

# Dual Design PID Controller for Robotic Manipulator Application

Phichitphon Chotikunnan<sup>1</sup>, Rawiphon Chotikunnan<sup>2\*</sup>

<sup>1,2</sup> College of Biomedical Engineering, Rangsit University, Pathum Thani, Thailand

<sup>2</sup> Department of Information Technology and Digital Innovation, North Bangkok University, Bangkok, Thailand

Email: <sup>1</sup> phichitphon.c@rsu.ac.th, <sup>2</sup> rawiphon.ch@northbkk.ac.th

\*Corresponding Author

**Abstract**— This research introduces a dual design proportional–integral–derivative (PID) controller architecture process that aims to improve system performance by reducing overshoot and conserving electrical energy. A dual design PID controller uses real-time error and one-time step delay to adjust the confidence weights of the controller, leading to improved performance in reducing overshoot and saving electrical energy. To evaluate the effectiveness of a dual design PID controller, experiments were conducted to compare it with the PID controller using least overshoot tuning by Chien–Hrones–Reswick (CHR) technique. The results showed that a dual design PID controller was more effective at reducing overshoot and saving electrical energy. A case study was also conducted as part of this research, and it demonstrated that the system performed better when using a dual design PID controller. Overshoot and electrical energy consumption are common issues in systems that can impact performance, and a dual design PID controller architecture process provides a solution to these issues by reducing overshoot and saving electrical energy. A dual design PID controller offers a new technique for addressing these issues and improving system performance. In summary, this research presents a new technique for addressing overshoot and electrical energy consumption in systems through the use of a dual design PID controller. A dual design PID controller architecture process was found to be an effective solution for reducing overshoot and saving electrical energy in systems, as demonstrated by the experiments and case study conducted as part of this research. A dual design PID controller presents a promising solution for improving system performance by addressing the issues of overshoot and electrical energy consumption.

**Keywords**—Robotic; PID control; Dual design; Dynamic model

## I. INTRODUCTION

Recently, robotic arms are dominant in industrial applications, hospitals, clinics, etc. and are very popular to use robots to assist in various types of tasks. Minimal error is a problem that requires a theoretical approach in many different control systems to solve problems, such as robotic arms that are used to determine the movements at the desired point of motion. The movement has an error within the system as moving parts move from one location to another with repetitive movements from the same position. In practice, each cycle has found that there will be repeating errors and often a range of values always the same.

The research has applications to the welding robot as discussed elsewhere [1], [2] in automatic welding for the joining of stainless steel or commercially pure titanium. By controlling the movement of the robotic arm, various robot

motion control systems have been developed. In most cases, a developer of a mechanical robotics arm tends to develop both mechanical systems [3] and electronic control systems. The most popular development of robotic arms is usually focused on algorithms to control the rotation or movement of electric motors. For example, using the PID control [4] is the basic control in various systems. In the controller of the robotics arm is the current development in PID control, which has been many improvements to this system. It will be a way to adjust the gain value suitable for the system to be flexible, such as using the Fuzzy Neural Network Algorithm as discussed by [5] to design the gain of PID control [6] and control input before coming to the system. There are also other controllers used in the development of system controllers. The adaptive fuzzy system [7], [8] proposed a controller to show higher precision and stronger robustness, optimal control [9], adaptive sliding mode [10], adaptive neural network tracking control [11], etc., to improve the internal control system.

In the robotics arm category, GrblGru [12] is designed with 5 axis manipulator robots and can use G-code to control the robot. Various techniques of the program have been proposed to design MATLAB GUI and arduino IDE [13] in the programming to control the mini robot for application to pick up things or the robotic apply for COVID-19 Specimen Collection Process [14] to help the medical personnel in the future. Simulink, a program of MATLAB, can simulate to test the system in controller design. Various research has been proposed to design the system including Micro-Robotics [15], MICROGRID [16], Mini Drone [17], DC Motor Control [18], [19], BLDC Motor [20], lane-keeping control [21], and hexapod robot [22].

In the design of a motion profile to control the movement of robotic arms, the operator must define the motion path as the movement of a robotic arm. In the control of a program, there will be a point-to-point control [23] and trajectory tracking control [24], [25], in which each control has different advantages and disadvantages. The repetitive process [26] is part of the robot's control to do the same thing again. It may be a batch process or a continuous process with periodic input (continuously periodic process). In the operation of robotic arms, there is often a problem in the error value caused by repetitive motion. It is always the same size as the robotic arms repeat each move. In fact, the user cannot adjust the internal controller in the system. Therefore, users are popular to develop the design using iterative learning control,



repetitive control, and run-to-run control is often applied to solve this problem.

Iterative learning control (ILC) and repetitive control (RC) have a similar workflow, where the difference is that the ILC always has initial parameters before it is restarted, the RC can continue to run the process. Trajectory tracking process control of the robotic arm by controlling the system in a batch process of the ILC control [27] can be applied to develop the robotic arm to achieve the most efficient movement. Basic ILC implementation is typically in the form of a P-type ILC. The learning gain is applied as a 1-dimension or first-order ILC [28], [29]. A dual design iterative learning control (Dual design ILC) [30], which has an idea from the previous works. By extending the technique of using dual integral learners [31], the concept of this study was a neural network control with adjusted learning rates and application to use in MORF robot [32], where two of which were fast learners and slow learners. The adjusted learning rate is fast and slow coordinate work, allowing the system to quickly reduce the error value, regardless of whether the error signal is large or small. The system will be able to adjust the response value appropriately in the implementation of the ILC system with a total of two learners consisting, in the first learner using the concept of fuzzy logic control (FLC) to track both the desired learning control input and the desired trajectory in the gain adjustment mechanism (GAM) [33] in slow learner by using this fuzzy logic control and fast learner uses the technique of designing a repeatable control system with inverse frequency response (IFR) for ILC [34] and RC [35]. It can be reduced RMSE to lower and it can be applied in the system to prevent the occurrence of the transient response in the system.

The use of PID controllers in the development of control systems for mobile robotics has a long history, with early studies dating to [36], [37], [38], [39]. These controllers have become popular in the field of mobile robotics due to their ease of design and testing [40]. In the development of motor systems for mobile robots, PID controllers have been used in a variety of motor types, including DC servo motors [41], DC motors [42], [43], brushless DC motors [44], [45], [46], and modified versions such as those used in electric wheelchairs [47], [48], [49], [50], [51], [52], [53] and balance vehicles with PID control [54], [55], [56]. In some cases, PID controllers have been combined with fuzzy control methods, such as [57], [58], [59], [60], to improve the performance of mobile robots with mecamum wheels and omnidirectional wheels [61], [62], [63], [64]. These advances in control systems have the potential to pave the way for the development of industrial robots in the future.

There are several techniques that have been proposed for tuning the gain of a PID controller, including the Ziegler and Nicholas (ZN) method [65], [66], [67], [68], the Modified ZN (MZN) method [69], [70], [71], the Tyreus-Luyben (TL) method [72], [73], [74], and the CHR tuning method [75], [76]. These techniques can be applied in both open-loop and closed-loop control modes and can be used to find the gain values for a system by substituting known values into pre-estimated equations or tables. This makes it easy to tune a system that requires initial control. However, the use of

optimization algorithms such as genetic algorithm (GA) [77], [78], [79] or particle swarm optimization (PSO) [80], [81] can also provide a more flexible or cost-effective approach to tuning a PID system for specific systems. These algorithms can help to find the optimal gain values for a PID system that are suitable for a particular system and its requirements, but they may require a longer sampling time in order to learn and optimize the system.

The research contribution of this study includes the development and implementation of a dual design PID controller architecture that aims to improve system performance by reducing overshoot and conserving electric energy in systems. This controller architecture uses real-time error and one-time step delay to adjust the confidence weights of the controller, resulting in improved performance in terms of overshoot reduction and electric energy conservation. The effectiveness of a dual design PID controller was also evaluated and compared to a PID controller using the least overshoot tuning technique developed by CHR. Another research contribution of this study is the demonstration of a dual design PID controller's ability to effectively reduce overshoot and save electrical energy in systems through experiments and a case study. These findings offer promising solutions for improving system performance through the use of a dual design PID controller.

## II. METHOD

### A. Robotic manipulator Seiko D-Tran RT3200

The cartesian robot and the block diagram in Fig. 1 and Fig. 2, is a robotic arm structure that is used to manipulate screw nuts. It consists of 4 joints that allow for movement in the X-axis plane (joint R), rotation in the X-Y plane (joints T and A), and lifting and lowering in the Z-axis (joint Z). The robotic arm is controlled by a set of instructions entered through a LabVIEW program using the NI-cRIO board and an Arduino board. These boards work together to control the movement of the robot's joints (A, R, T, and Z) with feedback control. The movement of all joints can be set simultaneously, as shown in Fig. 2. The Seiko D-Tran RT3200 device is used as the controller for the system.

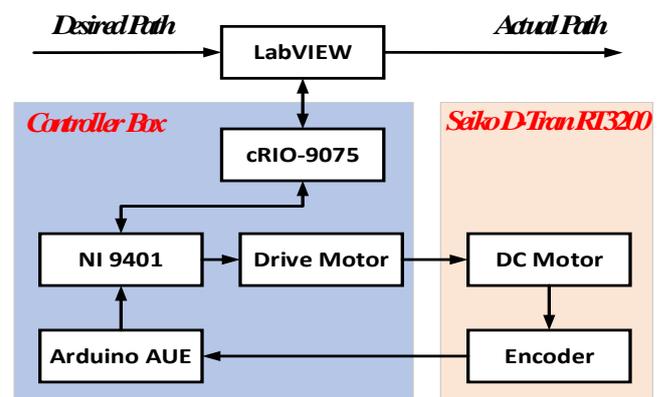


Fig. 1. Block diagram of the overall system.



Fig. 2. Seiko D-Tran RT3200 and device for controller.

The system can be divided into three main parts: the LabVIEW program execution, the controller box, and the Seiko D-Tran RT3200 robot. Fig. 1 and Fig. 2 provide an overview of the system's operation. The LabVIEW program controls the overall operation of the system, including the movement of the Seiko D-Tran RT3200 robotic manipulator. The program receives data from the controller box, which processes signals to control the robot's movement. The cRIO 9075 serves as the main processor, connecting the LabVIEW program to the NI-9401, which controls the H-bridge driver DC motor to rotate the motor. The motor is responsible for the movement of the Seiko D-Tran RT3200 robot's joint A, joint R, joint T, and joint Z. As the motor rotates, the encoder sends data to the Arduino DUE board to process the motion coordinate signal and send it back to the NI-9401. The NI-9401 then sends a signal back to the cRIO 9075, which processes the data and sends it back to the LabVIEW program to be displayed on the computer screen as the target coordinates.

The flowchart in Fig. 3 illustrates the operation of the program written in the LabVIEW program for the Seiko D-Tran RT3200. The program receives motion coordinates for joints A, R, T, and Z, with only joints R, T, and Z being considered in this research. The coordinates are then sent to the main processor. The program waits until the start button is pressed and then moves the robot to the specified coordinates. Once the displacement coordinates have been reached, the program execution ends.

### B. Hardware Implementations

This section explains the components that had been used including embedded controller, digital I/O interface cards, Arduino DUE, and DC motor drivers.

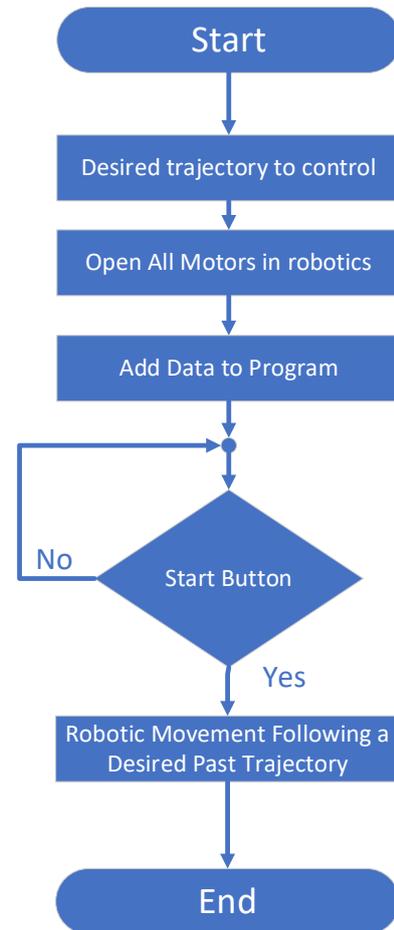


Fig. 3. Program flowchart of Seiko D-Tran RT3200.

The cRIO 9075 is a robust, fanless embedded controller that is well suited for advanced control and monitoring applications. It features a real-time processor and an FPGA, and offers a range of connectivity options, including one ethernet port and one serial port. As shown in Fig. 4, the controller has a 4-slot, 400 MHz CPU, 128 MB DRAM, and 256 MB storage. It also has a xilinx spartan-6 LX 25 FPGA. The cRIO 9075 connects the LabVIEW program to the Seiko D-Tran RT3200 robot, serving as the main processor to control the robot's movements.



Fig. 4. cRIO 9075.

The NI-9401, as shown in Fig. 5, is a digital I/O interface that can be configured for input or output in 4-bit increments. This allows for three different configurations: 8 digital inputs, 8 digital outputs, or 4 digital inputs and 4 digital outputs. It uses reconfigurable I/O (RIO) technology, which is only available on CompactRIO controllers, and can be programmed using the LabVIEW FPGA Module for custom applications such as high-speed counter/timers, digital

communication protocols, and pulse generation. Each channel is isolated from the others and the backplane using transient isolation. The NI-9401 operates at 5 V/TTL and has 8 bidirectional channels with a response time of 100 ns. It connects the drive motor controller to the motors of the Seiko D-Tran RT3200 robot, allowing the robot to move to the specified coordinates and receive encoder data. The Arduino Due board is used to send and receive encoder data to the NI-9401, and the received encoder data is then forwarded by the NI-9401 to the cRIO 9075 controller for further processing.



Fig. 5. NI-9401 card.

The Arduino DUE, shown in Fig. 6, is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board to feature a 32-bit ARM core microcontroller. The board has a total of 54 digital input/output pins, of which 12 can be used as pulse-width modulation (PWM) outputs, 12 analog inputs, 4 UARTs (hardware serial ports), an 84 MHz clock, a USB OTG-capable connection, 2 digital-to-analog converters (DACs), 2 TWI (two-wire interface) connections, a power jack, an SPI header, a JTAG header, a reset button, and an erase button. The Arduino DUE board connects to the encoder device to check the rotation of the motor as the Seiko D-Tran RT3200 robot moves to the specified coordinates. The encoder device sends a code to the NI-9401 board upon receiving it.



Fig. 6. Arduino DUE.

The H-bridge driver DC motor 200 A, as shown in Fig. 7, is a single motor driver that is capable of controlling a DC motor with a supply voltage of 12-48 V and a maximum current of 200A. It features a fully complementary power MOSFET driver and ultra-fast reverse recovery protection diode. The input interface signals are fully opto-isolated and operate at a TTL level of 5V with a current of 8 mA. The driver can be controlled using ON-OFF control, direction control, or speed control using pulse-width modulation (PWM) drives. The recommended PWM frequency range is 400-800 Hz, with a range of 400-1000 Hz possible. The H-bridge driver DC motor allows for the electronic switching of

the polarity of the DC motor, allowing it to rotate in both clockwise and counterclockwise directions.



Fig. 7. DC motor driver.

### C. Dynamic model of the robotic manipulator system

The robotic arm can be considered to find the system equations of input and output signals from [30] in the time domain. The robotic arm is a movement to move repeatedly on each axis. The equations obtained through the motion path generation model in the research presented in the equation are defined as a discrete time and the sampling time is 0.055 s, which results in estimating the transfer function for each axis. It complies with the equation as in (1).

$$P(z) = \frac{\gamma_1 z}{z^2 + \beta_1 z + \beta_0} \quad (1)$$

where the coefficients are used to appear in Table 1. The data set in the table is the variables of the robotic arm.

TABLE I. PARAMETERS USED IN THE OPEN-LOOP SYSTEM.

Joint	$\gamma_1$	$\beta_1$	$\beta_0$
Joint R	0.0333	-1.6871	0.6884
Joint T	0.0162	-1.7077	0.7111
Joint Z	0.0140	-1.7519	0.7526

### III. PID CONTROLLER DESIGN

In the basics of PID control, this is the design of individual motor control systems in industrial robots. By specifying  $G(Z)$  as the feedback control system for each axis of the robotic arm. As shown in (1), by  $\hat{G}_R(Z)$ ,  $\hat{G}_T(Z)$  and  $\hat{G}_Z(Z)$  is the value obtained from the approximation on each axis as in table 1, in which the output value in each axis is  $y(k)$  by setting the distance of error in motion is  $e(k)$  which can be calculated from (2).

$$e(k) = y_d(k) - y(k) \quad (2)$$

From (2) can be written in the state space and can be written in the form of a matrix. The system can display the time step response that occurs each time, where the following equation can be obtained.

$$y_j(k) = \bar{A}x(0) + Pu_j(k) + V_d \quad (3)$$

$$u_j(k) = [u_j(1)u_j(2)\dots u_j(p-1)]^T \quad (4)$$

$$P = \begin{bmatrix} CA^0B & 0 & \dots & 0 \\ CA^1B & CA^0B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \dots & CA^0B \end{bmatrix}; \quad (5)$$

$$\bar{A} = \begin{bmatrix} CA^1 \\ CA^2 \\ \vdots \\ CA^P \end{bmatrix}; \quad (6)$$

$$\begin{bmatrix} y_j(1) \\ \vdots \\ y_j(N) \end{bmatrix} = \underbrace{\begin{bmatrix} p_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ p_N & \dots & p_1 \end{bmatrix}}_P \begin{bmatrix} u_j(0) \\ \vdots \\ u_j(N-1) \end{bmatrix} + \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}}_Q x_j(0) \quad (7)$$

where  $p_i = CA^{i-1}B$  and  $q_i = CA^i$ , for  $i \in [1, N]$ . Let  $x_j(0)$  be the presumed initial state variable of zero parameter to test for iteration. Considering the system matrix  $P$  containing  $p_i$  along the diagonal lines, the coefficients  $p_i$  are referred to as Markov parameters of the system.

The control law of PID generating the feedforward control signal is given by

$$u_j(k) = K_p e_j(k) + K_i \sum_{k=N}^k e_j(k) + K_d (e_j(k) - e_j(k-1)) \quad (8)$$

where  $u_j$  is control inputs,  $K_p$  is a proportional gain,  $K_i$  is an integral gain,  $K_d$  is a derivative gain are in this study, and  $N$  is the number of data points being used in a summation (In this research, the value of  $N$  has been set to 5). It can show the basic structure of the PID as shown in Fig. 8.

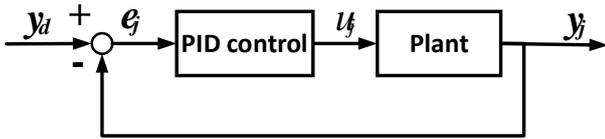


Fig. 8. Block diagram of PID control.

The motor system in joints R, T, and Z is controlled using a PID controller with CHR tuning to least overshoot. The R value, or R condition, for each axis was determined to be 12.75 for joint R, 21.33 for joint T, and 20.11 for joint Z, indicating that the P controller system should be used for control because the R value or R condition is greater than 10. The gain of the PID system for each axis is shown in Table 2 for joint R, Table 3 for joint T, and Table 4 for joint Z.

TABLE II. PID GAIN VALUES OF JOINT R.

Joint	$K_p$	$K_i$	$K_d$
P controller	4.25	-	-
PI controller	4.96	0.41	-
PD controller	4.96	-	0.40
PID controller	8.50	0.71	0.27

TABLE III. PID GAIN VALUES OF JOINT T.

Joint	$K_p$	$K_i$	$K_d$
P controller	8.00	-	-
PI controller	9.33	0.39	-
PD controller	9.33	-	0.90
PID controller	16.00	0.67	0.60

TABLE IV. PID GAIN VALUES OF JOINT Z

Joint	$K_p$	$K_i$	$K_d$
P controller	6.70	-	-
PI controller	7.82	0.29	-
PD controller	7.82	-	0.91
PID controller	13.41	0.49	0.60

#### IV. DUAL DESIGN PID CONTROLLER ARCHITECTURE

A dual design PID controller architecture is a design concept that aims to improve the performance of a PID controller by using two processors instead of one. The main processor, also known as controller 1, is a standard PID system that receives real-time input from the process and generates an output signal to control the process. The secondary processor, or controller 2, receives a one-time-step delay of the input of the process and adjusts it before it is sent as the control input to eliminate the control input from controller 1.

By continuously adjusting the output signal this way, a dual design PID controller is able to eliminate the tracking errors or overshoot and improve the performance of the PID controller. Fig. 9 shows a block diagram of a dual design PID controller. As shown in the figure, the main processor (controller 1) receives real-time errors as input from the process and generates an output signal based on the PID algorithm. The secondary processor (controller 2) receives a one-time-step delay as input to the process and both values are compensated as input into the system.

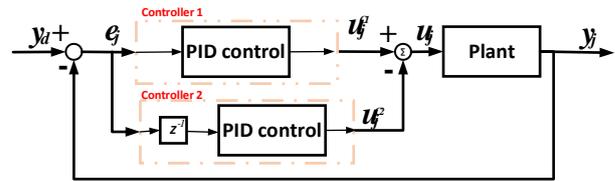


Fig. 9. Block diagram of a dual design PID controller.

In a dual design PID control, a PID control law is presented in controller 1, as shown in (9), and a PID control with one time-step delay is presented in controller 2 as in (10), and the sum of the answers of a dual design PID control structure as in (11) and can describe the structure that is designed in Fig. 4.

For controllers 1 and 2 can choose to use algorithm development by PID techniques to work in the system.

$$u_j^{c1}(k) = K_p e_j(k) + K_i \sum_{k=N}^k e_j(k) + K_d (e_j(k) - e_j(k-1)) \quad (9)$$

$$u_j^{c2}(k) = \frac{K_p}{\alpha} e_j(k-1) + \frac{K_i}{\alpha} \sum_{k=N}^k e_j(k-1) + \frac{K_d}{\alpha} (e_j(k-1) - e_j(k-2)) \quad (10)$$

$$u_j(k) = u_j^{c1}(k) - u_j^{c2}(k) \quad (11)$$

where  $\alpha$  is a gain of one time-step delay PID controller.

In the initial trial, where  $\alpha$  has not yet been activated, the initial error input  $e_j(-1)$  is equal to zero. As a result, the initial error input is equivalent to the reference command  $y_d$ . However, when  $\alpha$  can define numbers between 1 to  $\infty$ . In this research are defined  $\alpha$  in values of 2 for dual PID control.

The gain of controller 1, as shown in Fig. 9, is used in Tables 2, 3, and 4 for joint R, joint T, and joint Z. The gain of controller 2, also shown in Fig. 9, is used in Tables 5, 6, and 7 for joint R, joint T, and joint Z.

TABLE V. PID GAIN VALUES OF CONTROLLER 2 IN JOINT R.

Joint	$K_p$	$K_i$	$K_d$
P controller	2.13	-	-
PI controller	2.48	0.21	-
PD controller	2.48	-	0.20
PID controller	4.25	0.36	0.13

TABLE VI. PID GAIN VALUES OF CONTROLLER 2 IN JOINT T.

Joint	$K_p$	$K_i$	$K_d$
P controller	4.00	-	-
PI controller	4.67	0.19	-
PD controller	4.67	-	0.45
PID controller	8.00	0.33	0.30

TABLE VII. PID GAIN VALUES OF CONTROLLER 2 IN JOINT Z.

Joint	$K_p$	$K_i$	$K_d$
P controller	3.35	-	-
PI controller	3.91	0.14	-
PD controller	3.91	-	0.45
PID controller	6.70	0.25	0.30

## V. DESIRED TRAJECTORY

The robotic arm has been programmed to follow a trajectory consisting of 287 time-steps, with a sampling time of 0.055 seconds. This results in a total duration of 15.73 seconds for the trajectory. The starting position and target coordinates for the robot's movement can be seen in Fig. 10 and Fig. 11, respectively. The robot will use its joints R, T, and Z to move the screw to the desired position on the hard drive by using the nut as a reference point. The robot will begin at the start position and end at the target coordinates are shown in Fig. 10.

## VI. SIMULATION AND EXPERIMENTAL RESULTS

In this section, there are three experimental results that will be discussed. The first is an overview of the system simulation results, which were tested on the Seiko D-Tran RT3200 robot with both the PID Controller and the dual PID controller. The second result is the system analysis of the P controller and dual P controller in the simulation run, which was also tested on the Seiko D-Tran RT3200 robot. The third result is the analysis of the control input estimation of the P controller and dual P controller in the simulation run, which was also tested on the Seiko D-Tran RT3200 robot.

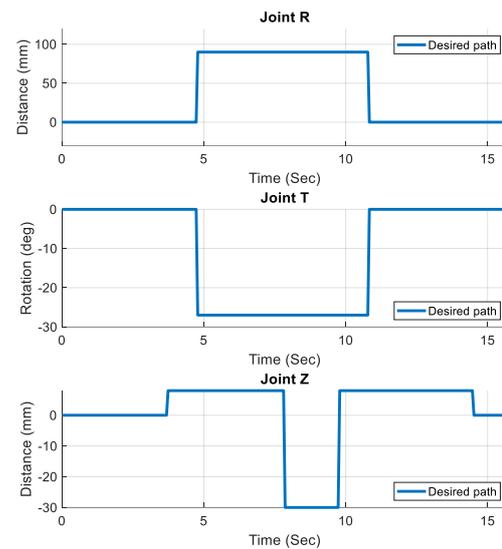


Fig. 10. Trajectory profile.

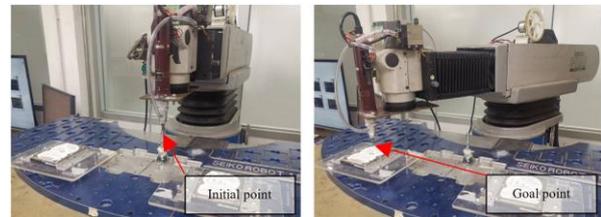


Fig. 11. Initial and goal points of the robotic manipulator.

### A. First result

The first set of results presents both simulation and experimental data on the performance of a PID controller and a dual PID controller on the Seiko D-Tran RT3200 robot. Fig. 12, Fig. 13, Fig. 14, and Fig. 15 show the simulation results, while Fig. 16, Fig. 17, Fig. 18, and Fig. 19 show the experimental results. These figures compare the performance of the two control methods using a step input to define the motion profile of the robotic arm. The trajectory profile in Fig. 10 is used to move the robot in each axis. The PID controller uses the gain values listed in Tables 2, 3, and 4, while the dual PID controller uses the gain values listed in Tables 2, 3, and 4 for controller 1 and Tables 5, 6, and 7 for controller 2. These gains of the PID controller were used to test the system's demonstrated performance of the system under each controller.

In the simulation results, the step inputs are shown in Fig. 12, Fig. 13, Fig. 14, and Fig. 15 as an overview of the system testing a dual design PID control. The PID control has the highest overshoot compared to the dual design P control method. Fig. 12 shows that the P control has a gain of 1, which is calculated from the proportional error in the system. The P control system is easy to design due to its basic nature. In the simulation experiments, the P control system had a slight overshoot while the dual P control system had a very low overshoot. The simulation results for the joints R, T, and Z were similar for all three joints.

Simulation results for PI control are shown in Fig. 13. This type of control is a 2-gain system that estimates values based on the proportion of errors in the system and error

values from past sums. During testing, it was found that PI control had a higher overshoot than P control because the sum of errors in integral term was collected, while dual PI control had a slightly higher overshoot than dual P control but less overshoot than the PI control system. Simulation results in joints R, T, and Z were similar in all three joints.

In Fig. 14, a PD control is a 2-gain controller that estimates values from the proportion of errors in the system and the difference of the error value. From the results of system testing, it was found that PD control had less overshoot but a higher value compared to P control because  $K_p$  was higher than  $K_p$  in P control. Dual PD control had a similar effect to dual P control and had a slightly higher overshoot. The simulation results of joints R, T, and Z were similar for all three joints.

Fig. 15 presents the results of testing a PID control system, which uses three gain values to approximate the proportional error in the system, the past summation error, and the difference in the error value. According to the results of the system test, it was found that the PID control system had a higher overshoot than the PI control system. This was due to the  $K_p$  gain value being double and the integral term having a large sum of errors in the system. The dual PID control had a higher overshoot than other dual controls, but it was able to reduce the overshoot faster than the PID control according to the simulation results in joints R, T, and Z, which were all similar.

In experiment 1, in-step inputs are shown in Fig. 5, Fig. 6, Fig. 7, and Fig. 8, a dual design P control is a decrease and holds a small overshoot for  $\alpha$  values are 2 and 4. P control has the highest overshoot compared with a dual design P control method, and the experiment is the same in the simulation, by efficiency is the last.

The simulation results suggest that the P control system is the best option for minimizing overshoot in the joints R, T, and Z. This is supported by CHR tuning and the consideration of the R-value or R condition, which shows that the R-value for all three axes is greater than 10. The P control system also had the lowest overshoot compared with PI control, PD control, and PID control. On the other hand, the PID control system had the highest overshoot among the tested control systems.

While a dual design PID control system was able to reduce overshoot faster than the standard PID control system, it still had higher overshoot compared to the other dual control systems. The simulation results for the PI and PD control systems were similar, with the PI control system having higher overshoot than the P control system and the PD control system having lower overshoot compared to the P control system. The single controller versions of these control systems had higher overshoot than dual design controllers. In conclusion, the simulation results support the selection of the P control system for use in the system, which is consistent with the consideration of CHR tuning.

The experimental results showed that a dual design PID control had the highest overshoot compared to the PID control method, as depicted in Fig. 16, Fig. 17, Fig. 18, and Fig. 19. These results were consistent with the system

simulation test. Although the P control performed the best in this system, as shown in Fig. 16, the PD control in Fig. 18 had a similar effect. On the other hand, when considering the other controllers in Fig. 17 and Fig. 19, it was found that the robot system used in the control system was unable to control the desired coordinates due to an overshoot within the system. However, the use of dual control helped to stabilize the system because the control estimation was more appropriate for the system. Overall, the system model and the experimental results were consistent.

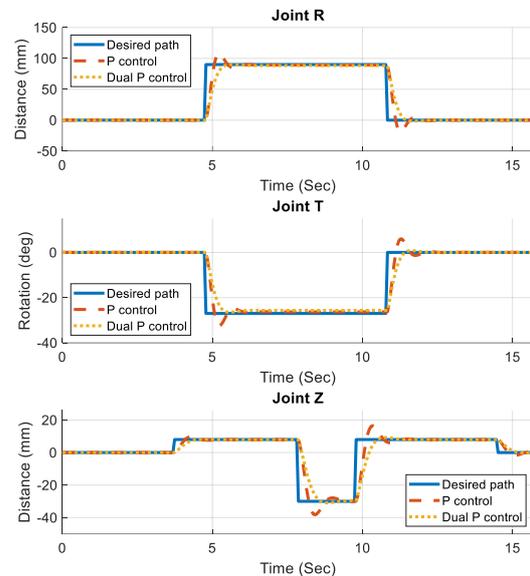


Fig. 12. The results of P controller in simulation.

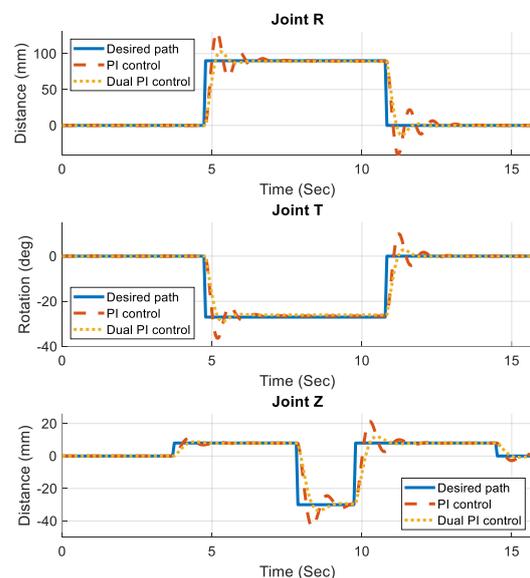


Fig. 13. The results of PI controller in simulation.

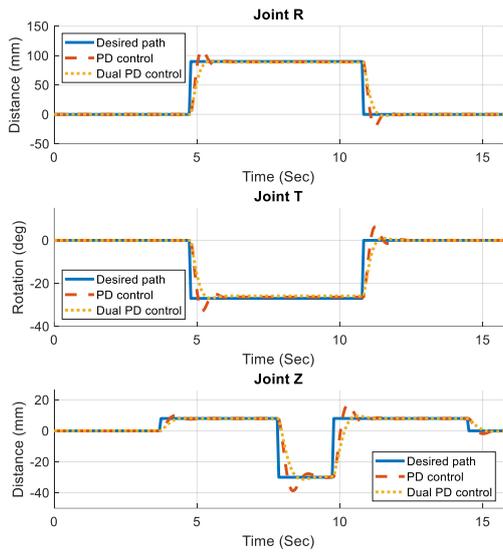


Fig. 14. The results of PD controller in simulation.

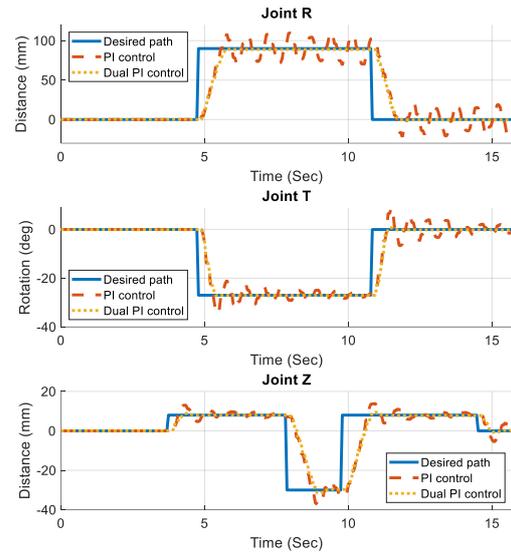


Fig. 17. The results of PI controller in experimental.

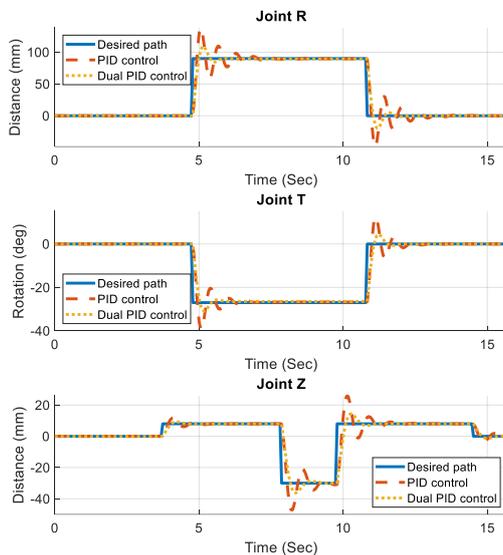


Fig. 15. The results of PID controller in simulation.

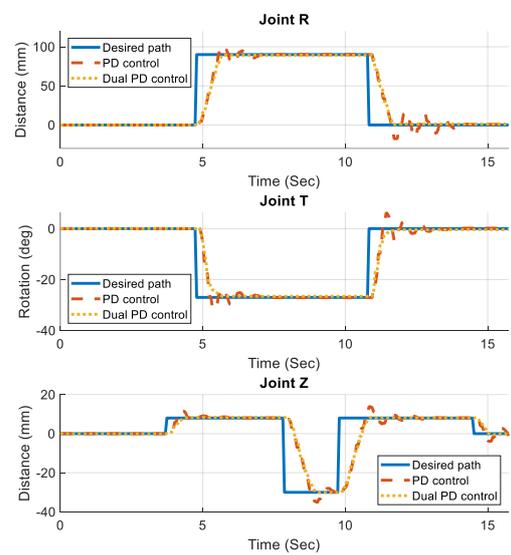


Fig. 18. The results of PD controller in experimental.

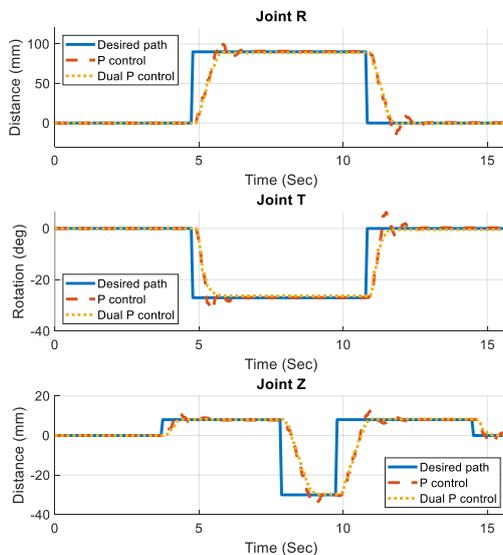


Fig. 16. The results of P controller in experimental.

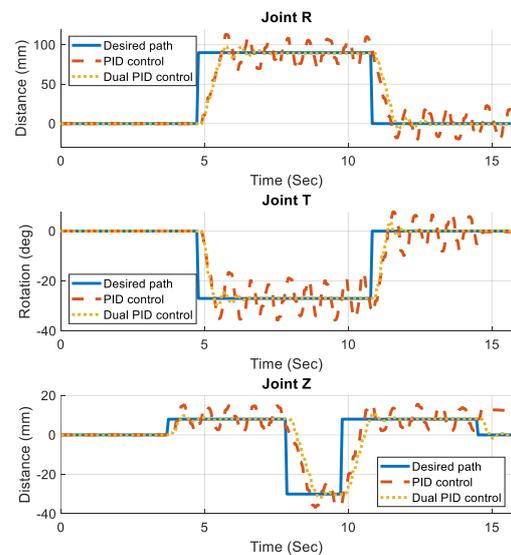


Fig. 19. The results of PID controller in experimental.

**B. Second result**

In the second result, the performance of a P controller and a dual P controller were evaluated through simulation and experimentation on a Seiko D-Tran RT3200 robot. The results showed that the dual P controller performed better overall. Fig. 20 and Fig. 21 show a comparison of the simulation and experimental results for the P controller and dual P controller in certain time periods. The rise time, settling time, and percentage of overshoot (%OS) were used to assess the performance of the two controllers.

Based on the results presented in Tables 8 and 9 and Fig. 20 and Fig. 21, it appears that the P controller is effective at improving the rise time, settling time, and overshoot of the robotic arm in both simulation and experiments.

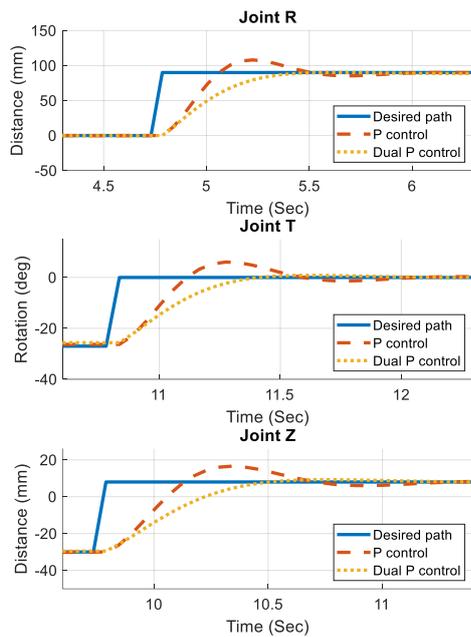


Fig. 20. Magnified Fig. 12 of the result from simulation.

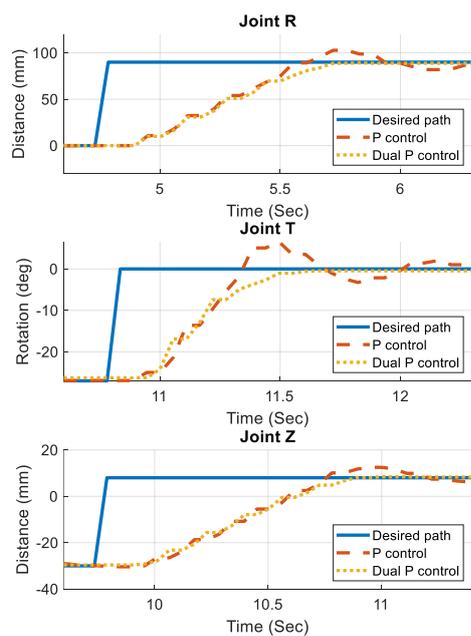


Fig. 21. Magnified Fig. 16 of the result from experimental.

In simulations, the P controller was able to achieve a shorter rise time and had overshoot rates of 18.61% for joint R, 43.58% for joint T, and 20.80% for joint Z.

In experiments, the P controller had overshoot rates of 10.22% for joint R, 17.14% for joint T, and 13.68% for joint Z. The dual design P controller also showed similar performance in both simulations and experiments, with overshoot rates of -0.82% and -2.22% for joint R, 2.46% and -0.23% for joint T, and 1.87% and -1.84% for joint Z, respectively.

The results suggest that the dual P controller is an effective choice for optimizing system estimation in the robotic arm. It is able to achieve the lowest overshoot and the shortest settling time. While the P controller may have a shorter rise time and higher overshoot rates, the dual P controller is able to provide the optimal results in terms of the lowest overshoot and shortest settling time. As a result, the dual P controller is a good choice for optimizing system estimation in the robotic arm.

TABLE VIII. OUTPUT OF STEP INPUT FROM SIMULATION.

Joint R			
Controller	Rise time	Settling time	%OS
P control	0.33	1.10	18.61
Dual P control	0.72	0.72	-0.82
Joint T			
Controller	Rise time	Settling time	%OS
P control	0.39	1.16	43.58
Dual P control	0.66	0.66	2.46
Joint Z			
Controller	Rise time	Settling time	%OS
P control	0.44	1.43	20.80
Dual P control	0.83	0.83	1.87

TABLE IX. OUTPUT OF STEP INPUT FROM EXPERIMENT.

Joint R			
Controller	Rise time	Settling time	%OS
P control	0.88	1.65	10.22
Dual P control	1.05	1.05	-2.22
Joint T			
Controller	Rise time	Settling time	%OS
P control	0.50	1.49	17.14
Dual P control	0.83	0.83	-0.23
Joint Z			
Controller	Rise time	Settling time	%OS
P control	1.02	2.31	13.68
Dual P control	1.21	1.21	-1.84

**C. Third result**

The third result of the analysis compared the control input estimation of the P controller and dual P controller in a simulation and on a Seiko D-Tran RT3200 robot. The results, shown in Fig. 10, reveal the percentage of pulse-width modulation (PWM) values used in the system. Fig. 22 and Fig. 23 show that the dual P controller consumes less electrical energy power than the P controller in the same trajectory. This suggests that the dual P controller is more accurate at estimating the control input, resulting in lower power consumption and electrical energy savings in the system.

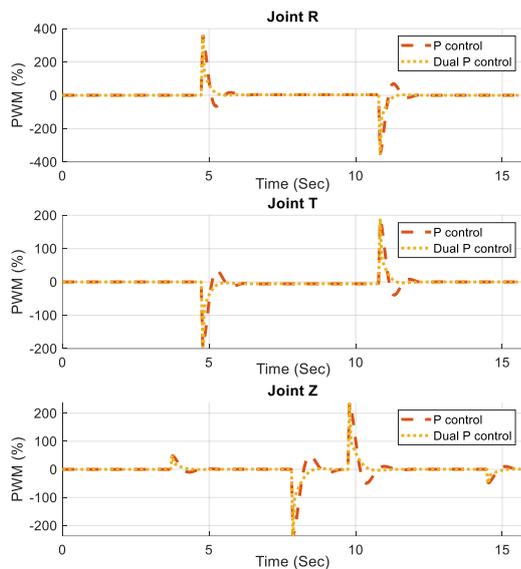


Fig. 22. The control input of P controller in simulation.

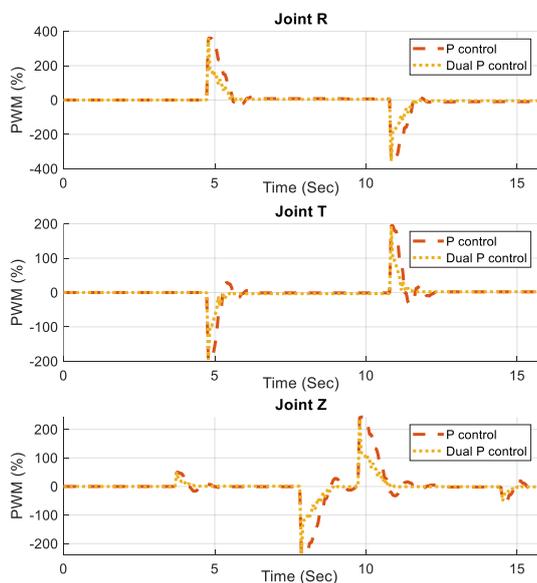


Fig. 23. The control input of P controller in experimental.

## VII. CONCLUSION

In conclusion, this paper proposed a dual design PID controller based on two PID controllers with real-time and one-time step delays and compared the results obtained from the control system operation to reduce the error values, rise time, settling time, and overshoot caused by trajectory tracking control. This study compared the performance of a single PID controller using the least overshoot tuning technique developed by CHR with that of a dual PID controller design. The scope of the PID controller using the CHR technique was also evaluated in this research. A dual design PID controller has the advantage of allowing the user to modify the selection weights to bring the delay of the controller's response into the work coordination and focus on adjusting this weight, which can select the delay or fast of the system's response. The results of the testing system were similar to the simulation results, indicating that a dual design PID controller is a viable choice for an efficient, stable, and

electrical energy saving for the control system. For future work, it would be beneficial to study the gain adjustment of a dual design PID controller to ensure its suitability for a variety of systems. Additionally, it would be useful to examine any limitations on the use of the controller and to optimize its tuning for specific applications. Further research in these areas could motivate other researchers to continue exploring the use of a dual design PID controller in various contexts.

## ACKNOWLEDGMENT

The researcher would like to thank the Research Institute, Academic Services Center, and College of Biomedical Engineering, Rangsit University for the grant of research funding to the research team.

## REFERENCES

- [1] J. An, F. You, M. Wu, and J. She, "Iterative learning control for nonlinear weighing and feeding process," *Mathematical Problems in Engineering*, vol. 2018, 2018, doi: 10.1155/2018/9425902.
- [2] X. Li, J. Liu, L. Wang, K. Wang, and Y. Li, "Welding process tracking control based on multiple model iterative learning control," *Mathematical Problems in Engineering*, vol. 2019, 2019, doi: 10.1155/2019/6137352.
- [3] J. Jung and K. Kong, "Mechanical parameter tuning based on iterative learning mechatronics approach," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 906-915, 2018, doi: 10.1109/TMECH.2018.2810196.
- [4] Y. Li, K. H. Ang, and G. C. Chong, "PID control system analysis and design," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 32-41, 2006, doi: 10.1109/MCS.2006.1580152.
- [5] Z. Dachang, D. Baolin, Z. Puchen, and C. Shouyan, "Constant force PID control for robotic manipulator based on fuzzy neural network algorithm," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/3491845.
- [6] B. Li, Y. Tan, J. Chen, X. Liu, and S. Yang, "Precise active seeding downforce control system based on fuzzy PID," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/5123830.
- [7] J. Zhou and Q. Zhang, "Adaptive fuzzy control of uncertain robotic manipulator," *Mathematical Problems in Engineering*, vol. 2018, doi: 10.1155/2018/4703492.
- [8] A. Elmogy, Y. Bouteraa, and W. Elawady, "An adaptive fuzzy self-tuning inverse kinematics approach for robot manipulators," *Journal of Control Engineering and Applied Informatics*, vol. 22, no. 4, pp. 43-51, 2020.
- [9] X. Long, Z. He, and Z. Wang, "Online optimal control of robotic systems with single critic NN-based reinforcement learning," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/8839391.
- [10] F. Ejaz, M. T. Hamayun, S. Hussain, S. Ijaz, S. Yang, N. Shehzad, and A. Rashid, "An adaptive sliding mode actuator fault tolerant control scheme for octorotor system," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019, doi: 10.1177/1729881419832435.
- [11] J. Yang, J. Na, G. Gao, and C. Zhang, "Adaptive neural tracking control of robotic manipulators with guaranteed nn weight convergence," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/7131562.
- [12] M. Marsono, Y. Yoto, A. Suyetno, and R. Nuralasari, "Design and programming of 5 axis manipulator robot with GrblGru open source software on preparing vocational students' robotic skills," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 539-545, 2021.
- [13] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale robot arm design with pick and place mission based on inverse kinematics," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 469-475, 2021.

- [14] C. Deniz and G. Gökmen, "A new robotic application for covid-19 specimen collection process," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 73-77, 2022.
- [15] E. S. Ghith and F. A. A. Tolba, "Design and optimization of PID controller using various algorithms for micro-robotics system," *Journal of Robotics and Control (JRC)*, vol. 3, no. 3, pp. 244-256, 2022.
- [16] A. K. Hado, B. S. Bashar, M. M. A. Zahra, R. Alayi, Y. Ebazadeh, and I. Suwarno, "Investigating and optimizing the operation of microgrids with intelligent algorithms," *Journal of Robotics and Control (JRC)*, vol. 3, no. 3, pp. 279-288, 2022.
- [17] E. H. Kadhim and A. T. Abdulsadda, "Mini drone linear and nonlinear controller system design and analyzing," *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 212-218, 2022.
- [18] A. Ma'arif and A. Çakan, "Simulation and arduino hardware implementation of dc motor control using sliding mode controller," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 582
- [19] A. Ma'arif and N. R. Setiawan, "Control of DC motor using integral state feedback and comparison with PID: simulation and Arduino implementation," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 456-461, 2021.
- [20] H. Maghfiroh, A. Ramelan, and F. Adriyanto, "Fuzzy-PID in BLDC motor speed control using MATLAB/Simulink," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 8-13, 2022.
- [21] M. Samuel, M. Mohamad, M. Hussein, and S. M. Saad, "Lane keeping maneuvers using proportional integral derivative (PID) and model predictive control (MPC)," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, pp. 78-82, 2021.
- [22] A. Wajiansyah, S. Supriadi, A. F. O. Gaffar, and A. B. W. Putra, "Modeling of 2-DOF hexapod leg using analytical method," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 435-440, 2021.
- [23] J. Jiang, S. Guo, L. Zhang, and Q. Sun, "Motor ability evaluation of the upper extremity with point-to-point training movement based on end-effector robot-assisted training system," *Journal of Healthcare Engineering*, vol. 2022, 2022. doi: 10.1155/2022/1939844.
- [24] 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.
- [25] 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.
- [26] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589-1600, 2009. doi: 10.1016/j.jprocont.2009.09.006.
- [27] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, 2006. doi: 10.1109/MCS.2006.1636313.
- [28] C. Guo, L. Zhong, J. Zhao, G. Gao, and Y. Huang, "First-order and high-order repetitive control for single-phase grid-connected inverter," *Complexity*, vol. 2020, 2020. doi: 10.1155/2020/1094386.
- [29] X. Wang and J. Wang, "Iterative learning control for one-sided lipschitz nonlinear singular conformable differential equations," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 17, pp. 7791-7805, 2020. doi: 10.1002/rnc.5191.
- [30] P. Chotikunnan, B. Panomruttanarug, and P. Manoonpong, "Dual design iterative learning controller for robotic manipulator application," *Journal of Control Engineering and Applied Informatics*, vol. 24, no. 3, pp. 76-85, 2022.
- [31] M. Thor and P. Manoonpong, "Error-based learning mechanism for fast online adaptation in robot motor control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2042-2051, 2019. doi: 10.1109/TNNLS.2019.2927737.
- [32] M. Thor and P. Manoonpong, "A fast online frequency adaptation mechanism for CPG-based robot motion control," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3324-3331, 2019. doi: 10.1109/LRA.2019.2926660.
- [33] P. Chotikunnan and B. Panomruttanarug, "Practical design of a time-varying iterative learning control law using fuzzy logic," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 3, pp. 2419-2434, 2022.
- [34] B. Panomruttanarug, R. W. Longman, and M. Q. Phan, "Steady state frequency response design of finite time iterative learning control," *The Journal of the Astronautical Sciences*, vol. 67, no. 2, pp. 571-594, 2020. Doi:10.1007/s40295-019-00198-9.
- [35] B. Panomruttanarug, "Position control of robotic manipulator using repetitive control based on inverse frequency response design," *International Journal of Control, Automation and Systems*, vol. 18, no. 11, pp. 2830-2841, 2020. Doi:10.1007/s12555-019-0518-2.
- [36] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818-827, 2021.
- [37] 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, pp. 1224, 2019.
- [38] D. Somwanshi, M. Bundele, 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.
- [39] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, "Design and Implementation of Speed Control System for DC Motor Based on PID Control and Matlab Simulink," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, pp. 127, 2020.
- [40] I. Carlucho, M. De Paula, and G. G. Acosta, "An Adaptive Deep Reinforcement Learning Approach for MIMO PID Control of Mobile Robots," *ISA Transactions*, vol. 102, pp. 280-294, 2020.
- [41] 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.
- [42] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, pp. 38100-38114, 2019.
- [43] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID System Regulator for Mini Conveyor Drive Based on MATLAB," *Journal of Robotics and Control (JRC)*, vol. 1, no. 6, pp. 185-190, 2020.
- [44] 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.
- [45] K. Vanchinathan and N. Selvagesan, "Adaptive Fractional Order PID Controller Tuning for Brushless DC Motor Using Artificial Bee Colony Algorithm," *Results in Control and Optimization*, vol. 4, 2021.
- [46] P. Dutta and S. K. Nayak, "Grey Wolf Optimizer Based PID Controller for Speed Control of BLDC Motor," *Journal of Electrical Engineering & Technology*, vol. 16, no. 2, pp. 955-961, 2021.
- [47] W. Batayneh and Y. AbuRmaileh, "Decentralized Motion Control for Omnidirectional Wheelchair Tracking Error Elimination Using PD-Fuzzy-P and GA-PID Controllers," *Sensors*, vol. 20, no. 12, pp. 3525, 2020.
- [48] M. A. Shamseldin, E. Khaled, A. Youssef, D. Mohamed, S. Ahmed, A. Hesham, and M. Badran, "A New Design Identification and Control Based on GA Optimization for an Autonomous Wheelchair," *Robotics*, vol. 11, no. 5, pp. 101, 2022.
- [49] L. A. Aguilar-Pérez, J. C. Paredes-Rojas, J. I. Sanchez-Cruz, J. A. Leal-Naranjo, A. Oropeza-Osornio, and C. R. Torres-SanMiguel, "Design of an Automated Multiposition Dynamic Wheelchair," *Sensors*, vol. 21, no. 22, pp. 7533, 2021.
- [50] V. Balambica, "PID Sensor Controlled Automatic Wheelchair for Physically Disabled People," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 5848-5857, 2021.
- [51] W. Budiharto, "Intelligent Controller of Electric Wheelchair from Manual Wheelchair for Disabled Person Using 3-Axis Joystick," *ICIC Express Letters, Part B: Applications*, vol. 12, no. 2, pp. 201-206, 2021.
- [52] H. H. Tesfamikael, A. Fray, I. Mengsteab, A. Semere, and Z. Amanuel, "Simulation of Eye Tracking Control Based Electric Wheelchair Construction by Image Segmentation Algorithm,"

- Journal of Innovative Image Processing (JIIP)*, vol. 3, no. 01, pp. 21-35, 2021.
- [53] L. Teng, M. A. Gull, and S. Bai, "PD-Based Fuzzy Sliding Mode Control of a Wheelchair Exoskeleton Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2546-2555, 2020.
- [54] E. Q. Ahmed, I. A. Aljazeera, A. F. Al-zubidi, and H. T. S. ALRikabi, "Design and Implementation of Control System for a Self-Balancing Robot Based on Internet of Things by Using Arduino Microcontroller," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 9, no. 3, pp. 409-417, 2021.
- [55] A. Latif, K. Shankar, and P. T. Nguyen, "Legged Fire Fighter Robot Movement Using PID," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, pp. 15-19, 2020.
- [56] K. Jayaswal, D. K. Palwalia, and S. Kumar, "Performance Investigation of PID Controller in Trajectory Control of Two-Link Robotic Manipulator in Medical Robots," *Journal of Interdisciplinary Mathematics*, vol. 24, no. 2, pp. 467-478, 2021.
- [57] B. Sumantri, E. H. Binugroho, I. M. Putra, and R. Rokhana, "Fuzzy-PID Controller for an Energy Efficient Personal Vehicle: Two-Wheel Electric Skateboard," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, pp. 5312, 2019.
- [58] F. A. Raheem, B. F. Midhat, and H. S. Mohammed, "PID and Fuzzy Logic Controller Design for Balancing Robot Stabilization," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 18, no. 1, pp. 1-10, 2018.
- [59] Y. Ramchandra, M. Salim, T. Ankush, and P. Vasantrao, "Self-Balancing Robot Using Raspberry Pi and PID Controller," *International Journal of Innovative Science and Research Technology*, vol. 6, no. 4, pp. 321-322, 2021.
- [60] T. Zhao, Y. Chen, S. Dian, R. Guo, and S. Li, "General Type-2 Fuzzy Gain Scheduling PID Controller with Application to Power-Line Inspection Robots," *International Journal of Fuzzy Systems*, vol. 22, no. 1, pp. 181-200, 2020.
- [61] G. Cao, X. Zhao, C. Ye, S. Yu, B. Li, and C. Jiang, "Fuzzy Adaptive PID Control Method for Multi-Mecanum-Wheeled Mobile Robot," *Journal of Mechanical Science and Technology*, vol. 36, no. 4, pp. 2019-2029, 2022.
- [62] C. Kim, J. Suh, and J. H. Han, "Development of a Hybrid Path Planning Algorithm and a Bio-Inspired Control for an Omni-Wheel Mobile Robot," *Sensors*, vol. 20, no. 15, pp. 4258, 2020.
- [63] S. F. Hasana and H. M. Alwan, "Modeling and Control of Wheeled Mobile Robot with Four Mecanum Wheels," *Engineering and Technology Journal*, vol. 39, no. 5, pp. 779-789, 2021.
- [64] N. Zijie, Z. Peng, Y. Cui, and Z. Jun, "PID Control of an Omnidirectional Mobile Platform Based on an RBF Neural Network Controller," *Industrial Robot: The International Journal of Robotics Research and Application*, vol. 49, no. 1, 2021.
- [65] V. V. Patel, "Ziegler-Nichols Tuning Method," *Resonance*, vol. 25, no. 10, pp. 1385-1397, 2020.
- [66] R. R. Alla, N. Lekyasri, and K. Rajani, "PID Control Design for Second Order Systems," *Int. J. Eng. Manuf.*, vol. 9, no. 4, pp. 45-56, 2019.
- [67] E. Bashier and O. Mohammed, "Optimally Tuned Proportional Integral Derivatives (PID) Controllers for Set-Point," *Journal of Engineering and Computer Science (JECS)*, vol. 13, no. 1, pp. 48-53, 2019.
- [68] T. Y. Wu, Y. Z. Jiang, Y. Z. Su, and W. C. Yeh, "Using Simplified Swarm Optimization on Multiloop Fuzzy PID Controller Tuning Design for Flow and Temperature Control System," *Applied Sciences*, vol. 10, no. 23, pp. 8472, 2020.
- [69] C. Dalen and D. Di Ruscio, "Performance optimal PI controller tuning based on integrating plus time delay models," *Algorithms*, vol. 11, no. 6, pp. 86, 2018.
- [70] S. Jain and Y. V. Hote, "Design of FOPID Controller Using BBBC via ZN Tuning Approach: Simulation and Experimental Validation," *IETE Journal of Research*, vol. 68, no. 5, pp. 3356-3370, 2022.
- [71] S. Kai-bo, H. Jian, and L. Lin-tao, "A New Method of DC Motor Speed Regulation based on STM32," *Journal of Astronautic Metrology and Measurement*, vol. 38, no. 2, pp. 87, 2018.
- [72] N. A. Rao and C. R. Kumar, "Speed Control of Brushless Dc Motor by Using PID and Fuzzy Logic Controller," *International Journal of Innovative Research in Technology*, vol. 4, no. 6, 2019.
- [73] P. Saini and C. Sharma, "Comparative Analysis of Controller Tuning Techniques for Dead Time Processes," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 4, no. 3, pp. 803, 2019.
- [74] E. Kesavan, K. M. Junaid, and B. Jaison, "Design and Analysis of Different Tuning Strategies of PI Controller for Distillation Column," *International Journal of Control Systems and Robotics*, vol. 4, 2019.
- [75] V. Dubey, "Comparative Analysis of PID Tuning Techniques for Blood Glucose Level of Diabetic Patient," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 2948-2953, 2021.
- [76] A. O. Amole, O. E. Olabode, D. O. Akinyele, and S. G. Akinjobi, "Optimal Temperature Control Scheme for Milk Pasteurization Process Using Different Tuning Techniques for a Proportional Integral Derivative Controller," *Iranian Journal of Electrical and Electronic Engineering*, vol. 2170, pp. 2170-2170, 2022.
- [77] S. Mahfoud, A. Derouich, N. El Ouanjli, M. El Mahfoud, and M. Taoussi, "A New Strategy-Based PID Controller Optimized by Genetic Algorithm for DTC of the Doubly Fed Induction Motor," *Systems*, vol. 9, no. 2, p. 37, 2021.
- [78] M. M. Nishat, F. Faisal, A. J. Evan, M. M. Rahaman, M. S. Sifat, and H. F. Rabbi, "Development of Genetic Algorithm (GA) Based Optimized PID Controller for Stability Analysis of DC-DC Buck Converter," *Journal of Power and Energy Engineering*, vol. 8, no. 9, p. 8, 2020.
- [79] H. Feng, C. B. Yin, W. W. Weng, W. Ma, J. J. Zhou, W. H. Jia, and Z. L. Zhang, "Robotic excavator trajectory control using an improved GA based PID controller," *Mechanical Systems and Signal Processing*, vol. 105, pp. 153-168, 2018.
- [80] E. S. Rahayu, A. Ma'arif, and A. Çakan, "Particle Swarm Optimization (PSO) Tuning of PID Control on DC Motor," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, pp. 435-447, 2022.
- [81] Z. Qi, Q. Shi, and H. Zhang, "Tuning of Digital PID Controllers Using Particle Swarm Optimization Algorithm for a CAN-Based DC Motor Subject to Stochastic Delays," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5637-5646.