2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #53088

Integrating TurtleBot, EMG, and AI for Experiential Robotics Learning Among High School and First-Year Students

Ray Chen Zeng, North Allegheny Senior High School

Ray Chen Zeng is a senior at North Allegheny Senior High School, Pennsylvania, with research interests in robotics and artificial intelligence.

Jacob Wang, Rensselaer Polytechnic Institute
Na Li, Harvard University

Integrating TurtleBot, EMG and AI for Experiential Robotics Learning among High School and Freshman Students

Ray Zeng[1], Jacob Wang[2], Na Li[3]

[1] North Allegheny Senior High School

[2] Rensselaer Polytechnic Institute

[3] School of Engineering and Applied Sciences at Harvard University

zengrayc@gmail.com, jacobwang2005@gmail.com, nali@seas.harvard.edu

Abstract

This paper presents an experiential robotics learning framework that integrates electromyography (EMG) sensing, artificial intelligence (AI), and mobile robotics using the TurtleBot3 platform. Designed for high school and first-year undergraduate students, the framework provides hands-on exposure to applied robotics, basic signal processing, and introductory machine learning through an end-to-end system that maps human muscle activity to real-time robotic motion in both simulated and physical environments.

By leveraging low-cost hardware and open-source software, the framework enables students to engage in data acquisition, feature extraction, supervised learning, and system integration without requiring extensive prior technical background. Experimental results demonstrate reliable EMG-based control of the mobile robot using simple machine learning models, validating the technical feasibility of the approach. Educational outcomes indicate that the framework effectively promotes student engagement, experimentation, teamwork, and the development of computational and engineering thinking skills among early-stage learners.

1 Introduction

Robotics, as an inherently interdisciplinary field, plays a critical role in fostering early interest in STEM disciplines among high school and first-year college students [1]. However, students at these levels often encounter robotics primarily through theoretical instruction, with limited opportunities for meaningful hands-on experience. Given their constrained exposure to engineering design and computer science practice, a carefully designed robotics platform—supported by structured, project-based activities—can offer tangible learning experiences that effectively bridge theoretical concepts and real-world applications. Such an approach not only enhances student engagement but also helps demystify complex engineering systems by illustrating how perception, data processing, learning, and control are integrated within intelligent systems. Ultimately, experiential robotics education can cultivate computational thinking, engineering intuition, and curiosity, motivating students to further explore advanced topics in robotics and related STEM fields [2, 3].

To this end, we, a team comprising a faculty mentor, one high school student, and one first-year undergraduate, developed an experiential robotics framework during the summer of 2025. The proposed framework integrates electromyography (EMG) sensing, a TurtleBot3 mobile robot, and the necessary communication and control components to support end-to-end system development. Designed to be accessible and intuitive, the platform provides a hands-on learning environment in which students actively engage with both hardware and software.

Using this framework, students construct a complete pipeline that acquires and processes human muscle activity signals, performs signal recognition and interpretation, and maps the resulting control commands to robotic motion in both simulated and physical environments. Although the technical depth required at each stage is intentionally kept modest, the framework exposes students to a broad range of foundational STEM concepts, including human physiology, bioengineering, artificial intelligence, and electrical and computer engineering. At the same time, it emphasizes system-level integration, iterative debugging, and problem-solving skills that are central to real-world engineering practice. Preliminary evaluations suggest that the framework effectively fosters curiosity, hands-on experimentation, and the development of computational and engineering thinking among students with limited prior exposure to STEM disciplines. By lowering barriers to entry while preserving meaningful technical content, the framework offers a scalable and flexible approach for engaging early-stage learners in interdisciplinary robotics education.

The remainder of the paper is organized as follows. Section 2 describes the setup of the robotics learning framework, including the cost analysis of hardware and software. Section 3 presents the experimental results. Section 4 concludes with a discussion of future work.

2 The Setup of Robotics Learning Framework

2.1 Framework Composition and Workflow

The framework comprises a set of integrated hardware and software components that are primarily set up, configured, and maintained by students, thereby reinforcing hands-on learning and system ownership.

The experimental setup used a TurtleBot3 Burger platform running ROS2 Humble on Ubuntu 22.04. Three MyoWare 2.0 EMG sensors were placed on the user's forearm and upper arm to capture muscle activity associated with distinct gestures. An Arduino Uno sampled the EMG signals at 1000 Hz and streamed the data via USB serial connection to the host PC, with readings normalized from a 0–1023 range to a standardized scale for further processing. The microcontroller performs initial signal acquisition and analog-to-digital conversion, enabling downstream processing and analysis. The digitized EMG data are then transferred to the PC and further analyzed using Google Colab, which provides access to a rich ecosystem of Python-based scientific computing and machine learning libraries. This modular pipeline allows students to explore signal preprocessing, feature extraction, and basic pattern recognition techniques while maintaining a clear connection between physiological signals and robotic control tasks.

The collected EMG data are labeled and stored to support supervised learning tasks. Using a variety of open-source software packages, students extract statistical and frequency-domain features—including the mean, root mean square (RMS), and variance. To process the continuous stream of EMG data in real time, the system employed a sliding window approach in which a fixed-duration window was advanced one time step at a time across the incoming signal. At each position, the window captured a short segment of EMG data from which statistical features were extracted before the window moved forward to the next tick. This overlap between consecutive windows ensured that no meaningful muscle activation was missed between classifications, and allowed the system to produce a continuous, low-latency stream of feature vectors for the classifier to act on in real time. Additionally, this method enabled us to amplify the amount of meaningful data obtained from each muscle flex. Multiple classification algorithms, such as Support Vector Machines (SVM) [4], Random Forests, K-Nearest Neighbors (KNN), and XGBoost [5, 6], are then employed to categorize EMG signals corresponding to different muscle activation patterns. To mitigate the effects of measurement noise and signal variability, careful feature selection and hyperparameter tuning are performed. These steps play a critical role in improving classification accuracy and robustness, while also exposing students to practical considerations in data-driven modeling and machine learning workflows.

Once a classifier is successfully trained to achieve the desired level of accuracy, it is deployed to process short, continuous intervals of live EMG data streamed from the Arduino microcontroller. Based on these inputs, the classifier generates real-time predictions of muscle activation patterns, which are then translated into control commands and transmitted to the ROS 2 environment. These commands ultimately drive the motion of the TurtleBot3 mobile robot in real time, thereby completing the end-to-end sensing, learning, and control pipeline [7]. The overall workflow of this system is illustrated in Figure 1.

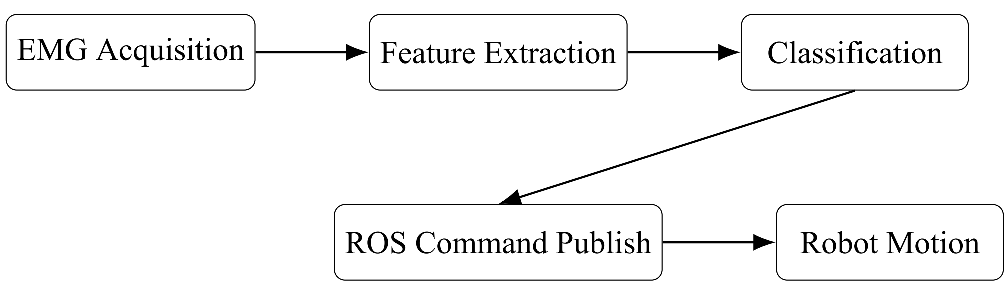


Figure 1: The Workflow of Robotics Learning Framework

2.2 Lab Environment and Cost-Analysis

We note that, by leveraging open-source software packages and low-cost hardware components, the proposed framework is highly cost-effective. A detailed breakdown of the required hardware components and their associated costs is provided in the following table. On the other hand, the software stack relies almost entirely on freely available open-source tools, including the operating system, middleware, and data analysis libraries, resulting in no additional licensing expenses. As a result, the overall system can be implemented at relatively low cost, making it particularly well suited for educational settings, while still supporting meaningful hands-on learning and experimentation.

Table 1: Hardware Cost Breakdown for the Robotics Learning Framework

Component	**Quantity**	**Estimated Cost (USD)**
TurtleBot3 Burger Mobile Robot Platform	1	\$750–\$850
Arduino Microcontroller (Uno / Nano)	1–2	\$10–\$30
MyoWare 2.0 EMG Sensor	3-5	\$40–\$60 each
EMG Sensor Cables and Electrodes	1 set	\$20–\$40
Battery Pack / Power Supply	1–2	\$20–\$50
Wi-Fi / Communication Module	1	\$10–\$30
Mounting Hardware and Mechanical Brackets	1 set	\$20–\$40
Breadboard and Prototyping Wires	1 set	\$10–\$20
Miscellaneous Electronics and Accessories	–	\$10–\$40
Total Hardware Cost		$\approx$ **\$1,000**

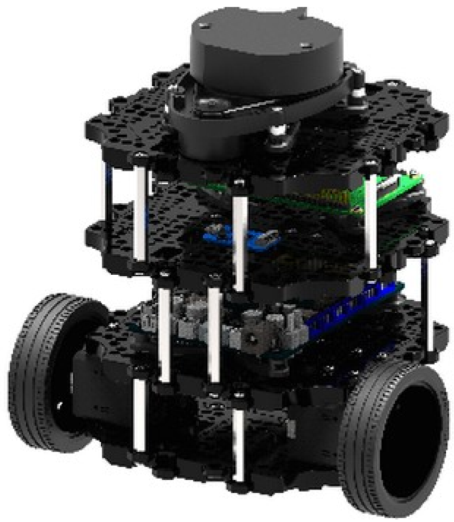
As shown in Table 1, the overall system cost remains at approximately USD 1,000, making the framework particularly suitable for high schools and colleges operating under limited budgets. In addition, all required devices are readily available from mainstream suppliers, which facilitates straightforward replication and deployment without reliance on proprietary or specialized equipment. Figure 2 illustrates the TurtleBot3 platform and the laboratory environment used in this study, demonstrating that the framework can be implemented in standard instructional spaces without the need for specialized facilities.

3 Empirical Demonstrations in the Lab

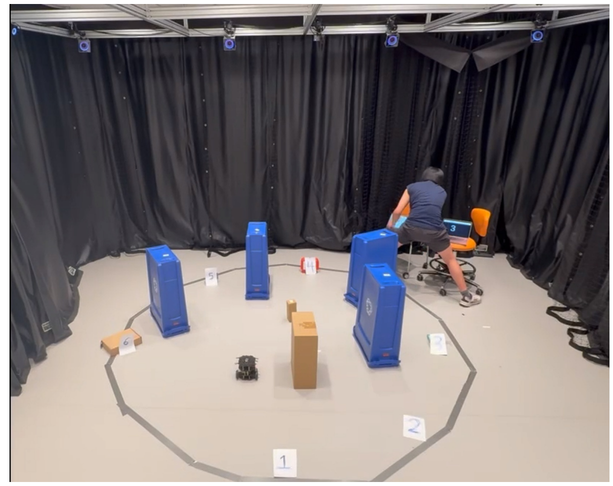
3.1 Experimental Setup

Students conducted a series of preliminary experiments to demonstrate and to evaluate the developed robotics learning framework through hands-on implementation and testing. As noted earlier, three MyoWare 2.0 EMG sensors were attached to the forearm and upper arm to capture muscle activity associated with distinct gestures. An Arduino Uno sampled and streamed EMG signals to a host PC via a USB serial connection. A custom Python-based ROS 2 node handled real-time data acquisition, feature extraction, classification, and velocity command publishing. The TurtleBot3 mobile robot was operated in an open indoor environment to isolate gesture recognition and control performance from navigation complexity.

To transform raw EMG signals into actionable control inputs, students implemented a feature extraction pipeline suitable for introductory STEM learning. Continuous EMG streams were



(a) Turtlebot 3



(b) The Experimental Lab

Figure 2: Turtlebot 3 and Lab Environment

segmented using a fixed-duration sliding window with overlap. Within each window, a set of amplitude- and shape-based features was computed. These features captured both the intensity and variability of muscle activation and served as inputs for supervised learning models. Classification models were trained and deployed to support real-time control of the robot.

3.2 Experimental Results

System performance was mainly evaluated by classification accuracy, defined as the percentage of EMG windows correctly labeled by the trained model, with confusion matrices, recall, and F1 scores serving as supplementary metrics. EMG signals were recorded in 5–10 second windows, sampled both continuously and in gaps to capture fresh and mildly fatigued muscle states, as fatigue was observed to reduce bicep contraction amplitude by approximately 20%. Across muscles, the biceps brachii produced the most consistent contractions and exhibited amplitudes roughly 20–30% higher than those of the Flexor Carpi Ulnaris. Using SVM, the system achieved an overall best classification accuracy of approximately 90% for shoulder-based motions, enabling reliable, low-latency, multidirectional control of the TurtleBot3, while other classifiers including K-Nearest Neighbors and Random Forest yielded significantly lower accuracies ranging from 50–80% for shoulder-based motions, with Random Forest consistently falling in the 64–72% range. Various features were extracted from each window, and students iteratively mixed and matched subsets alongside refinements to sensor placement to improve model robustness. Gestures involving larger muscle groups, such as bicep flexion for forward motion and wrist flexion for turning, exhibited the highest reliability and consistency, with some gestures distinguishable using simple metrics such as variance alone.

A key design improvement involved decoupling shoulder and forearm signals into independent classification models, with shoulder activity governing forward and backward motion and forearm activity controlling left and right turning, which improved SVM accuracy. Dataset balance also proved critical: an early imbalance of roughly 150,000 bicep flick samples against approximately 1,000 samples per remaining class produced 99% accuracy in testing but caused the robot to recognize only bicep flicks in deployment. Truncating all classes to 1,000 samples reduced accuracy to 89% but restored correct robot behavior.

Despite challenges associated with fine-grained gestures, the robot consistently responded with smooth and predictable motions, with response times generally within one second. Experimental tests further demonstrated precise responsiveness to muscle movements, enabling the robot to execute straight trajectories as well as sharp turns. In obstacle-course evaluations consisting of multiple boxes arranged in a large circular configuration, the robot successfully navigated all available paths with a 100% completion rate. A de-identified video regarding those experiments can be found in:`https://www.youtube.com/watch?v=qUq31F_yBtc`.

Overall, the experimental results validate both the technical feasibility of EMG-based robotic control and the effectiveness of the robotics learning framework as an instructional platform, demonstrating how human physiological signals can be transformed into robotic actions using accessible data science and machine learning techniques.

3.3 Educational Activities and Outcomes

This project has provided high school and first-year undergraduate students with an accessible introduction to core concepts in robotics, artificial intelligence, and signal processing. Through hands-on development and troubleshooting of an end-to-end EMG-controlled robotic system, students gained practical experience in data acquisition, feature extraction, supervised learning, and system integration, which are usually introduced in more advanced coursework. The major activities and their reported time allocations are summarized in Table 2, where "H/S" and "C/S" denote the high school student and college student, respectively. Clearly, these activities spanned multiple disciplinary areas and required a substantial time commitment to complete.

The experiential nature of the developed framework enhanced student engagement and reinforced theoretical concepts by linking abstract ideas, such as classification and control, to observable robotic behavior. In addition to technical skills, students developed teamwork, experimentation, and system-level problem-solving abilities. The high school student mentioned that, "Seeing the robot responds to my own muscle signals made abstract concepts such as machine learning and control feel tangible, and it greatly increased my confidence in choosing a STEM program in college and addressing complex technical challenges independently." The college student similarly noted that, "Perseverance was at the core of this project. Days of debugging, parameter tuning, and troubleshooting hardware and software issues made the moment the TurtleBot3 responded to live EMG commands especially rewarding. This project encapsulated the practical challenges of engineering and robotics."

Overall, this framework effectively supports early-stage learners in building computational and engineering thinking skills relevant to future STEM coursework and research. Future classroom deployments will incorporate formal pre- and post-activity surveys with larger cohorts to statistically evaluate learning outcomes.

Table 2: Major Activities and Time Allocations

Activity Category	Sub-Tasks	H/S Hours	C/S Hours
Literature Review and Concept Learning	EMG fundamentals, signal processing concepts, machine learning theory, ROS tutorials, TurtleBot3 documentation	70	123
Data Collection	Sensor placement trials, gesture recording sessions, labeling datasets, calibration experiments on multiple gestures	10	25
Machine Learning Development	Feature extraction, model selection (SVM/RF/KNN/XGBoost/ANN), hyperparameter tuning, validation experiments	200	190
System Development and Integration	Arduino communication, serial transfer, ROS/ROS2 node configuration, classifier to cmd_vel pipeline	200	123
Software Development	Python scripting, ROS/ROS2 scripting, Arduino programming, code refactoring, testing automation	80	88
Experimental Testing and Performance Optimization	Accuracy evaluation, latency measurement, repeated trials, robot testing	50	70
Documentation and Reporting	Writing notes, recording results, preparing figures and explanations	60	44
Total		**670**	**663**

4 Conclusion and Future Work

This work demonstrates that a combination of affordable hardware, open-source software, and interdisciplinary system design can effectively make abstract concepts in robotics and AI accessible to early-stage learners, including high school and first-year undergraduate students. By integrating EMG sensing, ML-based classification, and real-time control of a TurtleBot3 mobile robot, the developed framework provides a concrete and engaging platform for experiential STEM learning that bridges human physiological signals and robotic behavior.

Experimental results validate both the technical robustness of the system and the effectiveness of its instructional design. Students successfully implemented an end-to-end pipeline that transforms human muscle activity into robotic motion, achieving reliable real-time performance using accessible data science and AI/ML techniques. Beyond technical outcomes, the framework supported the development of key educational competencies, including computational thinking, system-level reasoning, experimentation, and collaborative problem solving.

Future work will focus on expanding the framework to further enhance both learning outcomes and system capabilities. Planned extensions include the control and coordination of multiple robots, real-time feedback visualization tools, and additional sensing modalities to deepen student understanding of human–machine interaction. We also aim to broaden deployment of the developed framework in high school and early undergraduate settings, enabling wider adoption of hands-on, interdisciplinary robotics education that prepares students for advanced STEM coursework, projects and research.

References

[1] S. Bruder and K. Wedeward, “Robotics in the classroom,” *IEEE Robotics & Automation Magazine*, vol. 10, no. 3, pp. 25–29, 2003.

[2] N. R. Council, D. of Behavioral, S. Sciences, B. on Testing, Assessment, B. on Science Education, and C. on Highly Successful Schools or Programs for K-12 STEM Education, *Successful K-12 STEM education: Identifying effective approaches in science, technology, engineering, and mathematics*. National Academies Press, 2011.

[3] A. Eguchi, “Bringing robotics in classrooms,” in *Robotics in STEM education: Redesigning the learning experience*, pp. 3–31, Springer, 2017.

[4] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[7] ROBOTIS, “Turtlebot3 quick start guide.” `https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/`, 2024. Accessed: 2026-01-21.