Gesture-Based Teleoperation of Stretch Robot

Shubhankar Katekari[#1], Atharva Jamsandekar[#2], Nikhil Gutlapalli[#3]

[#]*Robotics, Northeastern University*
Boston, MA, USA

[1]katekari.s@northeastern.edu

[2]jamsandekar.a@northeastern.edu

[3]gutlapalli.n@northeastern.edu

***Abstract* - We present a real-time gesture recognition and robotic control system using ROS2 and Mediapipe. Our platform recognizes human arm and hand gestures from standard webcam input and translates them into commands for a robotic arm and gripper. The system includes multiple modes: base movement, arm positioning, and gripper control, each using different hand pose landmarks for intuitive, markerless control. Additionally, we incorporate a "two open palms" gesture for stopping camera processing and returning to an idle state awaiting new instructions. Our approach is designed for ease of integration into existing ROS-based robotics platforms. We evaluate the system's performance across various lighting conditions and user differences, demonstrating reliable detection with minimal latency on common hardware. The results show that real-time gesture control can be achieved with off-the-shelf components, opening the door for more natural and interactive human–robot collaboration scenarios.**

Keywords - Real-time gesture recognition, ROS2-based robotic control, Human–robot interaction (HRI), Markerless gesture-based control

I. Introduction

Gesture-driven robotic control has emerged as a compelling alternative to conventional hardware devices, largely due to natural user experience and reduced physical constraints. Traditional approaches often rely on specialized sensors, such as wearable gloves or infrared markers, which may impede user comfort or increase complexity. In contrast, modern computer vision pipelines are now capable of extracting detailed key points from regular webcam images in real time, allowing for markerless, highly flexible interfaces. This capability has spurred widespread interest in human–robot interaction (HRI), where gestures can communicate a wide array of intentions without requiring physical contact or specialized infrastructure.

Our system seeks to exploit these advancements by integrating Mediapipe - a widely adopted library for real-time landmark detection - with ROS2, a messaging-based framework for robotic software. The synergy between these two technologies aims to isolate the user's control intent from the robot's actual motion planning, leading to an architecture that is both modular and extensible. Specifically, we split the design into two ROS2 nodes: a "Menu Node" focused on user-driven mode selection and a "Controller Node" dedicated to interpreting gestures. This dual-node setup ensures that user interface logic (such as button hovering) remains distinct from the gesture detection and kinematics mapping needed for the robot's arm, gripper, or base.

The system further addresses workflow continuity by providing a "two open palms" gesture that gracefully halts the camera-based controls, reverting to the menu so the operator can select new joint functionalities. Our preliminary results indicate that even non-expert users find this approach simple and intuitive, with minimal training or calibration required. By presenting the following details on methodology, experiments, and outcomes, we hope to illustrate how this approach contributes to robust yet user-friendly HRI.

II. Related Work

A growing body of research has leveraged Mediapipe for gesture and pose recognition, often with impressive real-time throughput and detection accuracy. For instance, N. H. Phat et al. [1] enhance a MediaPipe Holistic model with recurrent networks to improve dynamic gesture segmentation, demonstrating a reduced error rate in challenging settings. Similarly, Yaseen et al. [2]

integrate MediaPipe's keypoint extraction as the front-end for a deep architecture, emphasizing how pretrained CNN models (e.g., Inception-v3) can be cascaded with temporal classifiers (e.g., LSTM) to capture both spatial and motion cues in human gestures. These techniques confirm that robust hand tracking and gesture labeling can be achieved even with modest computational resources.

Beyond the domain of gesture segmentation, researchers have extended these methods to real-world human–robot interaction (HRI). Mazhar et al. [3] incorporate full-body pose estimation and gesture recognition to command robots in real-time, highlighting the crucial role of skeleton tracking in safe, close-proximity operations. Xie et al. [4] address a similar problem in the context of quadruped teleoperation, revealing that well-structured gestures enable fluid transitions among walking, manipulation, and stop behaviors - an idea akin to our multi-mode control for base, arm, and gripper. Meanwhile, purely vision-based manipulator control has been studied from both machine-learning and classical robotics standpoints. Sekkat et al. [5] explore a reinforcement learning algorithm that bypasses explicit inverse kinematics by mapping camera inputs to motor commands, whereas Lin et al. [6] focus on calibrated object detection for precise part placement. Both underscore that visual pipelines can handle increasingly complex tasks without resorting to specialized sensors. Our approach synthesizes these insights, employing MediaPipe's direct landmark extraction within a high-level ROS2 architecture geared toward intuitive gesture control of robotic arms and grippers.

III. Methods

3.1 System Architecture

The overall architecture of our approach is described in Figure 1. A webcam captures the user's gestures, feeding images to both the "Menu Node" and "Controller Node." The Menu Node offers a set of on-screen buttons - each corresponding to a subsystem of the robot (e.g., base movement or arm extension). By hovering a fingertip over a button for a set duration, the user selects or deselects that function. After the user confirms by hitting "continue," the chosen subsystem is published to /selected_joint.

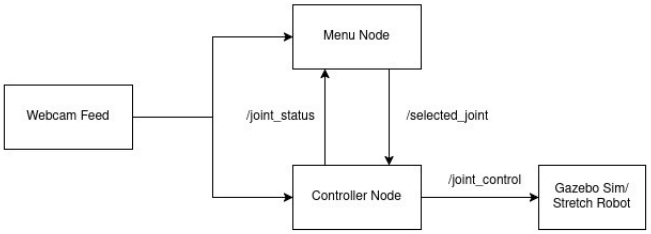

Fig. 1 System Architecture Diagram

In parallel, the Controller Node listens on /selected_joint to determine which gestures to interpret. For instance, if "base" is active, the system translates fingertip offsets into linear and angular velocity commands, whereas if "arm stretch/retract" is chosen, it computes an elbow angle from the user's posture and publishes a normalized joint command. A unique feature is the two open palms gesture: if detected, the Controller Node issues a True message on /joint_status, signaling the Menu Node to resume, effectively "pausing" direct gesture control and allowing the user to pick another mode.

3.2 Hover-Based Menu Interface

An example of the hover-based menu is shown in Figure 2. This interface uses color-coded rectangles to represent each button, changing appearance if the fingertip remains within that region for a specified hover time (e.g., two seconds). This design is less error-prone than immediate clicks because users can correct minor deviations before triggering a selection. Once "continue" is hovered, the final choice is broadcast to the relevant topic, thus neatly isolating UI tasks from robot-specific logic.

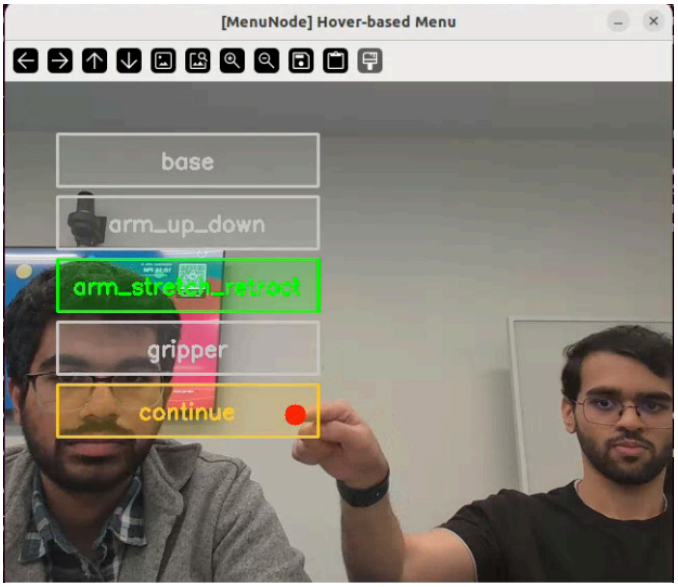

Figure 2.Illustration of the on-screen menu with labeled buttons and fingertip overlay for selection.

3.3 Gesture Detection for Arm and Hand

When the user selects a manipulator-related mode (e.g., up/down, stretch/retract, or gripper), the system leverages Mediapipe's real-time body and hand landmark detection. To keep the presentation accessible, we describe only key functional steps:

1. Arm Up/Down: Compares the vertical position of the wrist to the shoulder, mapping that difference into a 0.0-1.0 scale with incremental steps of 0.1.
2. Arm Stretch/Retract: Calculates an elbow angle from three body landmarks (shoulder–elbow–wrist), again quantized to a discrete set of possible extension levels.
3. Gripper: Monitors the thumb–index distance. If it is below a threshold, the gripper is commanded to close (e.g., 0.0), and otherwise remains open (e.g., 0.9).

A "two open palms" posture finalizes our synergy. If the system detects two separate hands, each with all fingertips sufficiently far from the palm center, it concludes that the user wishes to exit the current gesture mode. The node halts camera-based processing and sends a Boolean "quit" signal to the menu node, which reappears. This mechanism ensures that mode switching remains at the user's discretion without requiring additional physical buttons or complicated gestures.

Figure 3.Overlay showing detected shoulder, elbow, wrist, and fingertip landmarks used for arm control and gripper distance.

IV. Experiments and Results

For evaluation, we deployed our system on a Stretch 3 robot model running in the Ignition Gazebo simulator under ROS2 Humble. Figure 4 shows the simulated environment with the Stretch 3's manipulator, base, and gripper visible. We also used a typical mid-range PC, simulating a standard USB webcam feed at ~30 FPS for both the Menu and Controller Nodes.

To assess the performance of our gesture-control framework, we conducted a series of controlled trials involving n=5 participants. Each participant performed five gesture types - (1) Base Movement, (2) Arm Up/Down, (3) Arm Stretch/Retract, (4) Gripper (Open/Close), and (5) Two Palm Detection for idle/quit signaling - under moderate indoor lighting at a capture rate of approximately 30 FPS. We recorded a total of 100 gesture instances per category, evaluating accuracy, recall, average latency, and false positive rates (where applicable). The following sections detail our findings.

Figure 4.Snapshot of the Stretch 3 robot in the Ignition Gazebo simulation environment, controlled via our gesture-based system.

4.1 Quantitative Analysis

We define accuracy as the proportion of correct recognitions (i.e., the system's output matches the user's intended gesture), while recall captures the fraction of true positives successfully detected among all intended occurrences. Latency corresponds to the mean time from the user's gesture initiation to the visible command effect in the robot simulator. False positives occur when the system erroneously registers a gesture (e.g., "two palms" detection) despite the user not intending to trigger that event.

Table 1 summarizes the results across the five gesture types.

4.1.1 Base Movement

Base control achieved 95% accuracy, indicating that simple fingertip offset detection relative to the image center is robust under typical conditions. The 0.05 false positive rate largely stemmed from minor hand jitter or partial frames where the fingertip was momentarily out of bounds. Overall, participants found the base mode intuitive, with an average latency of 0.20 s, sufficient for smooth teleoperation.

Gesture Type	Accuracy	Recall	Avg. Latency (s)	False Positives
Base Movement	0.95	0.93	0.20	0.05
Arm Up/Down	0.90	0.88	0.22	0.07
Arm Stretch/Retract	0.88	0.85	0.25	0.09
Gripper (Open/Close)	0.96	0.94	0.19	0.04
Two Palms (Quit Signal)	0.85	0.90	0.29	0.15

Table 1. Gesture Accuracy, Recall, Latency and False Positives rate results

4.1.2 Arm Up/Down and Arm Stretch/Retract

Arm-related gestures showed slightly lower accuracy and recall (down to 0.88 - 0.90) due to potential rapid user movements. Nevertheless, the quantization to 0.1 increments of joint positions helped stabilize the control. The latency values of 0.220.22 and 0.250.25 seconds remain acceptable for non-critical tasks, such as placing objects. A modest increase in false positives (up to 0.070.07–0.090.09) can be attributed to partial frames where the user's arm was mid-gesture, momentarily resembling another posture.

4.1.3 Gripper (Open/Close)

The gripper gesture yielded the highest overall metrics, with 96% accuracy and 94% recall, reflecting the robustness of the thumb–index distance threshold. Users noted that they could reliably maintain or break a certain finger spacing to initiate open/close commands. The system's false positive rate of 0.04 was the lowest among all gestures, suggesting that simple binary thresholds can be especially reliable when focusing on fingertip separation.

4.1.4 Two Palms (Quit Signal)

While the system effectively recognized "two palms" in most cases - shown by a high recall of

0.90 - its accuracy was slightly lower at 0.85, with false positives climbing to 0.15. In numerous trials, users spread a single hand widely or unintentionally displayed a partial second hand in the field of view, leading to misclassification. The average latency of 0.290.29 s results partly from the additional verification step (ensuring both hands remain open for a stable duration), which adds some overhead. Despite these drawbacks, participants found the feature useful for gracefully returning to the menu-based interface.

4.2 User Feedback and Observations

Qualitative feedback indicated that the majority of participants found the system easy to learn, particularly appreciating the hover-based menu's forgiving 2 s selection window. Several users suggested that "two palm detection" be made more sensitive to reduce necessary posture time, though we note that raising sensitivity may further inflate false positives. Overall, the synergy between the menu selection and real-time gesture processing was well-received, aligning with our goal of a modular and user-friendly interface.

V. Discussions and Summary

Our experiments demonstrate that a menu-guided and gesture-driven approach can effectively control a multi-joint robot, shown here with a simulated Stretch 3 manipulator. The separation of user interface logic (menu node) from the posture-based command generation (controller node) ensures that new robot capabilities or modes can be appended without overhauling the entire system. Furthermore, the synergy of real-time pose detection with a simple quantization scheme yields a consistently interpretable link between user gestures and robot states, preventing abrupt or erratic movements.

In practice, we envision broadening the repertoire of recognized gestures - potentially adopting dynamic gestures or multi-finger configurations - to expand the operational envelope. Likewise, future work could explore synergy with speech or other sensors, creating a multimodal interface for even smoother HRI experiences. Nevertheless, the present setup is already sufficiently robust for common teleoperation tasks, emphasizing that off-the-shelf vision solutions, combined with ROS2's flexible architecture, can deliver a practical platform for intuitive human–robot collaboration.

Acknowledgment

We sincerely thank Prof. Bruce Maxwell for his invaluable guidance and for sharing his deep expertise in the field of Computer Vision, which played a crucial role in shaping the direction and execution of this work. His insights and feedback greatly enriched our understanding and approach throughout the project.

References

[1] N. H. Phat, P. D. L. Hong, D. D. Dang, et al., "Proposing Hand Gesture Recognition System Using MediaPipe Holistic and LSTM," in Proc. 2023 Int. Conf. on Advanced Technologies for Communications (ATC), IEEE, 2023.

[2] Yaseen, O.-J. Kwon, J. Kim, et al., "Next-Gen Dynamic Hand Gesture Recognition: MediaPipe, Inception-v3 and LSTM-Based Enhanced Deep Learning Model," Electronics, 2024.

[3] O. Mazhar, S. Ramdani, B. Navarro, R. Passama, A. Cherubini, "Towards Real-Time Physical Human-Robot Interaction Using Skeleton Information and Hand Gestures," in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2018.

[4] J. Xie, Z. Xu, J. Zeng, Y. Gao, K. Hashimoto, "Human–Robot Interaction Using Dynamic Hand Gesture for Teleoperation of Quadruped Robots with a Robotic Arm," Electronics, 2025.

[5] H. Sekkat, S. Tigani, R. Saadane, A. Chehri, "Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping," Applied Sciences, vol. 11, no. 17, 2021.

[6] C.-J. Lin, P.-J. Lin, C.-H. Shih, "Vision-Based Robotic Arm Control for Screwdriver Bit Placement Tasks," Sensors and Materials, vol. 36, no. 3, 2024.