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 shoulder flexion, shoulder abduction/adduction, and elbow flexion, 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 presence, interaction, and accessibility, MARVIN allows tangible rituals to emerge out of traditional confinement and into a process of technological mediation that reconnects individuals to the physical world and to one another.

2 HARDWARE

MARVIN uses components from the ROBOTIS OpenManipulator-X platform. Each arm has 5 DOF 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.

3 SOFTWARE

MARVIN runs on the Humble distribution of ROS2. High-level motion is handled by MoveIt, which translates user commands into low-level commands sent to Dynamixel controllers. A custom URDF/Xacro model, which was modified from ROBOTIS's package [3], encodes simultaneous kinematics, inertial, and control interfaces for both arms.

3.1 Pose Detection

OpenCV interfaces a MediaPipe-based pose detector to extract normalized 3D joint landmarks from a webcam stream [2]. We use shoulder (11,12), elbow (13,14), wrist (15,16), and hip (23,24) landmarks to compute our desired joint angles. Figure 1a displays the landmark labelling scheme.

3.1.1 Shoulder Flexion. Let S (shoulder), E (elbow), H (same-side hip), H_{opp} (opposite hip).

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

Project onto the midline plane orthogonal to $\hat{\mathbf{n}}$: $\mathbf{u}_\pi = \mathbf{u} - (\mathbf{u} \cdot \hat{\mathbf{n}})\,\hat{\mathbf{n}}, \quad \mathbf{v}_\pi = \mathbf{v} - (\mathbf{v} \cdot \hat{\mathbf{n}})\,\hat{\mathbf{n}}$.
Flexion:

$$\alpha = \arccos\left(\frac{\mathbf{u}_\pi \cdot \mathbf{v}_\pi}{\|\mathbf{u}_\pi\|\,\|\mathbf{v}_\pi\|}\right). \tag{1}$$

Map α to shoulder flexion/extension joint (J_1).

3.1.2 Shoulder Adduction/Abduction. Let $\mathbf{u}$, $\hat{\mathbf{n}}$ be as above. Project $\mathbf{u}$: $\mathbf{u}_\pi = \mathbf{u} - (\mathbf{u} \cdot \hat{\mathbf{n}})\,\hat{\mathbf{n}}$.
Adduction magnitude:

$$\beta = \arccos\left(\frac{\mathbf{u} \cdot \mathbf{u}_\pi}{\|\mathbf{u}\|\,\|\mathbf{u}_\pi\|}\right). \tag{2}$$

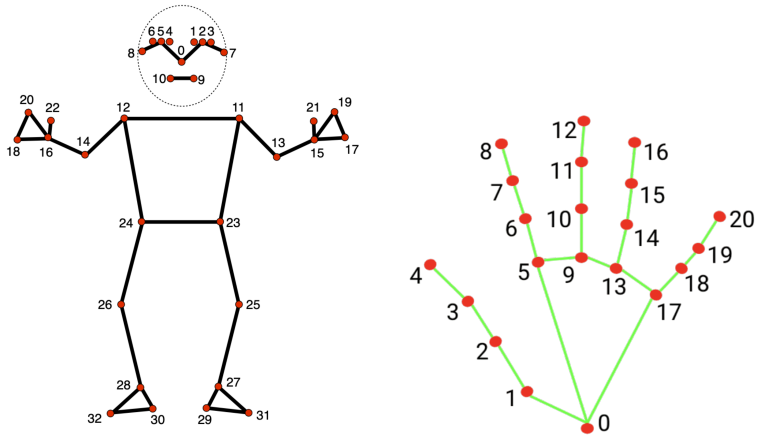
Signed ab/adduction can be obtained from sign $\big((\mathbf{u} \times \mathbf{u}_\pi) \cdot \hat{\mathbf{n}}\big)$. Map to J_2.

3.1.3 Elbow Flexion. Let $\mathbf{u}$ be as above and forearm $\mathbf{f} = E - W$. Then

$$\theta = \arccos\left(\frac{\mathbf{u} \cdot \mathbf{f}}{\|\mathbf{u}\|\,\|\mathbf{f}\|}\right). \tag{3}$$

Map θ to elbow joint J_3.

3.1.4 Hand Open/Close. Using the Hand Landmarker (labels shown in Figure 1b), 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.



(a) MediaPipe Pose Landmarks [4]. The model tracks 33 landmark locations, representing the approximate location of the labeled body parts.

(b) MediaPipe Hand Landmarks [1]. The detects the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions.

Fig. 1. Comparison of MediaPipe landmark models: (a) Pose and (b) Hand.

3.2 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, 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.

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 previously mentioned key pose landmarks directly to MARVIN, while the hand landmarks are processed locally asforementioned and the boolean status of each hand is then sent 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. Figure 3 illustrates the communication pipeline from user to MARVIN.

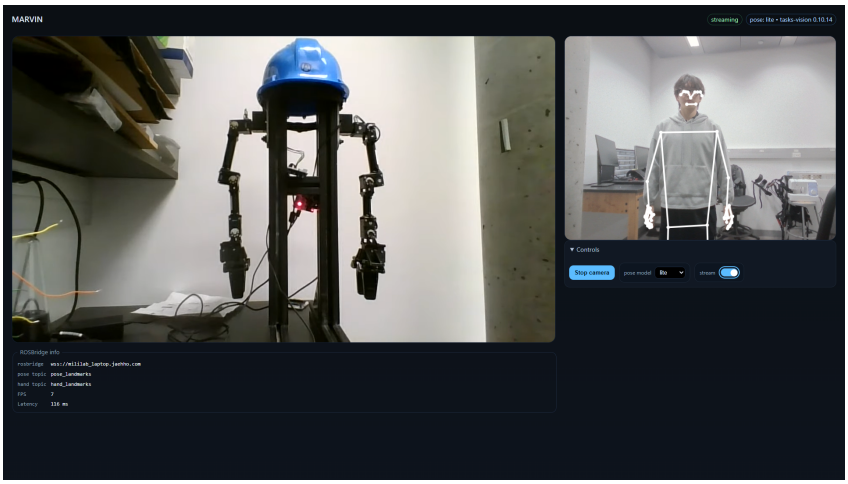


Fig. 2. Screenshot of MARVIN website: livestream of MARVIN on left, webcam feed and landmark overlay to the right. Connection information below MARVIN livestream and controls below webcam feed.

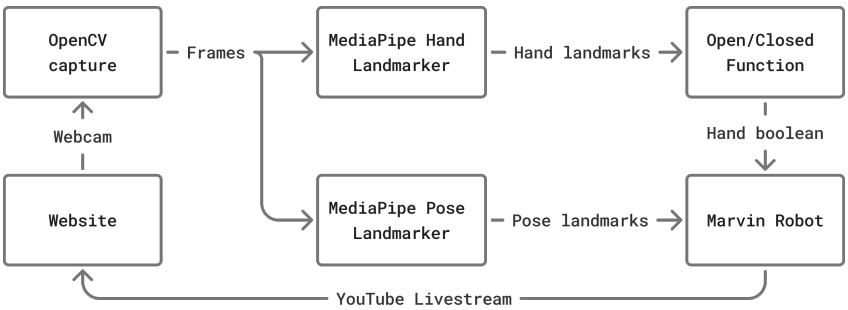


Fig. 3. Communication pipeline from user to MARVIN.

5 OPERATION AND USER STUDY

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, while the robot was stationed in the lab. Participants were interacted with the website and controlled MARVIN from a separate room. Despite attempts to optimize low latency communication, there is a noticeable lag between operator movement and live stream video feedback, with an average round-trip time of *3* s. The delay includes the the websocket transmission time, processinng time on the ROS2 side, and return youtube streaming lag.

6 DISCUSSION

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.

7 CONCLUSION

We provided a remote teleoperation system that minimizes user setup while retaining 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 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.]. *MediaPipe Solutions guide | Google AI Edge*. https://ai.google.dev/edge/mediapipe/solutions/guide
[3] Robotis [n. d.]. *Open Manipulator*. Robotis. https://github.com/ROBOTIS-GIT/open_manipulator
[4] [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