International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Integration and Optimization of Software to Control Robotic Arms: A Comprehensive Study on Modeling, Hardware Implementation, and PID Tuning

Sandeep Yadav[1]*, Sunil Kumar[2], Manoj Goyal[3]

Submitted: 11/12/2023 **Revised**: 26/01/2024 **Accepted**: 01/02/2024

Abstract: Software-controlled robots are the most common type of robotic arms, and their integration is critical to the progress of industries. In this study article, a robotic arm is created using SolidWorks and then its hardware implementation is finished following virtual inspection. Using techniques for higher-order system control such as Genetic Algorithm (GA), Ziegler-Nichols (Z-N), Ant Colony Optimisation (ACO), and Particle Swarm Optimisation (PSO), the movement and design of the DC servo motor's PID controller are accomplished through PID tuning. In particular, the Genetic Algorithm (GA) is used to improve robotic arm control. MATLAB Simulink simulates the software-based control of the robotic arm and offers a graphical user interface with easily navigable settings. This paper describes in detail how the robotic arm was created step by step and how its hardware and software were implemented. Additionally, Arduino hardware is used for data collecting and control system assessment when observing and evaluating the robotic arm's output features. A range of input variables are tested using the robotic arm.

***Keywords**: PID Controller, ARDUINO UNO, Simulink, Tuning, SolidWorks, Robot Arm*

1. Introduction –

Robotics technology, with its great performance and efficiency, provides many benefits for industrial systems. Many studies and applications of control theory have surfaced in recent years, with a special emphasis on robotic system control. Nowadays, robots are used in educational, medical, and industrial contexts. One of the most common types of robots is the manipulator. This topic is fascinating, particularly when robots are being used in difficult, hostile settings where it is prohibitive for humans to access. For example, working in areas such as nuclear or chemical reactors presents serious risks to human life; nevertheless, using robots reduces these risks while preserving operational effectiveness [1]. Analyzing and comprehending the dynamic responses of DC servo motors employed as actuators in robot manipulators, along with exploring control methodologies, is imperative for ensuring precise and seamless motion of robotic arm models. Certain references suggest that Genetic Algorithms (GA) exhibit minimal overshoot and enhanced stability in comparison to PID controllers [2].

The aim of the simulation of an uncharged spindle in a direct current (DC) motor is to approximate the real system's behaviour in the absence of external charging [3]. A simulation that closely mimics the features of an actual DC servo motor is the goal [3-5]. In MATLAB's Single Input, Single Output (SISO) design tool, the stability response of systems is analyzed. The root locus approach is employed to generate a Proportional-Integral-Derivative (PID) controller, and the system's stability is assessed in the process [4-7]. The essential structure of a Fuzzy Logic Controller (FLC) consists of fuzzification, knowledge base, decision-making, and defuzzification units [7, 8]. There are two inputs and one output in a fuzzy inference system (FIS). Error (E) and Delta Error (DE) make up the inputs [9]. Humans should work on enhancing and refining control techniques for industrial robot manipulators to manage and optimize plant performance. This can ultimately replace human involvement in risky, challenging, and monotonous tasks within hazardous environments [10]. The dynamic characteristics of robot manipulators involve the identification of nonlinearities and external input elements, such as friction, load variations, and disturbances [11]. The PID controller is a very powerful control tool in the field of industrial automation because of its resilience and simplicity [12]. Achieving precise control of a robotic arm, as exemplified by Mohammed Hussein, has posed significant challenges in recent times. The nonlinearity inherent in robot manipulators often renders the conventional PID controller inadequate for achieving the necessary tracking control performance [13-14]. Researchers were inspired to develop the network that combines a PID controller and a fuzzy logic controller [14–16]. Improved dependability and shorter computation times were the main draws of using the Differential Evaluation Algorithm [17], Bacterial Foraging Optimisation Algorithm [18], Gravitational Search Algorithm [19], Simulated Annealing [20], Artificial

Sant Longowal Institute of Engineering & Technology, Longowal -148106, India
**sandeepyadav.sliet@gmail.com, sunil_thappa@yahoo.com, manojgoyal.sliet12@gmail.com*

Immune System [21], Artificial Bee Colony Optimisation [22], African Buffalo Optimisation [23], Ant Colony Optimisation (ACO) [24], and Bat Algorithm [25]. Notably, in addition to a PID controller, the network itself included a fuzzy logic controller [26].

2. Methodology

There are two main ways that this research contributes. Firstly, the optimizer design tool in Matlab is used to create a PID controller for a higher-order system. Second, the same system is used to execute multiple computer algorithmsant colony optimisations (ACO), Zigler Nichols (Z-N), and particle swarm optimisation (PSO). The PID controller (GA) is adjusted with the help of a genetic algorithm. The response system is utilised to compare the automatic tuning of the PID controller with a suggested fuzzy logic controller, thereby illustrating simulation findings for higher-order systems. The structure of the paper is as follows: While Section II provides more detail on the arm robot manipulator model, Section III provides information on the design and significance of the DC servo motor's transfer function. In Section V, several computational strategies are used to produce the tuning for the DC servo motor using MATLAB/Simulink. The PID controller for the DC servo motor is explained in Section IV.

1.1 Robotic Arm Manipulator Model

Fig.1 demonstrates the effectiveness of the PID controller in achieving precise tracking through simulation on a model of a robotic arm manipulator. The model represents the experimental configuration of a research-scale arm robot manipulator with four degrees of freedom, as depicted in Fig. 1. Furthermore, Fig.2 illustrates the integral role of Arduino in the arm robot manipulator model, outlining its schematic connection to all DC motor points and electrical components.

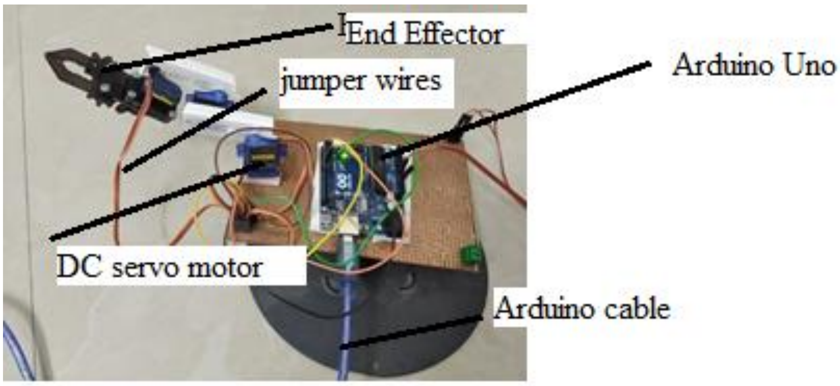


Fig.1 Model of a robotic arm manipulator

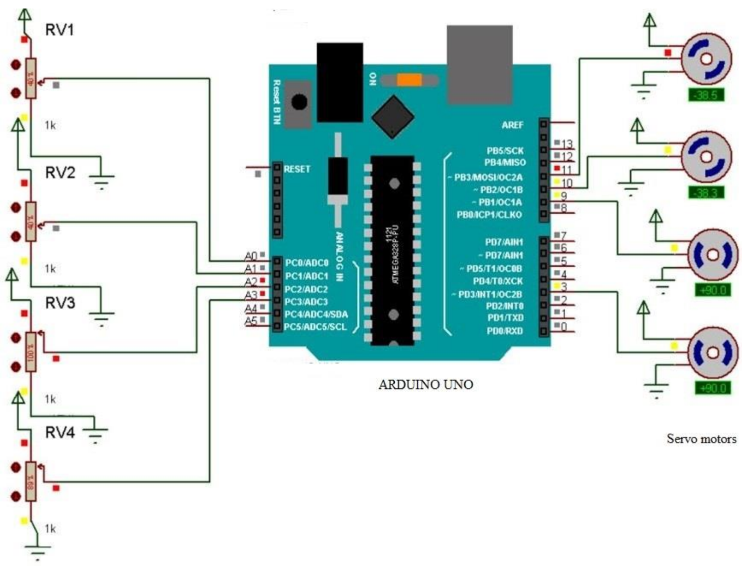


Fig.2 Schematic Representation of Wire Connections and Electronic Hardware (Arduino)

3. Dc Servo Motor Description

DC servo motors are the preferred choice for achieving wide-range speed control in various applications where variable speeds are required. In the simulation, the model of the DC servo motor is depicted in Figure 3. Table 1 provides a comprehensive list of the motor's parameters, nomenclature, and corresponding values. [6]

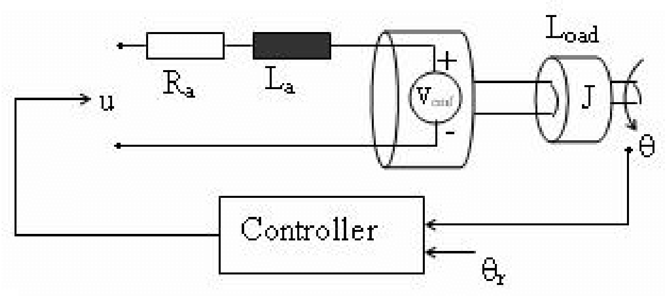


Fig.3 Schematic Configuration of the DC Servo Motor [6]

Table 1 DC Servo Motor Parameters, Nomenclature, and Corresponding Values [25]

Parameter	**Symbol**	**Value**
Moment of Inertia	Jm	$0.000052 kg.m2$
Friction coefficient	B_m	$0.01 N.ms$
Back EMFconstant	K_b	$0.235 V/rad.s-1$
Torque constant	K_a	$0.235 Nm/A$
Electric resistance	R_a	$2 ohm$
Electric endurance	L_a	$0.23 H$

The goal of modeling a DC servo motor is to emulate the behavior of the actual motor as accurately as possible. By considering the relevant parameters [1], one can derive the transfer function of the DC servo motor specifically for position regulation. Fig. 4 illustrates the block diagram of a DC servo motor, including the load torque value. [5]

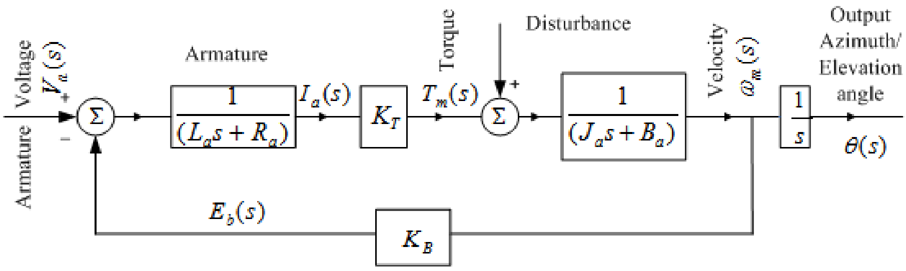


Fig.4 Block diagram of a DC servo motor

3.0 PID Controller Design

A PID controller is under development for a higher-order system described by the transfer function equation (1). Prior to initiating the simulation and implementation of the PID controller, utilize MATLAB's SISO design tool to create the PID controller design. Subsequently, perform a Root Locus (R-H) study on the stability response of the system [4-7]. The design architecture for the automatic PID tuning tool in MATLAB was implemented using Simulink.

$$\frac{F_{15}(s)}{f_{ref}(s)} = \frac{\mu_{H(K_{gp}s+K_{gI})(R_R s+K_R)}}{(K_{gd}s^2+K_{gP}s+K_{gi})(I_R s^3+R_R I_R s^2+K_R I_R s)+\mu_{H(K_{gp}s+K_{gI})(R_R s+K_R)}} \quad (1)$$

Table 2 Parameters of Various Tuning Methods

GA parameters	**ACO parameters**	**PSO parameters**
Population size-300	Population size-100	Population size-100

Mutation rate-0.01	No. of ants=50	Wmax=0.6
Arithmetic crossover	No of path=20	Wmin=0.1
Iteration:100	Iteration:100	Iteration:100

Table 3 Comparative Performance of Zigler-Nichols, Genetic Algorithm, Ant Colony Optimization, and Particle Swarm Optimization Methods

characteristics	**Z-N**	**GA**	**ACO**	**PSO**
SETTING TIME	0.775	0.000414	0.33	0.272
Rise time(sec)	0.101	0.000042	0.0358	0.0474
%overshoot	38	6	24	16
PID Proportional coefficient(Kp)	0.39	4.47*10-7	0.002	0.47
Integral coefficient(Ki)	210	484.49	321	426
Derivative coefficient (Kd)	32	0.4499	1.46	1.32

Fig. 5 and Table 3 present a comparative analysis of PID parameter tuning using genetic algorithm (GA), Ziegler-Nichols (Z-N), ant colony optimization (ACO), and particle swarm optimization (PSO) [27]. Each technique employs an equal number of populations. The MATLAB code aligns with the methodology illustrated in Figure 5 for the tuning process. Among all the strategies, the Genetic Algorithm demonstrates superior results.

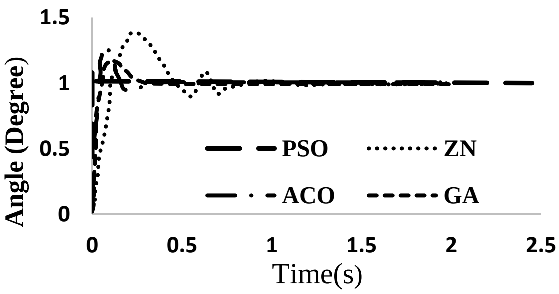


Fig. 5 Optimization of PID Controller Parameters using Various Methods and Comparative Analysis of GA/Z-N/ACO/PSO

Design a higher-order system with a transfer function equation (1) and develop a PID controller using MATLAB's Simulink design tool. Before simulating and implementing the PID controller, investigate the stability response systems using the Root Locus-Hearts approach (R-H approach). Utilize Simulink optimization techniques as the basis for the design tool to automatically tune the PID controller in MATLAB.

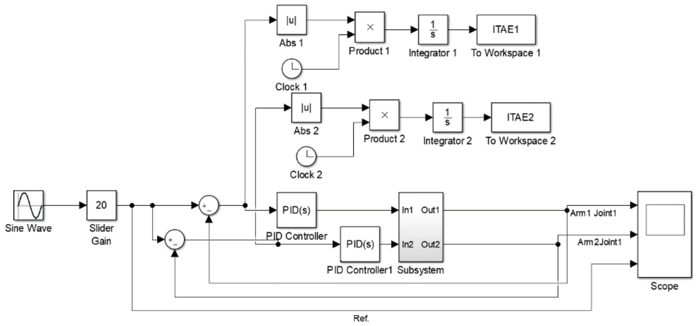


Fig.6 PID Controller Simulation for Robotic Arm in Simulink

Following the development and analysis of the PID controller response, Fig. 6 demonstrates the utilization of the Simulink model of the PID controller on Arduino. Using the Simulink Support Package for Arduino Hardware and integrating a potentiometer are part of this integration [29]. The computer language Simulink makes it possible to simulate complex systems. Fig. 9 shows the PID controller's implementation result according to the Simulink model.

3.1 Simulation and Experimentation

In this scenario, a simulated robotic model was developed using Simulink, a software environment created by MathWorks Inc. This model served to validate dynamic computations and simulate the motion of robotic devices under the influence of a PID controller, which stands for Proportional-Integral-Derivative controller. The PID controller parameters were determined using Genetic Algorithm (GA) tuning.

Fig. 7(a) shows how the single-arm manipulator model was created in SolidWorks, a popular 3D modelling programme, and then simulated in MATLAB using Simulink. When the model was run in MATLAB Simulink, it showed unconstrained mobility. Using the GA-derived parameters, the PID tool in Simulink was used to control each rotational joint (DC motor). This allowed the model to stabilise quickly, which is consistent with the GA method.

The response of the DC motor for all joints, including the base, elbow, and shoulder, is illustrated in Fig.10 and aligns well with the PID results obtained through the Hybrid controller's mathematical methods. Table 4 displays the stable configuration of the model. This underscores that the simulated model faithfully reproduces the behavior of real-world robotic devices when subjected to closed-loop control commands of joint torques using a PID controller. The geometric, dynamic parameters, and coordinate system are depicted in Fig. 7(b).

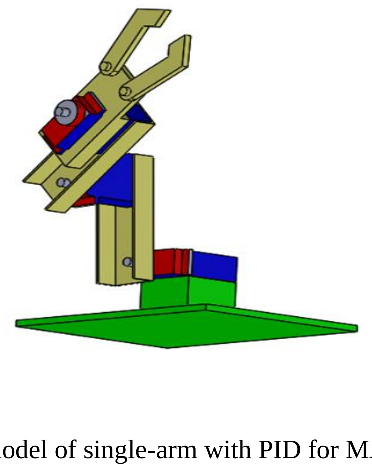


Fig. 7(a) SolidWorks model of single-arm with PID for MATLAB simulation

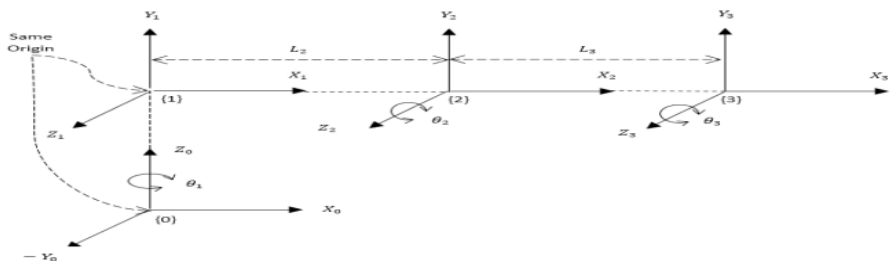


Fig. 7(b) Coordinate Frame assignment of the robotic manipulator each joint

The PID parameters, obtained through the Genetic Algorithm (GA) tuning method, were applied to simulate a single-arm manipulator model created in SolidWorks and implemented in MATLAB Simulink, as illustrated in Fig.7. The model, consisting of rotating joints driven by DC motors, demonstrated smooth movement during simulation in MATLAB Simulink. The PID controller, integrated with each rotating joint, effectively utilized the GA-derived parameters listed in Table 2. The simulation results revealed that the model achieved stability within a short time frame, aligning with the efficiency of the GA technique. The stable configuration of the model is detailed in Table 4. The DC motor responses for the base, elbow, and shoulder joints, as depicted in Fig. 7(a), exhibited excellent agreement with the PID results derived from the mathematical approach of the Hybrid controller.

4. Results and Discussion

Using the parameters listed in Table 4, a simulation was conducted for a three-degree-of-freedom robotic manipulatorA three-degree-of-freedom (3-DOF) robotic manipulator's PID controller parameters were selected by experimenting with several high-gain values in a trial-and-error manner. Using high gains of 3 and 4 was shown to cause a protracted 30-second oscillation of the robot's tip during simulations. Conversely, using a high gain of 1 produced an oscillation that lasted for 7 seconds at first, and then there was a slow convergence towards the target trajectory value. The PID controller parameters were determined using different methods, including Ziegler-Nichols (Z-N), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). Among these methods, the GA approach yielded the most favorable results for the system, with specific parameter values detailed in Table 3. Various gain values, ranging from μ_H=1 to 10, were tested in conjunction with different methods for PID controller parameter tuning. Interestingly, the Genetic Algorithm (GA) method produced the most favorable results specifically at μ_H=1. In contrast, the alternative methods and their respective parameterizations did not yield outcomes as promising as those achieved with the GA method at μ_H=1. Fig.5 provides a visual representation of the simulation results obtained using these different methods.

5. Experimentation

After fine-tuning the PID controller parameters and validating the system changes through simulation, the PID controller is implemented in the hardware system. This controller assesses the output voltage against the input voltage by utilizing the current motor position obtained through the Arduino card's voltage reading from the output potentiometer. Upon setting the desired position, the PID controller generates a PWM control signal to drive the motor to the specified position. The PID controller retrieves the motor's current position as a voltage reading from the output potentiometer, enabling comparison with the expected direction of motor rotation influenced by the PID algorithm's control signal.

5.1 Control of robotic arm using Arduino

To enhance our comprehension of the system and its response, we can establish communication between MATLAB and ARDUINO via the USB port. MATLAB will employ inverse kinematics to calculate the joint angles based on the input values provided in the Graphical User Interface (GUI). Subsequently, these angles will be transmitted to ARDUINO, which will instruct the hardware to move correspondingly. Initially, to manage the robotic arm, ARDUINO will be utilized before implementing the PID controller. The diagram of the robotic arm's constructed model is illustrated in Fig.8.

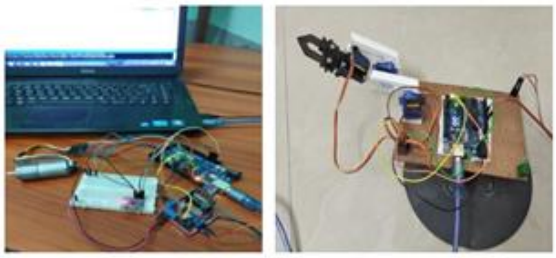


Fig. 8 Hardware setup (Experimental Setup) of control of the single-arm robot using Arduino Uno

5.2 Simulation Results

A simulation involving a three-degree-of-freedom (3-DOF) robotic manipulator was conducted, utilizing the parameters outlined in Table 3. The selection of PID controller parameters was accomplished through a trial-and-error approach, with different high gain values considered. During these simulations, the reference velocity for the robot's tip was determined based on its movement.

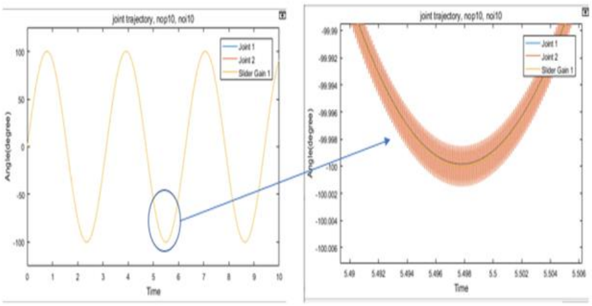
Notably, when high gains of 3 and 4 were employed, the robot's tip exhibited oscillations for a prolonged period of 30 seconds. In contrast, when a high gain value of 1 was used, the robot's tip initially oscillated for 5 seconds but subsequently reduced its oscillations, eventually closely aligning with the desired trajectory. The results of these simulations, with PID parameters fine-tuned using the Genetic Algorithm (GA) in MATLAB, are depicted in

Fig.9, showcasing the joint trajectory of the robotic manipulator.

The simulation involved a three-degree-of-freedom (3-DOF) robotic manipulator, utilizing parameters from Table 4. PID controller parameters were determined through various methods for different high gain values (μH) such as Ziegler-Nichols (Z-N), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). It was observed that among these methods, the GA approach produced the best results, particularly at μ_H=1. Other methods and gain values did not perform as well as the GA method at μ_H=1.

The experimental results, illustrated in Fig.10, depict the trajectory followed by the robot manipulator's tip. Initially, a manipulator trajectory plot was generated using a random number generator. Subsequently, the PID parameters were implemented on an Arduino, and the program was executed. Minor errors were observed, with the PID controller demonstrating a shorter rise time compared to other controllers. However, during the transient phase, it exhibited oscillatory behavior and a relatively large steady-state error. These oscillations resulted in a high overshoot of 9.1%, affecting system performance.

One could argue that the PID controller is suitable for precisely controlling the movement of the robotic arm manipulator model. Nonetheless, the results suggest that the GA technique is a superior choice when implementing a PID controller for this application.



NOP-10, NOI-10

Fig.9 Joint response with PID controller simulation

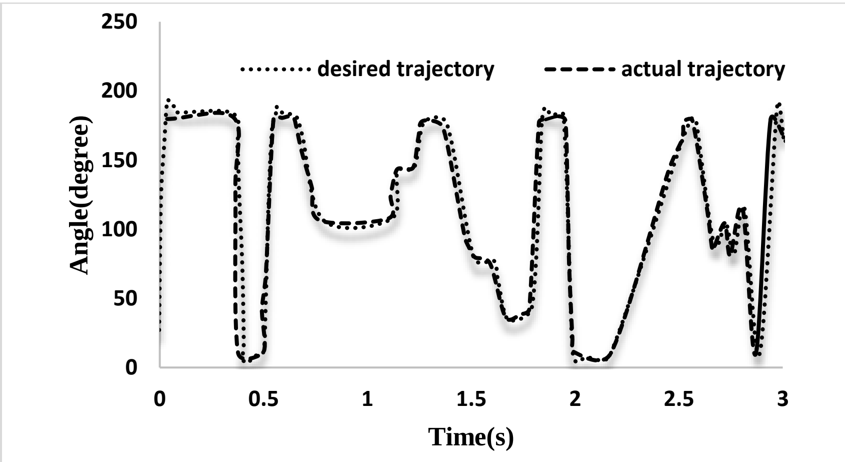


Fig. 10 Experimental end effector movement results with PID controller

6. Conclusion:

In this study, we have presented an advanced control design for achieving precise and smooth control of an arm robot manipulator model. By employing the Root-Locus (R-H) criterion, we have demonstrated the potential for enhanced control performance, robustness, and overall system stability. The application of the R-H criterion has yielded benefits such as quicker response times, reduced overshoot, and improved controller stability. Analyzing the temporal response parameters, it is evident that the PID controller, while exhibiting a shorter rise time compared to other controllers, experiences significant oscillations during the transient phase, leading to a substantial steady-state error. This behavior is accompanied by a notable 9.1 percent overshoot, which adversely impacts the overall system performance. Nevertheless, it can be argued that the PID controller is suitable for achieving precise control of the arm robot manipulator model. The study results strongly advocate for the Genetic Algorithm (GA) technique as the preferred method for implementing the PID controller in this context. he accomplishment of the robotic arm project, which entailed the incorporation of a PID controller based on Arduino Uno, host software, and DC motors controlled by a potentiometer, underscores the necessity for the careful design, implementation, and programming of both hardware and software components in the construction of robotic systems.

Acknowledgment

The Ministry of Education (MOE) and the Indian institution Sant Longowal Institute of Engineering and Technology (SLIET) are acknowledged by the authors for their kind financial support in the form of institute grants and Senior Research Fellows (SRF). The research project's viability and success were greatly enhanced by this support.

References

[1] Zhang, J., Zhuang, J., & Du, H. (2009). Self-organizing genetic algorithm based tuning of PID controllers. Information Sciences, 179(7), 1007-1018. International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-1, Issue-1, June 2012.

[2] Bindu, R., & Namboothiripad, M. K. (2012). Tuning of PID controller for DC servo motor using genetic algorithm. International Journal of Emerging Technology and Advanced Engineering, 2(3), 310-314.R. S. Burns, “Advances Control Engineering”, Jordan Hill, 2001.

[3] C. L. Philips and R. D. Harbor,” Feedback Control Systems”, PrPrentice-Hall2000.

[4] Muhammad Ahmad Baballe, Mukhtar Ibrahim Bello, Abubakar Abdullahi Umar, Abdullahi Kabiru Shehu, Dahiru Bello, & Faiz Tijjani Abdullahi. (2022). A Look at the Different Types of Servo Motors and Their Applications. https://doi.org/10.5281/zenodo.6529859

[5] T. V. Ramana, S. S. Manaktala, K. Valarmathi, Nitika Vats Doohan, Dasharathraj K. Shetty, Harish Kumar, Reynah Akwafo, "Energy Auditing in Three-Phase Brushless DC Motor Drive Output for Electrical Vehicle Communication Using Machine Learning Technique", Wireless Communications and Mobile Computing, vol. 2022, Article ID 9644795, 14 pages, 2022. https://doi.org/10.1155/2022/9644795

[6] S. N. Sivanandam, S. Sumathi, and S. N. Deepa,” Introduction to Fuzzy Logic using MATLAB”, Springer, 2007.

[7] Golcuk A. Design and implementation of a hybrid FLC+ PID controller for pressure control of sleep devices. Biomedical Signal Processing and Control. 2022 Jul 1;76:103702. https://doi.org/10.1016/j.bspc.2022.103702

[8] S.R.Vaishnav, Z.J.Khan, “Design and Performance of PID and Fuzzy Logic Controller with Smaller Rule Set for Higher-Order System,” Proceedings of the World Congress on Engineering and Computer Science, WCECS 2007, October 24-26, 2007, San Francisco, USA.

[9] Tudić V, Kralj D, Hoster J, Tropčić T. Design and Implementation of a Ball-Plate Control System and Python Script for Educational Purposes in STEM Technologies. Sensors (Basel). 2022;22(5):1875. Published 2022 Feb 27. doi:10.3390/s22051875

[10] Shih-Jer, H. and L. Ji-Shin ."A stable Self-Organizing Fuzzy Controller for Robotic Motion Control", Industrial Electronics, IEEE Transactions on 47(2), PP421-428, 2000.

[11] Kiam Heong, A., "PID Control System Analysis, Design, and Technology", Control Systems Technology, IEEE Transactions on 13(4), PP559-576, 2005.

[12] Kumar, S., Rastogi, V., & Gupta, P. (2018)The.The PID-based impedance control scheme for a flexible single-arm underwater robot manipulator. In International Conference on New Frontiers in Engineering, Science & Technology (pp. 481-488).

[13] Mohammed Hussien, "Design Controller System for Robotic Arm", Vo1. 6, No. 4, pp449–463, Nov 2016.

[14] I. China, N. Liouane, and P. Borne, "Tuning PID Controller Using Multiobjective Ant Colony Optimization," Appl. Comput. Intell. Soft Comput., vol. 2012, no. 1, pp. 1–7, 2012, DOI: 10.1155/2012/536326.

[15] K. Kinoshita, S. Wakitani, and S. Ohno, "Design of neural network PID controller based on E-FRIT," Electr. Eng. Japan (English Transl. Denki Gakkai Ronbunshi), vol. 205, no. 2, pp. 33–42, 2018, DOI: 10.1002/eej.23141.

[16] L. Liu, C. Pu, R. Barga, and T. Zhou, "Differential evaluation of continual queries," Proc. - Int. Conf. Distrib. Comput. Syst., pp. 458–465, 1996, DOI: 10.1109/icdcs.1996.507994.

[17] V. Rajinikanth and K. Latha, "I-PD Controller Tuning for Unstable System Using Bacterial Foraging Algorithm: A Study Based on Various Error Criterion," Appl. Comput. Intell. Soft Comput., vol. 2012, pp. 1–10, 2012, DOI: 10.1155/2012/329389.

[18] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," Inf. Sci. (NY)., vol. 179, no. 13, pp. 2232–2248, 2009, DOI: 10.1016/j.ins.2009.03.004.

[19] R. E. Haber, R. Haber-Haber, R. M. Del Toro, and J. R. Alique, "Using simulated annealing for optimal tuning of a PID controller for time-delay systems. An application to a high-performance drilling process," Lect. Notes Comput. Sci. (including Subsea. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4507 LNCS, pp. 1155–1162, 2007, DOI: 10.1007/978-3-540-73007-1_140.

[20] Y. Peng, X. Luo, and W. Wei, "A new control method based on artificial immune adaptive strategy," Elektron. ir Elektrotechnika, vol. 19, no. 4, pp. 3–8, 2013, doi: 10.5755/j01.eee.19.4.1246.

[21] P. W. Tsai, J. S. Pan, B. Y. Liao, and S. C. Chu, "Enhanced artificial bee colony optimization," Int. J. Innov. Comput. Inf. Control, vol. 5, no. 12, pp. 5081–5092, 2009, DOI: 10.1002/elps.200900194.

[22] J. B. Odili, M. N. M. Kahar, and A. Noraziah, "Parameters-Tuning of PID controller for automatic voltage regulators using the African buffalo optimization," PLoS One, vol. 12, no. 4, pp. 1–17, 2017, DOI: 10.1371/journal.pone.0175901.

[23] Kumar, S., Rastogi, V., & Gupta, P. (2014). Trajectory Control of Single Arm Underwater Flexible Welding Robot using Bond Graphs. In The Proceedings of the 2014 International Conference on bond graph modeling and simulation (ICBGM'2014), Simulation Series (Vol. 46, No. 8, pp. 79-84).

[24] Srinivasan, S., Latchman, H., Shea, J., Wong, T., & McNair, J. 2004, October. Airborne Traffic Surveillance Systems: Video Surveillance of Highway Traffic. In Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks. 131-135. DOI: http://dx.doi.org/10.11113/jt.v79.9987

[25] Sandeep Yadav, Sunil Kumar, Manoj Goyal "PID Tuning and Stability Analysis of Hybrid Controller for Robotic Arm Using ZN, PSO, ACO, and GA" (int. rev. of mechanical engineering),vol .16, No.5 May 2022, DOI: 10.15866/ireme.v16i5.21982

[26] Kumar, S., Rastogi, V., & Gupta, P. PID based impedance control scheme for flexible single arm underwater robot manipulator. In International Conference on New Frontiers in Engineering, Science & Technology [2018] (pp. 481-488).

[27] Jangid, Mahendra Kumar, Sunil Kumar, and Jagtar Singh. "Trajectory tracking optimization and control of a three three-links manipulator for application in casting." International Journal of Advanced Technology and Engineering Exploration 8, no. 83 (2021): 1255. http://dx.doi.org/10.19101/IJATEE.2021.874468