AUTONOMOUS 3DOF PICK-AND-PLACE ROBOT USING VISION-BASED CONTROL

Andrea Walker Perez (email: k2478626@kcl.ac.uk; Student ID: 1861152)

Author Affiliations: Anthony Ishaya Hana, Quitterie Lacome d'Estalenx

ABSTRACT

This report presents the design, development, and evaluation of a 3 Degree-of-Freedom (3DOF) autonomous pick-and-place robotic system guided by computer vision. The system integrates a Logitech camera, AprilTag-based object detection, and a ROS2-based modular control architecture to enable precise object handling. Motion planning was performed using inverse kinematics with a Jacobian-based Damped Least Squares (DLS) method, ensuring smooth and stable movements. Experiments assessed placement accuracy with varying tag sizes, analysed the effect of damping parameters, and evaluated collision detection mechanisms using both vision and motor current monitoring. Results demonstrated high placement precision, robust obstacle handling, and operational safety, highlighting the system's potential applicability in medical robotics for autonomous surgical tool handling. Future improvements include the integration of advanced object recognition algorithms to eliminate marker dependency and further enhance adaptability in dynamic environments.

1. INTRODUCTION

In high-pressure surgical environments, the efficiency, accuracy, and timing of instrument handling are crucial factors influencing patient outcomes. Studies have shown that automation in hospitals can reduce cognitive load and improve coordination, particularly when integrated with smart robotics and AI systems [1, 2]. Traditionally, human assistants are responsible for passing tools to surgeons, but studies show that this process can introduce delays, mistakes, and even impact post-surgical recovery [3, 4]. Recent innovations in robotic scrub nurses [5], instrument delivery systems [6, 7], and surgical automation [8] demonstrate the growing use of robots to improve surgical workflows and reduce human error.

This project aims to develop a vision-guided robotic system capable of autonomously performing pick-and-place operations with high precision and safety. By leveraging computer vision and real-time control algorithms, the system seeks to replicate the accuracy and consistency required in surgical environments for handling instruments or components.

To achieve this, the project was structured around the following objectives:

- Design and assemble a 3DOF robotic manipulator integrated with a motorised gripper.
- Implement a computer vision pipeline using AprilTags to localise objects in real time.
- Develop a ROS2-based modular control system for vision, motion planning, and actuation.
- Employ inverse kinematics with a Damped Least Squares (DLS) solver to ensure a smooth motion controller.
- Test the robot's placement accuracy under different visual conditions and damping parameters.
- Evaluate system reliability and safety using collision detection via vision-based path analysis and motor current monitoring.

By meeting these objectives, the project demonstrates the viability of a low-cost, modular robotic system for use in applications where precision and safety are paramount, such as autonomous surgical tool handling.

2. METHODS

2.1. Experimental Setup

The setup (Figure 3) comprised a custom-built robotic arm with 3 degrees of freedom, powered by two Dynamixel motors and a motorised clamp end-effector. The motors were controlled through a Raspberry Pi 4 and OpenRB-150 interface board. The end-effector (Figure 1) was 3D printed based on a custom CAD design optimised for secure cube grasping.

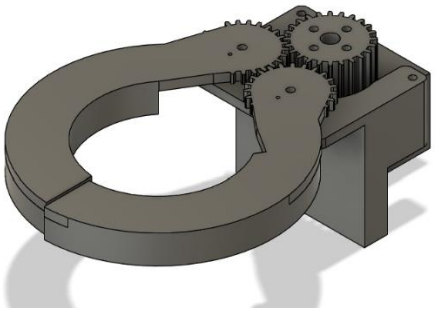

Figure 1 CAD model of the robotic end-effector (clamp mechanism) designed for object grasping

A Logitech webcam was mounted above the workspace, providing a complete overhead view of the environment. The workspace was populated with cubes marked by AprilTags to enable accurate object localisation. Careful camera calibration ensured the captured image coordinates could be reliably converted into the robot's base frame for precise movement execution. The robot's reachable workspace, calculated via forward kinematics, forms a semi-circular region as shown in Figure 2. This defines the limits of the robot's motion in Cartesian space and was used to constrain object placement and target zones during testing.

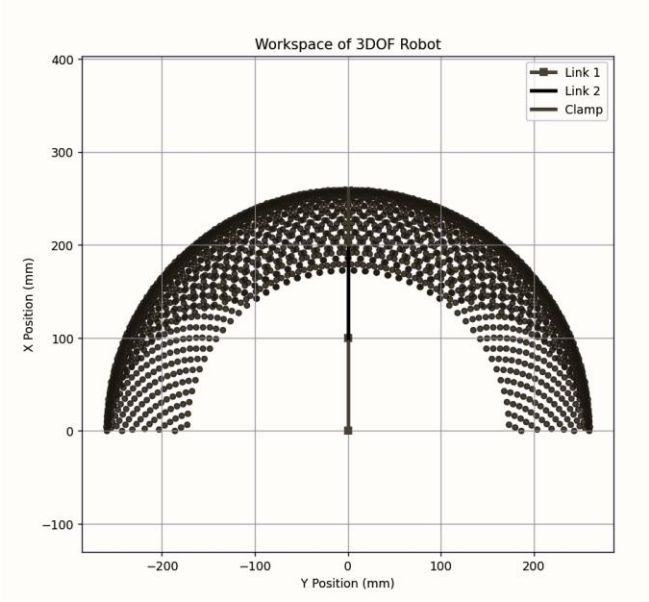

Figure 2 Available workspace for the 3DOF Robot

Figure 3 Camera and Robot Arm setup

2.2. Control System / Algorithms

The robot's control system was implemented using a modular ROS2-based architecture. Image data from the Logitech webcam was acquired using OpenCV and published as ROS topics. A dedicated vision-processing node performed AprilTag detection, extracting tag IDs and 6-DOF poses from the camera's coordinate frame. These detected poses were then transformed into the robot's base frame using extrinsic calibration parameters. AprilTags were selected over ArUco markers for improved robustness, detection range, and accuracy [9, 10]. Their higher-resolution binary encoding, stability at steep viewing angles, and reduced false favourable rates made them ideal for this system, which uses a top-down camera view and operates in variable lightning conditions. The OpenCV AprilTag module was used with the DICT_APRILTAG_16h5 dictionary, balancing tag ID capacity and robustness.

Figure 5 shows the modular ROS2 node architecture, with individual nodes handling image acquisition, AprilTag detection, trajectory generation, and motor control.

Motion planning was achieved using a Jacobian-based inverse kinematics solve, using the Damped Least Squares (DLS) method to compute joint velocity commands from the end-effector velocity requirements. DLS is particularly suitable for avoiding singularities and ensuring numerical stability during motion. The standard formulation is:

$$\Delta\theta = J^T(JJ^T + \lambda^2 I)^{-1}\Delta x \quad [11]$$

where $\Delta\theta$ is the joint velocity vector, J is the Jacobian matrix, λ is the damping factor, and Δx is the desired end-effector velocity. The damping factor was experimentally tuned. DLS was selected over direct pseudoinverse or Jacobian transpose methods due to its superior numerical stability near singularities and ability to manage the limited redundancy of the 3DOF arm.

Once a valid target pose was identified, the trajectory planner node generated an interpolated path and sent it to the motor control node. This node handled actuation, gripper control, and emergency stop functionality. Real-time motor currents were monitored to detect potential collisions with objects or workspace boundaries.

Figure 6 illustrates the step-by-step execution of a pick-and-place task, from initial positioning and object detection to grasping, transporting, and releasing the object at the target location.

2.2.1. Camera Calibration

Accurate pose estimation of AprilTags requires precise calibration of the camera. A checkboard pattern size 10x7 (internal corners) with 25 mm square spacing was used. Fifteen images were captured from different angles within the camera's field of view. The calibration algorithm extracted 2D image points and matched them to 3D world coordinates. Sub-pixel refinement was performed using OpenCV's *cornerSubPix* method, and the intrinsic parameters (camera matrix) and distortion coefficients were estimated using *cv.calibrateCamera* [12].

Reprojection error was computed by projecting 3D object points back onto the image and comparing them to the detected corners, resulting in a mean projection error below 0.5 pixels. All calibration parameters were saved to calibration.npz and loaded during runtime to support accurate AprilTag pose estimation.

Figure 4 shows the calibration setup and the corner detection grid for calculating the camera parameters.

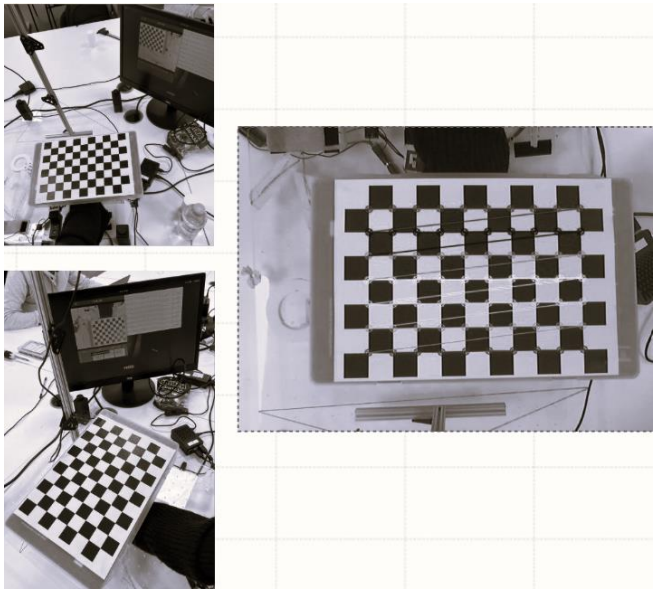

Figure 4 Calibration Setup and Corner Detection

2.2.2. ROS2 Node Implementation

The robot's control system was structured into four ROS2 nodes. This follows the ROS guideline that each node should encapsulate a single, well-defined responsibility [13] and keeps the code ready for in-process composition via the ROS 2 Components API [14]

- camera_publisher.py: Captures RGB frames with *cv2.VideoCapture* and publishes them relying entirely on OpenCV I/O utilities [15].
- aruco_posed.py: Detects AprilTags through OpenCV's ArUco subsystem (*cv::aruco::detectMarkers*) [15]. The tag corners are then fed to estimatePoseSingleMarkers, which internally solves the Perspective-n-Point (PnP) pose-estimation problem [16]. AprilTag 2 coding is used for its high robustness and low false-positive rate [10].
- marker_tracking.py: Acts as the central decision-making node. It subscribes to marker pose data, pairs object and target markers, performs inverse kinematics using DLS [11], and publishes joint angle commands and gripper control signals.
- hardware_interface.py: Handles low-level communication with Dynamixel motors via the Dynamixel SDK [17]. It also includes motor calibration, current/velocity monitoring, gripper control, and a collision response system that logs and recovers from unsafe conditions.

This modular node structure enables distributed processing and allows component-specific debugging and reuse in future robotic systems.
Real-time safety monitoring was implemented by continuously evaluating motor current values against defined thresholds, triggering a recovery procedure, and publishing emergency stop signals when necessary.

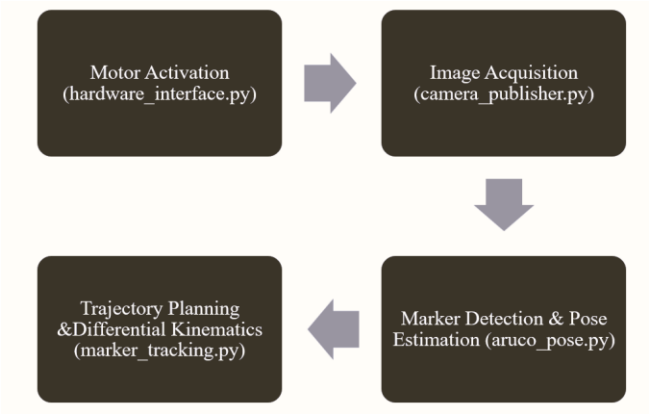

Figure 5 Block diagram showing the modular ROS2 architecture, with nodes for image acquisition, AprilTag detection, trajectory planning, and motor control.

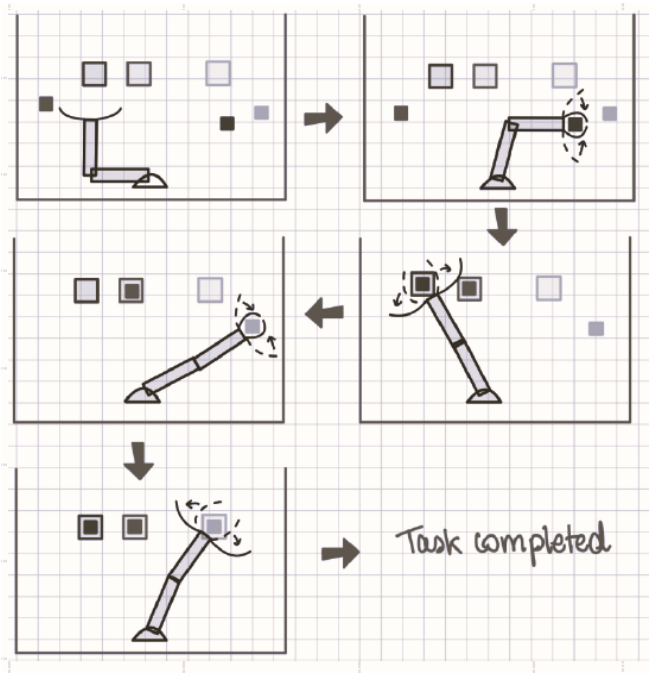

Figure 6 Pick-and-Place Operational Pipeline

2.3. Experimental Protocol

Four structured experiments were conducted to thoroughly evaluate the robot's performance and robustness. Each experiment involved systematic procedures, repeated trials, and specific performance measurements.

2.3.1. Placement Accuracy Test with 25mm AprilTags

All object and target AprilTags were positioned within the robot's mapped workspace (see Figure 2) to ensure full reachability and avoid joint limit violations during pick-and-place operations.

Small 25 mm AprilTags were affixed to cubes within the robot's workspace to evaluate baseline placement accuracy. The robot's vision system detected each object marker, calculated its position relative to the base frame, and executed a complete pick-and-place cycle using inverse kinematics and Damped Least Squares (DLS) control method. This involved moving from the home position, grasping the cube, and placing it at a predefined target location associated with another AprilTag. After each placement, the final Cartesian position of the end-effector was recorded. The process was repeated across ten trials to ensure statistical reliability. Mean positional errors and standard deviations between the expected and actual final positions were calculated to assess consistency and accuracy.

2.3.2. Placement Accuracy Test with 50mm AprilTags

The second experiment followed the same procedure, with larger 50mm AprilTags attached to the cubes. After recalibrating the camera system, the robot detected and tracked the larger tags and performed ten pick-and-place operations. The final end-effector positions were recorded after each placement, as with the previous test. This experiment aimed to determine whether larger tags enhanced detection robustness and placement precision, while also examining whether increased size introduced distortion-related errors, particularly toward the edges of the camera's field of view.

2.3.3. Damping Factor Variation Experiment

To investigate the influence of the damping factor parameter in the DLS solver, the robot was tasked with performing repeated pick-and-place operations under five different damping factor settings: 0.0, 0.05, 0.1, 1.0, and 5.0. For each setting, the damping value was updated in the control node, and the robot executed ten identical trials, moving an object from a fixed starting position to a predefined target. Final end-effector positions were measured after each trial, and positional errors were calculated to compare accuracy across different settings. This experiment identified the damping factor that best balanced trajectory stability and responsiveness.

2.3.4. Collision Control Tests

Collision handling was evaluated through two complementary tests. In the vision-based collision prevention test, foreign AprilTags, unrelated to the current task, were temporarily placed within the robot's trajectory path during operation. The robot was expected to detect the obstructing marker, abort its current movement, return safely to its home position, and resume the task once the obstacle was removed.

In the hardware-based collision detection test, external force was gently applied to the robot arm mid-operation to simulate physical interference. Motor currents were monitored in real time, and when the current exceeded predefined thresholds (see section 3. Results for these thresholds), the system was expected to trigger an emergency stop. The test was repeated under various current thresholds, including no limit conditions, to assess sensitivity and robustness. Success was determined by the robot's ability to safely halt or respond appropriately without sustaining hardware damage.

2.4. Data Collection and Analysis

Relevant quantitative data were collected during each experiment to evaluate the system's performance across different tasks and conditions. For the placement accuracy experiments, the Cartesian coordinates of the robot's end-effector were recorded after each pick-and-place operation using internal kinematics feedback. These positions were compared to the predefined target coordinates associated with AprilTag locations, allowing for calculation of the positional error in three dimensions. Each experimental condition was repeated ten times to ensure statistical robustness. The mean error and standard deviation were computed for each trial set to assess accuracy and consistency.

The same positional error metrics were used for the damping factor variation experiment to compare performance across control parameter settings. This analysis helped identify the damping values that yielded the lowest average error with minimal variability.

The collision control tests recorded two types of data: motor current readings and task interruption behaviour. During hardware collision tests, real-time current values from each Dynamixel motor were logged and compared against predefined safety thresholds. In vision-based collision prevention, task interruption and recovery behaviour were monitored and documented. These data points allowed for assessment of both detection sensitivity and the effectiveness of the safety protocols in dynamic environments.

All data were logged and processed using Python-based scripts, enabling systematic computation of performance metrics later reported in the Results section.

3. RESULTS

3.1. Placement Accuracy with 25mm AprilTags

The robot demonstrated high accuracy and consistency when placing the objects using 25mm AprilTags. As shown in Table 1, the mean Cartesian coordinates for each target were closely aligned with the desired positions, with standard deviations under 3.2 mm in all directions. This indicates that the vision system, trajectory planning, and actuation worked reliably within the camera's central viewing zone.

Figure 7 visualises this performance, where each point represents the target position and actual placement with error bars showing standard deviation. The small spread of errors across all five trials reinforces the system's robustness under consistent lighting and calibrated conditions.

Video Demonstration

Table 1 Placement accuracy results using 25mm AprilTags

Desired Position (x, y)	**Mean Cartesian (x, y)**	**Std Cartesian (x, y)**
(106.04, 194.71)	(106.01, 191.97)	(1.19, 1.34)
(179.46, 156.82)	(179.86, 155.19)	(1.09, 1.27)
(187.57, -160.65)	(184.64, -157.25)	(1.60, 2.34)
(221.47, -8.07)	(216.75, -7.58)	(2.60, 2.05)
(79.35, -210.96)	(79.45, -210.81)	(0.74, 3.16)

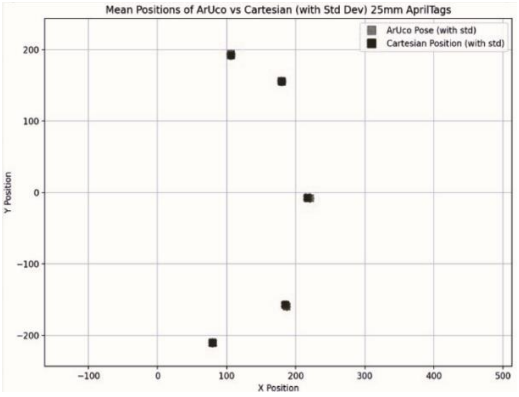

Figure 7 Graph of mean positional error (mm) for pick-and-place tasks with 25mm AprilTags

3.2. Placement Accuracy with 50mm AprilTags

Using 50mm AprilTags improved detection reliability but introduced placement variability at the camera's periphery. As shown in Table 2, the central targets yielded comparable performance to the 25mm test. However, the last two rows reveal significant deviations (e.g., ~17 mm error and high standard deviation) in areas where lens distortion was more pronounced.

Figure 8 highlights this trend, with larger error bars for placements near the edges of the workspace. These findings suggest that while larger tags enhance visibility, careful camera calibration and workspace layout are essential to maintain precision throughout the field of view.

Video Demonstration

Table 2 Placement accuracy results using 50mm AprilTags

Desired Position (x, y)	**Mean Cartesian (x, y)**	**Std Cartesian (x, y)**
(183.90, -154.04)	(184,98, -150.95)	(1.03, 1.49)
(197.64, 127.35)	(199.23, 122.46)	(1.30, 1.40)
(219.74, -1.43)	(217.18, -0.69)	(0.53, 0.45)
(94.54, -204.10)	(85.12, -175.12)	(17.54, 27.88)
(99.64, 189.77)	(104.23, 153.42)	(16.79, 85.20)

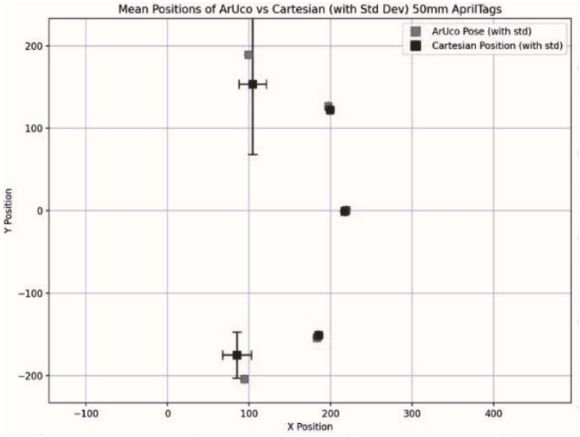

Figure 8 Graph of mean positional error (mm) for pick-and-place tasks with 50mm AprilTags

3.3. Effect of Damping Factor on Placement Accuracy

This experiment tested how the DLS damping factor influenced placement precision. Table 3 shows each damping value's average placement error and standard deviation. The lowest error (1.14 mm) occurred at a damping factor of 0.05, which also had the least variability (0.054 mm). Both extremely low (0.01) and high (5.0) damping values resulted in reduced accuracy and stability.

Figure 9 shows that this trend demonstrates an optimal damping region between 0.05 and 0.1. These results validate the importance of fine-tuning DLS parameters in balancing responsiveness and trajectory smoothness.

Table 3 Mean standard deviation and placement errors under different damping factor values

Damping Factor	**Mean Error (mm)**	**Std Dev (mm)**
0.01	2.55	0.80
0.05	**1.14**	**0.54**
0.1	1.70	0.98
0.5	2.84	0.87
1.0	2.43	1.09
5.0	2.62	0.94

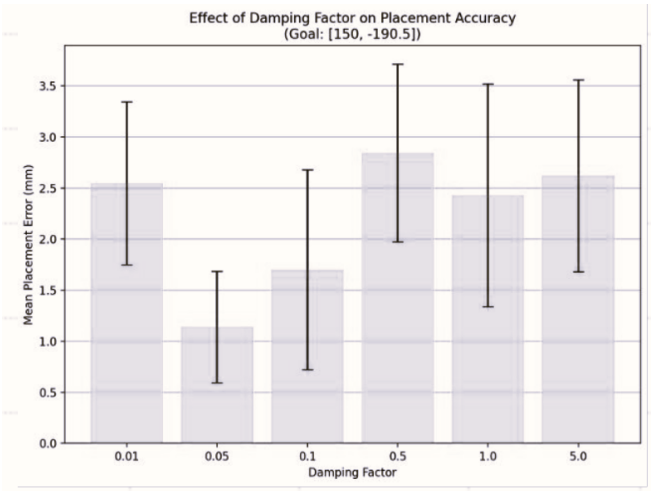

Figure 9 Graph of placement error versus damping factor values, illustrating optimal damping region

3.4. Collision Control Tests

3.4.1. Current Collision

Four collision control trials evaluated the system's ability to detect and respond to physical disturbances via motor current threshold. In Figure 10, no force was applied, and both motor velocities remained stable. In Figure 11, force was applied, but no current threshold was enforced; this resulted in continued motion and increased current draw, potentially risking hardware stress.

Figure 12 and Figure 13 show the system's performance with thresholds set at 190 mA / 170 mA and the optimal 150 mA / 100 mA, respectively. In both cases, the robot successfully halted its motion, with Figure 13 representing the most effective safety response.

These findings confirm that setting appropriate current limits enables the robot to react rapidly to unexpected collisions without damage.

Video Demonstration

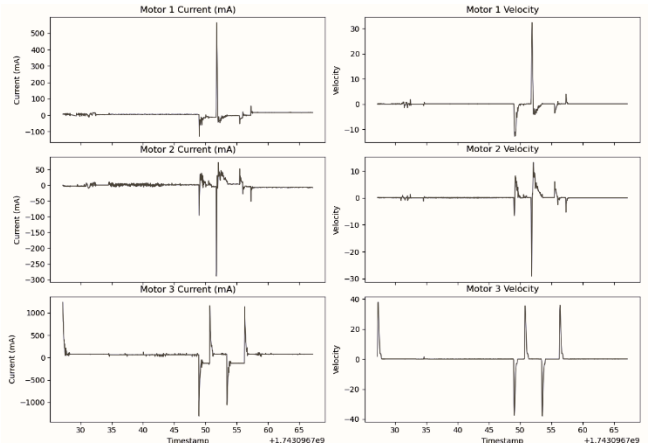

Figure 10 Current and Velocity reading when no force is applied

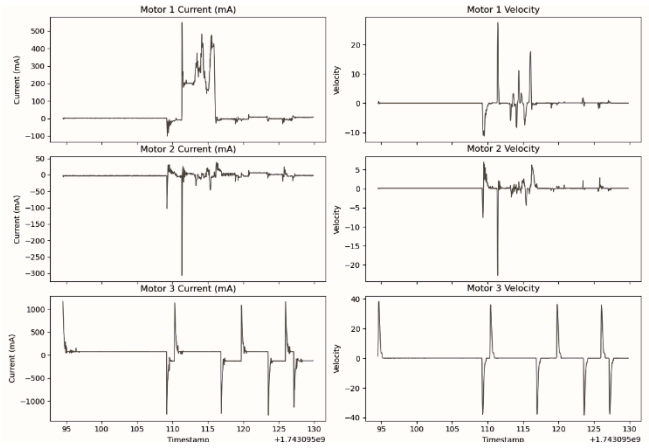

Figure 11 Current and Velocity readings are taken when a force is applied, but no restraining threshold is added.

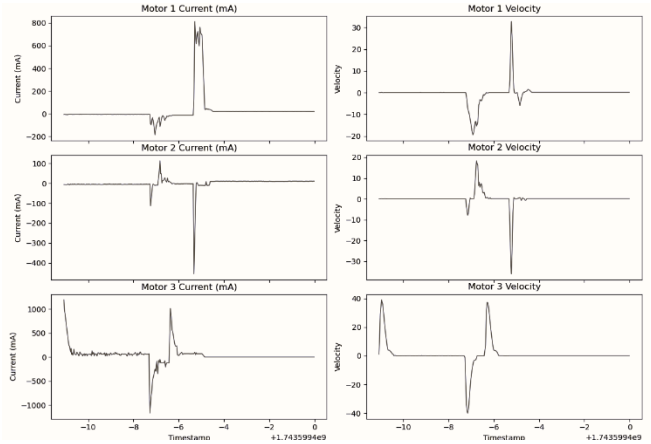

Figure 12 Current and Velocity Readings when force is applied, and a threshold of 190 for motor 1 and 170 for motor 2

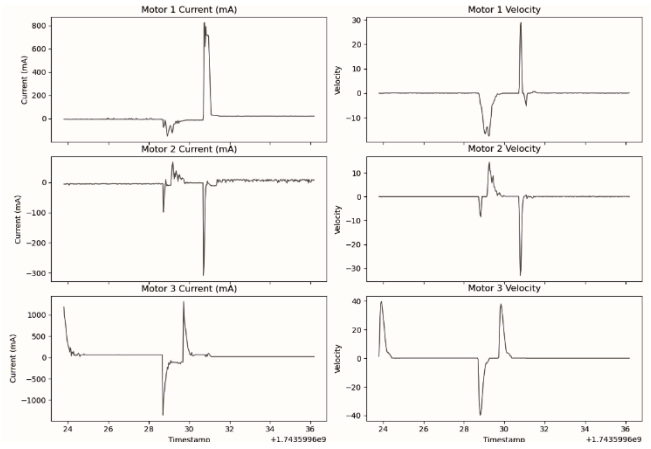

Figure 13 Current and Velocity readings with force applied and optimal thresholds of 150 for motor 1 and 100 for motor 2

3.4.2. Collision avoidance

In this experiment, a foreign AprilTag was deliberately placed within the robot's planned path to test its ability to avoid visual obstacles. The robot identified the obstructing tag as too close to the trajectory and immediately aborted the motion. It then returned to the predefined home position before attempting to re-plan its path once the obstacle was cleared.

As shown in Figure 14, the ROS2 terminal output logs illustrate the complete detection and response cycle. A warning is triggered as the robot calculates proximity to the obstacle, then publishes a command to move to the home position for safety. This behaviour replicates basic surgical assistant logic, where unexpected objects (e.g., instruments or personnel) must be automatically avoided without compromising safety.

Video Demonstration

Figure 14 ROS2 terminal output demonstrating vision-based collision avoidance

4. DISCUSSION

The experimental results validate the effectiveness of the developed 3DOF vision-guided robotic system for precise and autonomous pick-and-place operations. Across all tests, the system consistently achieved low placement errors under well-calibrated conditions while demonstrating resilience through software- and hardware-based collision detection strategies.

The placement accuracy experiments using 25mm AprilTags showed strong reliability, with low mean positional errors and minimal variation. These results confirm reliability, with low mean positional errors and minimal variation. These results confirm the quality of the camera calibration and the robustness of the inverse kinematics pipeline under optimal conditions. In contrast, the test using 50mm AprilTags revealed increased variability in end-effector positions, particularly when objects are near the edge of the camera's field of view. This deviation likely stems from lens distortion effects amplified with larger tags and wider coverage areas. While larger tags improve detection reliability, the results underscore the importance of precise camera calibration and workspace layout, primarily when operating close to image borders.

The experiment on damping factor variation reinforced the importance of tuning control parameters. Damping values of 0.05 and 0.1 yielded the lowest mean placement errors and smallest standard deviations. Excessively low or high damping factors led to erratic or overly sluggish motions, confirming that fine-tuning the DLS solver is essential to balance responsiveness and stability in the robot's motion planning [11].

Collision control tests demonstrated that the system could detect and respond to physical contact and visual obstructions in real time. The hardware-based collision detection, relying on current thresholds, successfully halted motion when the robot encountered external force. Tests showed that the optimal thresholds (150 mA for Motor 1 and 100 mA for Motor 2) allowed timely intervention without false positives. Vision-based collision avoidance was equally effective: the robot correctly identified unexpected markers in its path and adjusted its behaviour accordingly, mimicking behaviour required in high-stakes environments like surgical assistance.

Despite these promising results, several limitations were observed. The system depends on fiducial markers (AprilTags), which are impractical in many real-world scenarios. Future iterations could incorporate deep learning-based object recognition techniques (e.g., YOLOv4 or similar) to enable marker-less operation. Additionally, lighting conditions and camera placement notably affected detection accuracy. Slight misalignments or environmental variations could degrade system performance, indicating the need for more adaptive calibration or dynamic vision processing. These observations align with the findings in human-robot interaction studies, which emphasise the need for adaptive systems that reduce cognitive burden and support real-time decision-making in critical environments such as surgery [3, 4].

Finally, the robot's current configuration is limited to three degrees of freedom, constraining its reachable workspace and flexibility. Expanding the system to include additional degrees of freedom would allow for more complex movements and broader applicability in constrained or cluttered environments, such as an operating room.

For medical integration, improving interface usability and user trust, highlighted in surgical robotics literature [1, 2], will be crucial in future developments.

5. CONCLUSION

This project successfully demonstrated the design and implementation of an autonomous 3DOF robotic system capable of performing vision-guided pick-and-place operations with high precision and reliability. By integrating AprilTag-based object detection, a modular ROS2 control architecture, and trajectory planning using the Damped Least Squares (DLS) method, the robot could dynamically interact with its environment while responding safely to physical and visual obstacles.

Experimental results showed that the system achieved low placement error and consistent performance when operating with 25mm AprilTags, with slightly increased variability observed when using larger markers near the edge of the workspace. Parameter tuning experiments revealed that moderate damping values (0.05-0.1) produced optimal motion control, balancing smoothness and responsiveness. Additionally, the robot demonstrated effective collision handling through hardware current monitoring and vision-based avoidance, validating the robustness of its safety mechanisms.

This project provided valuable hands-on experience in integrating computer vision. Robotics control and real-time system feedback. It also highlighted the importance of careful calibration, parameter tuning, and environmental design to achieve consistent robotic performance.

Future improvements could include implementing deep learning-based object recognition to eliminate reliance on fiducial markers, extending the robot's degrees of freedom for increased reach and dexterity, and integrating more advanced human-robot interaction features such as voice control [1], shared autonomy, or surgeon-guided gestures [5]. Prior research in adaptive interfaces and robotic assistants supports the value of these additions in improving usability and real-time safety [6]. These enhancements would further align the system with the demands of real-world medical robotic applications, including surgical assistance and intelligent tool handling.

6. REFERENCES

[1] K. Jeong Hun *et al.*, "Development of a Smart Hospital Assistant: integrating artificial intelligence and a voice-user interface for improved surgical outcomes," in *Proc.SPIE*, 2021, vol. 11601, p. 116010U, doi: 10.1117/12.2580995. [Online]. Available: https://doi.org/10.1117/12.2580995

[2] A. Ezzat, A. Kogkas, J. Holt, R. Thakkar, A. Darzi, and G. Mylonas, "An eye-tracking based robotic scrub nurse: proof of concept," (in eng), *Surg Endosc*, vol. 35, no. 9, pp. 5381-5391, Sep 2021, doi: 10.1007/s00464-021-08569-w.

[3] F. Lai and E. Entin, "Robotic Surgery and the Operating Room Team," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 11, pp. 1070-1073, 2005/09/01 2005, doi: 10.1177/154193120504901115.

[4] K. Catchpole *et al.*, "Human Factors Integration in Robotic Surgery," *Human Factors*, vol. 66, no. 3, pp. 683-700, 2024/03/01 2022, doi: 10.1177/00187208211068946.

[5] M. G. Jacob, Y. T. Li, and J. P. Wachs, "A gesture driven robotic scrub nurse," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 9-12 Oct. 2011 2011, pp. 2039-2044, doi: 10.1109/ICSMC.2011.6083972.

[6] M. G. Jacob, Y. T. Li, and J. P. Wachs, "Surgical instrument handling and retrieval in the operating room with a multimodal robotic assistant," in *2013 IEEE International Conference on Robotics and Automation*, 6-10 May 2013 2013, pp. 2140-2145, doi: 10.1109/ICRA.2013.6630864.

[7] H. Tan *et al.*, "An integrated vision-based robotic manipulation system for sorting surgical tools," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 11-12 May 2015 2015, pp. 1-6, doi: 10.1109/TePRA.2015.7219664.

[8] C. Perez-Vidal *et al.*, "Steps in the development of a robotic scrub nurse," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 901-911, 2012/06/01/ 2012, doi: https://doi.org/10.1016/j.robot.2012.01.005.

[9] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 9-13 May 2011 2011, pp. 3400-3407, doi: 10.1109/ICRA.2011.5979561.

[10] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9-14 Oct. 2016 2016, pp. 4193-4198, doi: 10.1109/IROS.2016.7759617.

[11] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 123-134, 1994, doi: 10.1109/87.294335.

[12] O. Contributors. "Camera Calibration." https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (accessed.

[13] Open Robotics. "Understanding ROS 2 Nodes." https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html (accessed 1 May 2025.

[14] Open Robotics. "About Composition." https://docs.ros.org/en/foxy/Concepts/About-Composition.html (accessed 1 May 2025.

[15] S. Garrido and A. Panov. "Detection of ArUco Markers." https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html (accessed 1 May 2025.

[16] X. Lu, "A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation," *Journal of Physics: Conference Series*, vol. 1087, p. 052009, 09/01 2018, doi: 10.1088/1742-6596/1087/5/052009.

[17] Robotis. "DYNAMIXEL SDK Overview." https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/overview/ (accessed 1 May 2025.

7. APPENDIX

A: Supplementary Resources

The following resources contain supporting materials referenced throughout this report:

- GitHub Repository (Code & ROS2 Nodes)
- Video Demonstrations and Results