Gesture-Controlled Robotic Arm Manipulator

Dylan Joseph Cadigan, Wentworth Institute of Technology

Dylan Cadigan is currently pursuing a Master of Science degree in Electrical and Computer Engineering at the Douglas D. Schumann School of Engineering, Wentworth Institute of Technology, Boston, MA. He received his Bachelor of Science in Electrical Engineering from the same institution. His research interests include robotics and automation systems, control theory, sensor fusion, computer vision, machine learning, and artificial intelligence. He is a member of IEEE, IEEE-HKN, and Tau Beta Pi.

Robert Nguyen
Dr. Tahmid Latif, Wentworth Institute of Technology

Tahmid Latif is an assistant professor of electrical and computer engineering at the School of Engineering of Wentworth Institute of Technology, Boston, MA. He received his doctorate in electrical engineering from North Carolina State University, Raleigh, NC. His research interests lie at the intersection of electronics and biology, with a focus on bioelectronics and instrumentation, biohybrid insect robots, insect-machine interfaces, and educational robots. He is a member of IEEE and Sigma Xi.

©American Society for Engineering Education, 2026

Gesture-Controlled Robotic Arm Manipulator

Abstract

Robotic arm manipulators have traditionally been controlled using heavy and complicated teach pendants, scripted sequences, or pre-programmed routines. While these methods have been industry standards, they may often fail to provide intuitive or natural control mechanisms, despite the objective of replicating the dexterity and fluidity of the human arm. This work explored a novel control methodology that integrated computer vision-based gesture recognition with wearable inertial measurement units (IMUs) to enable a more natural, human-like motion of a robotic arm. By leveraging advances in machine learning and sensor fusion, the system translated user gestures and arm movements into precise commands for the manipulator.

This approach enhanced the dexterity of the arm, simplified the programming of repetitive tasks, and facilitated rapid adaptation to sudden environmental changes. The integration of real-time computer vision for dynamic adjustments, along with wearable sensors that track precise motion changes, ensured seamless interaction between the user and the robotic system. The results demonstrated an approach that augmented robotic manipulation by making it more adaptable, user-friendly, cost-effective, and efficient across various fields and education.

Introduction

Modern robotic arm manipulators are often designed to mimic human arms but usually lack intuitive and organic control mechanisms. Two common methods for controlling a robotic arm include teach pendants and hard coding. Unfortunately, both solutions have significant drawbacks. Teach Pendants, usually in the form of a handheld device, are often heavy (3-6 lbs.), causing excessive strain and fatigue on a user's wrist after several minutes of use. They are also expensive with basic low-end pendants starting around $1000 USD. Hard coding manipulators causes slower development due to the constant task of programming and reprogramming written instructions. This, in turn, also makes live testing, troubleshooting code, and implementing in educational settings more difficult. With tons of robotic arms on the market, each with their own native language, users must know a variety of programming protocols in order to effectively program the arm. These existing control methods may create barriers and steep learning curves due to complexity, cost, ergonomics, and lack of adaptability. There is a significant need for a more user-friendly and adaptive control method that can mitigate all of the aforementioned shortcomings.

Background Study and Literature Review

Recent scholarly work on robotic arm manipulators reveals a variety of strategies and technologies used in their development, particularly focusing on the enhancement of their control systems by means of wearable sensors and computer vision. Among the most commonly used computer vision tools are OpenMV and OpenCV. OpenMV is preferred for simpler, lightweight applications using camera modules and MicroPython, while OpenCV is a more powerful option

designed for use with systems that have a dedicated operating system and processing power [1], [2]. Combining either of these tools with an analysis algorithm enables robotic arms to perceive their environment and perform tasks with greater autonomy and precision [3], [4].

Wearable sensors also play a key role in the control of robotic arm manipulators, especially in applications requiring precise or intuitive human input. Placed on the user's body, typically near the arms, hands, or fingers, these sensors capture motion data like orientation and muscle activity, which is then translated into real-time arm movement. Common types of sensors include inertial measurement units (IMUs) and flex sensors that vary electrical resistance depending on the degree they are bent at [5]. Both sensors offer varying levels of sensitivity and responsiveness. This technology enables more natural human-robot interaction, particularly useful in assistive tech and precision-based tasks. As sensor technology evolves, wearable-controlled systems are expected to become even more responsive and accessible [6], [7], [8].

In terms of complete movement control, there's a big divide that exists in design approaches. Some robotic arms rely entirely on computer vision to interpret and respond to visual cues, while others are operated through wearable sensors that capture human motion to guide the robotic limb [9], [10], [11], [12], [13]. Interestingly, only a small number of prototypes combine both methods, and even fewer incorporate alternative control techniques such as eye tracking, stationary sensors, or EKG monitoring, which suggests a space for further exploration and innovation in multi-channel control systems [14], [15].

From a manufacturing standpoint, 3D printing has emerged as a crucial technology in the prototyping phase of robotic arm development. It offers a cost-effective and time-efficient means of producing custom parts, allowing for rapid iteration and customization [12]. Most robotic arm prototypes are powered by servo or stepper motors, with control systems often built using Arduino platforms, ROS (Robot Operating System), or integrated with OpenCV for real-time vision-based control [1], [16], [17].

The motivations behind these projects are often rooted in enhancing accessibility, especially for people with physical impairments, automating everyday tasks, and pushing the boundaries of how humans and robots can interact in shared spaces [18], [19]. These efforts reflect a broader interest in making robotics more adaptable and user-centered, hinting at significant future developments in assistive technologies and human-robot collaboration [20], [1]. Beyond accessibility considerations, education-focused robotics has experienced significant growth over the past decade. Research indicates that integrating robotics into K–16 classrooms has expanded particularly within STEM education, where robot-assisted learning environments are associated with moderate positive effects on students' academic performance and attitudes compared to traditional instruction. A recent multilevel meta-analysis of studies published between 2010 and 2022 reported that educational robotics had a notable positive impact on STEM learning outcomes, especially in measures of learning performance and student attitudes [21], [22]. Global scholarly output on robotics in education has steadily increased over the same period, reflecting not only a rise in publication volume but also an expanding research focus

encompassing early childhood through higher education, interdisciplinary learning, creativity, and computational thinking [23]. Notably, some researchers have focused on the development of low-cost (<$200) educational robotics kits to improve accessibility and allow more students to engage with complex systems [24], [25].

These trends mirror broader shifts in educational practice toward interactive, hands-on learning that engages students more deeply than lecture-based instruction alone. This expansion highlights the need for more intuitive control interfaces, aligning with the core philosophy of this project.

Existing Solutions

To enable more intuitive operation of robotic arm manipulators, researchers have focused on three primary approaches, each presenting distinct strengths and limitations.

The first method relies solely on computer vision, offering a low-cost option that typically requires only a standard webcam and no additional hardware. Despite its affordability, this technique is vulnerable to occlusion and reduced detection reliability when visual confidence declines. Furthermore, restricted viewing angles can hinder comprehensive motion tracking, often necessitating dual-camera (stereo) configurations to achieve adequate coverage.

The second approach depends exclusively on sensor-based tracking, most commonly through IMUs. This method delivers reliable initial performance without the use of cameras and remains relatively inexpensive. However, accumulated drift over extended use reduces precision, making periodic recalibration necessary. In addition, capturing complex movements with a single unit is impractical, and incorporating multiple devices to enhance fidelity raises system cost.

The third strategy employs a scaled-down physical replica outfitted with potentiometers or encoders to measure joint movement. This configuration provides high accuracy and a natural user experience, particularly during direct manipulation. Nevertheless, implementation requires additional hardware, including a secondary model, supplemental components, and extensive wiring, leading to higher expenses. Long-term durability also poses challenges, as repeated physical interaction can result in mechanical degradation.

Proposed Solution

This work, developed as part of a two-semester ECE senior design capstone, proposed a hybrid control system for robotic arm manipulators that integrates wearable motion sensors with computer vision. By leveraging the strengths of both approaches, this system aimed to improve precision, responsiveness, and overall usability. Such a combination remains relatively unexplored in current engineering research, offering opportunities for advancements in teleoperation and accessibility. The hybrid system enables users to control robotic arms remotely, reducing the risk of injury, illness, or fatality in hazardous environments. This functionality is particularly valuable for tasks including maintenance, experiments, or inspections in areas unsafe for human presence. Additionally, the system could assist individuals with

limited arm mobility due to medical conditions, allowing them to manipulate objects and perform tasks without physical strain. These capabilities were made possible by gesture-based control, which does not require direct physical contact with the robotic arm or the exertion of lifting weight by the user, enhancing both safety and accessibility.

Project Constraints and Limitations

To simplify the overall approach and implementation, two key design constraints were adopted in this project.

The first constraint concerns the physical construction of the robotic arm manipulator. While the human arm has seven degrees of freedom (DOF), most robotic manipulators are designed with six DOF, which is the minimum required to fully define the position and orientation of a rigid body in three-dimensional space. In this work, the arm was constructed with five DOF, thereby reducing redundancy between shoulder and wrist yaw (horizontal rotation). Rather than employing separate actuators for both joints, the wrist yaw actuator was omitted in favor of a stronger and more capable shoulder yaw mechanism. This design choice decreases mass near the end effector and marginally simplifies the control strategy.

The second constraint relates to the vision system used during experimental evaluation. The built-in laptop webcam was unable to provide reliable depth information, which is essential for accurate spatial perception. This limitation stems from the absence of an infrared (IR) sensor, a feature commonly included in modern devices to generate depth maps for applications such as facial recognition. To compensate for this hardware constraint, a low-cost ultrasonic proximity sensor was integrated to supplement the webcam with depth measurements.

Methodology and Approach

The overarching goal of the project was to create an effective control program for a robotic arm manipulator based on a user's gestures. In theory, the robot's movements should match the user's movements with minimal delay between the two.

System Architecture

The system was comprised of four primary subsystems: the robotic arm, the wearable sensor module, the signal processing system, and the power distribution subsystem (Figure 1).

The robotic arm subsystem contained a microcontroller (Arduino Uno) that controlled stepper and servo motors responsible for joint movement. It also included the robotic manipulator and the corresponding motor drivers for the stepper motors. The microcontroller received data from the signal processing system via a wired serial connection and actuated the arm to the specified coordinates.

The signal processing subsystem performed two main functions. First, it determined the position of the user's hand using computer vision for horizontal and vertical tracking, and an ultrasonic sensor for depth measurement. Second, it synchronized the positional data with input from the

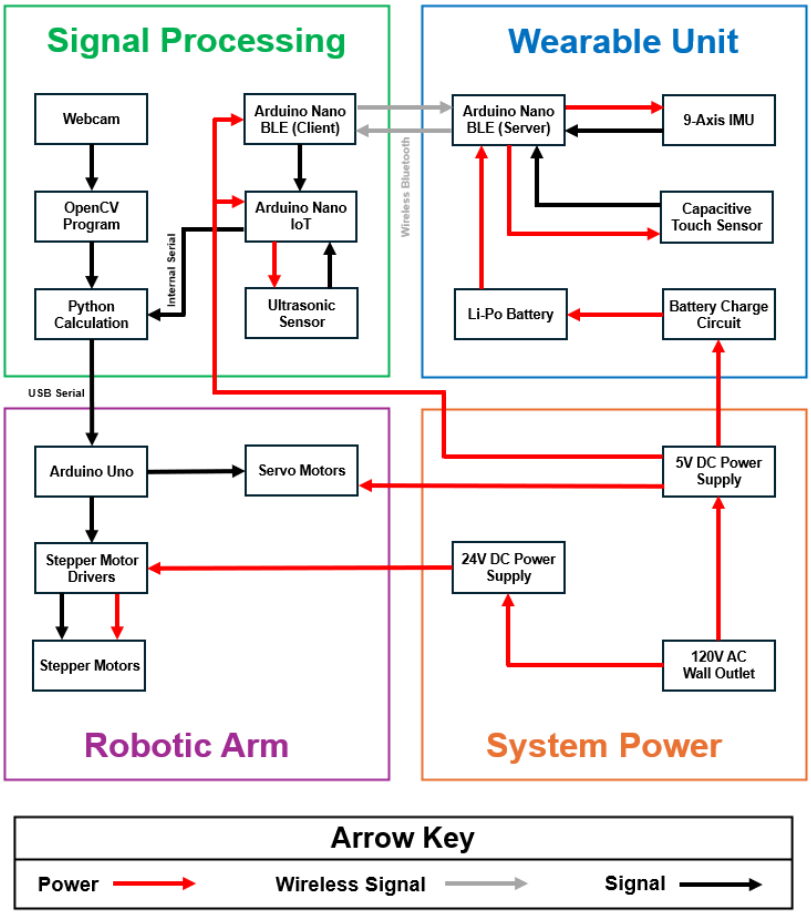

Figure 1. The system-level block diagram for the design.

wearable sensor module to generate a single, coordinated data packet, which was transmitted to the robotic arm. This synchronization minimized timing errors and ensured smooth, coordinated arm movement.

The wearable sensor subsystem included a dedicated microcontroller (Arduino Nano 33 BLE) with a built-in 9-axis IMU. It communicated wirelessly with the signal processing system via Bluetooth Low Energy (BLE). A touch sensor detected when the user intends to close the robotic claw (the end-effector). Sensor fusion was performed using a Madgwick filter, allowing computation of the user's hand orientation (roll, pitch, and yaw). The orientation data and touch sensor state were transmitted to the robotic arm for execution.

System Setup

In an ideal configuration, the system would consist of two main setups, as illustrated in Figure 2.

The Remote Setup would be where the robotic arm and its primary controller (the Arduino Uno) reside. A downward-facing camera (Camera 1) could be used to monitor the arm's movements, providing a live feed to the user at the operation site.

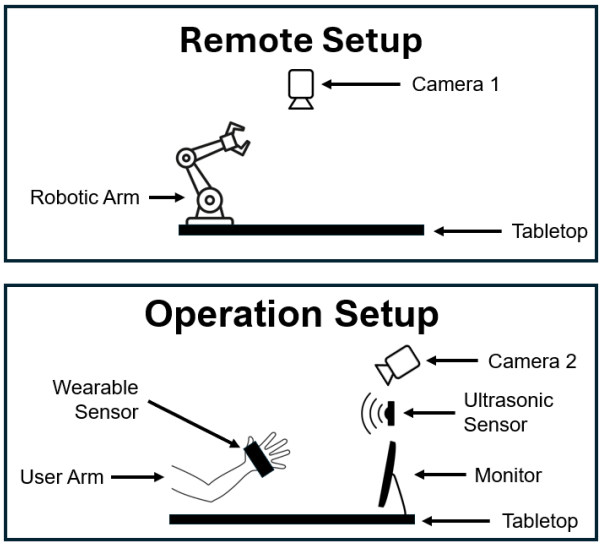

Figure 2. The system setup for the design.

In the Operation Setup, the user would wear the motion-sensing module while being monitored by the computer vision system through a secondary camera (Camera 2), which in this case is the laptop's webcam. Camera 2 would be positioned at approximately a 45-degree angle to optimize gesture tracking. A monitor could display the feed from Camera 1, which would allow the user to observe the robotic arm's movements in real time. Additionally, an ultrasonic sensor was used to measure the distance between the user and the screen, facilitating depth-based adjustments in the control system.

Hardware Design

The Arduino Uno was selected as the primary microcontroller for the robotic arm due to its low cost, wide availability, and robust GPIO support. Its input/output capabilities were well-suited for controlling servo motors, stepper motor drivers, and additional feedback sensors. The board also had low power consumption and could operate entirely from a USB connection, simplifying system requirements. The wearable sensor module was based on the Arduino Nano 33 BLE, a compact microcontroller with integrated BLE for efficient wireless communication and reduced power consumption, making it ideal for battery-powered operation. The board included a 9-axis IMU for accurate orientation tracking and provided sufficient GPIO to interface with a TTP223 capacitive touch sensor, which served as a reliable solid-state switch for controlling the end-effector mechanism. Since the wearable module communicated via BLE and most laptops lack native BLE support, an Arduino Nano 33 IoT was used as an intermediary device. This board was chosen for its compatibility with the Nano 33 BLE programming environment and lower cost. It functioned solely as a data relay for the wearable module and ultrasonic sensor. The HC-SR04 ultrasonic sensor was chosen for its simplicity and reliability, requiring no external libraries. Connecting it to the Nano 33 IoT allowed for simultaneous reception of wearable and distance data, minimizing timing discrepancies.

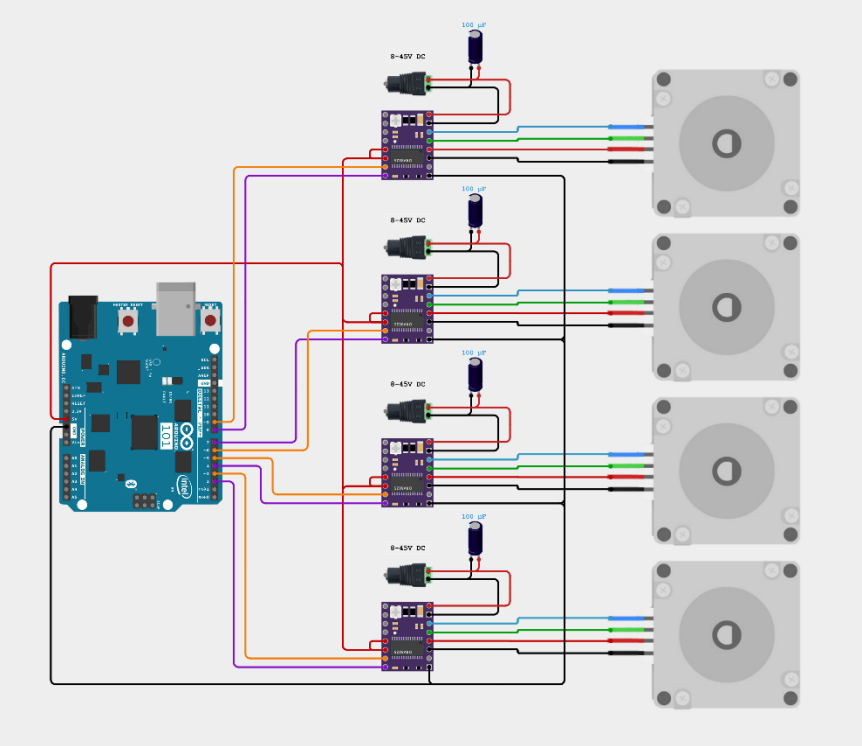

Figure 3. The wiring of the DRV8825 stepper motor drivers.

NEMA 17 stepper motors were chosen for the three lower joints due to their high torque, reliability, and extensive documentation, providing sufficient strength to support the arm and payload. A smaller NEMA 11 stepper motor was chosen for wrist rotation because of its compact form factor and higher torque output compared to a servo motor. DRV8825 motor drivers were utilized to drive the stepper motors due to their compact size, Arduino compatibility, and ability to meet the power requirements of the NEMA motors (Figure 3). Each driver was powered by an external 24V DC supply, as the motors required approximately 1.6 A of current. For the wrist and end effector, MG995 servo motors were used. Their light weight and power requirements made them suitable for low-torque joints. The end effector was a simple claw mechanism, programmed to stop applying force upon detecting resistance to prevent mechanical strain.

Software Design

The positioning algorithm for the computer vision system ran on a laptop, which was chosen because of its built-in webcam and superior processing power compared to microcontrollers. This setup ensured better performance for both computer vision tasks and data forwarding. Although the control system was compatible with any device capable of running Python 3.12, a laptop offered the best balance between portability and performance.

The computer vision component was implemented using OpenCV and MediaPipe. Since the program ran in Python, threading was used to allow the vision system and the routing code to execute concurrently. Communication with the Arduino Uno and the Arduino Nano 33 IoT was

handled via serial connections. This allowed the laptop to combine data from the computer vision system with sensor data received from the IoT devices and forward it to the Uno. The Arduino Nano 33 IoT communicated via USB serial and primarily handled the BLE data. It merged this data with readings from an ultrasonic sensor and sent the combined information to the Python script on the laptop for further processing.

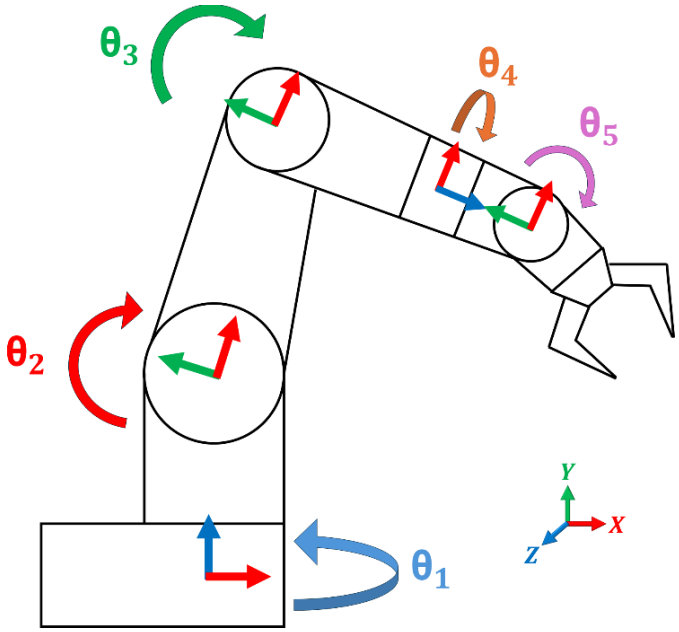

Figure 4. A simplified drawing of the robotic arm manipulator with joint frames labeled for clarity. The reference frame {0} and frame {1} are aligned.

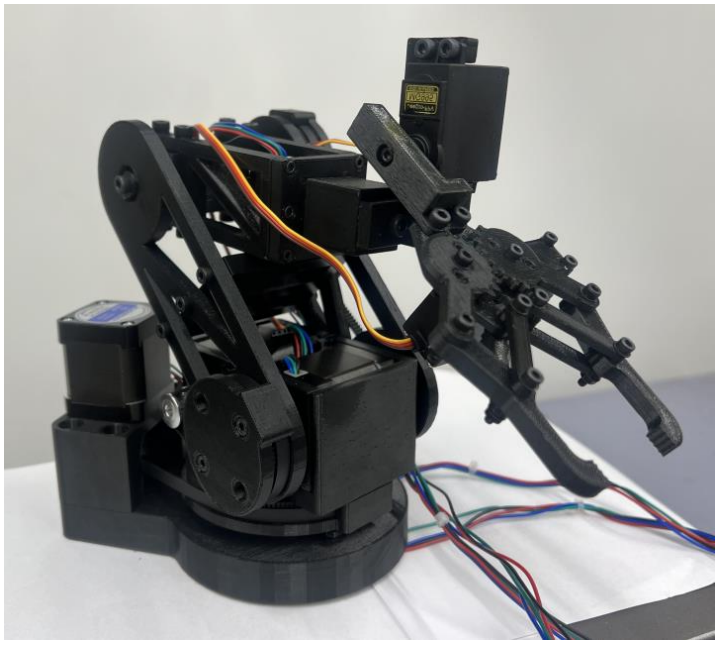

Figure 5. The fully assembled robotic arm manipulator.

Physical Design of the Robotic Arm Manipulator

As shown in Figure 4, the manipulator was chosen to have 5 degrees of freedom to reduce redundant actuation, thus simplifying the control program and physical design.

Each joint also corresponded to a specific motion in a human arm:

- Joint 1 (θ_1) replicated horizontal shoulder motion – Horizontal movement from the computer vision.
- Joint 2 (θ_2) replicated vertical shoulder motion – Vertical movement from the computer vision and depth from the proximity sensor.
- Joint 3 (θ_3) replicated vertical elbow motion – Vertical movement from the computer vision and depth from the proximity sensor.
- Joint 4 (θ_4) replicated horizontal wrist motion – IMU roll data from the wearable unit.
- Joint 5 (θ_5) replicated vertical wrist motion – IMU pitch data from the wearable unit.

In order to compensate for the significant weight of the stepper motors, the motors themselves were placed as close to the bottom of their respective joint as possible. This meant that, in order to achieve motion in the joints, belts and pulleys were used. Finally, 3D printing was chosen as the primary fabrication method since it allowed for rapid and cost-effective prototyping (Figure 5). PETG filament was selected for its favorable balance of strength, durability, and thermal resistance compared to other common 3D printing materials.

System Testing

To facilitate easy troubleshooting, each system was tested individually. First, the computer vision system was tested by gathering coordinate data and printing it to the terminal. Then, the IMU and sensor fusion were tested by tilting the unit while observing the Arduino Nano BLE output on the serial monitor. Following that, the BLE connection was tested by having the IoT device print received BLE data to the serial monitor. The routing and data combination code was then verified by running a threaded Python script that collected data from the IoT device and printed the combined output to the terminal.

Each joint of the robotic arm was also tested to ensure smooth, unrestricted movement. After the components were assembled and properly wired, a verification program was uploaded to move the motors back and forth at varying speeds. The resulting motion was documented for future reference. After confirming that both the coordinate system and physical joint movement functioned correctly, the tracked coordinate data was sent to the robotic arm's Arduino Uno via a serial connection. The arm joints were then tested to assess how accurately they responded to the received coordinate data.

Results and Discussion

The results presented were based on system-level proof-of-concept testing and course-based evaluations rather than a controlled human-subjects study and are intended to illustrate feasibility

and educational value rather than establish benchmark performance. Overall, the system operated correctly as it tracked a user's gestures and appropriately matched with minimal delay.

Computer Vision

As shown in Figure 6, the computer vision was capable of recognizing joints like the elbow and wrist. The system performed fairly accurately when tracking movement within the frame. The most critical aspect of this process was the extraction of the joint coordinates. During testing, the X and Y coordinates (horizontal and vertical axes, respectively) proved to be accurate. While the computer vision system function is mostly correct, it is not accurate enough to handle more advanced inputs, such as detecting the intent to open or close the end-effector. As shown in Figure 7, the system could become confused when an object obscures the hand, sometimes hallucinating fingers or failing to detect them entirely due to low confidence levels. This was the main reason behind the implementation of a touch sensor for controlling the opening and closing of the claw, replacing the unreliable finger-tracking approach. Thankfully, extracting coordinates from the computer vision system was successful in both the X and Y directions.

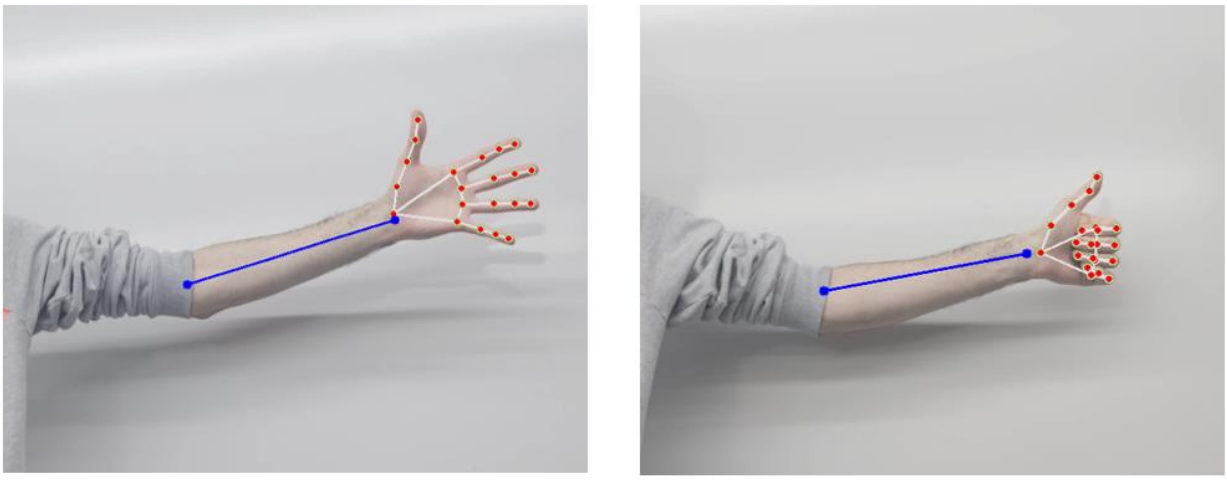

Figure 6. Two instances of the computer vision program properly tracking a user's hand and arm.

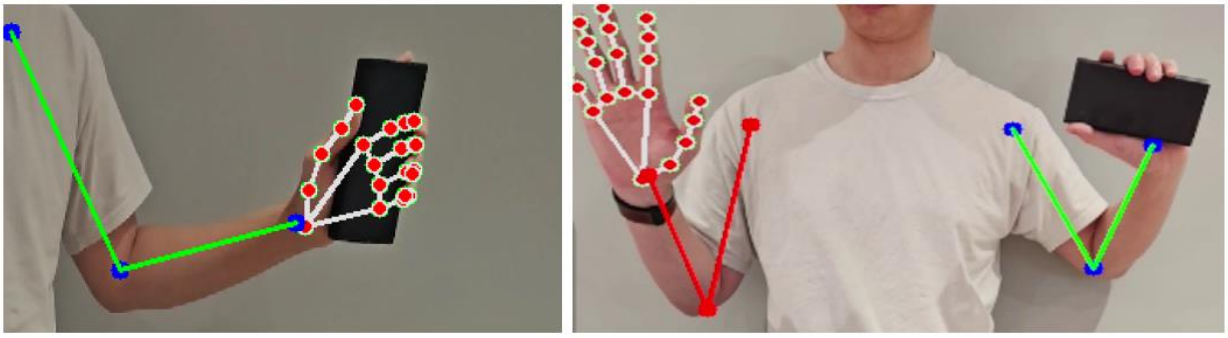

Figure 7. Two instances of the computer vision program unable to track a user's hands properly.

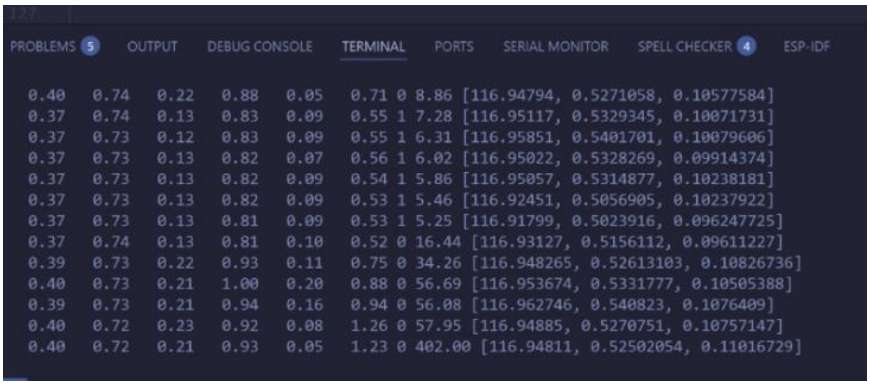

Figure 8. The sensor fusion terminal. From left to right, Wrist X-coordinate, Wrist Y-coordinate, Elbow X-coordinate, Elbow Y-coordinate, Shoulder X-coordinate, Shoulder Y-coordinate, Touch sensor, Ultrasonic, IMU roll, IMU pitch, IMU yaw.

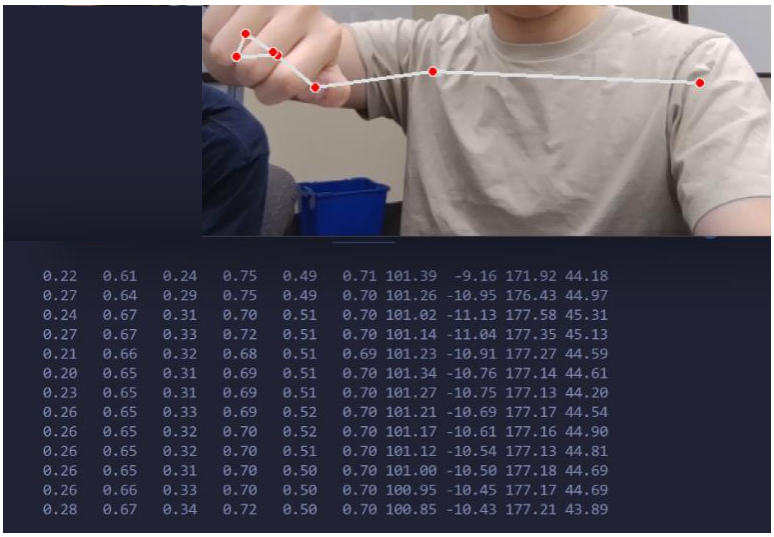

Figure 9. The sensor fusion system with the IMU, computer vision, and ultrasonic sensor data (this image did not include touch sensor).

IMU Data

Rotational data from the IMU (Figure 8) was obtained effectively using a Madgwick filter. Additional sensors, including touch and ultrasonic sensors, were integrated later in the project and functioned as expected (Figure 9). However, combining all pieces of data together proved to be a challenge. The computer vision system and the wearable/ultrasonic system operated at different update rates and could not interact directly without causing timing issues. While delays and timeouts could have been used to manage synchronization, they risked the system falling out of sync over time if not implemented perfectly. To address this, threading was introduced so that both systems could run in parallel. A separate program was created to collect and merge the data. Each system pushed its data into a queue using Python's queue library. When preparing to send data to the Arduino Uno, the program cleared the queue to ensure only the most recent data was used, avoiding outdated information.

Gesture-Control of the Manipulator

To evaluate the effectiveness of the entire system, a combination of quantitative and qualitative assessment methods were employed. Gesture recognition performance was evaluated by analyzing live data readouts and visually inspecting the placement of detected landmarks and skeletal links overlaid on the camera feed. Hand motion demonstrations were conducted under varying lighting conditions, backgrounds, and user orientations to assess robustness and generalizability. During system testing, the vision-based gesture recognition module achieved a high classification and spatial positioning accuracy exceeding 95% in real-time for a predefined set of discrete hand gestures. Minor performance degradation was observed under extreme lighting variations; however, the system remained stable and functional.

IMU tracking accuracy was evaluated by comparing users' intended hand and wrist motions with the corresponding system outputs across each rotational axis. When fused with visual input, IMU data significantly improved motion continuity and control precision, compensating for temporary visual occlusions and reducing jitter. This multimodal fusion reduced end-to-end system latency to under 600 milliseconds and minimized positioning errors during robotic arm operation. As a result, smoother trajectories and more accurate end-effector placement were observed during manipulation tasks.

System responsiveness and task-level performance were assessed using a standardized set of robotic manipulation tasks, including pick-and-place operations and predefined trajectory-following exercises. The system consistently completed tasks with high success rates while maintaining responsiveness under dynamic conditions, such as sudden changes in background, lighting, or user position. These results demonstrated the system's reliability and suitability for real-world deployment.

Beyond technical performance metrics, user feedback was assessed on usability, intuitiveness, and learning curve. Based on informal user feedback and reflective discussion, the gesture-based interface was perceived to be more intuitive than conventional control approaches (Figures 10 and 11). Overall, both objective measurements and subjective evaluations indicate that the proposed system provides accurate, responsive, and user-friendly gesture-based robotic control.

Future Improvements

While the system functioned as intended, there were several design improvements that could enhance its reliability and usability. One notable improvement would be offering the option to use a serial connection on the wearable module as an alternative to Bluetooth, which could help reduce signal interference when needed.

The current ultrasonic sensor was fairly accurate but suffered from a limited field of view and reduced accuracy when detecting objects that are offset. To address this, a second camera could be added to the computer vision system. This setup would require a reference card (a pattern of known size and shape) to calibrate the offset between the cameras. Once configured, this would significantly improve depth perception and overall accuracy.

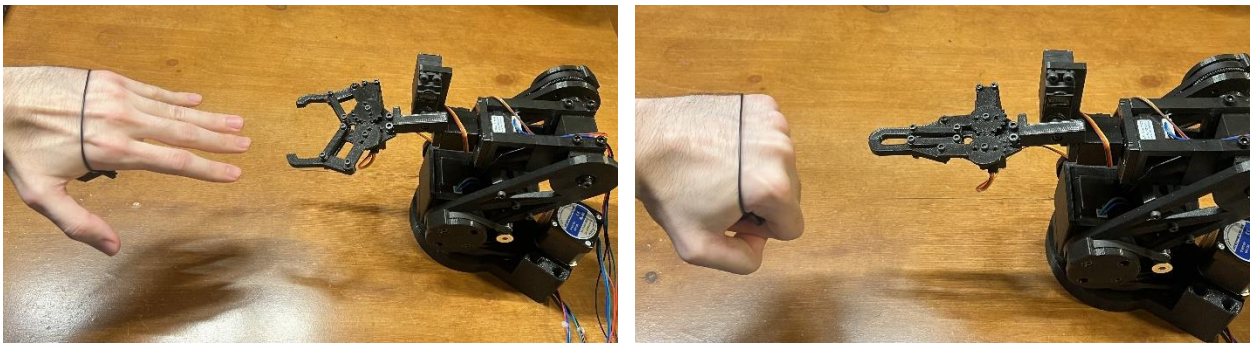

Figure 10. When pressing the button on the wearable module, the end-effector closes.

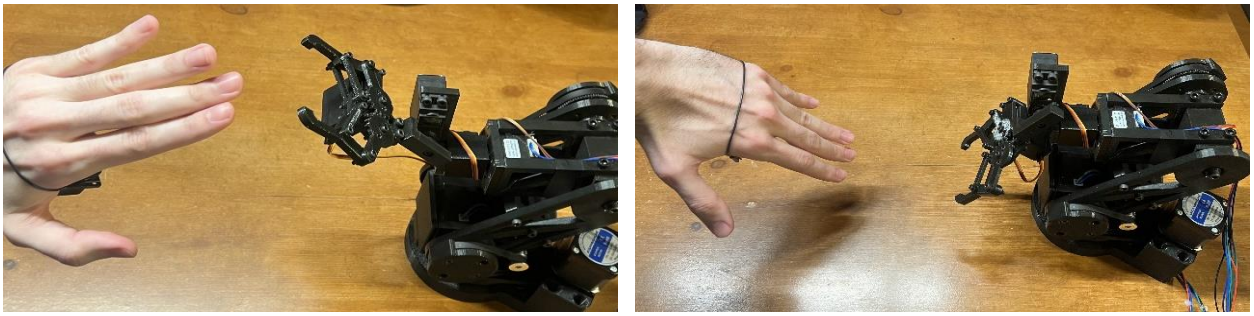

Figure 11. When tilting the wearable module up, the robot's wrist also moves up (left). When tilting the wearable module down, the robot's wrist also moves down (right)

Finally, the system could benefit from improvements to its mechanical design. In its current configuration, there is no mechanism for adjusting the tension of the belts connected to the stepper motors. Over time, these belts are subject to stretching, and insufficient tension may lead to inaccurate or unintended arm motion. Incorporating adjustable belt tensioners would allow proper re-tensioning after wear, eliminating the need to replace the belts entirely less often.

Educational Framework and Assessment

This work was developed by two undergraduate students as part of a two-semester ECE senior design capstone under the direction of an ECE professor. The work reinforced experiential learning through open-ended, interdisciplinary system design, including areas in electrical, computer, robotics, and mechanical engineering. While the system demonstrated effective gesture-based robotic control, the primary contribution of the paper laid in its role as an experiential learning platform within an undergraduate ECE curriculum. Students developed a hybrid robotic arm control system combining computer vision–based gesture recognition with wearable IMUs and an ultrasonic proximity sensor to enable intuitive, human-centered manipulation. The project required integration of robotics, embedded systems, machine learning, control systems, computer vision, and human–robot interaction, while managing real-world constraints such as cost, timelines, and component selection.

Learning Objectives

The learning objectives for this project included: (1) synthesizing knowledge across multiple engineering domains, (2) developing systems-level thinking through hardware–software co-design and sensor fusion, (3) evaluating human-centered metrics such as usability, user experience, responsiveness, and latency, (4) practicing iterative prototyping and debugging, (5) coordinating project schedules, (6) managing budgets through component research and selection, and (7) communicating technical decisions through design reviews and documentation.

The project followed an iterative, design-based structure. Students defined system requirements, selected sensing modalities, implemented real-time data acquisition, and developed a sensor fusion pipeline combining vision, IMU, and ultrasonic data to improve accuracy and reliability. Iterations addressed challenges including gesture misclassification, IMU drift, depth-sensing limitations, and device synchronization. Faculty mentorship focused on guidance rather than prescriptive instruction and fostering student ownership of design decisions.

Educational Assessment

Educational impact was evaluated using direct and indirect methods. Direct assessments included rubric-based design reviews, technical reports, and live demonstrations that focused on interdisciplinary integration, hardware/software justification, and problem-solving. The rubric was developed for consistency in evaluation and feedback across four levels: Exemplary, Satisfactory, Developing, and Unsatisfactory. The design rubric evaluated the technical understanding and design objective, the iterative design process, functionality, innovation, the completion of the design, and the integration of all components and/or sub-systems into the developed system. Additionally, detailed testing methods, experimentation, and analysis and discussion of results were evaluated. Presentation and communication required a technical report, oral and poster presentations, and project demonstrations. Indirect assessments included pre- and post-project self-efficacy reflections on system integration, sensor fusion, and debugging, along with written statements on challenges and lessons learned.

Results indicated meaningful experiential learning outcomes. Students reported increased confidence in integrating hardware and software, managing timelines and budgets, and evaluating system-level trade-offs. Design reviews demonstrated improved ability to justify technical decisions and analyze performance metrics such as latency (<600 ms), gesture recognition accuracy (>95%), and motion reliability. Reflections highlighted the value of iterative prototyping, failure-driven learning, and interdisciplinary collaboration. Achievements included successful device communication, accurate manipulator tracking, and physical creation of a robotic arm despite minimal prior mechanical experience.

Technical findings include: IMUs provided high-resolution motion tracking but were prone to drift; ultrasonic sensors improved depth sensing, making it a cheaper alternative to adding a second camera; and computer vision provided contextual awareness but was limited by obstructions. Combining these methods improved control reliability, responsiveness, and

adaptability, with potential applications in assistive technologies, kinesthetic programming, and remote handling of hazardous materials.

The framework, an open-ended problem definition with iterative development, milestone-based assessment, and reflection, is broadly transferable, demonstrating how technically complex capstone projects can serve as effective platforms for experiential learning in undergraduate engineering education.

Acknowledgments

This work was partially supported by the Wentworth Launch Grant and the Wentworth SoE-KEEN Mini Grant. The authors thank the Douglas D. Schumann School of Engineering at Wentworth Institute of Technology for their logistical support and assistance.

References

[1] J. Paterson and A. Aldabbagh, "Gesture-controlled robotic arm utilizing OpenCV," in 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021. Available: https://ieeexplore.ieee.org/document/9461389. DOI: 10.1109/HORA52670.2021.9461389.

[2] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer vision-based hand gesture recognition for human-robot interaction: a review," Complex & Intelligent Systems, vol. 10, (1), pp. 1581–1606, 2024. Available: https://doi.org/10.1007/s40747-023-01173-6. DOI: 10.1007/s40747-023-01173-6.

[3] J. Wang and H. Peng, "Object grabbing of robotic arm based on OpenMV module positioning," in 2023 2nd International Conference on Artificial Intelligence and Computer Information Technology (AICIT), 2023. Available: https://ieeexplore.ieee.org/document/10277796. DOI: 10.1109/AICIT59054.2023.10277796.

[4] H. Zhang and L. Sheng, "Motion capture and analysis algorithm of drama stage performance based on computer vision and deep learning," in 2024 3rd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS), 2024. DOI: 10.1109/AIARS63200.2024.00107.

[5] S. R. Deb and S. Deb, "Robot kinematics and dynamics," in Robotics Technology and Flexible Automation, 2nd Edition ed. Anonymous, 2010. Available: https://www.accessengineeringlibrary.com/content/book/9780070077911/chapter/chapter2.

[6] M. Çoban and G. Gelen, "Wireless teleoperation of an industrial robot by using myo arm band," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018. Available: https://ieeexplore.ieee.org/document/8620789. DOI: 10.1109/IDAP.2018.8620789.

[7] L. Tao and D. Hongwang, "A camera-IMU system extrinsic parameter calibration method," in 2017, DOI: 10.1109/ITNEC.2017.8284902.

[8] D. Xu, Z. Wang and X. Zhu, "Real-time human motion capture based on BiRNN and inertial sensor," in 2023, DOI: 10.1109/ICMA57826.2023.10215889.

[9] M. A. Cabrera et al, "OmniCharger: CNN-Based Hand Gesture Interface to Operate an Electric Car Charging Robot through Teleconference," vol. 13, (3), 2024. Available: https://doi.org/10.1145/3659060. DOI: 10.1145/3659060.

[10] T.-C. Chang, Y.-C. Wu, C.-C. Han, and C.-S. Chang, "Real-time arm motion tracking and hand gesture recognition based on a single inertial measurement unit," in Proc. 11th Int. Conf. Internet of Things: Systems, Management and Security (IOTSMS), Malmö, Sweden, Sept. 2024, doi: 10.1109/IOTSMS62296.2024.10710280.

[11] H. -H. Jung, M. -K. Kim and J. Lyou, "Realization of a hybrid human motion capture system," in 2017, DOI: 10.23919/ICCAS.2017.8204238.

[12] S. Kurundkar, K Kshirsagar, V. Kulkarni, H. Lambat, and M. Lambe, "3D printed robotic arm using hand gestures," in 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2023. Available: https://ieeexplore.ieee.org/document/10290643. DOI: 10.1109/I-SMAC58438.2023.10290643.

[13] X. Zhang and X. Wu, "Robotic control of dynamic and static gesture recognition," in 2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), 2019. Available: https://ieeexplore.ieee.org/document/9004188. DOI: 10.1109/WCMEIM48965.2019.00100.

[14] X. Fan, "Fine motion capture algorithm based on virtual reality and its application in sports training research," in 2024, DOI: 10.1109/IPEC61310.2024.00032.

[15] Y. -S. L. -K. Cio, M. Raison, C. L. Ménard, and S. Achiche, "Proof of Concept of an Assistive Robotic Arm Control Using Artificial Stereovision and Eye-Tracking," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 27, (12), pp. 2344–2352, 2019. Available: https://ieeexplore.ieee.org/document/8887462. DOI: 10.1109/TNSRE.2019.2950619.

[16] P. D. S. H. Gunawardane, N. T. Medagedara, B. G. D. A. Madusanka, and S. Wijesinghe, "The development of a gesture controlled soft robot gripping mechanism," in 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), 2016. Available: https://ieeexplore.ieee.org/document/7946573. DOI: 10.1109/ICIAFS.2016.7946573.

[17] C. Zeng, H. Zhou, W. Ye, and X. Gu, "iArm: Design an Educational Robotic Arm Kit for Inspiring Students' Computational Thinking," Sensors (14248220), vol. 22, (8), pp. N.PAG, 2022. Available: https://research.ebsco.com/linkprocessor/plink?id=1c4c6dcc-9c5c-30cc-8c05-c030946476c4. DOI: 10.3390/s22082957.

[18] H. Jiang, J. P. Wachs and B. S. Duerstock, "Integrated vision-based robotic arm interface for operators with upper limb mobility impairments," in 2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR), 2013. Available: https://ieeexplore.ieee.org/document/6650447. DOI: 10.1109/ICORR.2013.6650447.

[19] M. Mashagbeh, M. Al-Khawaldeh, S. Hussein, H. AlAbdellat, and A. Swidan, "A low-cost robotic arm manipulator model for enhancing student learning in an industrial robots course," in 2023, DOI: 10.1109/FIE58773.2023.10343313.

[20] E. Aksoy, A. D. Çakr, B. A. Erol, and A. Gumus, "Real time computer vision based robotic arm controller with ROS and gazebo simulation environment," in 2023 14th International Conference on Electrical and Electronics Engineering (ELECO), 2023. Available: https://ieeexplore.ieee.org/document/10416078. DOI: 10.1109/ELECO60389.2023.10416078.

[21] V. A. C. Arismendi and D. B. Aranibar, "Imitation learning and dimensionality reduction for the resolution of inverse kinematics in anthropomorphic robotic arms," in 2024, DOI: 10.1109/LARS64411.2024.10786432.

[22] F. Ouyang and W. Xu, "The effects of educational robotics in STEM education: a multilevel meta-analysis," *International Journal of STEM Education*, vol. 11, no. 1, Feb. 2024, doi: 10.1186/s40594-024-00469-4.

[23] S. Lu, Z. Nuryana, X. Ni, W. Xu, and M. N. Alias, "A decade of educational robotics: trends and SDG contributions," *Humanities and Social Sciences Communications*, vol. 12, no. 1, Aug. 2025, doi: 10.1057/s41599-025-05663-5.

[24] A. Reyes, S. Reinhardt, T. Wise, N. Rawashdeh, and S. Paheding, "Gesture controlled collaborative robot arm and lab kit," in *Proc. 2022 ASEE Conf. Industry and Education Collaboration (CIEC)*, Tempe, AZ, Feb. 4–11, 2022.

[25] E. Markvicka, J. D. Finnegan, K. Moomau, A. S. Sommers, M. S. Peteranetz, and T. A. Daher, "Designing learning experiences with a low-cost robotic arm," in *Proc. 2023 ASEE Annu. Conf. Expo.*, Baltimore, MD, June 2023, doi: 10.18260/1-2--42983.