Low-Cost EMG-IMU Controlled Quadruped

Daniel Grant [*1] **Fritz Kürmayr** [*1] **Shou-Jen (Lawrance) Chen** [*1] **Jayanth Sadhasivan** [*1]

Abstract

EMG-Quad investigates two complementary directions toward accessible legged robotics for educational and maker settings. First, we design a mechanically simplified, cost-conscious quadruped using only 3D-printed parts, standard fasteners, and off-the-shelf components, requiring no machining or custom PCBs. By adopting the *Q8Bot* architecture[1] (Wu & Hong, 2025b), we reduce actuation from 12 to 8 motors while preserving effective locomotion via differential turning. Second, we develop a wrist-worn controller integrating surface electromyography (sEMG) and inertial sensing (IMU) to map wrist gestures and muscle activity to real-time locomotion commands. This interface is inspired by Meta Reality Labs' wrist-worn sEMG work for AR interaction. EMG-Quad demonstrates a reproducible pipeline integrating wearable sensing, embedded processing, and locomotion control for low-cost, intuitive legged-robot teleoperation.[2]

1. System Design

Our system architecture is shown in Figure 1 illustrates the overall architecture.

User gestures is captured through a wearable EMG/IMU wristband, which integrates surface EMG electrodes and an onboard IMU. Raw biosignals and IMU responses are digitized by an OpenBCI Cyton board and transmitted wirelessly via a programmable Bluetooth dongle.

All signal processing and control logic are executed on a Raspberry Pi 4, which acts as the central computation unit. Incoming EMG and IMU data are processed through a gesture recognition pipeline. IMU signals are sent in and used as threshold to map to high-level motion commands (e.g., forward, turn, jump, stop). We also attempted to trained classifiers map biosignals to the commands. These commands are then translated into locomotion actions via a gait table lookup and trajectory generation module inspired by (Wu & Hong, 2025a).

Motor commands are sent from the Raspberry Pi to the quadruped through a USB-to-serial (U2D2) interface, using the Dynamixel communication protocol. Joint trajectories are streamed at a fixed control rate of 100 Hz, and all motors are updated synchronously to ensure coordinated motion.

Power is provided by a 12 V LiPo battery, which directly supplies the motors and, through a voltage regulator, powers the Raspberry Pi.

2. Hardware

2.1. Quadruped

Design

The quadruped hardware was designed to be compact, mechanically robust, and easily reproducible using only 3D-printed parts, standard fasteners, and off-the-shelf electrical components. In particular, the chassis was optimized for a dense actuator arrangement while minimizing 3D-print material usage by integrating the actuators as structural elements of the frame. This was achieved through a layout that preserves full leg mobility while allowing the eight actuators to sit directly adjacent to one another, resulting in a tightly packed geometry. The actuators are connected via daisy-chained cables that are intentionally routed through guided channels inside the chassis to protect wiring and simplify assembly.

The legs are composed of two primary components: an upper leg and a lower leg. The design goal was to keep the geometry as simple as possible while enabling easy assembly using standard screws only, and ensuring long-term durability through press-fit ball bearings. The upper and lower leg components are assembled in mirrored pairs (rotated by $180°$), such that all sixteen printed leg elements are realized using only two unique part designs.

Finally, the electronics mounting structure supports the

[*]Equal contribution [1]Footnotes shouldand Computer Science, University of California, Berkeley, United States. Correspondence to: Daniel Grant <dgrant@berkeley.edu>.

Proceedings of the student presentations delivered during the Fall 2025 offering of EECS 149A/249A: Introduction to Embedded and Cyber-Physical Systems.

[1]https://arxiv.org/abs/2508.01149
[2]https://github.com/lawraa/WristControlledQuadruped

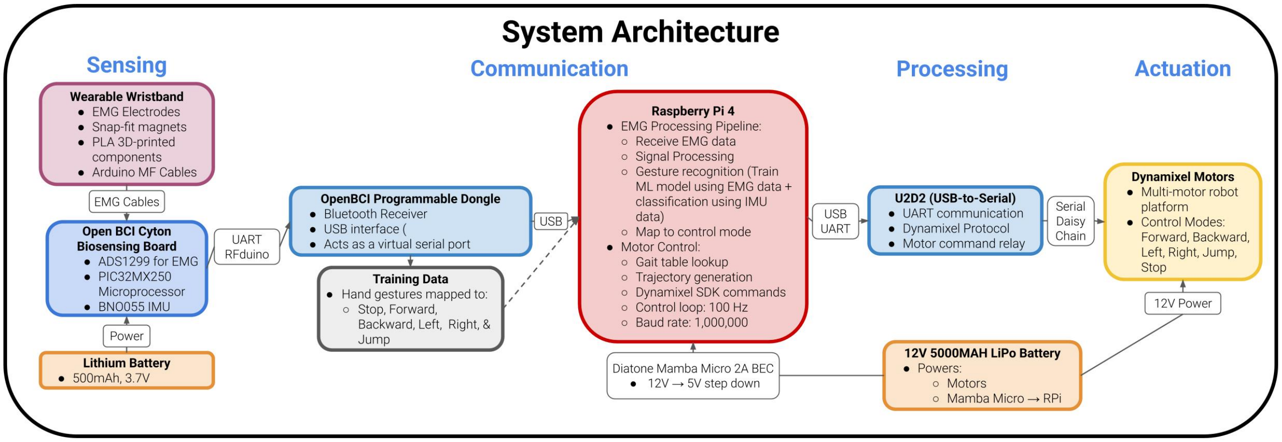

Figure 1. System Architecture

Figure 2. EMG–Quad quadruped prototype showing the 3D-printed chassis and leg assemblies, compact actuator layout, and stacked electronics mounting structure enabling straightforward replication without machining or custom PCBs.

Raspberry Pi, motor controller, battery, Bluetooth modem, and a voltage splitter / BEC for dual power delivery. The structure is optimized for minimal height and maximum compactness, and simultaneously acts as an additional structural reinforcement for the robot. A multi-level stacking layout provides access to key contact points on the battery, motor controller, and Raspberry Pi while keeping the footprint small. As with the rest of the platform, all mounting components are 3D-printed and assembled using standard screws for straightforward replication.

Components

- 8× upper leg (3D-printed)
- 8× lower leg (3D-printed)
- 16× ball bearings (press-fit)
- M2 screws and nuts
- M2.5 screws and nuts
- M3 screws and nuts
- 1× chassis (3D-printed)
- 6× distance holder, M2.5, 20 mm (3D-printed)
- 6× distance holder, M2.5, 10 mm (3D-printed)
- 1× upper chassis (3D-printed, 3-part print)
- 1× battery holder (3D-printed)
- 1× motor controller holder (3D-printed)
- 8× Dynamixel RX-24F robot actuators
- 1× 12 V 5000 mAh battery
- 1× Raspberry Pi 4
- 9× Robot Cable X4P
- 1× USB-C to USB-B cable
- 1× Diatone Mamba Micro 2A BEC
- 1× U2D2 (USB-to-Serial)
- 1× OpenBCI Cyton biosensing board
- 1× Arduino USB dongle

2.2. Wristband

Design

The wrist-worn controller is designed as a research-oriented yet easily reproducible sEMG interface inspired by recent Meta Reality Lab's "neural" wristband. The primary design

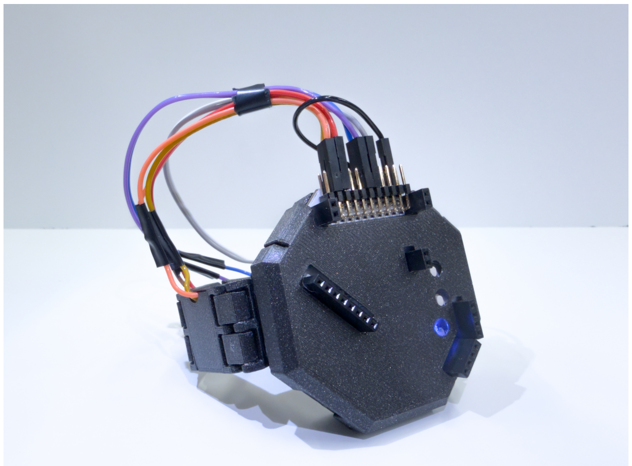

Figure 3. Prototype EMG–IMU wristband showing the 3D-printed enclosure, linked wrist strap with integrated dry EMG electrodes, and external electrode cabling interfaced to the OpenBCI Cyton biosensing board.

objective was to maximize the number of differential EMG channels available from a compact, wearable form factor while maintaining ease of fabrication and assembly. To this end, the wristband uses a two-column, dual-hinge linkage architecture composed of four primary strap sections, each housing a pair of gold-plated dry electrodes, with four differential EMG channels distributed circumferentially around the wrist. A ninth electrode electrode is positioned beneath the lower housing mount to act as a bias electrode. (Fig. 3).

The strap linkage and enclosure are fabricated as a single 3D-printed assembly using black PLA filament on a Prusa Mk4 with a 0.4 mm nozzle, prioritizing reproducibility and minimizing assembly complexity. The linked geometry allows the strap to conform passively to a range of wrist sizes while maintaining consistent electrode–skin contact during motion. Each electrode pair is mounted in a dedicated enclosure measuring $24 \times 12 \times 5$ mm, with individual electrodes measuring 10 mm in diameter and 3 mm in height. The modular layout also enabled evaluation of alternative strap configurations with three, five, and six sections also tested during early prototyping.

Electrode connectivity is achieved via color-coded, 26 AWG stranded PVC-insulated cables (1–1.5 m length) terminated with soldered single female headers. The electrodes are gold-plated; these electrodes would need to be replaced once surface wear exposes the underlying substrate, as exposed base metal introduces undesirable noise into EMG recordings. Internal electronics are secured using 2 mm double-sided adhesive tape, while small, strong neodymium disk magnets assist with ensuring a tight fit around the wrist. The complete wristband, excluding electronics, uses approximately 63 g of PLA material and can be assembled in under one hour using basic tools.

The OpenBCI Cyton board is mounted on the dorsal side of the wrist, resulting in a bulkier form factor than a fully custom design; however, this tradeoff simplifies replication and supports the use of an open-source, off-the-shelf biosensing platform. During data collection, the wristband was worn comfortably for sessions lasting 15–30 minutes, with longer operation expected to be feasible. While the prototype prioritizes modularity and signal accessibility over aesthetics and environmental sealing, it provides a practical baseline for low-cost, wearable EMG-driven robot control.

Components

The wristband hardware consists of the following primary components:

- 1× OpenBCI Cyton biosensing board (8-channel sEMG + 3-axis accelerometer)
- 8× gold-plated dry EMG electrodes (arranged as 4 differential pairs)
- 1× bias electrode
- 4× wrist strap sections (3D-printed PLA)
- 1× linked wristband enclosure (3D-printed PLA)
- 26 AWG stranded PVC-insulated cabling (1–1.5 m, color-coded)
- Neodymium disk magnets for fastening
- 2 mm double-sided adhesive tape for mounting electronics

3. Software

All of our code can be found at the following repository: https://github.com/lawraa/WristControlledQuadruped

3.1. Quadruped

This section describes the software architecture and control pipeline used to generate quadruped locomotion. Our code translates high-level movement commands into synchronized low-level motor position commands.

Software Architecture Overview

The quadruped software is organized into a modular pipeline with four main components:

1. **Foot Trajectory Generation**: produces parameterized Cartesian foot paths for a single leg.
2. **Inverse Kinematics Solver**: converts Cartesian foot positions into joint-space commands.

3. **Gait Management and Coordination**: constructs multi-leg gaits via phase offsets.
4. **Motor Control**: converts joint commands into Dynamixel motor commands and transmits them synchronously.

Foot Trajectory Generation

Locomotion is completed by precomputed foot trajectories defined in Cartesian space. For each leg, a complete gait cycle is discretized into two phases $N = s_1 + s2$: (1) Swing (lift) phase: the foot moves forward while lifting off the ground. (2) Stance (return) phase: the foot returns to its initial horizontal position while maintaining or increasing vertical support.

$$y(x) = y_0 - y_{range1} sin(\pi x)$$

$$y(x) = y_0 + y_{range2} sin(\pi x)$$

The horizontal foot position is centered around a nominal value x_0 with stride length x_{range}. Vertical motion is generated using sinusoidal function shown in the equation above where in the swing phase, it has an amplitude of y_{range1} to provide foot clearance and a initial standing height of y_0.

Inverse Kinematics

Each Cartesian foot position (x, y) generated by the trajectory planner is converted into joint angles using a closed-form inverse kinematics (IK) solution for a planar 2-DoF leg. The solver models the leg as a two-link mechanism with known link lengths and motor separation distance, and computes joint angles using geometric relationships derived from the law of cosines. The solver stores the last valid solution and automatically reduces stride or lift parameters if a requested foot position lies outside the feasible workspace. (**?**) contains more information on the motor gait computation.

Gait Construction via Phase Coordination All quadruped gaits are constructed from a single-leg base trajectory by applying fixed phase offsets between legs. For example, in TROT mode diagonal leg pairs move in phase with a 50% cycle offset, which allows two legs on the floor each stride. In WALK mode, three legs each leg is offset by 25% of the gait cycle, which allows three legs to be on the floor at all times. Turning is achieved by asymmetrically scaling the stride length of the inside legs which allows us to make directional control without modifying the underlying gait structure. Each gait produces a sequence of joint-space commands of the form:

$$[q_{FL1}, q_{FL2}, q_{FR1}, q_{FR2}, q_{BL1}, q_{BL2}, q_{BR1}, q_{BR2}$$

corresponding to the eight actuated joints.

Real-Time Gait Execution

Gait execution is managed by a gait manager that tracks the current gait and direction. Joint trajectories are sampled at a fixed control frequency of 50 Hz. At each control tick, the next joint configuration is selected from the active gait trajectory. For jump motions, when triggered, the system temporarily switches to a jump gait, executes exactly one full gait cycle, and then returns the robot to a neutral standing pose before resuming other gaits.

Motor Control Interface

Joint-space commands are converted into Dynamixel RX-24F motor commands using position control. Joint angles are recentered around a 150°, which is decided since the motor's valid angle is 0° to 300°. It is then converted into motor ticks with approximately 0.29° resolution per tick, and adjusted for motor direction inversions and calibration offsets. All eight motors are commanded simultaneously using synchronized writes over a 1 Mbps rate.

Code Reuse and Adaptation Parts of the quadruped gait generation approach were inspired by the open-source Q8bot repository: https://github.com/EricYufengWu/q8bot

3.2. Wristband

The software architecture for the smart wristband describes the acquisition and processing of IMU accelerometer data alongside captured EMG signals. The wristband's Cyton Biosensing board's RFD22301 microcontroller captures 8 24-bit signed integer samples from the ADS1299, converts the data to two's complement, and packs each channel's signal with accelerometer data into a 33-byte packet. The 33-byte packet is then sent to the quadruped's Raspberry Pi 4 microcontroller, where it is split into relevant EMG and IMU data. The following subsections describe the internal data processing of pipeline of the Cyton Biosensing Board.

LIS3DH 3 axis Accelerometer

The microcontroller samples the 3-axis LIS3DH accelerometer at a rate of 25 Hz via I2C. The accelerometer data is then processed in the quadruped's Raspberry Pi 4 microcontroller, which then maps forward, backward, left, and right wrist motions to corresponding forward and backward movements and left and right rotational commands. Acceleration in the z-direction corresponded to a jump, where the z direction is defined as the axis perpendicular to the floor.

EMG Processing

The microcontroller samples fixed-length 200-sample windows at a rate of 250 Hz from 8 EMG channels. Four of the EMG channels are directly connected to differential electrode pairs to capture bioelectrical signals. At each sampling instant, the ADS1299 sends data to the RFD22301 via SPI.

Like the accelerometer, EMG processing occurs in the quadruped's microcontroller. We evaluated two machine learning approaches for mapping EMG signals to hand gestures: Support Vector Machines (SVM), and an Deep Neural Network (DNN). The EMG signals are first processed using a RMS envelope to decrease noise and emphasize muscle activation. Since muscle activity primarily occurs within the 20-450 Hz range, and bioelectrical signals are sampled at 250 Hz, no additional frequency-domain filtering is applied.

For the SVM, the following features are computed across all 200 samples for each of the 4 channels:

1. **Root Mean Square (RMS)**: $\sqrt{\mathrm{mean}(x^2)}$
2. **Mean Absolute Value (MAV)**: $\mathrm{mean}(|x|)$
3. **Waveform Length (WL)**: $\sum |\Delta x|$
4. **Zero Crossings (ZC)**: the number of times the signal crosses zero within a window, with crossings due to noise excluded using a magnitude threshold.

Each $4\text{x}4$ feature matrix is flattened and passed to a SVM.

The DNN operates differently than the SVM. It processes the 200-sample window by grouping the data into ten segments and extracts features such as the maximum absolute value, minimum absolute value, peak-to-trough difference, and signal variance from each group. Each 200-sample window produces 20 feature matrices, which are provided as input to train the DNN.

Training was conducted by prompting users to perform hand gestures corresponding to wrist extension, flexion, left rotation, right rotation, rest, and finger spreading. Because training achieved a maximum classification accuracy of 40%, it was insufficient for reliable performance; therefore, only accelerometer based control was utilized for our demo.

4. Future Work

We thought of several promising directions to improve upon both the quadruped platform and the wristband interface.

Locomotion and Mechanical Improvements. The current gait implementation uses kinematically-defined trajectories with fixed sinusoidal profiles and skid-steer turning. Future work should explore more biomimetic and dynamically-stable gaits that exploit natural leg compliance and body dynamics, potentially incorporating center-of-mass trajectory optimization and adaptive foot placement strategies. Also, the current 3D-printed foot geometry exhibits insufficient ground contact friction during rapid maneuvers; integrating compliant, high-friction foot pads (e.g., textured rubber or silicone inserts) would improve traction.

Enhanced EMG Sensing and Processing. While the current system integrates both EMG and IMU data, the gesture recognition pipeline relies primarily on accelerometer-based wrist motion detection due to signal quality constraints. Fully leveraging the differential EMG channels requires addressing several hardware and signal processing challenges. First, the wristband should be redesigned with a fully enclosed EMG electrode cavity to improve skin contact consistency and reduce motion artifacts. Second, expanding from four to six or eight differential electrode pairs would increase spatial sampling density and improve gesture discrimination, particularly for complex hand and finger movements. Producing real-time EMG-driven control would enable finer-grained gesture differentiation and eliminate the dependency on exaggerated wrist motions, moving closer to the subtle, low-effort interfaces demonstrated in recent neural wristband research.

5. Conclusion

The quadruped features 8 actuated DoF and executes synchronized position control at 100 Hz with 0.29° resolution per tick over a 1 Mbps Dynamixel bus. Five gaits are supported, with stride lengths up to 50 mm, foot clearance up to 20 mm, and jump vertical push up to 40 mm. Gait cycles range from 0.6–1.2 s, and the robot achieves a maximum forward velocity of 17 cm/s (>1 body length/s).

Course-Related Concepts. This project exercised several core embedded-systems concepts. *Sensor & serial interfacing* was central to the end-to-end control loop: wristband IMU/EMG signals were acquired over a digital sensor bus (I^2C), while motor commands were issued over a serial link (UART) to the Dynamixel actuators. *Real-time scheduling* considerations ensured that sensing, preprocessing, and command transmission ran at a stable rate with bounded latency, enabling responsive motion and preventing control instability from delayed updates. Finally, *finite state machines* were used to structure system behavior into explicit control modes (e.g., idle, gesture-driven control, and safety stop), improving reliability by providing clear transitions and predictable fail-safe behavior.

Learnings. A key hardware takeaway is that the chassis and leg structure should not trade structural stiffness too aggressively for light weight and compactness. In practice, insufficient rigidity concentrates stress at actuator interfaces and increases fracture risk during dynamic locomotion. With the current reinforcements and geometry, the robot operated reliably and remains easy to replicate using standard fasteners and 3D-printed parts, supporting durability in repeated use.

For the wristband, the dominant challenge is maintaining consistent electrode placement and contact over the same muscle groups while the forearm's cross-sectional profile changes significantly during wrist flexion, pronation/supination, and palm motion. This makes mechanically stable strap geometry and repeatable electrode pressure critical for usable EMG signals across sessions. Without this contact consistency, even carefully tuned filters and ML-based decoding pipelines are not able to yield a decisive improvement in control performance.

Finally, combining EMG with IMU sensing proved to be a robust strategy: the IMU provides stable, low-latency motion intent, while EMG contributes higher-resolution cues for fine-grained control. Together, they enable an interface that is both intuitive and capable of more detailed command modulation than either modality alone.

Cost. The total system cost for the EMG–Quad prototype was approximately $442 (USD), excluding access to shared lab tooling such as 3D printers.

6. Individual Contributions

The following describes individual group member contributions for this project.

Daniel Grant

Daniel completed the initial Raspberry Pi integration and motor testing, creating the walking modes demonstrated for the quadruped during Milestone 1. He also custom designed and printed all of the hardware housing for the wristband and did the integration for that system. He additionally wrote the IMU control algorithm used during our final demo, and collected + analyzed training data for an EMG-based control algorithm that was not pursued further.

Fritz Kürmayr Fritz independently 3D-designed the quadruped's chassis and iterated through multiple design cycles to optimize both usability and compactness. He also CAD-designed the legs and developed a novel construction approach to reduce the number of mechanical components to an absolute minimum. In addition, he designed the remaining robot components end-to-end, led the full mechanical assembly, and optimized several electronics integration details to improve overall robustness and reliability.

Shou-Jen (Lawrance) Chen Lawrance did the initial setup of the Raspberry Pi to motor communication, debugged the Raspberry Pi–to–motor communication pipeline, allowing it to send signals properly. He also wrote, integrated, calibrated all the gait designs allowing the robot to move smoothly to match our specific hardware. He also wrote the overall structure of the code to intake classification results that maps to different actions to control the motors. He also created the poster design.

Jayanth Sadhasivan Jayanth worked on the the machine learning algorithms used to map EMG signals to hand gesture recognition. He investigate hand gestures that produced strong bioelectrical responses, explored multiple approaches to EMG-based gesture classification, and evaluated techniques to further filter EMG data for particular muscle movements. His work encompassed signal preprocessing, feature extraction, and the design and testing of several deep neural networks for gesture recognition.

References

Wu, Y. and Hong, D. Remote gait computation and wireless data collection on the q8bot miniature quadruped. In *2025 22nd International Conference on Ubiquitous Robots (UR)*, pp. 362–367, 2025a. doi: 10.1109/UR65550.2025.11078123.

Wu, Y. and Hong, D. Design of q8bot: A miniature, low-cost, dynamic quadruped built with zero wires. *arXiv preprint arXiv:2508.01149*, 2025b.