Journal of Advanced Engineering Trends
ISSN : 2682 - 2091

Vol. 44, No. 1, January 2025

Optimal Sliding Surface PID for Position Control of Robotic Arm

Ahmed Abdel-Sattar [1,2 *], Abu-Hashema M. El-Sayed [2], Shehab R. Tawfeic [3], Ahmed M. Kassem[1]

[1] *Electrical Engineering Dep., Faculty of Engineering, Sohag University, Sohag, Egypt*

[2] *Electrical Engineering Dep., Faculty of Engineering, Minia University, Minia, Egypt*

[3] *Mechanical Power Engineering and Energy Dep., Faculty of Engineering, Minia University, Minia, Egypt*

** Corresponding Author E-mail: ahmed_abdelstar@eng.sohag.edu.eg*

A R T I C L E I N F O

Article history:
Received: 24 September 2024
Accepted: 9 December 2024
Online: 28 February 2025

Keywords:
Sliding Surface PID,
PSO,
PID,
FOPID,
SMC

A B S T R A C T

This article presents a comprehensive study on the control system of a single-joint robotic arm by comparing the performance of an optimal Sliding Surface PID (SSPID) controller, tuned via Particle Swarm Optimization (PSO), with both optimal PID Controller and Fractional Order PID (FOPID) controller. The proposed SSPID controller integrates sliding mode control principles, where the equivalent control is derived from the system dynamics to stabilize the system on the sliding surface, while the switching control component utilizes a PID structure to ensure robustness against uncertainties and external disturbances. Simulation and experimental results indicate that the SSPID controller significantly outperforms both PID and FOPID controllers regarding rise time and tracking accuracy. The simulation results also showed that the sliding surface PID controller enhances robustness, especially under variations in system physical parameters of 20% and 50%. The SSPID exhibits faster response times, reduced overshoot, and stable performance.

1. Introduction

Robotic arms have become an essential component in modern industries. They help in manufacturing, healthcare, and logistics, due to their precision, flexibility, and ability to perform repetitive tasks [1-3]. The control of a robotic arm is a complex task that requires precise positioning and trajectory tracking. Among the various control techniques, Proportional-Integral-Derivative (PID) control and Sliding Mode Control (SMC) are widely used due to their simplicity, robustness, and effectiveness.[4-7]

PID control is a popular choice for robotic arm control because of its simplicity and tuning. However, PID control may not provide satisfactory performance in the presence of uncertainties, disturbances, and nonlinearities [8-10]. On the other hand, SMC is effective in handling these challenges due to its ability to reject disturbances and provide robust control, However, SMC often suffers from chattering, leading to wear and tear on the system [11-13].

Researchers have developed various SMC techniques to enhance the performance and robustness of robotic manipulators. For instance, a fuzzy sliding mode controller was proposed for a flexible single-link robotic manipulator, demonstrating its effectiveness in suppressing vibrations [14]. An integral sliding mode control approach was also presented for the position control of a robotic manipulator, showcasing its ability to reduce tracking errors [15]. Comparative studies have also been conducted, such as a study that evaluated the performance of PID, fuzzy logic control, and SMC for a 2-DOF robotic manipulator [16]. Furthermore, researchers have focused on enhancing the robustness of SMC, with the development of a robust sliding mode control strategy for robots driven by compliant actuators [17]. Additionally, adaptive incremental SMC approaches have been proposed, such as one that demonstrated improved tracking performance for a robot manipulator [18]. These studies highlight the potential of SMC-based controllers in achieving high-performance control of robotic manipulators.

Despite these advances, there remains a need for further research on the optimal design and tuning of SMC for robotic arm control. In particular, there is a need to develop SMC schemes that can provide fast and robust responses without chattering and overshoot. This paper aims to address this research gap by proposing an alternative SMC scheme that combines the benefits of PID and SMC.

The main contribution of this paper is the development of an optimal Sliding Surface PID (SSPID) controller with an equivalent controller derived from the system dynamics and a switching controller PID controller to reduce chattering. The proposed controller is optimized using Particle Swarm Optimization (PSO) to ensure optimal performance. PSO was chosen for its efficiency, faster convergence, and simpler implementation compared to other heuristic methods. The

proposed controller is benchmarked against optimal PID and optimal Fractional Order PID (FOPID).

This paper is organized into 6 sections as follows: section 1 introduces the fundamental concepts of controlling robotics by PID and SMC controllers, Section 2 presents the system outline and mathematical modeling, Section 3 discusses the optimization and parameter tuning, Section 4 presents the simulation and results and it is subdivided into 3 subsections, 4.1 compares between the performance of optimal SSPID against optimal PID, 4.2 compares between the performance of optimal SSPID against optimal FOPID, 4.3 compares between the robustness of SSPID and traditional PID controllers, Section 5 discuss the experimental setup and results, and Section 6 concludes the paper.

2. System Dynamics and Outline

The mathematical modeling of the robotic arm is critical to describe its behavior and determine the relationship between joint position and velocity to torque/force or current/voltage. A single-joint robotic arm connected to a fixed field-DC motor is considered, the electrical model of the DC motor is shown in Figure 1. The robotic arm parameters used in this study were adopted from [19] and listed in Table 1. This provides a detailed description of the parameters relevant to the single-joint robotic arm connected to a fixed-field DC motor, including armature resistance, inductance, inertia, damping, torque constant, and back EMF constant.

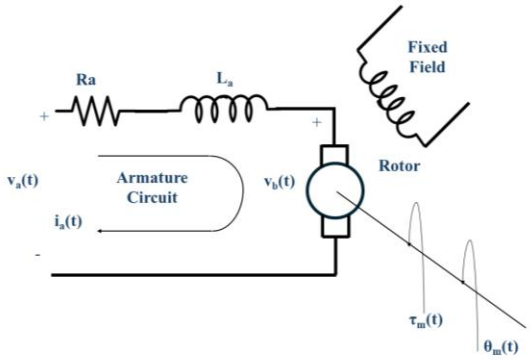

Figure 1: Fixed-field DC motor circuit diagram

Table 1: Single-joint robotic arm parameters.

Parameter	Value	Explanation
R_a	*1 Ω*	**Armature Resistance**
L_a	*0.23 H*	**Armature Inductivity**
J_{eqv}	*0.0333 kg.m²*	**Equivalent Inertia**
b_{eqv}	*0.012 N.m.s/rad*	**Equivalent Damping**
K_t	*0.023 N.m/A*	**Torque Constant**
K_b	*0.023 V.S/rad*	**Back EMF Constant**

The back-EMF is induced by the armature winding of the DC motor by rotation in the magnetic field, and the generated EMF is proportional to the motor speed [20]:

$$V(t) = L_a \frac{di(t)}{dt} + R_a i(t) + K_b \dot{\theta}(t) \quad (1)$$

where, V(t) is the applied voltage, measured in volts (v), La is the armature inductance, measured in Henry (H), Ra is the armature resistance, measured in Ohms (Ω), $i(t)$ is the motor current, measured in amperes (A), Kb is the back EMF constant, measured in volts per second per rad (v.s/rad), and $\dot{\theta}(t)$ is the angular velocity of the robotic arm, measured in radians (rad).

The mechanical equation of the robotic arm is:

$$J_{eqv} \ddot{\theta}(t) + b_{eqv} \dot{\theta}(t) = K_t i(t) \quad (2)$$

where, J_{eqv} is the equivalent inertia, measured in kilogram-metres squared (kg.m2), b_{eqv} is the equivalent damping, measured in newton-meters per radians per second (N.m.s/rad), K_t is the torque constant, measured in newton-meters per ampere (Nm/A), and $i(t)$ is the motor current.

Substituting the robotic arm model parameters from Table 1, the mechanical equation becomes:

$$0.0333 \ddot{\theta}(t) + 0.12 \dot{\theta}(t) = 0.023 i(t) \quad (3)$$

and the electrical equation becomes:

$$V(t) = 0.23 \frac{di(t)}{dt} + i(t) + 0.023 \dot{\theta}(t) \quad (4)$$

Taking Laplace transformation and solving equations (3) and (4), we got the transfer function as:

$$G = \frac{0.023}{0.0077659s^3 + 0.609s^2 + 0.1205s} \quad (5)$$

To derive the control law, first the system is expressed in state-space form. Define the state variables as:

$$x_1 = \theta(t) \quad (6)$$

$$x_2 = \dot{\theta}(t) \quad (7)$$

$$x_3 = i(t) \quad (8)$$

where, $\theta(t)$ is angular position, $\dot{\theta}(t)$ is the angular velocity, $i(t)$ is the motor current

The state-space equations become:

$$\dot{x}_1 = x_2 \quad (9)$$

$$\dot{x}_2 = \frac{K_t x_3 - b_{eqv} x_2}{J_{eqv}} = \frac{0.023 x_3 - 0.12 x_2}{0.0333} \quad (10)$$

$$\dot{x}_3 = \frac{V(t) - R_a x_3 - K_b x_2}{L_a} = \frac{V(t) - x_3 - 0.023 x_2}{0.23} \quad (11)$$

This system describes the third-order dynamics of the robotic arm and motor.

For Higher Order Sliding Mode Control (HOSMC), the sliding surface is given by:

$$s(t) = \ddot{e}(t) + 2\lambda \dot{e}(t) + \lambda^2 e(t) \quad (12)$$

where, $e(t) = x_1 - \theta_d$, is the position error, $\dot{e}(t) = x_2 - \dot{\theta}_d$ is the velocity error, $\ddot{e}(t) = \dot{x}_2 - \ddot{\theta}_d$, is the acceleration error

To keep the system on the sliding surface, we set $s(t) = 0$. This ensures that the system remains on the sliding surface at all times.
From the system dynamics, we know:

$$\ddot{x}_1 = \dot{x}_2 = \frac{0.023\,x_3 - 0.12\,x_2}{0.0333} \quad (13)$$

Substituting this into the sliding surface Equation (12):

$$s(t) = \frac{0.023\,x_3 - 0.12\,x_2}{0.0333} 2\lambda x_2 + \lambda^2(x_1 - \theta_d) \quad (14)$$

Setting $s(t) = 0$ gives us:

$$\frac{0.023\,x_3 - 0.12\,x_2}{0.0333} 2\lambda x_2 + \lambda^2(x_1 - \theta_d) = 0 \quad (15)$$

Solving for x_3:

$$0.023x_3 = 0.0333[-2\lambda x_2 - \lambda^2(x_1 - \theta_d)] + 0.12x_2 \quad (16)$$

$$x_3 = \frac{-0.0666\lambda x_2 - 0.0333\lambda^2(x_1 - \theta_d) + 0.12x_2}{0.023} \quad (17)$$

$$x_3 = (5.65 - 2.9\lambda)x_2 - 1.45\lambda^2(x_1 - \theta_d) \quad (18)$$

From Equation (8), Setting $\dot{x}_3 = 0$,to derive the equivalent control that keeps the system on the sliding surface and denoting V(t) as ueq(t) as it represents the equivalent control signal, and solving for ueq(t):

$$0 = \frac{u_{eq}(t) - x_3 - 0.023\,x_2}{0.23} \quad (19)$$

$$u_{eq}(t) = x_3 + 0.023\,x_2 \quad (20)$$

Substituting the expression for x_3 from equation (15):

$$u_{eq}(t) = (5.65 - 2.9\lambda)x_2 - 1.45\lambda^2(x_1 - \theta_d) + 0.023\,x_2 \quad (21)$$

Simplifying,

$$u_{eq}(t) = (5.673 - 2.9\lambda)x_2 - 1.45\lambda^2(x_1 - \theta_d) \quad (22)$$

For the switching control we use a PID controller to guarantee the robustness of the system:

$$u_{SW}(t) = k_p s(t) + k_i \int s(t) + k_d \frac{d}{dt} s(t) \quad (23)$$

where, k_p is the proportional gain, k_i is the integral gain, k_d is the derivative gain, Figure 2. shows the block diagram of PID controller.

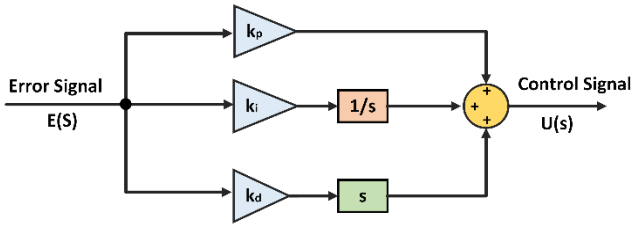

Figure 3: Block diagram of the PID controller

The final control law u(t) for position control of the single joint robotic arm will be a combination of the equivalent control and switching control as follows:

$$u(t) = u_{eq}(t) + u_{SW}(t) \quad (24)$$

Figure. 3 illustrates the block diagram of control system of single-joint robotic arm based on SSPID controller represented by equation (24).

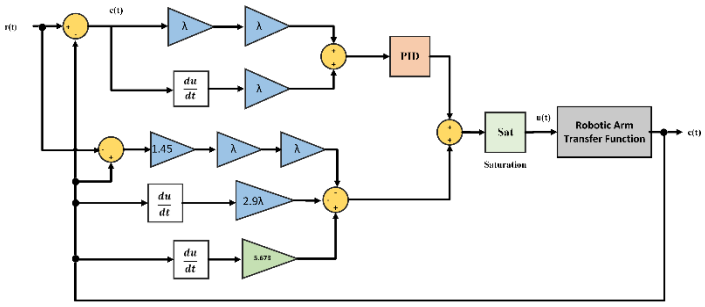

Figure 3: Block diagram of the sliding surface PID controller

3. Optimization of Controller Parameters

In this research, the Particle Swarm Optimization (PSO) algorithm is utilized to optimize the parameters of the SSPID, PID, and FOPID controllers. PSO is a population-based random optimization technique, it depends on the collective behavior of particles in a swarm to search for the optimal solution. PSO is effective in identifying optimal controller parameters, which is crucial for achieving precise system control. Figure 4 shows The PSO optimization process. It depicts the iterative process of particles updating their positions based on their personal best positions and the global best position [21-23].

PSO is selected for tuning the controllers' parameters due to its efficiency and robustness, particularly in complex control systems like robotic arms. PSO offers several advantages over traditional optimization techniques such as Genetic Algorithms (GA). Unlike GA, which involves computationally intensive operations like crossover and mutation, PSO employs a simpler mechanism based on the movement of particles in a search space, allowing for faster convergence to optimal solutions. Additionally, PSO requires fewer parameters to tune, which simplifies the optimization process and enhances its global search capabilities, thereby reducing the risk of premature convergence. This makes PSO particularly effective for minimizing performance metrics like the Integral of Time-weighted Absolute Error (ITAE), as it balances exploration and exploitation effectively. The iterative nature of PSO ensures precise optimization of SSPID parameters, leading to superior system performance characterized by improved rise time, minimized overshoot, and enhanced robustness, as evidenced by our simulation and experimental results.

The performance of PSO algorithm is impacted by several parameters. These parameters are carefully selected to ensure efficient convergence to the optimal solution. The parameters used in this research are listed in Table 2. These parameters include the swarm size or population size, number of iterations,

inertia weight, inertia damping ratio, personal learning coefficient, and global learning coefficient. The tuning process consists of adjusting these parameters to strike a balance between exploration and exploitation, ensuring that the algorithm approaches to the optimal solution. By carefully selecting these parameters, the PSO algorithm can identify the optimal controller parameters that minimize the fitness function.

Table 2: PSO parameters

Parameter	Value	Explanation
n_{size}	50	Swarm size
I	100	Iteration number
w	1	Inertia weight
w_{damp}	0.99	Inertia weight damping ratio
c_1	1	Personal Learning Coefficient
c_2	1	Global Learning Coefficient

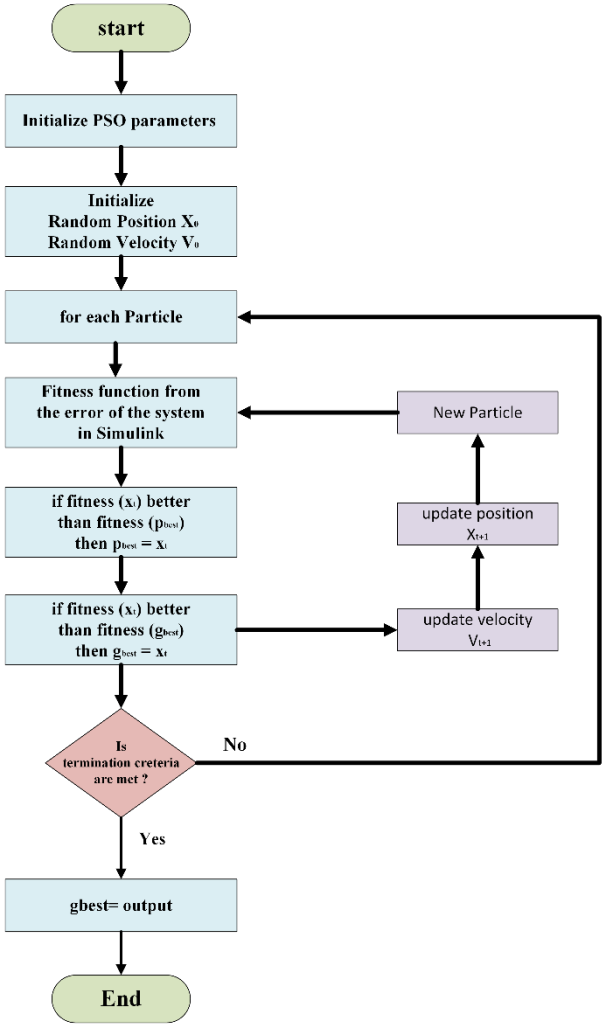

Figure 4: Flow chart of Particle Swarm Optimization

To evaluate the performance of the proposed controller, this research utilizes the Integral of Time-weighted Absolute Error (ITAE) fitness function shown in equation (25). The ITAE criterion is a widely used performance measure for control system design. ITAE minimizes the absolute error multiplied by time, reducing both the magnitude and duration of errors. By utilizing the ITAE fitness function the controller parameters can be optimized. this achieves improved performance and robustness, improves tuning control parameters, and enhances system performance and stability. The ITAE fitness function provides a balanced trade-off between achieving a fast response and minimizing overshoot, making it an effective metric for evaluating controller performance [24-25].

$$ITAE = \int_0^\infty t.|e(t)|dt \quad (25)$$

4. Simulation and Results

In this section, a comprehensive simulation study was performed for evaluating the performance of the proposed control strategy for a single-joint robotic arm system. The simulation was performed using MATLAB/Simulink version 2020a. The mathematical model of the system described by the transfer function in equation (5) served as the base for the simulation. The main objective of this simulation is to benchmark the performance of the optimal sliding surface PID (SSPID) controller against two other optimally tuned controllers: the PID and FOPID controllers. To achieve this, the output responses of each control system are systematically compared and analysed to evaluate their efficacy in controlling the robotic arm system. The simulation methodology employed in this study is illustrated in Figure 5, which outlines the step-by-step approach used to evaluate the performance of the proposed control strategies.

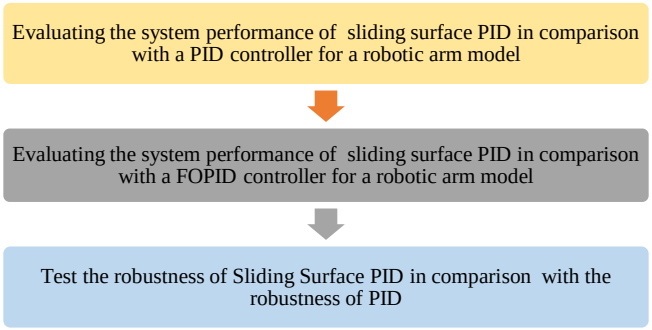

Figure 5. The scenarios for the simulation results

4.1. Performance Comparison of Optimal SSPID and Optimal PID Controllers for a Robotic Arm

In this section, the system performance of a single-joint robotic arm model controlled by an optimal SSPID controller is evaluated in comparison to an optimal PID controller. Significant improvements are observed in terms of faster response time and the elimination of overshoot when utilizing the SSPID controller. The optimal control parameters for the SSPID controller and their upper and lower bounds that are used in optimization are presented in Table 3.

Table 3: Optimal SSPID parameters

Parameter	Upper bound	Value	Lower bound
λ	0.01	15	20
k_p	0	29.8	50
k_i	0	3.72	5
k_d	0	2.14	5

where λ is the sliding surface coefficient, kp is the proportional gain, ki is the integral gain, and kd is the derivative gain of the PID controller.

The Optimal PID controller parameters are presented in Table 4.

Table 4: Optimal PID controller parameters

Parameter	Upper bound	Value	Lower bound
k_p	0	12.5	50
k_i	0	0.01	5
k_d	0	4.5	5

To evaluate the performance of the proposed control technique, the angular position output is measured under three distinct input scenarios. The system was subjected to step, square wave, and sine wave input reference signals. The simulation results offer an accurate analysis of the system's response to these varying input conditions. Additionally, the following figures highlight the advantages of the SSPID controller compared to the optimal PID controller in these scenarios.

Figure 6 compares the angular position response of a robotic arm using an optimal SSPID controller and a traditional PID controller with a step wave input as the reference signal. The results show that the optimal SSPID controller exhibits a faster rise time and minimal delay in reaching the desired angular position. It outperforms the conventional PID controller, which has a slower response.

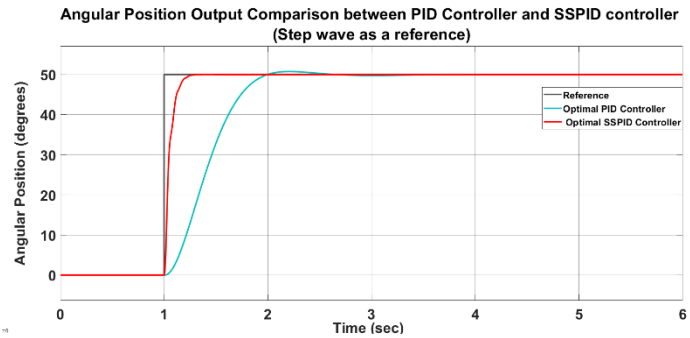

Figure 6.: The system response for step input considers optimal PID and optimal SSPID techniques

The SSPID controller demonstrates superior transient behavior, as there is no observable overshoot, whereas the conventional PID controller experiences a small amount of overshoot before settling. Both controllers ultimately reach a steady state, but the SSPID controller achieves a quicker convergence to the reference signal, indicating better overall system stability and responsiveness.

Furthermore, Figures 7 and 8 compare the performance of the optimal PID and optimal SSPID controllers when tracking square wave and sine wave reference signals, respectively. In both cases, the optimal SSPID controller demonstrates superior tracking accuracy. For the square wave input in Figure 7, the optimal SSPID controller outperforms the Optimal PID controller with faster rise times and more accurate responses at the sharp transitions of the square wave. The PID controller lags, with overshoot and slower tracking of the sudden changes.

Similarly, for the sine wave input in Figure 8, the SSPID controller shows reduced phase lag compared to the PID controller, resulting in better synchronization with the reference signal. The PID controller, however, exhibits a noticeable delay, leading to a phase shift that hinders its tracking precision.

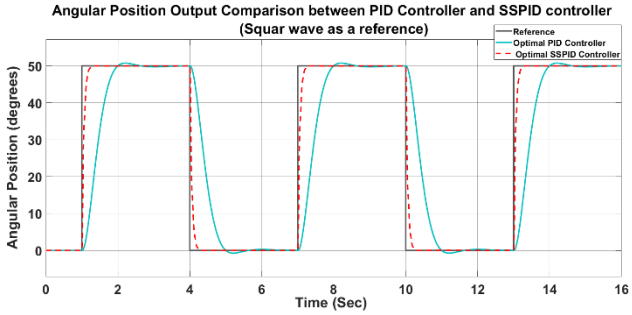

Figure 7: The system response for square wave input considering optimal PID and optimal SSPID techniques

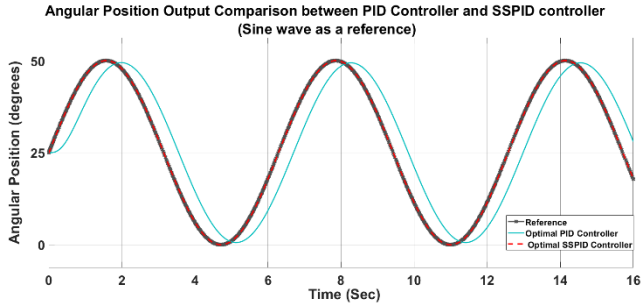

Figure 8: The system response for sine wave input considering optimal PID and optimal SSPID technique.

Overall, the optimal SSPID controller is more effective in maintaining accurate and fast responses for both periodic sine wave and discontinuous square wave reference signals.

Figure 9 gives some interesting information about how the control signal of the optimal PID controller and optimal Sliding Surface PID controller performs. The two signals are almost equal in peak because of the inclusion of the saturation module that saturates the control signal of the SSPID, but it was found very important to remember that SSPID has a higher response

rate and robustness. The saturation module ensures that the control actions remain within a specified range, preventing excessive control efforts that could lead to instability or actuator saturation.

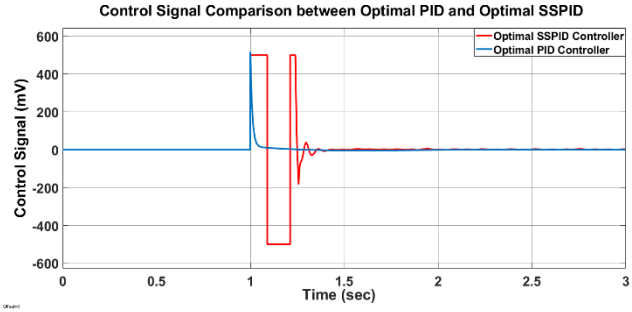

Figure 9: Comparison of the control signal of Optimal PID controller against Optimal SSPID Controller

The system specifications (rise time, settling time, peak, and peak time) for both the optimal SSPID and optimal PID are presented in Table 5.

Table 5: Specifications of the considered systems based on Optimal SSPID and Optimal PID controller

Parameter	Optimal SSPID	Optimal PID
Rise Time (sec)	0.10539	0.59677
Settling Time (sec)	1.1979	1.9125
Peak (degrees)	50	50.8
Peak Time (sec)	1.353	2.2083

In terms of the previous results and system specification comparison, the optimal SSPID controller outperforms the optimal PID controller in terms of rise time, tracking accuracy, and overshoot minimization. The difference is most pronounced in the step and square wave responses, where the SSPID controller shows significant advantages in fast and precise response, while the sine wave case reveals the SSPID's ability to reduce phase lag, thus enhancing its performance across a range of input types.

4.2. Performance Comparison of Optimal SSPID and Optimal FOPID Controllers for a Robotic Arm

Fractional Order PID (FOPID) is an enhanced version of the conventional PID controller that utilizes fractional calculus to generalize the orders of the integral and derivative components. Unlike traditional PID controllers, where the orders of both the integral and derivative terms are fixed as integers (usually 1), the FOPID allows these orders to be any real number. This added flexibility enables finer tuning and can lead to improved performance, especially in managing complex systems such as robotic arms [26-27]. In this simulation, the optimal SSPID controller for the robotic arm is compared against the FOPID controller. This benchmarking aims to evaluate the performance differences between the two control strategies, highlighting the advantages and limitations of each in managing the robotic arm's dynamics. The FOPID parameters are listed below in Table 6, where Λ is the integral term fractional order, and μ is the derivative term fractional order.

Table 6: Optimal FOPID controller parameters.

Parameter	Upper bound	Value	Lower bound
k_p	0	7	50
k_i	0	0.01	5
k_d	0	2	0
Λ	0	0.75	1
μ	0	0.82	1

The performance comparison between the SSPID and FOPID controllers is evaluated using three reference input signals: step, square, and sine waves, the subsequent figures present the angular position output for each input type, highlighting the superiority of the SSPID controller in key areas. Figure 10 illustrates the angular position output with a step signal as the reference. Based on Table 7, the SSPID controller significantly outperforms the FOPID controller in terms of rise time and settling time, achieving a rise time of 0.10539 seconds compared to 1.0389 seconds for FOPID, and a settling time of 1.1979 seconds versus 2.6633 seconds for FOPID. The SSPID controller also reaches the peak value in just 1.3067 seconds, whereas the FOPID requires 3.5632 seconds, making SSPID much faster in responding to the step input.

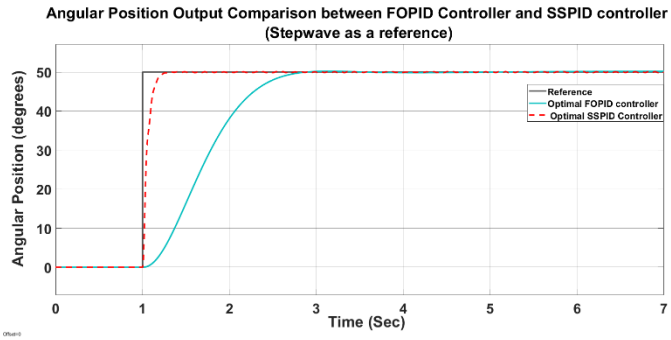

Figure 10: The system response for step input considers optimal FOPID and optimal SSPID techniques

Figure 11 shows the angular position output using a square wave as the reference. The SSPID controller excels in handling the sharp transitions of the square wave, responding with higher accuracy and minimal delay. In contrast, the FOPID controller's slower rise and settling times result in a delayed response to these abrupt changes. Despite FOPID's inherent flexibility in fractional tuning, the SSPID demonstrates superior performance in following rapid variations in the input.

Figure 12 compares the angular position output for a sine wave input. However, the SSPID displays more precise phase tracking and reduced phase lag compared to the FOPID.

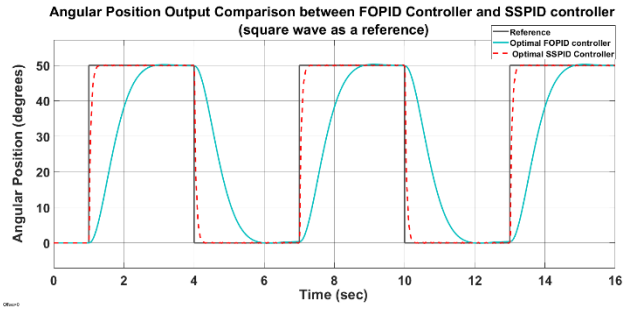

Figure 11: The system response for square wave input considering optimal FOPID and optimal SSPID techniques

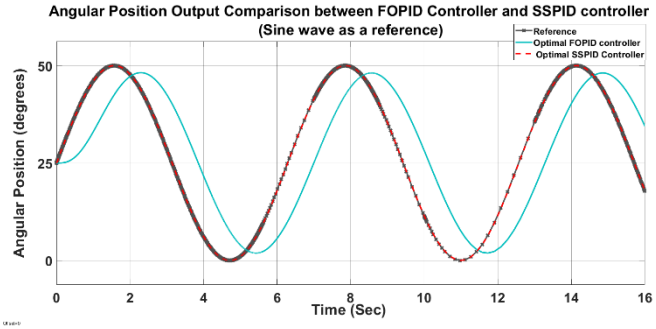

Figure 12. The system response for sine wave input considering optimal FOPID and optimal SSPID techniques

In Figure 13, the comparison of control signals between the optimal Fractional Order PID (FOPID) controller and the optimal Sliding Surface PID (SSPID) controller reveals a significant difference in their magnitudes. The control signal generated by the FOPID controller is approximately half that of the SSPID. This disparity can be attributed to the inherent design and tuning of the controllers; while the FOPID is optimized for a smooth response, it does not exert as much control effort as the SSPID. The SSPID, leveraging sliding mode control principles, is designed to react more aggressively to disturbances and variations in system dynamics, resulting in a higher control signal output. Despite the lower signal from the FOPID, it is essential to note that the SSPID's enhanced control effort translates into faster response times and improved robustness, allowing it to maintain better performance under dynamic conditions. This comparison underscores the effectiveness of the SSPID approach in achieving optimal control performance for robotic arm applications, particularly in scenarios requiring quick adjustments and resilience against external disturbances.

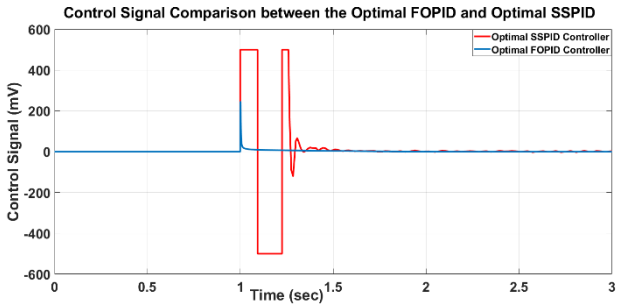

Figure 13: Comparison of the control signal of Optimal FOPID controller against Optimal SSPID Controller

Table 7 reinforces the fact that the optimal SSPID controller is more efficient in terms of rise time, settling time, and overall responsiveness compared to the FOPID controller, making it a more robust choice for controlling systems such as robotic arms, where fast and accurate tracking is critical.

Table 7: Specifications of the considered systems based on Optimal SSPID and Optimal FOPID controller

Parameter	Optimal SSPID	Optimal FOPID
Rise Time (sec)	0.10539	1.0389
Settling Time (sec)	1.1979	2.6633
Peak (degrees)	50	50
Peak Time (sec)	1.3067	3.5632

4.3. Robustness Evaluation of an Optimal SSPID Controller Versus an Optimal PID Controller in a Robotic Arm Model

In this section, we analyze the robustness of the optimal SSPID and optimal PID controllers for the robotic arm model under parameter variations. Two case studies are presented where the system parameters - specifically armature resistance, armature inductance, and load mass - are varied by 20% and 50%. These variations affect the transfer function of the robotic arm, allowing us to assess the controllers' ability to maintain performance under non-ideal conditions.

In the first case study, the system parameters are varied by 20%. The performance of both the optimal PID and optimal SSPID controllers is evaluated and compared. Figure 14 shows the output variation of the robotic arm controlled by the optimal PID controller when the parameters are changed by ±20%. The angular position varies significantly for each case, indicating that the optimal PID controller's performance is affected by the parameter changes.

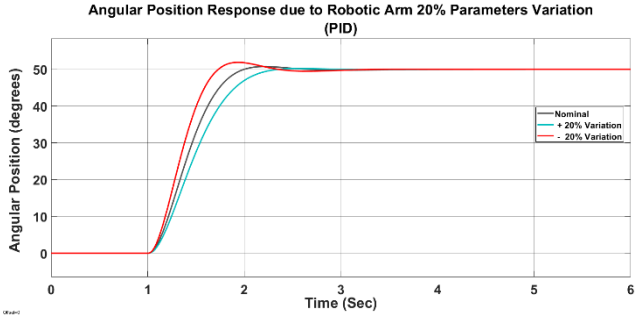

Figure 14: The system response due to parameters 20% variation for optimal PID controller

Table 8 presents the robotic system specifications, where the rise time, settling time, and peak values for different variations (0.8x, 1.0x, 1.2x of the original values) are listed.

Table 8. Specifications comparison of Optimal PID controller robustness in case of 20% parameter variations

Parameter	0.8 x	x	1.2 x
Rise Time (sec)	0.46447	0.59677	0.74003
Settling Time (sec)	2.1543	1.9125	2.1653
Peak (degrees)	51.8	50.8	50.2
Peak Time (sec)	1.9495	2.2083	2.5698

Figure 15 shows the output variation of the robotic arm controlled by the optimal SSPID controller for the same 20% parameter variation. Unlike the PID controller, the SSPID controller maintains a nearly constant angular position across all variations, demonstrating its robustness.

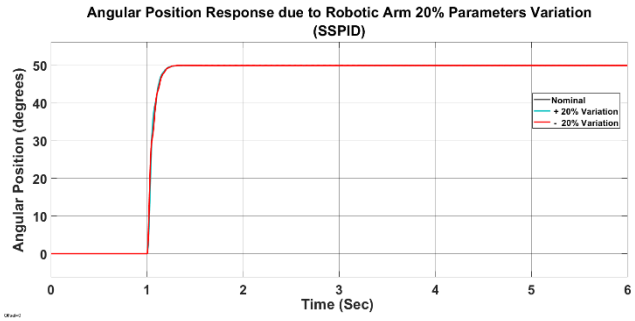

Figure 15: The system response due to parameters 20% variation for optimal SSPID controller

Table 9 shows the corresponding numerical results, confirming minimal changes in rise time, settling time, and peak values.

Table 9. Specifications comparison of Optimal SSPID controller robustness in case of 20% parameter variations.

Parameter	0.8 x	x	1.2 x
Rise Time (sec)	0.11414	0.10539	0.10475
Settling Time (sec)	1.2002	1.1979	1.1972
Peak (degrees)	50	50	50
Peak Time (sec)	1.3781	1.353	1.3015

The results from this case study demonstrate that the SSPID controller is significantly more robust than the PID controller, as its performance remains stable despite the parameter changes, whereas the PID controller exhibits noticeable variations in key performance metrics.

In the second case, the system parameters are varied by 50% to further test the robustness of the controllers. Figure 16 illustrates the output variation of the robotic arm controlled by the optimal PID controller under a 50% parameter variation. As seen in the previous case, the PID controller's performance is notably impacted, with changes in rise time, settling time, and peak angular position.

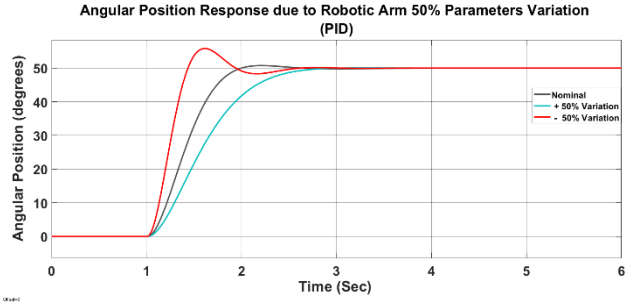

Figure 16: The system response due to parameters 50% variation for optimal PID controller

Table 10 provides the quantitative data highlighting the significant variations in performance.

Table 10: Specifications comparison of Optimal PID controller robustness in case of 50% parameter variations

Parameter	0.5 x	x	1.5 x
Rise Time (sec)	0.28709	0.59677	0.96261
Settling Time (sec)	2.3633	1.9125	1.913
Peak (degrees)	55.8	50.8	50
Peak Time (sec)	1.6192	2.2083	3.3034

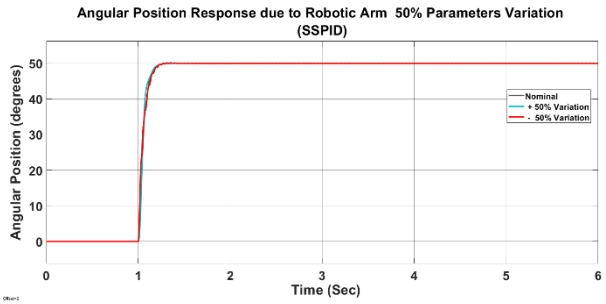

Figure 17: The system response due to parameters 50% variation for optimal SSPID controller

Figure 17 presents the output variation for the optimal SSPID controller under the same 50% variation. Similar to the 20% variation case, the SSPID controller maintains consistent performance, with very little change in the angular position response.

Table 11 details the numerical values, which show minimal deviation from the nominal case, further confirming the robustness of the SSPID controller.

Table 11: Specifications comparison of Optimal SSPID controller robustness in case of 50% parameter variations

Parameter	0.5 x	x	1.5 x
Rise Time (sec)	0.12316	0.10539	0.08774
Settling Time (sec)	1.2133	1.1979	1.1841
Peak (degrees)	50	50	50
Peak Time (sec)	1.345	1.353	1.3658

From the results of both case studies, it is clear that the optimal SSPID controller exhibits superior robustness compared to the optimal PID controller. While the PID controller experiences significant variations in its response characteristics when the system parameters are varied, the SSPID controller maintains nearly identical performance. This robustness is critical for applications where system parameters are uncertain or subject to change, making the SSPID controller a more reliable choice for controlling robotic arms in such environments.

5. Experimental Design and Results

To evaluate the real-world performance of the optimal SSPID and optimal PID controllers, an experimental model of a single-joint robotic arm was constructed. This section describes the experimental setup, the control circuit, and the comparison of angular position outputs from both controllers.

The experimental setup consisted of a single-joint robotic arm with a mass of 0.85 kg, driven by a brushed DC motor. The arm's reference position was set to the horizontal (0 degrees), serving as the starting point for all experiments. The angular position of the robotic arm was measured using a carbon-film potentiometer with a measurement range of 220 degrees, which was coupled to the motor shaft via copper gears for accurate position feedback. The output response was observed and recorded using a PeakTech P 1337 oscilloscope, capable of measuring signals with 100 MHz bandwidth and supporting 2 channels for simultaneous data collection.

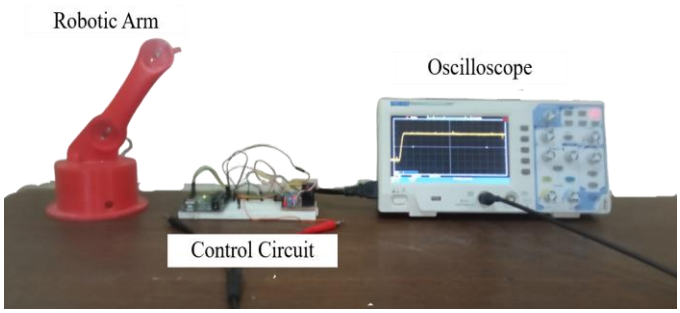

Figure 18: Experimental Setup of Robotic Arm Model

The control circuit, designed for this experiment, was implemented using the Arduino IDE and the Arduino C programming language.

Figure 18 provides an overview of the experimental setup, while Figure 19 shows the detailed control circuit diagram.

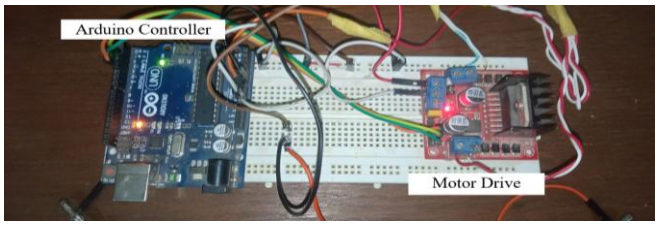

Figure 19: Control Circuit of the Model.

Figure 20 presents the angular position output comparison between the optimal PID and optimal SSPID controllers during the experiment. The results show a consistent performance advantage for the SSPID controller in terms of tracking accuracy, rise time, and response stability.

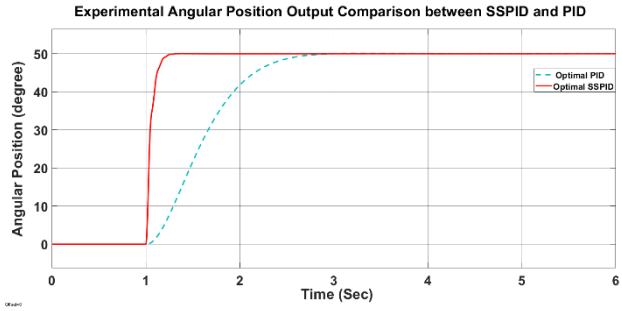

Figure 20: Experimental output comparison between Optimal SSPID and Optimal PID controllers

The experimental results align with the simulation findings, reinforcing the superior performance of the SSPID controller over the PID controller. The SSPID consistently showed faster response times and improved tracking accuracy across different types of input signals. Moreover, the controller effectively minimized overshoot and steady-state error, critical for achieving high-precision control in robotic arm applications.

In summary, the experimental results confirm that the optimal SSPID controller provides enhanced performance over the optimal PID and optimal FOPID controller, particularly in terms of rise time, settling time, and overall tracking accuracy.

6. Conclusion

This research provides a detailed investigation into the control of a single-joint robotic arm, benchmarking the performance of an optimal Sliding Surface PID (SSPID) controller against both an optimal PID controller and an optimal Fractional Order PID (FOPID) controller. Through rigorous simulation and experimental validation, the SSPID controller demonstrated superior performance in terms of robustness, response time, tracking accuracy, and overshoot minimization.

The simulation results highlighted the significant advantages of the optimal SSPID controller, particularly under varying reference input signals such as step, square, and sine waves. In particular, the optimal SSPID controller exhibited faster rise times, negligible overshoot, and greater accuracy in tracking the reference signal when compared to both the optimal PID and optimal FOPID controllers. The robustness of the optimal SSPID was further highlighted by its ability to maintain stable performance despite 20% and 50% variations in system parameters such as armature resistance, inductance, and load mass. These variations had a more pronounced negative impact on the performance of the PID and FOPID controllers, underscoring the SSPID's resilience.

Future work could extend the proposed control technique to multi-joint robotic arms to evaluate its effectiveness in more complex systems. Additionally, further exploration of new forms of sliding mode control could enhance the adaptability and robustness of the SSPID controller in broader control applications.

These results demonstrate the potential of SSPID control to significantly outperform traditional PID and FOPID techniques, establishing a strong foundation for future research and practical implementation in advanced robotic systems.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Y. H. T. Htun, M. S. Hlaing, and T. T. Hla, “Master-Slave Synchronization of Robotic Arm using PID Controller,” Indonesian Journal of Electrical Engineering and Informatics, vol. 11, no. 1, 2023, doi: 10.52549/ijeei.v11i1.4171.

[2] P. Sutyasadi, “Control Improvement of Low-Cost Cast Aluminium Robotic Arm Using Arduino Based Computed Torque Control,” Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, vol. 8, no. 4, 2022, doi: 10.26555/jiteki.v8i4.24646.

[3] A. J. Ishak, A. C. Soh, and M. A. Ashaari, “Position control of arm mechanism using pid controller,” J Theor Appl Inf Technol, vol. 47, no. 2, 2013.

[4] 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, 2021, doi: 10.18196/jrc.26140.

[5] D. Singh Rana and A. Professor, “Modelling, stability analysis and control of flexible single link robotic manipulator,” 2007. [Online]. Available: www.ijareeie.com

[6] H. Yildiz, N. Korkmaz Can, O. C. Ozguney, and N. Yagiz, “Sliding mode control of a line following robot,” Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 42, no. 11, 2020, doi: 10.1007/s40430-020-02645-3.

[7] L. S. Mezher, “Position control for dynamic DC MOTOR with robust PID controller using MATLAB,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 3, 2019, doi: 10.30534/ijatcse/2019/92832019.

[8] Nihad Wazzan A.Basil N.Raad M.Kasim Mohammed H, “PID Controller with Robotic Arm using Optimization Algorithm,” International Journal of Mechanical Engineering Education, 2022.

[9] A. Bârsan, “Position Control of a Mobile Robot through PID Controller,” Acta Universitatis Cibiniensis. Technical Series, vol. 71, no. 1, 2020, doi: 10.2478/aucts-2019-0004.

[10] W. W. Naing, K. Z. Aung, and A. Thike, “Position Control of 3-DOF Articulated Robot Arm using PID Controller,” International Journal of Science and Engineering Applications, vol. 7, no. 9, 2018, doi: 10.7753/ijsea0709.1001.

[11] C. J. Fallaha, M. Saad, H. Y. Kanaan, and K. Al-Haddad, “Sliding-mode robot control with exponential reaching law,” IEEE Transactions on Industrial Electronics, vol. 58, no. 2, 2011, doi: 10.1109/TIE.2010.2045995.

[12] S. Islam and X. P. Liu, “Robust sliding mode control for robot manipulators,” IEEE Transactions on Industrial Electronics, vol. 58, no. 6, 2011, doi: 10.1109/TIE.2010.2062472.

[13] J. D. J. Rubio, “Sliding mode control of robotic arms with deadzone,” IET Control Theory and Applications, vol. 11, no. 8, 2017, doi: 10.1049/iet-cta.2016.0306.

[14] B. A. Bazzi and N. G. Chalhoub, “Fuzzy sliding mode controller for a flexible single-link robotic manipulator,” JVC/Journal of Vibration and Control, vol. 11, no. 2, 2005

[15] S. Nadda and A. Swarup, “Integral Sliding Mode Control for Position Control of Robotic Manipulator,” in 2018 5th International Conference on Signal Processing and Integrated Networks, SPIN 2018, 2018. doi: 10.1109/SPIN.2018.8474267.

[16] M. Tomar, S. Mandava, N. Hemalatha, V. R. Rao, and R. K. Mandava, “Design of PID, FLC and Sliding Mode Controller for 2-DOF Robotic Manipulator: A Comparative Study,” International Journal of Mathematical, Engineering and Management Sciences, vol. 8, no. 1, 2023, doi: 10.33889/IJMEMS.2023.8.1.006.

[17] H. Wang, Y. Pan, S. Li, and H. Yu, “Robust Sliding Mode Control for Robots Driven by Compliant Actuators,” IEEE Transactions on Control Systems Technology, vol. 27, no. 3, 2019, doi: 10.1109/TCST.2018.2799587.

[18] Y. Wang, Z. Zhang, C. Li, and M. Buss, “Adaptive incremental sliding mode control for a robot manipulator,” Mechatronics, vol. 82, 2022, doi: 10.1016/j.mechatronics.2021.102717.

[19] F. A. Salem, “Modeling, Simulation and Control Issues for a Robot ARM; Education and Research (III),” International Journal of Intelligent Systems and Applications, vol. 6, no. 4, 2014, doi: 10.5815/ijisa.2014.04.03.

[20] M. Vaijayanti, “Robotic Arm Control Using Pid Controller And Inverse Kinematics,” International Journal of Engineering Development and Research, vol. 5, no. 2, 2017.

[21] O. Djaneye-Boundjou, X. Xu, and R. Ordonez, “Automated particle swarm optimization based PID tuning for control of robotic arm,” in Proceedings of the IEEE National Aerospace Electronics Conference, NAECON, 2016. doi: 10.1109/NAECON.2016.7856792.

[22] A. Sathish Kumar et al., “An intelligent fuzzy-particle swarm optimization supervisory-based control of robot manipulator for industrial welding applications,” Sci Rep, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-35189-2.

[23] K. J. Wu and M. Y. Chen, “Controller with the PID Parameters Optimization by PSO for a 6-DOF Robotic Arm,” in 2023 International Conference on Fuzzy Theory and Its Applications, iFUZZY 2023, 2023. doi: 10.1109/iFUZZY60076.2023.10324208.

[24] W. Lv, D. Li, S. Cheng, S. Luo, X. Zhang, and L. Zhang, “Research on PID control parameters tuning based on election-survey optimization algorithm,” in 2010 International Conference on Computing, Control and Industrial Engineering, CCIE 2010, 2010. doi: 10.1109/CCIE.2010.198.

[25] Y. K. Soni and R. Bhatt, “Simulated Annealing optimized PID Controller design using ISE, IAE, IATE and MSE error criteria,” International Journal of Advanced Research in Computer Engineering & Technology, vol. 2, no. 7, 2013.

[26] A. K. M and R. Younus Fadhil, “Performance Improvement of the Robotic Arm using Fractional Order PID,” Engineering and Technology Journal, vol. 34, no. 14, 2016, doi: 10.30684/etj.34.14a.2.

[27] S. Husnain and R. Abdulkader, “Fractional Order Modeling and Control of an Articulated Robotic Arm,” Engineering, Technology and Applied Science Research, vol. 13, no. 6, 2023.