MARVIN: Remote Teleoperation of a Dual-Arm Robot

Real-time human motion mirroring with ROS2 and client-side ML

JAEHO CHO*, The Cooper Union for the Advancement of Science and Art, USA
SOPHIA KLYMCHUK*, The Cooper Union for the Advancement of Science and Art, USA

MARVIN is a dual-arm teleoperational robot that mirrors a human operator's upper-body motion in real time using just a standard webcam. Client-side MediaPipe models extract 3D pose and hand landmarks, which are transmitted via websocket to a ROS2-MoveIt servoing stack that commands two OpenManipulator-X arms. We detail perception-to-actuation mappings, including geometric formulations for joint angles, and describe a web interface that enables intuitive interaction. We report user survey results from local and remote operation, and discuss ethical and societal impacts.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; *Gestural input*; *Web-based interaction.*

Additional Key Words and Phrases: teleoperation, human-robot interaction, ROS2, MoveIt, MediaPipe

ACM Reference Format:
Jaeho Cho and Sophia Klymchuk. 2026. MARVIN: Remote Teleoperation of a Dual-Arm Robot: Real-time human motion mirroring with ROS2 and client-side ML. In *Proceedings of TEI Student Design Challenge (TEI SDC '26).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

MARVIN is a teleoperational robot that mirrors the upper-body movements of a human operator in real time. Operation works via a local webcam or a remote connection, where a web application captures the user's webcam stream for pose landmarking, enabling MARVIN to act as a physical avatar across any distance.

MARVIN explores the theme of the 2026 SDC theme of Sensory Rituals through the design of a remotely-operated humanoid robot that reintroduces physicality into digital communication. Contemporarily, much of interpersonal digital interaction occurs via screen-based interfaces that flatten embodied and sensory engagement. Via MARVIN, we seek to restore a sense of physical presence to virtual meetings by allowing its users to inhabit a robot equipped with two controllable arms. By remotely manipulating the robot via one's own mirrored actions, tangible social connection can be had at-distance, enabling the operator to perform actions, express intentions, and interact with physical environments shared by the receiver.

Furthermore, MARVIN is an accessible platform, as it requires only a standard webcam and Internet interface, which extends this ritual to many users, with only minimal technological barriers.

*Both authors contributed equally to this research.

Authors' addresses: Jaeho Cho, jaeho.cho@cooper.edu, The Cooper Union for the Advancement of Science and Art, New York, New York, USA; Sophia Klymchuk, sophia.klymchuk@cooper.edu, The Cooper Union for the Advancement of Science and Art, New York, New York, USA.

TEI SDC '26, March, 2026, Chicago, Illinois

ACM ISBN 978-1-4503-XXXX-X/2026/06
https://doi.org/XXXXXXX.XXXXXXX

Via its integration of physical interaction and accessibility, MARVIN allows tangible rituals of connection to emerge out of separation.

2 METHODOLOGY

2.1 Hardware And Software Integration

MARVIN uses components from the ROBOTIS OpenManipulator-X platform. Each arm has 5 degrees-of-freedom and is powered by Dynamixel XM430-W350 servos. We utilize two arms attached to an aluminum-beam torso and powered via a Dynamixel U2D2 power hub.

MARVIN runs on the Humble distribution of ROS2. MoveIt handles high-level motion planning and translates user commands into low-level Dynamixel controller commands. A custom Unified Robot Description File (URDF) model, which was modified from ROBOTIS's package [4], encodes kinematic, inertial, and control interfaces for both arms.

2.2 Pose Detection

OpenCV interfaces a MediaPipe-based pose detector to extract normalized 3D joint landmarks from a webcam stream [3]. We use shoulder (11,12), elbow (13,14), wrist (15,16), and hip (23,24) landmarks to compute our desired joint angles. Figure 1 displays the landmark labelling scheme.

For each side (left and right), let S, E, W, H, H_{opp} be the 3D position vectors of the shoulder, elbow, wrist, same-side hip, and opposite hip, respectively. Then define:

$$\mathbf{f} = E - W, \quad \mathbf{u} = E - S, \quad \mathbf{v} = H - S, \quad \mathbf{h} = H_{opp} - H, \quad \hat{\mathbf{h}} = \frac{\mathbf{h}}{\|\mathbf{h}\|} \tag{1}$$

2.2.1 Shoulder Flexion/Extension. To determine the anatomical shoulder flexion/extension angle, we start by projecting the upper arm and torso vectors $\mathbf{u}$ and $\mathbf{v}$ onto the shifted anatomical sagittal plane orthogonal to $\hat{\mathbf{h}}$:

$$\mathbf{u}_\pi = \mathbf{u} - (\mathbf{u} \cdot \hat{\mathbf{h}})\,\hat{\mathbf{h}}, \quad \mathbf{v}_\pi = \mathbf{v} - (\mathbf{v} \cdot \hat{\mathbf{h}})\,\hat{\mathbf{h}}. \tag{2}$$

which estimates Shoulder Flexion as:

$$\alpha = \arccos\left(\frac{\mathbf{u}_\pi \cdot \mathbf{v}_\pi}{\|\mathbf{u}_\pi\|\,\|\mathbf{v}_\pi\|}\right). \tag{3}$$

We map α to the first joint J_1 of each OpenManipulator kinematic chain.

2.2.2 Shoulder Abduction/Adduction. Using the original and projected upper arm vectors $\mathbf{u}$ and $\mathbf{u}_\pi$, we estimate the ab/adduction magnitude as:

$$\beta = \arccos\left(\frac{\mathbf{u} \cdot \mathbf{u}_\pi}{\|\mathbf{u}\|\,\|\mathbf{u}_\pi\|}\right). \tag{4}$$

The direction of β can be obtained from sign $\left((\mathbf{u} \times \mathbf{u}_\pi) \cdot \hat{\mathbf{h}}\right)$ which is positive for abduction and negative for adduction. We map this to the second joint J_2 of each OpenManipulator kinematic chain.

2.2.3 Elbow Flexion. Using upper arm and forearm vectors $\mathbf{u}$ and $\mathbf{f}$, we estimate elbow flexion as:

$$\theta = \arccos\left(\frac{\mathbf{u} \cdot \mathbf{f}}{\|\mathbf{u}\|\,\|\mathbf{f}\|}\right). \tag{5}$$

and map θ to the third joint J_3 of each kinematic chain.

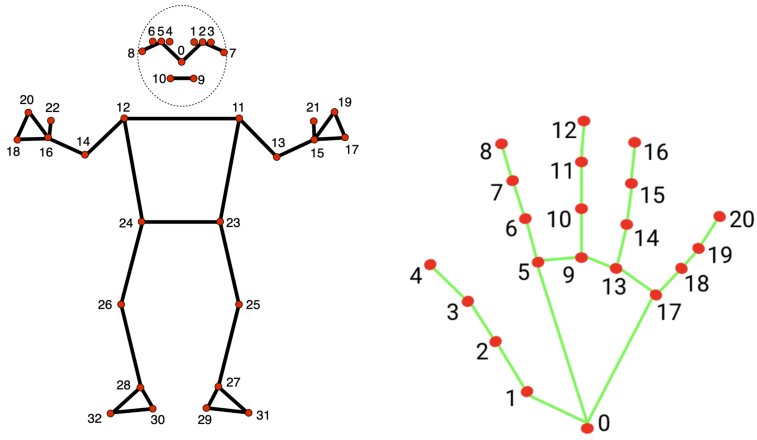

Fig. 1. Comparison of MediaPipe landmark models. (a) Pose Landmarks [5]: the model tracks 33 landmark locations representing the approximate positions of labeled body parts. (b) Hand Landmarks [1]: the model detects 21 hand-knuckle coordinates within the detected hand regions.

2.2.4 Hand Open/Close. Using the Hand Landmarker (labels shown in Figure 1), define a reference length from wrist (0) to middle-finger MCP (9). We mark a hand as open if fewer than three fingertips among indices {4,8,12,16,20} lie closer to the wrist than the reference. We then publish binary open/closed messages to drive the gripper.

2.3 Control

We use MoveIt real-time servoing with one independent kinematic chain per arm. Each chain is associated with a servoing node that subscribes to desired joint velocities. These velocities are computed from the error between target (mirrored) joint angles and current joint states from the `/joint_states` topic, before being scaled by gains and sent as velocity `JointJog` commands. With a previous incident of MARVIN damaging itself during testing [2], we implemented several safety measures. We enforce velocity, position, and current limits at the controller, while MoveIt predicts and prevents self-collisions via planning scene simulations and pre-computed collision matrices. The ROS2 node architecture can be found at MARVIN.ee.cooper.edu/assets/rosgraph.png.

2.4 Website

The website functions as the browser-based front end of MARVIN's remote teleoperation system. It captures the user's webcam stream, performs local pose and hand landmark inference, and transmits the key pose landmarks, as well as the boolean status of each hand, directly to MARVIN. The website also embeds a live video stream of the robot, allowing visual feedback during operation.

Implemented entirely in client-side JavaScript using the MediaPipe Tasks API, the system requires no native installation or external dependencies. Figure 2 shows a screenshot of the web interface. A compact control panel enables users to start or stop the camera, select inference model complexity, and toggle streaming to the ROSBridge server, which exposes `pose_landmarks` and `hand_landmarks` topics for downstream motion control.

The connection to MARVIN is made via ROSBridge over WebSocket, allowing seamless communication between the browser and the ROS2 backend. The communication pipeline is illustrated at MARVIN.ee.cooper.edu/assets/communication.png

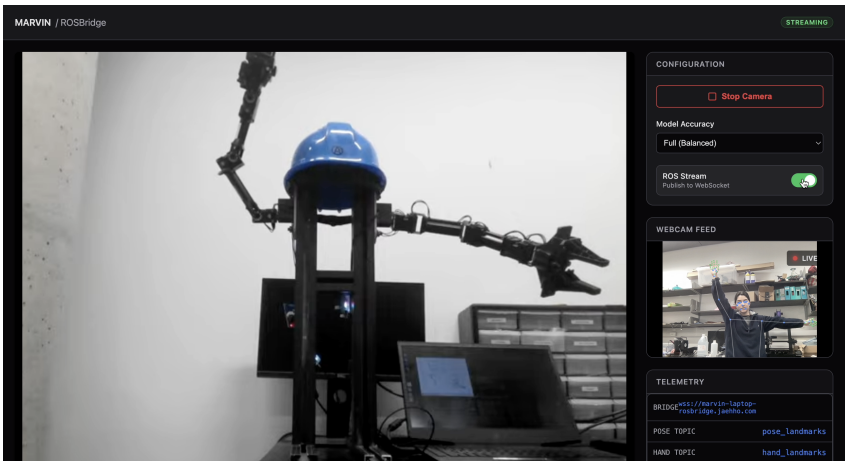

Fig. 2. Screenshot of MARVIN website: livestream of MARVIN on left, webcam feed and landmark overlay to the right. Connection information below MARVIN livestream and controls above webcam feed.

3 RESULTS AND DISCUSSION

MARVIN was tested in local operation during the Cooper Union End of Year Show of May 2025. The robot mirrored onlooker movements in real time, and we received qualitative feedback requesting the addition of hand gestures for a more natural feeling. This informed subsequent implementation of the hand landmark module and gripper control.

We additionally tested remote operation of MARVIN with peer volunteers from the Cooper Union community. Participants controlled MARVIN from a separate room via the website. Despite latency optimization attempts, there is noticeable lag between operator movement and livestream video, with an average round-trip time of *3* s. The delay includes the the websocket transmission time, processing time on the ROS2 side, and return Youtube streaming lag.

MARVIN demonstrates that fully client-side inference is viable for dual-arm mirroring. Remaining gaps include wrist pronation/supination estimation, depth ambiguity in monocular input, and gripper force control. Future work includes (1) integrating depth perception, (2) incorporating joint configurations and kinematic structures more closely aligned with human anatomy, (3) self-calibration, and (4) extending to mobile bases for extended telepresence.

We recognize that telepresence expands access but raises safety and privacy concerns. As all image processing happens client-side, we transmit only necessary landmark data that does not include any identifying user information, including the webcam feed.

We created a remote teleoperation system that minimizes user setup to provide intuitive, low-latency control of a dual-arm robot. All code and hardware designs are open-sourced at https://github.com/jaehho/marvin.

ACKNOWLEDGMENTS

Many thanks go to our advisor, Prof. Mili Shah, for her continued guidance and support. We also credit previous students who worked on MARVIN, Aaron Schmitz, Rose Gebhardt, and Do Hyung (Dave) Kwon. Finally, we thank the Cooper Union community as a whole for volunteering and supporting facilities.

REFERENCES

[1] Google AI for Developers [n. d.]. *Hand Landmarks Detection Guide | Google AI Edge*. Google AI for Developers. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker
[2] [n. d.]. HRI 2021 - An Affordable, Accessible Human Motion Controlled Interactive Robot and Simulation. https://www.youtube.com/watch?v=zCMmiuUJx5M
[3] [n. d.]. *MediaPipe Solutions guide | Google AI Edge*. https://ai.google.dev/edge/mediapipe/solutions/guide
[4] Robotis [n. d.]. *Open Manipulator*. Robotis. https://github.com/ROBOTIS-GIT/open_manipulator
[5] [n. d.]. *Pose Landmark Detection Guide | Google AI Edge | Google AI for Developers*. https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

Received NA; revised NA; accepted NA