FROM DESIGN TO CONTROL: DEVELOPMENT AND EXPERIMENTAL VALIDATION OF A MEDICAL ROBOTICS ARM

Zou Han

Department of Surgical & Interventional Engineering
School of Biomedical Engineering & Imaging Sciences
Faculty of Life Sciences & Medicine
King's College London, London, UK
K24012533@kcl.ac.uk

Abstract. This study aims to develop a two-degree-of-freedom (2-DOF) medical robotic arm system through CAD modeling and 3D printing for structural design, integrating Arduino microcontrollers, DC motors, sensors, and H-Bridge motor driver modules for hardware implementation. The system employs a PID control algorithm for parameter tuning, combined with forward/inverse kinematics and the Jacobian matrix for trajectory planning. Experimental validation includes target point tracking and multi-point path tracking, utilizing real-time MATLAB-Arduino communication for control and data visualization. The results demonstrate that the system exhibits high stability and precise path-tracking capabilities under PID control, effectively reducing tracking errors while showing strong responsiveness across varying control parameters. Future work will explore adaptive control algorithms to optimize PID parameters, further enhancing real-time responsiveness and tracking accuracy in complex scenarios. Overall, this study successfully achieves an end-to-end workflow from design to control, validating the system's feasibility and effectiveness in medical task scenarios, laying a technical foundation for future optimization and practical clinical applications.

1. INTRODUCTION

With the rapid advancement of medical robotics, robotic systems have become essential tools in modern healthcare, enhancing precision and improving patient outcomes. As highlighted in studies [1], [2], these systems offer unprecedented precision, stability, and reliability in minimally invasive surgery and medical diagnostics. In this context, two-degree-of-freedom (2-DOF) robotic systems have emerged as key enablers in tasks requiring fine control and high-precision positioning, such as needle insertion [3], tissue manipulation, and surgical tool alignment [4]. However, maintaining stability and precision in such operations poses a significant challenge. This essay examines the effectiveness of PID (Proportional-Integral-Derivative) control strategies in enhancing the performance of 2-DOF robotic arms. By analyzing their strengths, addressing counterarguments, and exploring potential future developments, we aim to highlight their importance in surgical applications.

1.1.1 Question

How can PID control strategies enhance the stability and precision of 2-DOF medical robotic arms in surgical applications?

1.1.2 Arguments

The simplicity and versatility of PID control makes it one of the most widely used algorithms in robotics due to its simplicity and reliability. The proportional term provides immediate corrective action, the integral term minimizes steady-state error, and the derivative term predicts error trends to prevent overshoot. These characteristics make PID ideal for time-sensitive surgical tasks, where precision is critical [5]. Studies have shown its effectiveness in robotic arms for accurate path tracking and error compensation [3]. For instance, in trajectory planning, PID enables smooth motion, reducing jerky movements that could damage delicate tissues.

In surgical environments, real-time responsiveness is essential. PID algorithms, when properly tuned, provide dynamic corrections to changing conditions, such as sudden variations in load or unexpected resistance during surgery. The ability to adapt in real time ensures that the robotic arm maintains its target position with minimal error, which is crucial for operations requiring sub-millimeter precision [6]. While traditional PID relies on fixed parameters, integrating adaptive algorithms (e.g., fuzzy logic or neural networks) can extend PID's capabilities [7], allowing it to adjust to dynamic conditions seamlessly.

The cost-effectiveness and accessibility of PID control set it apart from more computationally intensive algorithms like Model Predictive Control (MPC). Its resource-efficient nature makes it ideal for systems based on Arduino or other embedded platforms with limited processing power. Additionally, its simplicity reduces implementation and maintenance costs, making PID a practical choice for budget-constrained surgical systems[8].

1.1.3 Counterarguments

The lack of predictive capability in PID control is a significant limitation compared to advanced algorithms like

MPC. While PID excels in real-time control, it cannot anticipate future states or calculate optimal inputs, which ensures smoother transitions and fewer oscillations. However, MPC's computational demands often limit its application in real-time surgical systems [8,9,10]. This highlights a trade-off between real-time feasibility and predictive accuracy.

The challenges of fixed PID parameters in dynamic environments often lead to performance degradation under rapidly changing conditions [11]. For example, sudden variations in load during robotic surgery can cause overshoot or instability, requiring frequent manual re-tuning [12]. While adaptive or self-tuning PID controllers show promise, they add complexity and require further development to meet real-time constraints [13,14].

The limitations of alternative algorithms like Sliding Mode Control (SMC) further complicate the landscape of robotic control. Although SMC provides robustness against external disturbances, it suffers from the chattering phenomenon, which can destabilize delicate surgical tasks [15]. Similarly, reinforcement learning-based controllers offer potential but are computationally expensive and lack interpretability [16], making them challenging to adopt in high-stakes medical environments.

1.1.4 Conclusion

PID control strategies provide an effective and balanced solution for enhancing the stability and precision of 2-DOF robotic arms in surgical applications. Despite its limitations, such as fixed parameter reliance and lack of predictive capability, PID remains a cornerstone due to its simplicity, cost-effectiveness, and adaptability. Future advancements, including adaptive algorithms and AI integration, hold the potential to overcome current challenges, paving the way for more accessible and reliable surgical robotic systems.

1.2 AIMS

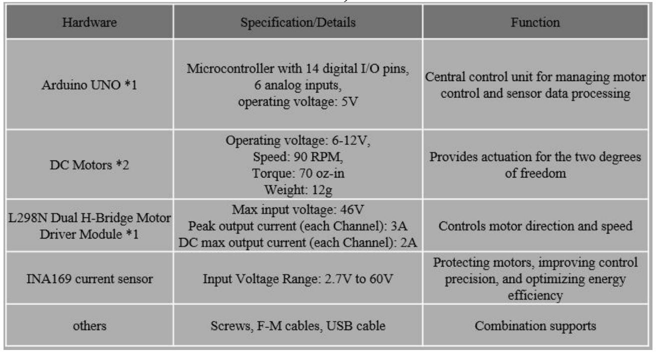
This study aims to improve the structural design of a two-degree-of-freedom (2-DOF) medical robotic arm system, optimize the feedback control system, and refine the PID control algorithm to enhance precision and stability in surgical applications. This involves designing and fabricating the robotic arm using CAD modeling and 3D printing to ensure structural stability and lightweight properties. The study further focuses on integrating Arduino microcontrollers, DC motors, sensors, and H-Bridge motor drivers for hardware implementation. To achieve precise motion control, the PID control algorithm will be optimized for dynamic corrections, reduced steady-state errors, and smooth trajectory tracking. The system's performance will be validated through experimental evaluations, including target point tracking and multi-point path tracking. Additionally, the research aims to explore adaptive control strategies for future enhancement, ensuring greater responsiveness and accuracy in complex surgical scenarios.

2. METHODOLOGY

2.1 System Design

The CAD design of the two-degree-of-freedom (2-DOF) medical robotic arm system is carefully structured to balance space utilization, structural stability, and ease of maintenance.

The hardware components, including the Arduino microcontroller, DC motors, sensors, and H-Bridge motor driver modules, are carefully selected and integrated to support the system's structural and functional design (refer to the hardware list in Table 2.1).



Hardware	Specification/Details	Function
Arduino UNO *1	Microcontroller with 14 digital I/O pins, 6 analog inputs, operating voltage: 5V	Central control unit for managing motor control and sensor data processing
DC Motors *2	Operating voltage: 6-12V, Speed: 90 RPM, Torque: 70 oz-in Weight: 12g	Provides actuation for the two degrees of freedom
L298N Dual H-Bridge Motor Driver Module *1	Max input voltage: 46V Peak output current (each Channel): 3A DC max output current (each Channel): 2A	Controls motor direction and speed
INA169 current sensor	Input Voltage Range: 2.7V to 60V	Protecting motors, improving control precision, and optimizing energy efficiency
others	Screws, F-M cables, USB cable	Combination supports

TABLE 2.1 Hardware List

2.1.1 CAD And Hardware Integration

The base (see Figure 2.1) incorporates a modular slot design for efficient assembly and debugging. It features a dual-layer layout: the first layer houses the Arduino UNO, while the second layer accommodates the dual H-Bridge motor driver module, optimizing space and ensuring component organization. A 5V power supply is directly soldered onto a compact chip to save space and enhance power stability, eliminating the need for a breadboard (shown in the *Results* section). The Arduino serves as the primary control unit, processing sensor data and controlling motor direction and speed through the H-Bridge motor driver. All components are connected via soldering and modular interfaces, ensuring reliable signal transmission and compact integration, forming a robust foundation for system stability.

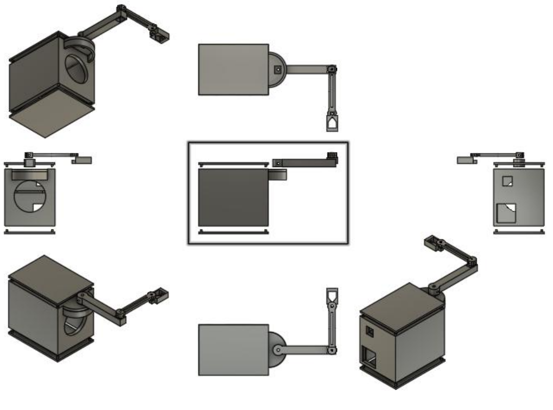


Figure 2.1: The multi-view representation of the final design

The robotic arm consists of Arm1 and Arm2, achieving a balance between stability and lightweight design. Arm1 demonstrates structural stability (see Figure 2.2(a)), while Arm2 features a lightweight structure with precise axis alignment (see Figure 2.2(b)). During the design process, careful consideration was given to the outer diameter and overlapping length of the joints, resulting in effective lengths of approximately 96mm for Arm1 and 60mm for Arm2, enabling the robotic arm to reach a maximum working range of 156mm, meeting the requirements of most precision control tasks[17].

Arm1 adopts a thicker structure to provide enhanced load-bearing capacity, while Arm2 utilizes a flat, lightweight design to reduce endpoint load pressure and minimize the risk of structural deformation. This dual-arm design ensures precision and flexibility in practical applications, providing a reliable foundation for medical tasks.

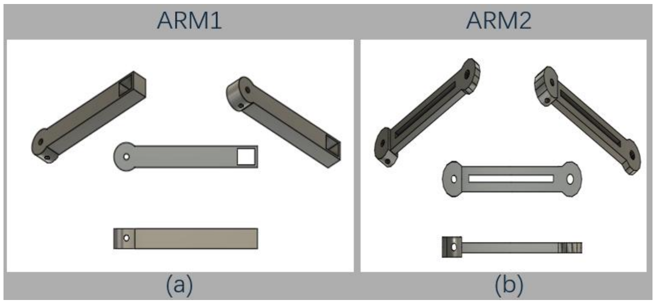


Figure 2.2: CAD Design of Robotic Arm Components

The penholder follows a lightweight design principle (see Figure 2.3), which reduces stress on Arm2, further minimizing the risk of structural deformation. The axis alignment of the penholder coincides with Arm2's rotational axis, effectively reducing positional offset errors during testing and improving precision.

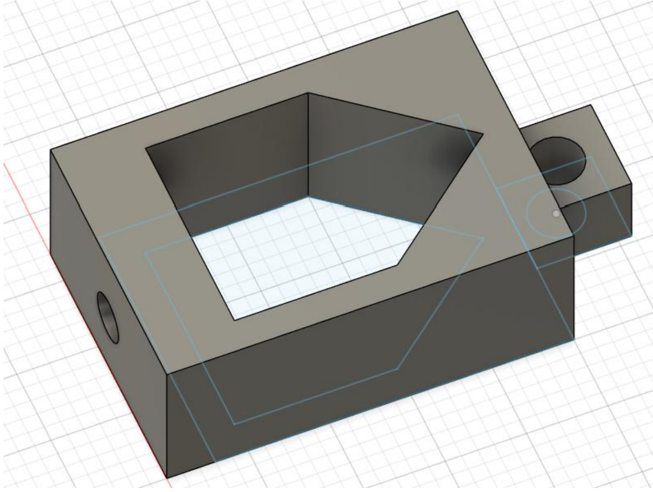
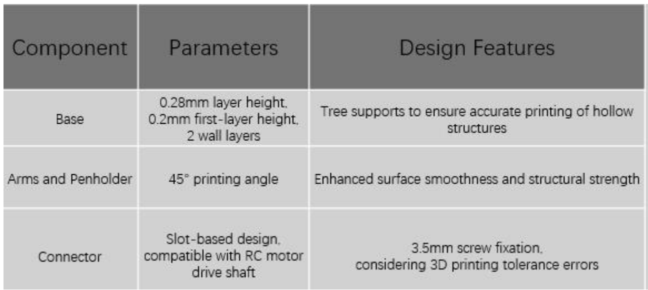


Figure 2.3: Penholder Design of Robotic Arm Components (all dimensions in mm)

The detailed printing specifications, including layer height, print angle, and structural support strategies, are comprehensively presented in Table 2.2 for clarity and reference. The average percentage of infill is 15%. And the material is PLA.



Component	Parameters	Design Features
Base	0.28mm layer height, 0.2mm first-layer height, 2 wall layers	Tree supports to ensure accurate printing of hollow structures
Arms and Penholder	45° printing angle	Enhanced surface smoothness and structural strength
Connector	Slot-based design, compatible with RC motor drive shaft	3.5mm screw fixation, considering 3D printing tolerance errors

TABLE 2.2: 3D Printing Parameters and Connector Design (all dimensions in mm)

All joints and connectors utilize a slot-based design that perfectly fits the RC motor’s drive shaft. A 3.5mm screw securely fastens one side of the D-shaft, while the motor body is housed in a square slot measuring 12.03mm × 10.03mm. This precise slot dimension accounts for 3D printing tolerances, ensuring a snug fit for the DC motor. The end-effector (penholder) is secured using 3.5mm screws and nuts, enhancing overall reliability and facilitating easy maintenance and scalability for potential mass production.

Overall, this CAD design achieves a balance between functionality, stability, and manufacturability, providing a robust foundation for the robotic arm's hardware integration and control systems.

2.2 Low-level Motor Control

2.2.1 PID Control Algorithm

The PID control algorithm combines three components—Proportional (P), Integral (I), and Derivative (D)—to regulate the dynamic response of the robotic arm.

The PID controller can be mathematically described using the following transfer function:

$$G_c \quad (s) = K_p \quad + \frac{K_i}{s} \qquad + K_d \quad s$$

To regulate the system's behavior, the error e(t), defined as the difference between the reference r(t) and the output y(t), is minimized:

$$e(t) = r(t) - y(t)$$

The output, u(t), is mathematically represented as:

$$\text{u(t)}=K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

where:

- e(t): Error signal (difference between the target and actual position).
- K_p, K_i, K_d : Gains for proportional, integral, and derivative components.

This formulation ensures that the robotic arm can dynamically adjust its response to achieve the desired joint angles with high precision.

2.2.2 Parameter Tuning Methodology

To achieve optimal control parameters (K_p, K_i, K_d) for the PID controller of the 2-DOF robotic arm system, a systematic methodology was adopted. The tuning process combined theoretical analysis from motor control principles with practical observations to ensure precise motion control and minimal error.

The process began with initial parameter settings based on theoretical analysis. The proportional gain K_p was initially set at a high value to provide strong proportional correction and was gradually decreased during dynamic response testing to reduce overshoot Max_V and oscillations. The integral gain K_i was kept constant at 0 throughout the process, as practical tests revealed that any non-zero K_i caused significant oscillatory behavior, destabilizing the system. The derivative gain K_d was initially set to a low value and adjusted iteratively after K_p was stabilized. Adjusting K_d effectively suppressed oscillations and improved dynamic stability. These initial settings provided a stable foundation for the iterative tuning process.

During each iteration, the robotic arm was subjected to step input signals to test the system's response under various K_p, K_i and K_d combinations. Key performance metrics were recorded, including overshoot Max_V, undershoot (Min_V), steady-state error (Error_V)—the difference between the target value and the system's actual response—and the final stabilized system response (Final_V). By systematically reducing K_p , the amplitude of oscillations was significantly minimized while maintaining an acceptable response speed. Based on the optimized K_p , K_d was further adjusted to smooth the response curve and suppress unnecessary oscillations. The final parameter configuration ensured a balanced trade-off between precision and stability.

The evaluation process relied on key metrics (the result is shown in the Results section): Max_V reflected the peak system response, where lower values indicated better stability; Error_V measured the difference between the target and actual response, representing control accuracy; and convergence time represented the time taken for the system to reach a steady state with minimal oscillations. By analyzing and optimizing these metrics, the final selected K_p, K_i and K_d values successfully achieved rapid response, minimal error, and high stability.

2.3 Forward and Inverse Kinematics

For a 2-DOF robotic arm, the forward kinematics can be expressed as the product of homogeneous transformation matrices. The general transformation matrix for a link i is:

$$A_{i-1i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & r_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & 0 & r_1 \cdot \sin(\theta_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:
θ_i : $Joint\ angle\ for\ link$ i.
r_i : Length of link i.

The total transformation matrix $T_{0,2}$ from the base frame to the end-effector frame is:

$$\mathrm{T} = A_{01} \cdot A_{12} = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & P_x \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

$$P_x = r_1 \cos(\theta 1) + r_2 \cos(\theta 1 + \theta 2)$$
$$P_y = r_1 \sin(\theta 1) + r_2 \sin(\theta 1 + \theta 2)$$

This matrix captures both the rotation and translation components of the end-effector's position and orientation relative to the base.

Inverse kinematics requires solving for θ_1 and θ_2 given the desired P_x and P_y . Using the Pythagorean theorem and trigonometric relationships (The diagram is presented in Figure 2.4):

$$\theta_1 = \tan^{-1}\left(\frac{p_y^0}{p_x^0}\right) - \tan^{-1}\left(\frac{r_2 \sin\theta_2}{r_1 + r_2\cos\theta_2}\right)$$

$$\theta_2 = cos^{-1}\left(\frac{p_x^{0^2} + p_y^{0^2} - r_1^2 - r_2^2}{2r_1r_2}\right)$$

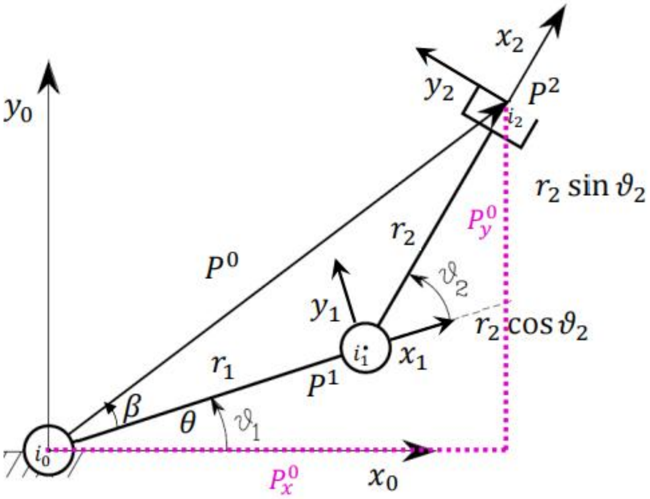


Figure 2.4: Schematic diagram

2.4 Workspace

The approach to determine the workspace involves systematically varying the joint angles within predefined ranges, calculating the reachable points using forward kinematics, and plotting these points to visualize the operational area. By iterating through all possible angle combinations, the method ensures a complete representation of the workspace, covering an area of 156mm x 156mm (see Figure 2.5) and accurately documenting the robotic arm's functional range. For arm1, it reaches 0-180 degrees while arm2 0-350 degrees.

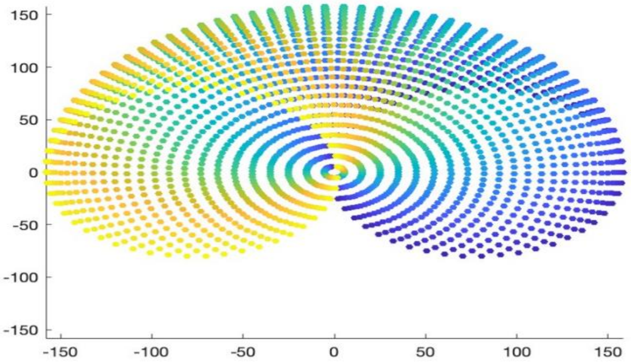


Figure 2.5: Workspace (Dimension: mm)

2.5 Differential Inverse Kinematics Methods

Differential inverse kinematics methods, including inverse Jacobian, damped least squares, and Jacobian transpose, map end-effector velocities to joint velocities for precise control.

2.5.1 Inverse Jacobian

Computes joint velocities as $\dot{q} = J^{-1}\dot{x}_e$ requiring J to be square and non-singular. Singularities can cause instability.

2.5.2 Damped Least Squares

The Damped Least Squares (DLS) method enhances stability near singularities by modifying the Jacobian pseudo-inverse:

$$A^{+} = A^{T}(AA^{T} + k^{2}I)^{-1}$$

Here, k^2 is the damping factor ensuring well-defined solutions even when the Jacobian A loses rank.

2.5.3 Jacobian transpose

Offers computational simplicity with $\dot{q} = J^{T}f$, avoiding direct inversion of J. And f controls input force vector.

2.5.4 Manipulability

Manipulability measures dexterity and is quantified as:

$$\mu = \det\left((JJ^{T}\right)$$

Lower values indicate reduced control near singularities.

2.6 High-Level Control

i: Target point input. The desired position is provided via a graphical user interface (GUI) or predefined coordinates. Then the system calculates the required joint angles using inverse kinematics to achieve the desired end-effector position in the workspace.

ii: Trajectory Planning: Smooth motion paths are generated to move from the current to the target joint angles while avoiding sudden movements. Then Encoder data is used to monitor the current joint positions, which are compared with the desired trajectory to minimize error.

iii: Proportional-Integral-Derivative (PID) controllers adjust motor inputs to ensure precise angle adjustments. Gains are optimized to achieve high stability and minimal steady-state error. Then signals from the PID controller drive the DC motors through H-Bridge circuits to achieve the required joint angles.

iv: The robotic arm moves to the desired position based on the calculated joint angles. The GUI continuously updates the system status, including motor feedback, trajectory progress, and real-time error metrics.

2.7 GUI

The GUI is designed with four input fields (Figure 2.6): two for entering the X and Y coordinates, and two for specifying the target angles of Motor1 and Motor2. It also features functionalities such as calculating the current coordinates, setting the origin, and returning to the origin. The current coordinate information during operation is displayed in the diagram.

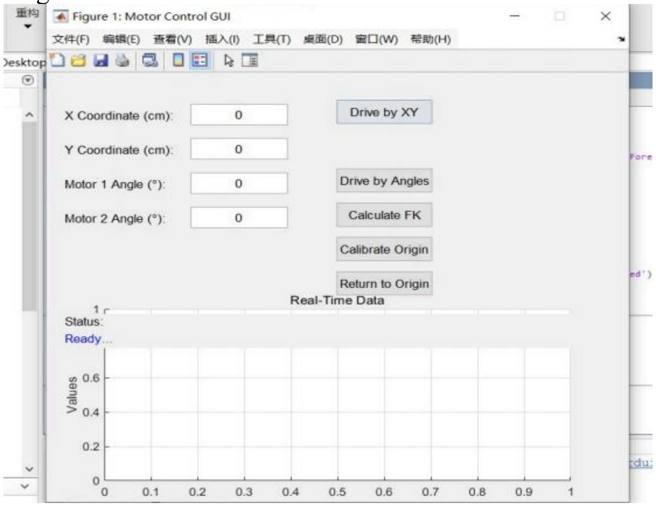


Figure 1: Motor Control GUI

X Coordinate (cm): 0

Y Coordinate (cm): 0

Motor 1 Angle (°): 0

Motor 2 Angle (°): 0

Status:
Ready...

Figure 2.6: GUI

2.8 Experimental Design

2.8.1 Experiment 1:

Calculate the current coordinates, determine the next coordinate within the range based on the current coordinates, and move to it.

Performance Metrics: Accuracy to target points, and Time Efficiency

2.8.2Experiment 2:

Assess the robot's capability to follow a predefined trajectory consisting of multiple target points.

Performance Metrics: Path accuracy, Smoothness and Completion Time.

2.9 Group GitHub repo

https://github.com/vivaanrobotics/roboticsgrp2.git

3. RESULTS

3.1 Finial Design of Robotic Arms

The complete final base is shown in Figure 3.1, with dimensions of 110mm × 80mm × 90mm. The multi-view diagram, as shown in Figure 3.2, provides a comprehensive visualization of the robotic system's design from different perspectives, highlighting key structural features and component placements. The dual-layer base design of the robot effectively balances space utilization and structural stability, successfully accommodating all hardware components (see Figure 3.3). The arm's unique design, combined with a 45-degree printing strategy, results in a lightweight yet highly durable structure with a smooth and aesthetically pleasing surface. The adjustable penholder securely grips pens of various sizes, while the streamlined linkage strategy ensures a robust connection between the motors, arms, and base. Additionally, soldered wiring eliminates the need for a breadboard, significantly enhancing the base's spatial efficiency and reducing its overall weight (see Figure 3.4).

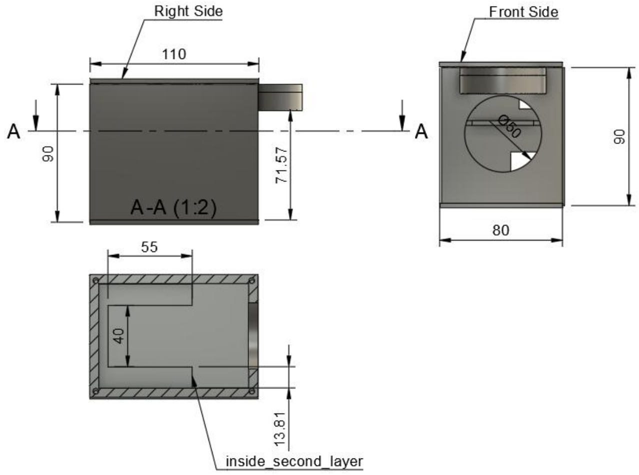


Figure 3.1: Final base design (Unit: mm)

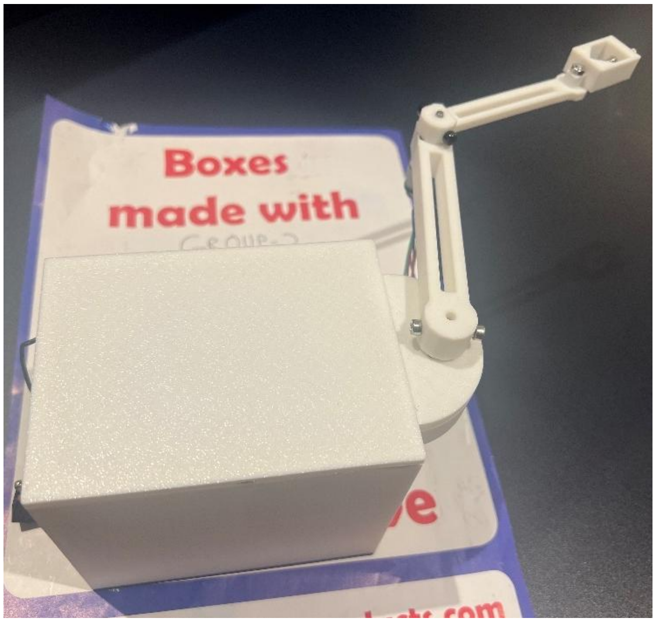


Figure 3.3: Fully assembled

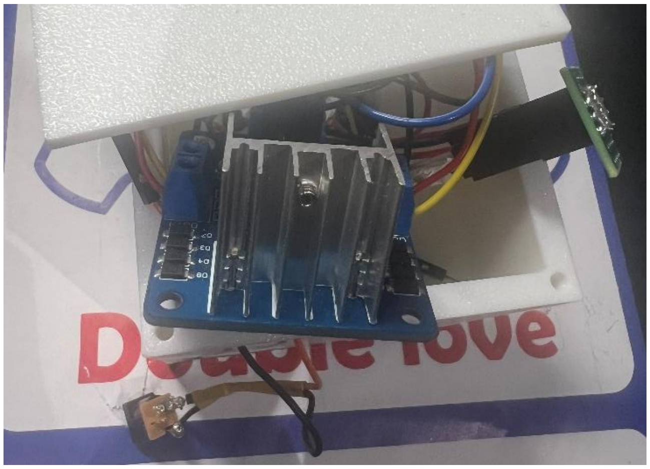


Figure 3.4: Soldered circuit

The detailed arms data are shown in Figure 3.5 below.

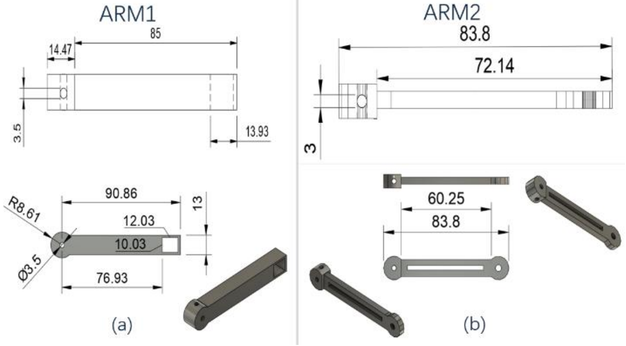


Figure 3.5: Final Robotic Arm Components (Unit: mm)

The detailed parameters of the penholder are shown in Figure 3.6.

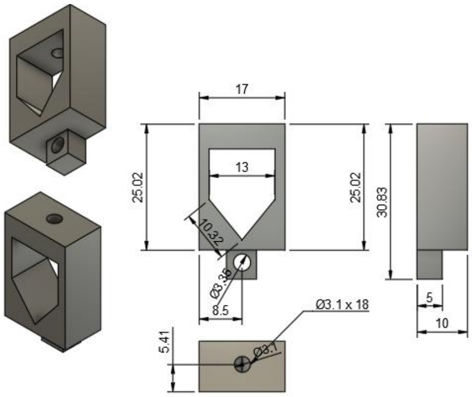


Figure 3.6: Penholder details (Unit: mm)

3.2 Hardware Integration

The first layer houses the Arduino Uno, and the back cover is removable (Figure 3.7), aligning with the original goal of facilitating debugging. The second layer accommodates the H-Bridge and sensors (Figure 3.8). The base successfully connects to Motor 1, with Motor 1's drive shaft linked to Arm 1, while Arm 2 is connected to both Arm 1 and Motor 2. The internal hardware and overall assembly have been successfully completed.

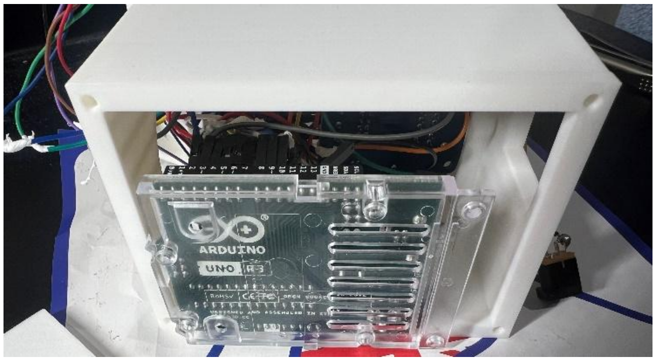


Figure 3.7: Bottom of the robot

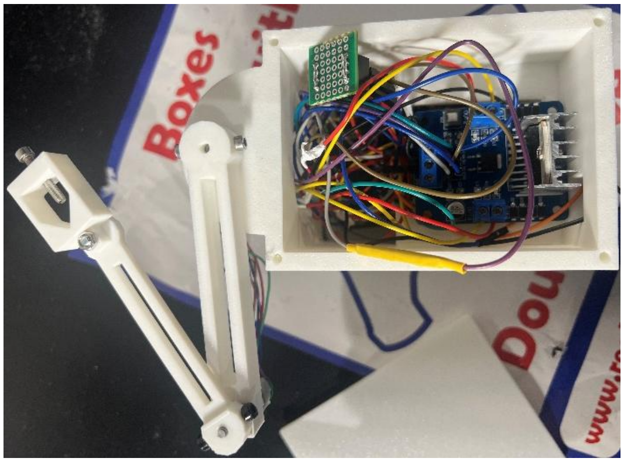


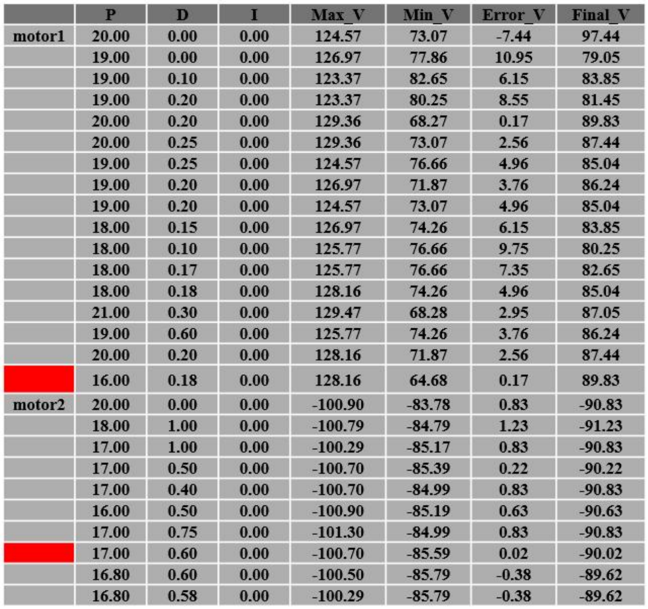
Figure 3.8: Top of the robot

3.3 Low-level Motor Control Results

The results of the low-level motor control demonstrate the effectiveness of the optimized PID parameters in achieving precise and stable joint movements. The following data (Table 3.1) highlights key performance metrics, including K_p, K_i, K_d , Error_V, Max_V, Min_V and Final_V, under various test condition**s.**

If the Final_V is closer to the target angle, the system is more accurate. Similarly, a smaller Error_V value indicates higher accuracy. For stability, it is crucial to achieve the smallest Max_V and the largest Min_V. This minimizes oscillations, resulting in a more stable system see Figure 3.2 (The smaller the value, the more stable it is).

So, we need to balance the stability and accuracy of the robot.



	P	D	I	Max_V	Min_V	Error_V	Final_V
motor1	20.00	0.00	0.00	124.57	73.07	-7.44	97.44
	19.00	0.00	0.00	126.97	77.86	10.95	79.05
	19.00	0.10	0.00	123.37	82.65	6.15	83.85
	19.00	0.20	0.00	123.37	80.25	8.55	81.45
	20.00	0.20	0.00	129.36	68.27	0.17	89.83
	20.00	0.25	0.00	129.36	73.07	2.56	87.44
	19.00	0.25	0.00	124.57	76.66	4.96	85.04
	19.00	0.20	0.00	126.97	71.87	3.76	86.24
	19.00	0.20	0.00	124.57	73.07	4.96	85.04
	18.00	0.15	0.00	126.97	74.26	6.15	83.85
	18.00	0.10	0.00	125.77	76.66	9.75	80.25
	18.00	0.17	0.00	125.77	76.66	7.35	82.65
	18.00	0.18	0.00	128.16	74.26	4.96	85.04
	21.00	0.30	0.00	129.47	68.28	2.95	87.05
	19.00	0.60	0.00	125.77	74.26	3.76	86.24
	20.00	0.20	0.00	128.16	71.87	2.56	87.44
🟥	16.00	0.18	0.00	128.16	64.68	0.17	89.83
motor2	20.00	0.00	0.00	-100.90	-83.78	0.83	-90.83
	18.00	1.00	0.00	-100.79	-84.79	1.23	-91.23
	17.00	1.00	0.00	-100.29	-85.17	0.83	-90.83
	17.00	0.50	0.00	-100.70	-85.39	0.22	-90.22
	17.00	0.40	0.00	-100.70	-84.99	0.83	-90.83
	16.00	0.50	0.00	-100.90	-85.19	0.63	-90.63
	17.00	0.75	0.00	-101.30	-84.99	0.83	-90.83
🟥	17.00	0.60	0.00	-100.70	-85.59	0.02	-90.02
	16.80	0.60	0.00	-100.50	-85.79	-0.38	-89.62
	16.80	0.58	0.00	-100.29	-85.79	-0.38	-89.62

TABLE 3.1: Gain Parameter Tuning Results for

PID Controller

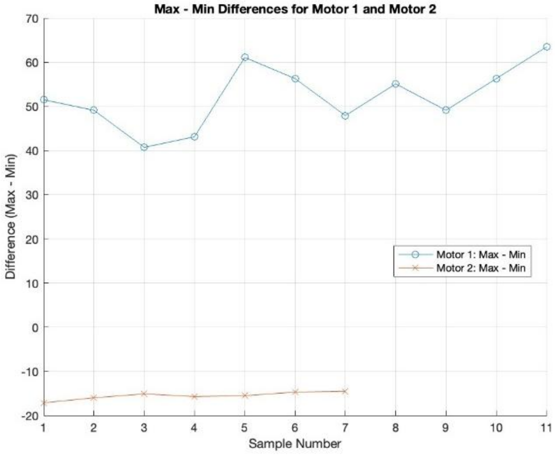


TABLE 3.2: Stability of robot

the optimal parameters were determined, and the response curve was plotted (see Figure 3.3.1 and Figure 3.3.2). After comparison, the most accurate final value with relatively small oscillation was selected. (according to the steps referred in 2.2.2)

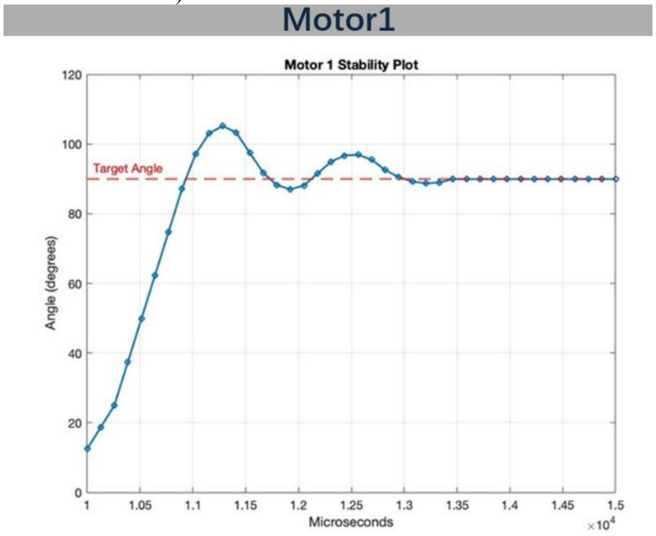


Figure 3.3.1: Time Response of Motor1

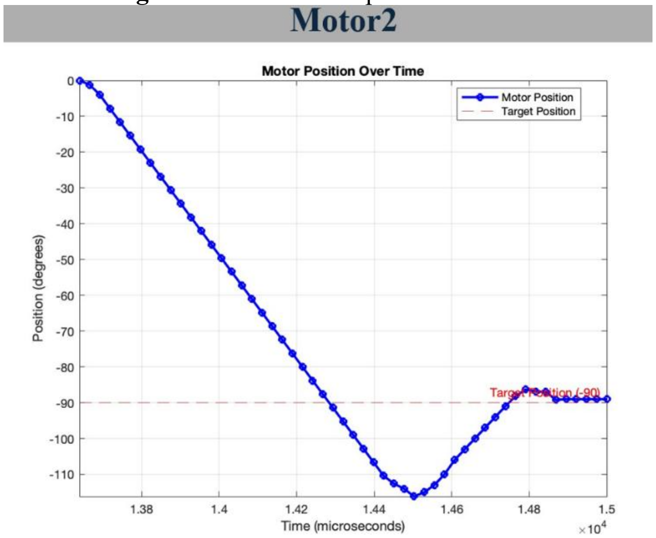


Figure 3.3.2: Time Response of Motor2

3.4 Results for Forward and Inverse Kinematic

The manipulability of the robotic arm along the trajectory from the initial position (0,0) to (1.5,1) was analyzed and visualized using MATLAB (Figure 3.4.1). This analysis employed the inverse Jacobian differential inverse kinematics approach, ensuring precise calculation of manipulability values across the path. By sampling multiple intermediate points, the trajectory was divided into discrete segments, and the Jacobian matrix was computed for each position.

The manipulability index, μ=$\sqrt{\det(JJ^T)}$, was calculated to quantify the robotic arm's dexterity and control efficiency at each point. Regions with higher μ indicate greater flexibility and movement capability, while lower values highlight areas near singular configurations where control might be limited. The analysis provides critical insights into the system's performance along the trajectory (Figure 3.4.2), helping to optimize motion planning and stability. Below are graphs depicting the manipulability variations along this trajectory, illustrating the robotic arm's adaptability in dynamic scenarios.

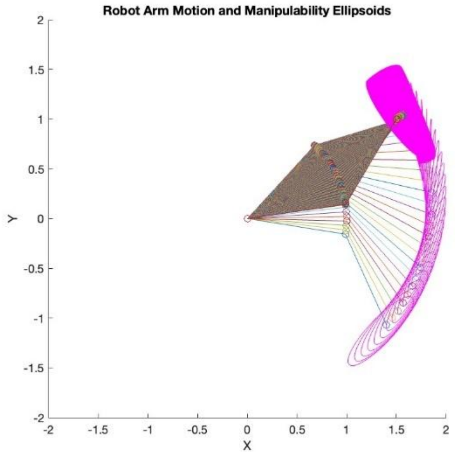


Figure 3.4.1: robotic arm motion and manipulability ellipsoids (Unit: cm)

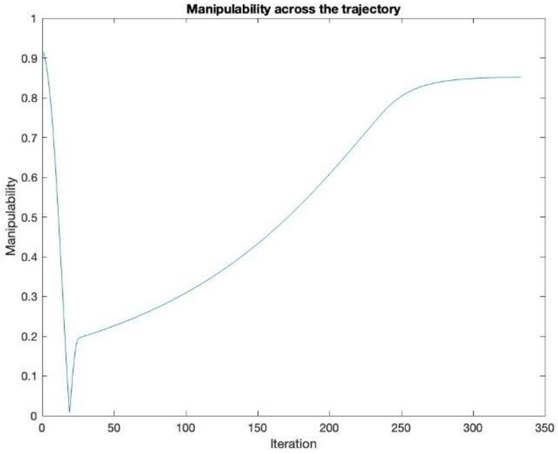


Figure 3.4.2: **M**anipulability across the trajectory

3.5 Experiment 1

Setting predicted coordination and running the robot arm to reach the point and record it (see Figure 3.5.1). To further analyze the system's performance, a normal distribution curve was generated to represent the percentage data obtained from experimental results. Average error in running Gui is 17.8% (see Figure 3.5.2).

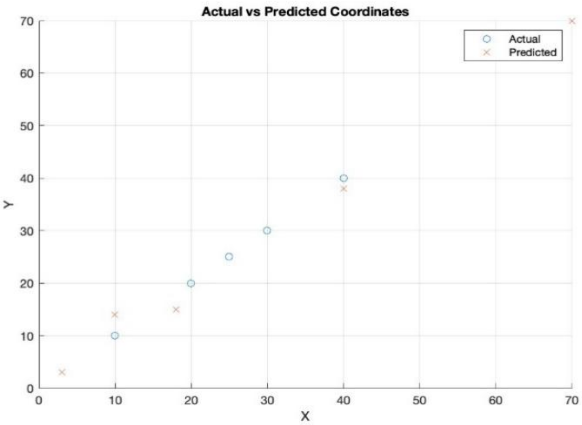


Figure 3.5.1: Plot of testing moving to points (5 times)

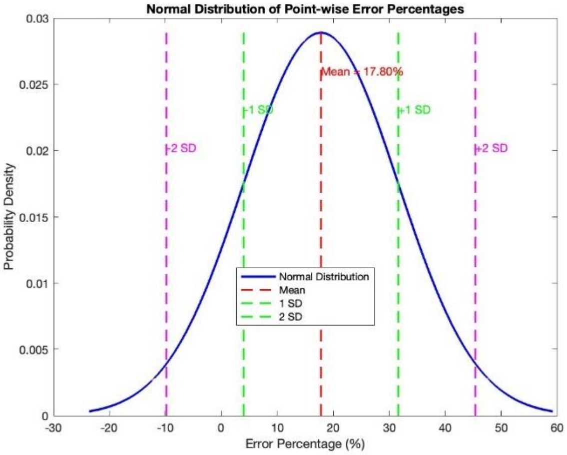


Figure 3.5.2: Plot of testing moving to points

4. DISCUSSION AND CONCLUSIONS

4.1 Critical Interpretation of The Results

The interplay between structural design and control strategies significantly impacted system performance [18]. The lightweight design of Arm 2 reduced endpoint load, enhancing control precision, while the modular base design optimized space utilization and facilitated maintenance. These design choices supported the stability achieved through PID control, demonstrating the interconnected nature of mechanical design and control algorithms.

The PID control results highlight the importance of optimal parameter tuning in achieving stable and precise robotic motion. As demonstrated in Section 3.3, the tuned parameters minimized overshoot Max_V and steady-state error (Error_V) while ensuring rapid convergence to target positions. These findings align with previous studies [5] which emphasize the effectiveness of PID controllers in time-sensitive applications requiring high precision.

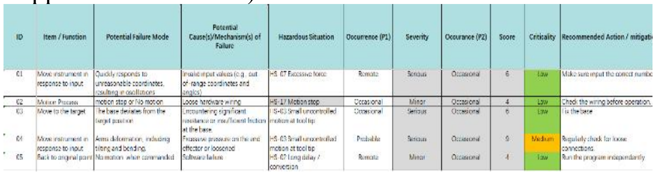
The trade-off between stability and accuracy is evident in the tuning process. While reducing K_p minimized oscillations, fine-tuning K_d further dampened system response, achieving the balance required for dynamic control. These results underline the adaptability of PID algorithms in addressing the demands of surgical robotics [6].

The workspace validation in Section 3.4 confirmed that the robotic arm effectively covered a 156 * 156 mm area, meeting design specifications. The manipulability analysis provided deeper insights into the robotic arm's dexterity along specific trajectories, such as the path from (0,0) to (1.5,1). High manipulability values in central regions indicate optimal performance, while reduced values near singular configurations highlight areas requiring careful control adjustments. This analysis aligns with research by Meir et al, which stresses the importance of dexterity in robotic tasks [17].

Despite the system's success in achieving stability, precision, and range, certain limitations were observed. Boundary performance issues caused slight deviations near workspace edges, mainly due to mechanical tolerances in joints and motor alignment, reducing accuracy in extreme positions. Additionally, the reliance on fixed PID parameters limited adaptability to dynamic conditions, as the controller struggled with sudden load changes or resistance variations. These limitations highlight the need for adaptive control strategies, as suggested by Chen et al. (2022), to dynamically adjust parameters in real-time [9]. Implementing advanced control techniques and refining the mechanical design could further improve the system's robustness and versatility.

4.2 Risk Assessment

The basic system-level failure mode effects analysis (FMEA) is shown below (Table 4.1, the clear one is shown in the Supplemental Material).



ID	Item / Function	Potential Failure Mode	Potential Cause(s)/Mechanism(s) of Failure	Hazardous Situation	Occurrence (P1)	Severity	Occurance (P2)	Score	Criticality	Recommended Action / mitigati
01	Move instrument in response to input	Quickly responds to unreasonable coordinates, resulting in oscillations	Invalid input values (e.g., out of-range coordinates and angles)	HS. 07 Excessive force	Remote	Serious	Occasional	6	Low	Make sure input the correct numbe
02	Motion Process	motion stop or No motion	Loose hardware wiring	HS-17 Motion stop	Occasional	Minor	Occasional	4	Low	Check the wiring before operation.
03	Move to the target	The base deviates from the target position	Encountering significant resistance or insufficient friction at the base.	HS-03 Small uncontrolled motion at tool tip	Occasional	Serious	Occasional	6	Low	Fix the base
04	Move instrument in response to input	Arm deformation, including tilting and bending.	Excessive pressure on the end effector or loosened	HS-03 Small uncontrolled motion at tool tip	Probable	Serious	Occasional	9	Medium	Regularly check for loose connections.
05	Back to original point	No motion when commanded	Software failure	HS-07 Long delay / conversion	Remote	Minor	Occasional	4	Low	Run the program independently

Table 4.1: Risk Assessment by FMEAS

4.3 Limitations

As mentioned in the above section, the system's reliance on fixed PID parameters limits its adaptability to dynamic conditions, leading to instability and overshooting under varying loads. And high-speed operations with low damping factors in the PID control settings result in oscillations, reducing the precision of trajectory tracking.

Exposed wires and fragile motors make the system less robust, increasing the risk of mechanical or electrical failure.

What’s more, the lightweight design, while suitable for prototyping, may not meet the robustness required for real-world medical applications.

4.4 Future work

According to risk assessment (section 4.2), to enhance the system's robustness, stability, and user experience, several improvements are proposed. Implementing input filtering in the GUI will help reject outliers and prevent invalid data from causing errors or unexpected behavior. Strengthening wire connections with tighter and more secure connectors will ensure reliable operation by preventing disconnections or detachment. Increasing the friction of the base will provide greater stability, minimizing unintended movements and ensuring a solid foundation for the robot. Additionally, optimizing the materials of the robotic arm and reinforcing its joints will enhance load-bearing capacity, reducing the risk of deformation, tilting, or structural instability. Finally, refining the GUI will improve user interaction, making the system more intuitive and efficient to operate. Implement adaptive control strategies, such as fuzzy logic [5] or neural network-based PID tuning, to enhance real-time responsiveness and accuracy.

These enhancements aim to deliver a more reliable, stable, and user-friendly robotic system.

4.5 Conclusion

This study successfully developed and validated a 2-DOF medical robotic arm system, achieving the objectives of structural stability, precision, and effective control. The integration of CAD modeling, 3D printing, and modular hardware components resulted in a lightweight yet robust design. The optimized PID control algorithm demonstrated precise motion control with minimal errors, as evidenced by target point tracking and trajectory-following experiments. The system’s manipulability analysis and workspace validation confirmed its suitability for precise surgical tasks within a 156 × 156 mm operational area.

While the system performed effectively under controlled conditions, limitations such as fixed PID parameters, reduced performance near workspace boundaries, and hardware fragility were identified. Future enhancements will focus on adaptive control integration, improved material choices, and refined GUI functionality to address these challenges. Overall, the project achieved its aims, laying a solid foundation for further optimization and potential clinical applications.

REFERENCES

[1] C. Walther, S. Fichtlscherer, T. Holubec, M. Vasa-Nicotera, M. Arsalan, and T. Walther, ‘New developments in transcatheter therapy of mitral valve disease’, *J Thorac Dis*, vol. 12, no. 4, pp. 1728–1739, Apr. 2020, doi: 10.21037/jtd.2019.12.137.

[2] P. Picozzi *et al.*, ‘Advances in Robotic Surgery: A Review of New Surgical Platforms’, *Electronics*, vol. 13, no. 23, p. 4675, Nov. 2024, doi: 10.3390/electronics13234675.

[3] C. Yang, Y. Xie, S. Liu, and D. Sun, ‘Force Modeling, Identification, and Feedback Control of Robot-Assisted Needle Insertion: A Survey of the Literature’, *Sensors*, vol. 18, no. 2, p. 561, Feb. 2018, doi: 10.3390/s18020561.

[4] M. G. Fujie and B. Zhang, ‘State-of-the-art of intelligent minimally invasive surgical robots’, *Front. Med.*, vol. 14, no. 4, pp. 404–416, Aug. 2020, doi: 10.1007/s11684-020-0743-3.

[5] S. Guo, Q. Chen, N. Xiao, and Y. Wang, ‘A fuzzy PID control algorithm for the interventional surgical robot with guide wire feedback force’, in *2016 IEEE International Conference on Mechatronics and Automation*, Harbin, Heilongjiang, China: IEEE, Aug. 2016, pp. 426–430. doi: 10.1109/ICMA.2016.7558601.

[6] H. M. Tuan, F. Sanfilippo, and N. V. Hao, ‘Modelling and Control of a 2-DOF Robot Arm with Elastic Joints for Safe Human-Robot Interaction’, *Front. Robot. AI*, vol. 8, p. 679304, Aug. 2021, doi: 10.3389/frobt.2021.679304.

[7] R. Bouzaiene, S. Hafsi, and F. Bouani, ‘Adaptive neural network PID controller for nonlinear systems’, in *2021 IEEE 2nd International Conference on Signal, Control and Communication (SCC)*, Tunis, Tunisia: IEEE, Dec. 2021, pp. 264–269. doi: 10.1109/SCC53769.2021.9768352.

[8] S. Richter, C. N. Jones, and M. Morari, ‘Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method’, *IEEE Trans. Automat. Contr.*, vol. 57, no. 6, pp. 1391–1403, Jun. 2012, doi: 10.1109/TAC.2011.2176389.

[9] S. Chen, K. Werling, A. Wu, and C. K. Liu, ‘Real-time Model Predictive Control and System Identification Using Differentiable Physics Simulation’, Nov. 23, 2022, *arXiv*: arXiv:2202.09834. doi: 10.48550/arXiv.2202.09834.

[10] S. A. N. Nouwens, B. de Jager, M. M. Paulides, and W. P. M. H. Heemels, ‘Constraint-adaptive MPC for large-scale systems: Satisfying state constraints without imposing them’, *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 232–237, 2021, doi: 10.1016/j.ifacol.2021.08.550.

[11] ‘Limitations of PID controllers’, in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, San Diego, CA, USA: IEEE, 1999, pp. 3843–3847 vol.6. doi: 10.1109/ACC.1999.786236.

[12] L. J. D. Silva Moreira, A. P. V. D. A. Aguiar, P. R. Barros, and G. Acioli Júnior, ‘Data-Driven PID Closed-Loop Evaluation and Retuning Time and Frequency Domain Approaches’, *J Control Autom Electr Syst*, vol. 32, no. 1, pp. 82–95, Feb. 2021, doi: 10.1007/s40313-020-00654-0.

[13] M. Kvasnica, C. N. Jones, I. Pejcic, J. Holaza, M. Korda, and P. Bakaráč, ‘Real-Time Implementation of Explicit Model Predictive Control’, in *Handbook of Model Predictive Control*, S. V. Raković and W. S. Levine, Eds., in Control Engineering. , Cham: Springer International Publishing, 2019, pp. 387–412. doi: 10.1007/978-3-319-77489-3_17.

[14] M. N. Zeilinger, C. N. Jones, and M. Morari, ‘Real-Time Suboptimal Model Predictive Control Using a Combination of Explicit MPC and Online Optimization’, *IEEE Trans. Automat. Contr.*, vol. 56, no. 7, pp. 1524–1534, Jul. 2011, doi: 10.1109/TAC.2011.2108450.

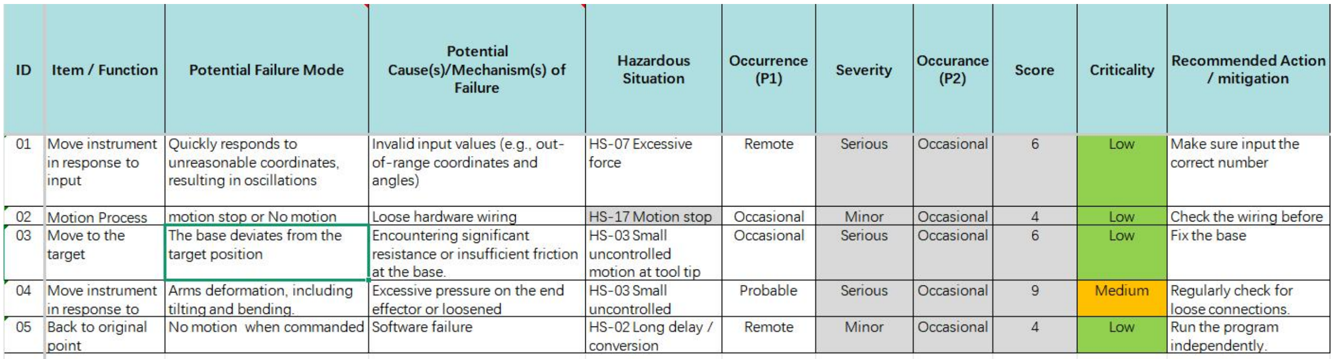
[15] M. Rox *et al.*, ‘Decoupling Steerability From Diameter: Helical Dovetail Laser Patterning for Steerable Needles’, *IEEE Access*, vol. 8, pp. 181411–181419, 2020, doi: 10.1109/ACCESS.2020.3028374.

[16] C. Glanois *et al.*, ‘A Survey on Interpretable Reinforcement Learning’, Feb. 24, 2022, *arXiv*: arXiv:2112.13112. doi: 10.48550/arXiv.2112.13112.

[17] I. Meir, A. Bechar, and A. Sintov, ‘Kinematic Optimization of a Robotic Arm for Automation Tasks with Human Demonstration’, in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan: IEEE, May 2024, pp. 7172–7178. doi: 10.1109/ICRA57147.2024.10610924.

[18] H. Zeng, ‘Lightweight Design and Simulation of Robot Arm’, in *2024 International Conference on Electrical Drives, Power Electronics &amp; Engineering (EDPEE)*, Athens, Greece: IEEE, Feb. 2024, pp. 201–206. doi: 10.1109/EDPEE61724.2024.00045.

Supplemental Material



ID	Item / Function	Potential Failure Mode	Potential Cause(s)/Mechanism(s) of Failure	Hazardous Situation	Occurrence (P1)	Severity	Occurance (P2)	Score	Criticality	Recommended Action / mitigation
01	Move instrument in response to input	Quickly responds to unreasonable coordinates, resulting in oscillations	Invalid input values (e.g., out-of-range coordinates and angles)	HS-07 Excessive force	Remote	Serious	Occasional	6	Low	Make sure input the correct number
02	Motion Process	motion stop or No motion	Loose hardware wiring	HS-17 Motion stop	Occasional	Minor	Occasional	4	Low	Check the wiring before
03	Move to the target	The base deviates from the target position	Encountering significant resistance or insufficient friction at the base.	HS-03 Small uncontrolled motion at tool tip	Occasional	Serious	Occasional	6	Low	Fix the base
04	Move instrument in response to	Arms deformation, including tilting and bending.	Excessive pressure on the end effector or loosened	HS-03 Small uncontrolled	Probable	Serious	Occasional	9	Medium	Regularly check for loose connections.
05	Back to original point	No motion when commanded	Software failure	HS-02 Long delay / conversion	Remote	Minor	Occasional	4	Low	Run the program independently.