. These results were achieved due to the small shift phases and lower sensitivity to gain variations. On the other hand, the response from the fuzzy-PID-GA (rule & output) controller was to oscillate constantly over time and to be on the verge of instability, due to its constant phase shift and gain in magnitude.
- These proposed controllers were limited by the high computational cost during the tuning processes.
- Since the tuning processes were performed separately before the implementation of the proposed controllers, the computational load was relatively low when the FSW robot was working.

In conclusion, the proposed controller design was confirmed to be effective in the simulation phase of the FSW robot based on a 3-PUU parallel manipulator. As a next step, the proposed controller design should be implemented in a real system to take into consideration actual environmental conditions and disturbances. Moreover, in future work, FSW robots can also be combined with image processing for advanced visual sensing. As for the tuning process, an advanced algorithm such as ANN is highly recommended.

ACKNOWLEDGMENT

This work was supported by the Distinguished International Associates of the Royal Academy of Engineering, Contract No. DIA-2022-146. The first author would like to thank the support from BPI-BPPT, LPDP, and Universitas 17 Agustus 1945 Surabaya, Indonesia.

REFERENCES

- [1] M. Soron and I. Kalaykov, "A Robot Prototype for Friction Stir Welding," *2006 IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1-5, 2006, doi: 10.1109/RAMECH.2006.252646.
- [2] Z. Lin, C. Cui, and G. Wu, "Dynamic Modeling and Torque Feedforward based Optimal Fuzzy PD control of a High-Speed Parallel Manipulator," *Journal of Robotics and Control*, vol. 2, no. 6, pp. 527-538, 2021, doi: 10.18196/jrc.26133.
- [3] A. Alipour, M. Mahjoob, and A. Nazarian, "A New 4-DOF Robot for Rehabilitation of Knee and Ankle-Foot Complex: Simulation and Experiment," *Journal of Robotics and Control*, vol. 3, no. 4, pp. 483-495, 2022, doi: 10.18196/jrc.v3i4.14759.
- [4] H. R. Nohooji, "Constrained neural adaptive PID control for robot manipulators," *Journal of the Franklin Institute*, vol. 357, no. 7, pp. 3907-3923, 2020, doi: 10.1016/j.jfranklin.2019.12.042.
- [5] C. E. Luis and A. P. Schoellig, "Trajectory Generation for Multiagent Point-To-Point Transitions via Distributed Model Predictive Control," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 375-382, 2019, doi: 10.1109/LRA.2018.2890572.
- [6] M. M. Ghazaei Ardakani, B. Olofsson, A. Robertsson and R. Johansson, "Model Predictive Control for Real-Time Point-to-Point Trajectory Generation," in *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 972-983, 2019, doi: 10.1109/TASE.2018.2882764.
- [7] S. Xue, Z. Zhao, and F. Liu, "Latent variable point-to-point iterative learning model predictive control via reference trajectory updating," *European Journal of Control*, vol. 65, 2022, doi: 10.1016/j.ejcon.2022.100631.
- [8] Y. Wang, K. Zhu, B. Chen, and H. Wu, "A New Model-Free Trajectory Tracking Control for Robot Manipulators," *Mathematical Problems in Engineering*, vol. 2018, 2018, doi: 10.1155/2018/1624520.
- [9] X. Ma, Y. Zhao, and Y. Di, "Trajectory Tracking Control of Robot Manipulators Based on U-Model," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/8314202.
- [10] M. R. Khoshdarregi, S. Tappe and Y. Altintas, "Integrated Five-Axis Trajectory Shaping and Contour Error Compensation for High-Speed CNC Machine Tools," in *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1859-1871, 2014, doi: 10.1109/TMECH.2014.2307473.
- [11] S. Shentu, F. Xie, X.-J. Liu, and Z. Gong, "Motion Control and Trajectory Planning for Obstacle Avoidance of the Mobile Parallel Robot Driven by Three Tracked Vehicles," *Robotica*, vol. 39, no. 6, pp. 1037-1050, 2021, doi:10.1017/S0263574720000880.
- [12] Y. Li and Q. Xu, "Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation," in *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 265-273, 2007, doi: 10.1109/TMECH.2007.897257.
- [13] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, vol. 79, pp. 172-188, 2018, doi: 10.1016/j.isatra.2018.04.010.
- [14] A. A. El-samahy and M. A. Shamseldin, "Brushless DC motor tracking control using self-tuning Fuzzy PID control and model reference adaptive control," *Ain Shams Engineering Journal*, vol. 9, no. 3, pp. 341-352, 2018, doi: 10.1016/j.asej.2016.02.004.
- [15] M. A. Shamseldin and A. A. EL-Samahy, "Speed control of BLDC motor by using PID control and self-tuning fuzzy PID controller," *15th International Workshop on Research and Education in Mechatronics (REM)*, pp. 1-9, 2019, doi: 10.1109/REM.2014.6920443.
- [16] N. Farouk and T. Bingqi, "Application of self-tuning fuzzy PID controller on the AVR system," *2012 IEEE International Conference on Mechatronics and Automation*, pp. 2510-2514, 2012, doi: 10.1109/ICMA.2012.6285741.
- [17] C. -T. Chao, N. Sutarna, J. -S. Chiou, and C. -J. Wang, "An Optimal Fuzzy PID Controller Design Based on Conventional PID Control and Nonlinear Factors," *Applied Sciences*, vol. 9, no. 6, 2019, doi: 10.3390/app9061224.
- [18] A. Ramya, A. Imthiaz, and M. Balaji, "Hybrid Self Tuned Fuzzy PID controller for speed control of Brushless DC Motor," *Automatika*, vol. 57, no. 3, pp. 672-679, 2017, doi: 10.7305/automatika.2017.02.1769.
- [19] J. D. Backer, *Feedback Control of Robotic Friction Stir Welding*, Doctoral dissertation, University West, 2014.
- [20] I. F. Zidane, Y. Khattab, M. El-Habrouk, and S. Rezeka, "Trajectory control of a laparoscopic 3-PUU parallel manipulator based on neural network in SIMSCAPE SIMULINK environment," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9335-9363, 2022, doi: 10.1016/j.aej.2022.03.024.
- [21] X. Liu, J. Yao, Q. Li, and Y. Zhao, "Coordination dynamics and model-based neural network synchronous controls for redundantly full-actuated parallel manipulator," *Mechanism and Manipulator Theory*, vol. 160, 2021, doi: 10.1016/j.mechmachtheory.2021.104284.
- [22] Z. Chen, Y. Liu, W. He, H. Qiao and H. Ji, "Adaptive-Neural-Network-Based Trajectory Tracking Control for a Nonholonomic Wheeled Mobile Robot With Velocity Constraints," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5057-5067, 2021, doi: 10.1109/TIE.2020.2989711.
- [23] M. Mukhtar, D. Khudher, and T. Kalganova, "A control structure for ambidextrous robot arm based on Multiple Adaptive Neuro-Fuzzy Inference System," *IET Control Theory & Applications*, vol. 15, no. 11, pp. 1518-1532, 2021, doi: 10.1049/cth2.12140.
- [24] M. R. A. Refaai, "Using Multiple Adaptive Neuro-Fuzzy Inference System to Solve Inverse Kinematics of SCARA Robot," *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 154-159, 2021, doi: 10.1109/SSD52085.2021.9429498.

- [25] S. Gobinath and M. Madheswaran, "Deep perceptron neural network with Fuzzy PID controller for speed control and stability analysis of BLDC motor," *Soft Computing*, vol. 24, no. 13, pp. 10161–10180, 2020, doi: 10.1007/s00500-019-04532-z.
- [26] A. Ramya, M. Balaji, and V. Kamaraj, "Adaptive MF tuned Fuzzy logic speed controller for BLDC motor drive using ANN and PSO technique," *The Journal of Engineering*, vol. 2019, no. 17, pp. 3947–3950, 2019, doi: 10.1049/joe.2018.8179.
- [27] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, "Backstepping Trajectory Tracking Based on Fuzzy Sliding Mode Control for Differential Mobile Robots," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 1, pp. 109–121, 2019, doi: 10.1007/s10846-019-00980-9.
- [28] J. Y. Zhai and Z. B. Song, "Adaptive sliding mode trajectory tracking control for wheeled mobile robots," *International Journal of Control*, vol. 92, no. 10, pp. 2255–2262, 2019, doi: 10.1080/00207179.2018.1436194.
- [29] A. Ma'arif and A. Çakan, "Simulation and Arduino Hardware Implementation of DC Motor Control Using Sliding Mode Controller," *Journal of Robotics and Control*, vol. 2, no. 6, pp. 582–587, 2021, doi: 10.18196/jrc.26140.
- [30] A. T. Azar, F. E. Serrano and A. Koubaa, "Adaptive Fuzzy Type-2 Fractional Order Proportional Integral Derivative Sliding Mode Controller for Trajectory Tracking of Robotic Manipulators," *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 183–187, 2020, doi: 10.1109/ICARSC49921.2020.9096163.
- [31] M. Yang, Z. Du, L. Sun, and W. Dong, "Optimal design, modeling and control of a long stroke 3-PRR compliant parallel manipulator with variable thickness flexure pivots," *Robotics and Computer-Integrated Manufacturing*, vol. 60, pp. 23–33, 2019, doi: 10.1016/j.rcim.2019.05.014.
- [32] C. Lauretti *et al.*, "Learning by Demonstration for Motion Planning of Upper-Limb Exoskeletons," *Frontiers in Neurobotics*, vol. 12, 2018, doi: 10.3389/fnbot.2018.00005.
- [33] K. S. Devi, R. Dhanasekaran and S. Muthulakshmi, "Improvement of speed control performance in BLDC motor using fuzzy PID controller," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 380–384, 2016, doi: 10.1109/ICACCCT.2016.7831666.
- [34] J. Xu, L. Xiao, M. Lin, and X. Tan, "Application of Fuzzy PID Position Control Algorithm in Motion Control System Design of Palletizing Robot," *Security and Communication Networks*, vol. 2022, pp. 1–11, 2022, doi: 10.1155/2022/8720960.
- [35] N. Y. Allagui, F. A. Salem, and A. M. Aljuaid, "Artificial Fuzzy-PID Gain Scheduling Algorithm Design for Motion Control in Differential Drive Mobile Robotic Platforms," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, doi: 10.1155/2021/5542888.
- [36] S. Krishna and S. Vasu, "Fuzzy PID based adaptive control on industrial robot system," *Materials Today: Proceedings*, vol. 5, no. 5, pp. 13055–13060, 2018, doi: 10.1016/j.matpr.2018.02.292.
- [37] M. Rabah, A. Rohan, Y. J. Han, and S. H. Kim, "Design of Fuzzy-PID Controller for Quadcopter Trajectory-Tracking," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 204–213, 2018, doi: 10.5391/IJFIS.2018.18.3.204.
- [38] G. S. Maraslidis, T. L. Kottas, M. G. Tsipouras and G. F. Fragulis, "A Fuzzy Logic Controller for Double Inverted Pendulum on a Cart," *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNM)*, pp. 1–8, 2021, doi: 10.1109/SEEDA-CECNM53056.2021.9566228.
- [39] A. M. Şerifoğlu and C. Kasnaoğlu, "Fuzzy Logic Controlled Active Hydro-Pneumatic Suspension Design Simulation and Comparison for Performance Analysis," *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–5, 2021, doi: 10.1109/HORA52670.2021.9461373.
- [40] N. H. Singh and K. Thongam, "Mobile Robot Navigation Using Fuzzy Logic in Static Environments," *Procedia Computer Science*, vol. 125, pp. 11–17, 2018, doi: 10.1016/j.procs.2017.12.004.
- [41] R. Arulmozhiyal, "Design and Implementation of Fuzzy PID controller for BLDC motor using FPGA," *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1–6, 2022, doi: 10.1109/PEDES.2012.6484251.
- [42] H. Maghfiroh, M. Ahmad, A. Ramelan, and F. Adriyanto, "Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink," *Journal of Robotics and Control*, vol. 3, no. 1, pp. 8–13, 2022, doi: 10.18196/jrc.v3i1.10964.
- [43] P. Chotikunnan, R. Chotikunnan, A. Nirapai, A. Wongkamhang, P. Imura, and M. Sangworasil, "Optimizing Membership Function Tuning for Fuzzy Control of Robotic Manipulators Using PID-Driven Data Techniques," *Journal of Robotics and Control*, vol. 4, no. 2, pp. 128–140, 2023, doi: 10.18196/jrc.v4i2.18108.
- [44] R. Kristiyono and W. Wiyono, "Autotuning Fuzzy PID Controller for Speed Control of BLDC Motor," *Journal of Robotics and Control*, vol. 2, no. 5, pp. 400–407, 2021, doi: 10.18196/jrc.25114.
- [45] D. Somwanshi, M. Bunde, G. Kumar, and G. Parashar, "Comparison of Fuzzy-PID and PID Controller for Speed Control of DC Motor using LabVIEW," *Procedia Computer Science*, vol. 152, pp. 252–260, 2019, doi: 10.1016/j.procs.2019.05.019.
- [46] A. Dhyani, M. K. Panda, and B. Jha, "Moth-Flame Optimization-Based Fuzzy-PID Controller for Optimal Control of Active Magnetic Bearing System," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 42, no. 4, pp. 451–463, 2018, doi: 10.1007/s40998-018-0077-1.
- [47] A. Shurajji and S. Shneen, "Fuzzy Logic Control and PID Controller for Brushless Permanent Magnetic Direct Current Motor: A Comparative Study," *Journal of Robotics and Control*, vol. 3, no. 6, pp. 762–768, 2022, doi: 10.18196/jrc.v3i6.15974.
- [48] G. L. Demidova, D. V. Lukichev, and A. Y. Kuzin, "A Genetic Approach for Auto-Tuning of Adaptive Fuzzy PID Control of a Telescope's Tracking System," *Procedia Computer Science*, vol. 150, pp. 495–502, 2019, doi: 10.1016/j.procs.2019.02.084.
- [49] P. Duraisamy, M. N. Santhanakrishnan, and R. Amirtharajan, "Genetic Algorithm Optimized Grey-Box Modelling and Fuzzy Logic Controller for Tail-Actuated Robotic Fish," *Neural Processing Letters*, vol. 55, no. 8, pp. 11577–11594, 2023, doi: 10.1007/s11063-023-11391-1.
- [50] C. Ntakolia, K. S. Platanitis, G. P. Kladis, C. Skliros and A. D. Zagoranos, "A Genetic Algorithm enhanced with Fuzzy-Logic for multi-objective Unmanned Aircraft Vehicle path planning missions," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 114–123, 2022, doi: 10.1109/ICUAS54217.2022.9836068.
- [51] H. Hu, T. Wang, S. Zhao, and C. Wang, "Speed control of brushless direct current motor using a genetic algorithm-optimized Fuzzy proportional integral differential controller," *Advances in Mechanical Engineering*, vol. 11, no. 11, 2019, doi: 10.1177/1687814019890199.
- [52] E. Pourjavad and R. V. Mayorga, "A comparative study and measuring performance of manufacturing systems with Mamdani Fuzzy inference system," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1085–1097, 2019, doi: 10.1007/s10845-017-1307-5.
- [53] N. H. Singh and K. Thongam, "Mobile Robot Navigation Using Fuzzy Logic in Static Environments," *Procedia Computer Science*, vol. 125, pp. 11–17, 2018, doi: 10.1016/j.procs.2017.12.004.
- [54] Z. Hou, Z. Li, C. Hsu, K. Zhang and J. Xu, "Fuzzy Logic-Driven Variable Time-Scale Prediction-Based Reinforcement Learning for Robotic Multiple Peg-in-Hole Assembly," in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 218–229, 2022, doi: 10.1109/TASE.2020.3024725.
- [55] R. D. Puriyanto and A. K. Mustofa, "Design and Implementation of Fuzzy Logic for Obstacle Avoidance in Differential Drive Mobile Robot," *Journal of Robotics and Control*, vol. 5, no. 1, pp. 132–141, 2024, doi: 10.18196/jrc.v5i1.20524.
- [56] R. Hassan, B. Cohanin, and O. de Weck, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 2, 2005, doi: 10.2514/6.2005-1897.
- [57] Z. Wang, Q. Wang, D. He, Q. Liu, X. Zhu and J. Guo, "An Improved Particle Swarm Optimization Algorithm Based on Fuzzy PID Control," *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 835–839, 2017, doi: 10.1109/ICISCE.2017.178.
- [58] Y. Liu *et al.*, "Self-Tuning Control of Manipulator Positioning Based on Fuzzy PID and PSO Algorithm," *Frontiers in Bioengineering and Biotechnology*, vol. 9, 2022, doi: 10.3389/fbioe.2021.817723.
- [59] J. Mendel, *Uncertain Rule-Based Fuzzy Systems*, Springer Cham, 2017, doi: 10.1007/978-3-319-51370-6.

-
- [60] J.C. Muresan and C. Ionescu, "Generalization of the FOPDT Model for Identification and Control Purposes," *Processes*, vol. 8, no. 6, 2020, doi: 10.3390/pr8060682.
- [61] Q. Bi, W. J. Cai, E. L. Lee, Q. G. Wang, C. C. Hang, and Y. Zhang, "Robust identification of first-order plus dead-time model from step response," *Control Engineering Practice*, vol. 7, no. 1, pp. 71–77, 1999, doi: 10.1016/S0967-0661(98)00166-X.
- [62] Z. Zhu *et al.*, "Fuzzy PID Control of the Three-Degree-of-Freedom Parallel Mechanism Based on Genetic Algorithm," *Applied Sciences*, vol. 12, no. 21, 2022, doi: 10.3390/app12211128.
- [63] A. Singh and V. K. Giri, "Design and Analysis of DC Motor Speed Control by GA Based Tuning of Fuzzy Logic Controller," *International Journal of Engineering Research and Technology*, vol. 1, 2012.