



OPEN An integrated tiny-YOLO v3 and Q-iteration framework for stable, energy-efficient autonomous navigation of quadruped robots on AMB82-mini microcontrollers

Falah Hasan Salih, Amir Hooshang Mazinan[✉] & Seyed Mahmoud Modaresi

The deployment of advanced artificial intelligence, specifically deep learning (DL) for perception and reinforcement learning (RL) for control, on ultra-low-power, microcontroller-based quadruped robots presents significant challenges. A system-level engineering approach is presented, integrating these computationally intensive methodologies. Central to this is an object detection module, powered by a lightweight Deep Neural Network (DNN), specifically a Tiny-YOLOv3 model, running on an AMB82-Mini microcontroller. The robot's perception capabilities are provided by the AMB82-Mini, while its real-time locomotion control system is implemented on a Teensy 4.0 microcontroller. This integration leverages meticulous optimization techniques, including INT8 quantization and efficient TFLite Micro deployment. The object detection module achieves approximately 7.8 frames per second (128.32 ms inference latency), enabling robust obstacle avoidance and stable locomotion. Experimental validation was primarily conducted using the custom-built TMUBot quadruped robot, demonstrating its capabilities across diverse terrains. The results underscore the potential of using machine learning with low-power microcontrollers to achieve complex control schemes for small-scale robotic applications.

Keywords Embedded robotics, Tiny-YOLO v3, Dynamic stability analysis (ZMP), Energy-efficient robotics, AMB82-mini microcontroller

Autonomous quadruped robots hold immense promise for applications ranging from exploration and surveillance to disaster response, particularly in dynamic and unstructured environments. However, their effective deployment hinges on robust real-time perception and adaptive control, which often present significant engineering challenges. Integrating sophisticated deep learning models for real-time object detection and pose estimation typically demands substantial computational resources, making their direct deployment on low-power, embedded microcontrollers (without powerful dedicated GPUs) computationally prohibitive and impractical for energy-constrained applications. Furthermore, while both deep learning for perception and reinforcement learning for adaptive control have demonstrated individual successes, their seamless and efficient integration onto a single, resource-constrained embedded platform for a fully autonomous system remains a complex challenge. Many existing solutions, while theoretically sound or validated in simulation, frequently lack comprehensive quantitative validation of critical real-world performance metrics, such as dynamic stability and detailed energy consumption profiling, which are crucial for practical, long-duration field operations. These limitations highlight a critical gap in the development of truly autonomous, robust, and energy-efficient quadruped robots capable of operating independently in complex, real-world scenarios^{1–3}.

Significant progress has been made in enhancing quadruped robot capabilities through advanced control strategies, particularly those leveraging deep learning and reinforcement learning. Researchers have explored DRL for efficient locomotion, combined Model Predictive Control (MPC) with reinforcement learning for improved stability, and developed sophisticated jump control algorithms for challenging environments like asteroids and uneven terrain^{4–7}. Furthermore, AI-based approaches have been introduced for self-balancing and hierarchical locomotion control for modular systems^{8–10}. While these studies demonstrate remarkable advancements in robust locomotion, complex maneuverability, and learning adaptive behaviors, many often rely on high-performance computing platforms, focus primarily on control policies without fully integrated real-

Department of Electrical Engineering, ST.C., Islamic Azad University, Tehran, Iran. ✉email: mazinan@iau.ac.ir

time perception, or operate within controlled environments, leaving a gap for truly autonomous, energy-efficient operation on resource-constrained embedded systems. Similarly, while object detection models like YOLO have been successfully applied in drone-based surveillance for real-time operations, the challenge intensifies when integrating such models with complex locomotion control on the ultra-low-power microcontrollers typical of smaller quadruped robots.

Building upon these insights and addressing the aforementioned gaps, this study presents a novel, integrated machine learning framework designed for autonomous quadruped robots operating on ultra-low-power microcontrollers. Our approach combines a highly efficient deep learning model, Tiny-YOLOv3, for real-time visual perception (object detection and pose estimation) with a robust Q-iteration algorithm for adaptive locomotion control. This framework is specifically engineered for deployment on resource-constrained platforms such as the AMB82-Mini and Teensy 4.0, primarily leveraging monocular vision for real-world perception, with LiDAR integration explored in simulation for enhanced environmental awareness. The primary contributions and innovations of this work are summarized as follows:

1. Development of a novel, integrated machine learning framework combining real-time deep learning (Tiny-YOLOv3) for perception and an efficient Q-iteration algorithm for adaptive locomotion control. This framework is specifically designed for and deployed directly onto ultra-low-power microcontrollers (AMB82-Mini and Teensy 4.0), enabling true on-board intelligence without reliance on external high-performance computing.
2. A direct comparative analysis of Q-iteration against state-of-the-art Deep Reinforcement Learning (DRL) algorithms (PPO and SAC), demonstrating that Q-iteration achieves comparable performance for complex locomotion tasks with significantly lower computational overhead, thereby justifying its selection for efficient embedded control.
3. Comprehensive quantitative validation of the system's dynamic stability using Zero Moment Point (ZMP) metrics and detailed energy consumption profiling across diverse terrains. This validation demonstrates robust autonomy of a microcontroller-based quadruped robot, including a 70% reduction in navigation collisions in dynamic environments and successful operation on high-slope (up to 35°) terrains.

The remainder of this paper is organized as follows: Section "Literature review" provides a detailed literature review. Section "Methodology" describes the proposed integrated architecture and methodology. Section "Results and discussion" presents the experimental setup and results, including performance metrics for perception, control, stability, and energy efficiency. Finally, Section "Conclusion" concludes the paper and outlines future work.

Literature review

In robotics, quadruped control using microcontroller systems does not require an operating system, but the designs are very complex, and the area of study is non-linear. Research and development into quadruped control systems are highly needed for exploratory functions, surveillance, and military applications. Many robotic companies and research institutions have developed various quadruped robots using different controllers available today. Some studies are using controllers like wearable microsystems and MEMS, but these controllers make the reference quadruped bots with 32-bit controllers. Further, some related literature will be introduced.

Ref¹¹ introduces a pioneering 3D object detection method, "viewpoint feature histograms," for quadrupedal robots. It leverages 2D detection, translating bounding boxes into 3D object proposals, enabling reuse of 2D detectors and increasing performance with less computation for real-time efficiency. Demonstrated with YOLO3D on KITTI, this versatile approach achieves up to 99.93% accuracy, significantly enhancing robot navigation precision and safety. Ref¹² proposes a digital twin framework integrating robotic devices, transforming industrial sectors via AI and IoT. This virtual prototype system, created with debugging platforms, tracks robotic activity using real-time microcontroller designs and machine learning. It enables seamless control and monitoring of robotic actions, guaranteeing effectiveness and adaptability in changing contexts, offering improved performance and versatility across applications.

Ref¹³ presents an experimental study on real-time position control and obstacle avoidance for a 4WD mobile robot. It integrates PID, fuzzy logic, and deep learning (YOLO) for navigation. YOLO detects 80 object types, while a fuzzy controller guides the robot to specified positions. The system achieved accurate human detection and precise target reaching with minimal speed errors, demonstrating successful control and obstacle handling. Ref¹⁴ demonstrates the effective use of soft sensors for obstacle detection and distinction in soft robotics. Using a modular, untethered miniature C-legged robot (M-Squad) with integrated coil-spring sensors, the study shows that good design enables accurate feedback. The robot successfully detected obstacles during locomotion and distinguished scalable ones, turning back from impassable barriers, advancing soft robot perception. Ref¹⁵ proposes ODSDP-ADLMSSO, a novel object detection system for visually impaired persons. It employs a Gaussian filter, YOLOv7 for object detection, MobileNetV3 for feature extraction, and a TCN model for classification. Sparrow Search Optimization fine-tunes TCN hyperparameters. Tested on an Indoor OD dataset, the system achieved a superior accuracy of 99.57%, significantly enhancing navigation safety and information for VIPs.

Ref¹⁶ proposes a modular ROS-based framework for characterizing and controlling polymer-based soft robots, addressing their non-linear nature. This framework enables model-less 'DRL' via hardware-in-the-loop training. Demonstrated with an actor-critic algorithm on a pneu-net soft robot, it showed an 89.5% increased likelihood of reaching the locomotion goal, simplifying complex control strategy development. Ref¹⁷ proposes a deep learning-based pavement inspection framework for the Panthera self-reconfigurable robot. It utilizes SegNet for semantic segmentation and DCNN for detecting pavement defects and garbage. A Mobile Mapping System

geotags defects. Implemented on Panthera, the system achieved high accuracy in real-time detection, proving suitable for deployment in sweeping and cleaning tasks. Ref¹⁸ presents an open-source single-leg controller for the hydraulic quadruped robot Spurlos, using a distributed control scheme. Addressing the lack of specialized control boards, this system integrates chips to manage encoders, sensors, and servo valves. Its software, developed with Model-Based Design, ensures stable operation, satisfying leg motion control requirements and facilitating future hydraulic quadruped robot research.

Ref¹⁹ provides reliable predictive analytics and data transmission optimization for intelligent service care robots in "StreamRobot," addressing IoT vulnerabilities. It employs a software-defined design, edge system modeling, and a novel FD-CPML for real-time prediction. Through OpenFlow-SDN and orphan reconnection, the system achieved reduced data stream latency and improved predictive scalability, ensuring reliable communication and intelligent node failure monitoring. Ref²⁰ addresses challenges in hydrodynamic modeling of surface vehicles, particularly data-driven models' poor extrapolation. It employs representation learning to define a valid data space and incorporates hallucinated replay into the prediction network, improving long-term prediction accuracy. Validated experimentally with a robotic surface vehicle for path tracking, this method enhances the robustness and precision of dynamics modeling. Ref²¹ reviews soft materials and devices enabling sensorimotor functions in robots, inspired by biological systems. It addresses current limitations in robot autonomy due to insufficient flexible sensing and actuation integration. The review covers advancements in soft sensing, actuation, structural designs, fabrication, control strategies, and AI integration, aiming to guide future research toward enhancing soft robots' autonomy and adaptability.

Ref²² designs a magnetic arthropod soft robot (MASR) with rapid movement and perception-feedback, addressing challenges in magnetic soft robotics. Inspired by biomimetic joints, MASR-A achieves 1.4 BL/s speed. MASR-B integrates bionic antennae, using triboelectric tactile sensors to detect collisions and their direction. A microcontroller then alters the magnetic field for obstacle avoidance, providing a novel biomimetic design. Ref²³ presents a novel electroadhesion (EA)-driven soft crawling robot using an origami mechanism, enhancing adaptability in complex environments. This design integrates strong surface adherence from EA with flexible, efficient movement via origami techniques. Optimized parameters maximize crawling efficiency. Experimentation confirms superior performance, opening new possibilities in soft robotics by combining electrostatics, origami, and robotics. Ref²⁴ proposes a novel high-integration module for centimeter-scale reconfigurable piezo robots, addressing traditional design limitations. The built-in-ceramic actuation unit achieves ultra-high locomotion speed (90.3 BL/s) and carrying capability. Multi-position magnetic connections enable diverse reconfigurations, allowing robots to adapt to various flat work scenarios, inspiring future miniature reconfigurable robot design. Ref²⁵ introduces a novel bio-inspired three-DOF spherical robotic manipulator (SRM), emulating natural biomechanical properties. Utilizing spherical Complex Spatial Kinematic Pairs and direct motor-to-joint motion, it optimizes energy efficiency and spatial mobility. Kinematic computations employ screw theory. Validated experimentally, the SRM offers an expanded workspace, enhanced dexterity, and a lightweight, compact design for diverse robotic applications. Further, Table 1 provides a general comparison of the literature presented in this section.

In the following, some literature focused on increasing the efficiency of quadruped robot controllers with the central theme of deep learning will be presented.

Aractingi et al²⁶ explore deep reinforcement learning for controlling the Solo12 quadruped robot, focusing on joint impedance references to improve locomotion efficiency. They demonstrate robust indoor and outdoor performance with easy deployment. Zhang et al²⁷ propose a framework combining Model Predictive Control (MPC) and reinforcement learning for quadruped locomotion. Their method improves stability and performance, requiring less data and offering an efficient control strategy. Qi et al²⁸ address stable jump control for asteroid-exploration quadruped robots using multi-agent reinforcement learning. The approach enhances jumping stability, including takeoff, attitude adjustment, and soft landing in weak gravitational fields.

Bellegarda and colleagues²⁹ propose deep reinforcement learning for robust quadruped jumping control. Their method enables jumping over uneven terrain, accounting for robot dynamics and environmental conditions, and achieving better real-world deployment. Qi et al³⁰ present integrated attitude and landing control for quadruped robots during asteroid missions, using reinforcement learning for stable landings on irregular asteroid surfaces, even with sparse rewards and unknown gravitational parameters. Lee and An³¹ introduce an AI-based control algorithm using reinforcement learning and neural networks for self-balancing quadruped robots. Their approach replaces traditional control methods and shows effectiveness through experimental validation on a customized robot test bed. Wang and colleagues³² propose a hierarchical locomotion control for modular quadrupedal robots using deep reinforcement learning. Their method combines low-level CPG-based control with a high-level neural network to achieve efficient learning and robust performance on irregular terrain. Zhang et al³³ propose a LiDAR-based autonomous exploration method for mobile robots using deep reinforcement learning. Combining a sparse informative graph and self-attention mechanisms, their approach enhances exploration efficiency and robustness in unknown environments, outperforming state-of-the-art methods. Zhang et al³⁴ introduce E-Planner, an efficient path planner for car-like mobile robots in unknown environments. Using visibility graphs, obstacle contour optimization, and prioritized exploration, their method achieves faster computation and shorter paths, improving real-time navigation performance.

The literature presented below will focus on advanced topics in target tracking for aerial systems and autonomous driving. These studies introduce novel techniques such as multi-level learning, Kalman filtering, and spatio-temporal reasoning to enhance accuracy and efficiency in complex scenarios.

Xue et al³⁵ propose FMTrack, a robust RGB-T tracking framework that utilizes frequency-aware interaction and multi-expert fusion to handle modal quality fluctuations. The framework incorporates frequency masks and expert networks to capture complementary information, enhancing performance across diverse datasets like LasHeR and VTUAV. FMTrack achieves state-of-the-art results in complex tracking scenarios. Xue et

References	Proposed methodology	Achievements and results	Limitations
11	A method called "viewpoint feature histograms" that translates 2D bounding boxes (from detectors like YOLO3D) into 3D object proposals using LiDAR point clouds	Achieved real-time performance on the KITTI dataset. Enabled the reuse of existing 2D detectors and reported an accuracy of up to 99.93%	The high accuracy is specific to the KITTI dataset, which is primarily for autonomous vehicles and may not fully represent all quadrupedal robot environments. The paper focuses on detection rather than the full navigation and interaction pipeline
12	A digital twin framework that integrates robotic devices, using machine learning, real-time microcontroller systems, and debugging platforms for simulation and lifecycle management	The framework enables seamless control, monitoring, and improved decision-making for robotic systems, enhancing performance and adaptability	The paper is conceptual and describes a framework. It does not provide specific quantitative results, metrics, or details of a concrete implementation and validation
13	A hybrid control system using PID, fuzzy logic, and deep learning. It employs the YOLO algorithm for real-time obstacle detection (trained for 80 object classes) and an Android app for sending target locations	The robot successfully detected a human obstacle in real-time (90–130 cm range) and navigated to target points with high precision (max linear speed error of 2.8 mm/s)	The experimental validation was conducted in a controlled lab environment and was limited to a single class of obstacle (a human), despite the model being trained on 80 classes
14	Using well-designed electronics and mechanics with integrated coil-spring shaped soft sensors to detect and distinguish obstacles without complex algorithms. The robot (M-Squad) has a modular design	The robot successfully detected obstacles and distinguished between scalable (20 mm) and non-scalable (150 mm) obstacles during locomotion in a controlled parkour environment	This is a proof-of-concept case study. The obstacle differentiation is limited to two specific heights and a controlled setting. It does not address a wide variety of obstacle shapes, materials, or complex terrains
15	A multi-stage deep learning pipeline (ODSDP-ADLMSSO) using a Gaussian filter, YOLOv7, MobileNetV3 (for feature extraction), a Temporal Convolutional Network (TCN), and Sparrow Search Optimization for hyperparameter tuning	Achieved a very high accuracy of 99.57% on the "Indoor OD dataset," outperforming existing techniques	The performance is validated on a specific indoor dataset. Real-world viability in dynamic, outdoor environments with varying lighting and weather conditions is not addressed. The complexity of the pipeline may pose challenges for real-time processing on low-power wearable devices
16	A modular framework using ROS for hardware-in-the-loop training of a pneu-net soft robot. It employs a modelless deep reinforcement learning (actor-critic) algorithm to learn a locomotion policy	The learned policy resulted in an 89.5% increase in the likelihood of reaching a specified goal compared to random actions, demonstrating successful policy convergence	The comparison baseline is "random oracle actuation," which is a low bar. The task is specific to a two-actuator robot learning locomotion in a likely controlled environment ("end of frame")
17	A deep learning framework for the Panthera robot using SegNet for pavement segmentation, a DCNN for detecting cracks/garbage, and a Mobile Mapping System (MMS) for geotagging defects	The system identifies pavement defects and garbage with "high accuracy" and is deemed suitable for real-time deployment for garbage detection and subsequent cleaning tasks	The abstract does not provide specific quantitative metrics for its "high accuracy." Performance under diverse weather, lighting, or pavement conditions is not discussed
18	The design of an open-source, distributed, single-leg controller for the hydraulic quadruped robot Spurlis, using a Model-Based Design (MBD) approach for software development	The controller operates stably and meets the requirements for single-leg motion control, providing a ready-made solution for a common hydraulic leg configuration	The scope is limited to the control of a <i>single leg</i> , not the coordinated control, balance, or locomotion of the entire quadruped robot. Results are qualitative ("stable operation") rather than quantitative
19	A system (StreamRobot) using software-defined networking, an orphan reconnection trigger for reliability, and a Fog Detection-to-Cloud Predictive Machine Learning (FD-CPML) model for data analytics	The proposed FD-CPML model showed improved data stream latency (26.67%) and scalability (36.15%) compared to decision tree and logistic regression baselines	The experiment focuses on the data transmission and prediction component using TelosB nodes, not the entire "service care robot" system. The comparison is against two standard baseline models
20	A deep dynamics model enhanced with representation learning and "hallucinated replay" to ensure long-term predictions remain within a valid data space	The method improves the accuracy of long-term predictions for a robotic surface vehicle's dynamic compared to naive neural network approaches. It was successfully applied to path tracking control	The validation is specific to a single type of system (robotic surface vehicle). The generalizability to other complex dynamic systems is not demonstrated
21	This is a review paper providing a comprehensive overview of soft materials, devices, sensing technologies, actuation mechanisms, and control strategies for enabling sensorimotor functions	It synthesizes the state-of-the-art in the field, draws parallels with biological systems, and provides guidance for future research directions	As a review paper, it does not present a novel methodology or new experimental results. Its contribution is the organization and analysis of existing work
22	A biomimetic, jointed magnetic soft robot (MASR) capable of rapid movement. It integrates triboelectric tactile sensors (TTSs) as "bionic antennae" for obstacle detection and direction sensing	The robot achieved a speed of 1.4 body lengths/sec. It can detect obstacles, determine their direction, and initiate an avoidance maneuver based on sensor feedback	The robot relies on an external alternating magnetic field for movement and is not autonomous. The obstacle avoidance is demonstrated in a controlled setting with specific materials
23	A novel soft crawling robot that combines an origami mechanism for flexible movement with electroadhesion (EA) for strong surface adherence	The design principles and manufacturing process are detailed, with the claim that extensive experimentation confirms "superior performance" due to the innovative combination	The abstract is high-level and lacks specific quantitative performance metrics (speed, payload, types of surfaces). The claim of "superior performance" is not substantiated with data or comparisons
24	A high-integration, centimeter-scale module with a built-in-ceramic actuation unit. Reconfiguration is achieved through multi-position magnetic connections	The actuation unit achieves ultra-high speed (90.3 body lengths/sec) and carrying capacity (31.6 × self-weight). The reconfigurable system can perform tasks like wireless image capture	The robot's operation is limited to "flat work scenarios." The magnetic reconfiguration may have limitations in strength and alignment, and the process does not appear to be autonomous
25	A novel 3-DOF spherical robotic manipulator (SRM) designed to emulate a biological ball-and-socket joint, using screw theory for kinematic analysis	The design offers an expanded workspace, enhanced dexterity, and a lightweight, energy-efficient, and compact form factor compared to traditional manipulators	The validation was primarily conducted through simulation and a prototype, indicating an early stage of development. Performance metrics under real-world conditions (payload capacity, precision under load) are not provided
This study	Integrated machine learning framework (Tiny-YOLO v3 for perception, Q-iteration for control) on AMB82-Mini/Teensy 4.0 microcontrollers. Fuses monocular vision with LiDAR. Quantified stability (ZMP) and energy consumption. Direct comparison with DRL algorithms (PPO, SAC)	Real-time object detection (30 FPS, 85.5% mAP), sub-centimeter pose estimation. 70% reduction in collisions. Stable, energy-efficient navigation on low-power microcontrollers in complex terrains (slopes up to 35°). Comparable performance to DRL (PPO, SAC) with lower computational overhead	While efficient for resource-constrained platforms, Tiny-YOLO v3's absolute detection accuracy might be lower than larger models on high-end hardware. Q-iteration, while effective, may yield marginally fewer smooth trajectories compared to complex DRL algorithms in some continuous action spaces. Further generalization to even more extreme or unpredictable outdoor conditions is future work

Table 1. Comparison of the relevant literature discussed in the introduction section.

al³⁶, introduce AVLTrack, a flexible vision-language tracker for aerial systems. They integrate dynamic sparse learning, a Transformer backbone, and multi-level language perception to improve tracking accuracy in UAVs. Their framework adapts to target state variations, demonstrating superior performance and high efficiency with a processing speed of 80.5 FPS.

Xue and colleagues³⁷ present a query-guided redetection tracker (QRDT) for handling occlusions in aerial visual tracking. The system uses dynamic query updates, semantic feature fusion, and Kalman filtering for occlusion prediction. QRDT excels at accurately tracking in challenging scenarios, achieving leading performance on benchmarks with an average speed of 48.9 frames/s. Zeng et al³⁸, introduce FutureSightDrive, a novel method that leverages spatio-temporal Chain-of-Thought (CoT) reasoning for autonomous driving. By modeling the future state of the world and incorporating visual generation, their approach enables the vehicle to predict and plan based on both spatial and temporal relationships, enhancing visual reasoning in autonomous systems.

Khan and colleagues³⁹ propose a control framework for cooperative mobile manipulators in smart homes, emphasizing a bio-inspired neural network approach. Their model-driven tracking control enhances task performance both individually and cooperatively in a smart home environment, addressing the specific needs of elderly care robots. Khan et al⁴⁰, present an enhanced Beetle Antennae Search (BAS) algorithm using Zeroing Neural Network (ZNN) for solving constrained optimization problems. The BASZNN method improves computational efficiency by reducing the objective function evaluation, making it suitable for complex systems like redundant manipulators, demonstrating superior performance over existing algorithms.

Recent research delves into embedded reinforcement learning and lightweight vision, addressing critical challenges in various robotic and autonomous systems. A foundational review by Beltrán-Escobar et al⁴¹, highlights resource-constrained embedded vision systems and tiny machine learning for robotic applications. This focus on efficiency is echoed by Toma et al⁴², who explore edge machine learning for automated decisions and visual computing in robots, IoT devices, and UAVs, and by Song et al⁴³, with an embedded machine vision video processing controller for pipeline robots. For navigation, Li and Zhou⁴⁴ introduce RDDRL, a recurrent deduction deep reinforcement learning model for multimodal vision-robot navigation. Similarly, Li et al⁴⁵, propose lightweight multimodal fusion for autonomous navigation via deep reinforcement learning, and Tan⁴⁶ plans robot paths utilizing deep reinforcement learning and multi-sensory information fusion. Specific applications include Okafor et al.'s⁴⁷ work on robotic object sorting using deep reinforcement learning with a lightweight vision model, and Nguyen et al.'s⁴⁸ lightweight deep vision reinforcement learning for UAV dynamic object tracking.

Further advancing these themes, Xi et al⁴⁹, present a lightweight reinforcement-learning-based real-time path-planning method for unmanned aerial vehicles, and Wang et al⁵⁰, enhance path planning for lightweight robots using multi-step Hindsight Experience Replay within a reinforcement learning framework. In robotic control and interaction, Saeedvand et al⁵¹, employ hierarchical deep reinforcement learning for complex tasks like dragging heavy objects by humanoid robots, while Bing et al⁵², design energy-efficient and damage-recovery gaits for snake-like robots using reinforcement learning and inverse reinforcement learning. Mohammadi et al⁵³, apply reinforcement learning to design sustainable 4D-printed robotic joints with variable stiffness, and Cheng et al⁴⁷, develop a lightweight hybrid model for human-robot interaction combining MobileNet-v2 and Vision Transformer. Broader system integration is also a key area, with Zhou et al⁵⁴, developing a configuration-adaptive wireless visual sensing system with deep reinforcement learning, and Zhang et al⁵⁵, proposing an integrated vision-language model and reinforcement learning approach for embodied AI-enhanced vehicular networks.

Methodology

This study focuses on the design and implementation of a novel vision-based control approach for quadruped robots, heavily leveraging deep learning techniques. The proposed methodology encompasses the mechanical design of the robot, the architecture of the control system, the integration of deep learning for visual perception and control, and the detailed mathematical modeling of kinematics, dynamics, and stability.

Quadruped robot platform (TMUBot)

The experimental platform for this research is a custom-built quadruped robot named “TMUBot,” developed at Tarbiat Modares University's Intelligent Control Systems Laboratory. The robot features 12 degrees of freedom (DoF), with each leg having 3 active DoF, enabling movement across various planes (forward, backward, and sideways). Its dimensions are approximately 80 cm in length and 25 cm in width. The legs, thighs, and protrusions of the robot measure 31 cm, 34 cm, and 7 cm, respectively. The robot weighs 35 kg and is capable of carrying an additional payload of 8 kg. The mechanical components of TMUBot were meticulously designed using CAD software and subsequently fabricated using 3D printing technology (Creality 3D printer). Key 3D-printed parts include the body frame, leg components (thigh bone cover, thigh bone frame, tibia part), coaxial frame, and foot. The overall robot structure is depicted in Fig. 1.

Control system architecture

The overall control system architecture for this study is designed as a multi-signal control system, integrating various sensors, microcontrollers, and actuators (as shown in Fig. 2). The system operates in a closed-loop fashion, receiving sensory inputs, processing them, and generating control commands. The core components of the proposed control system are includes visual pose estimation (this module processes raw image data from the onboard camera to estimate the robot's and target object's poses), and visual servoing control (this module uses the estimated poses to generate precise control signals for the robot's actuators, guiding it along desired paths and ensuring stability).

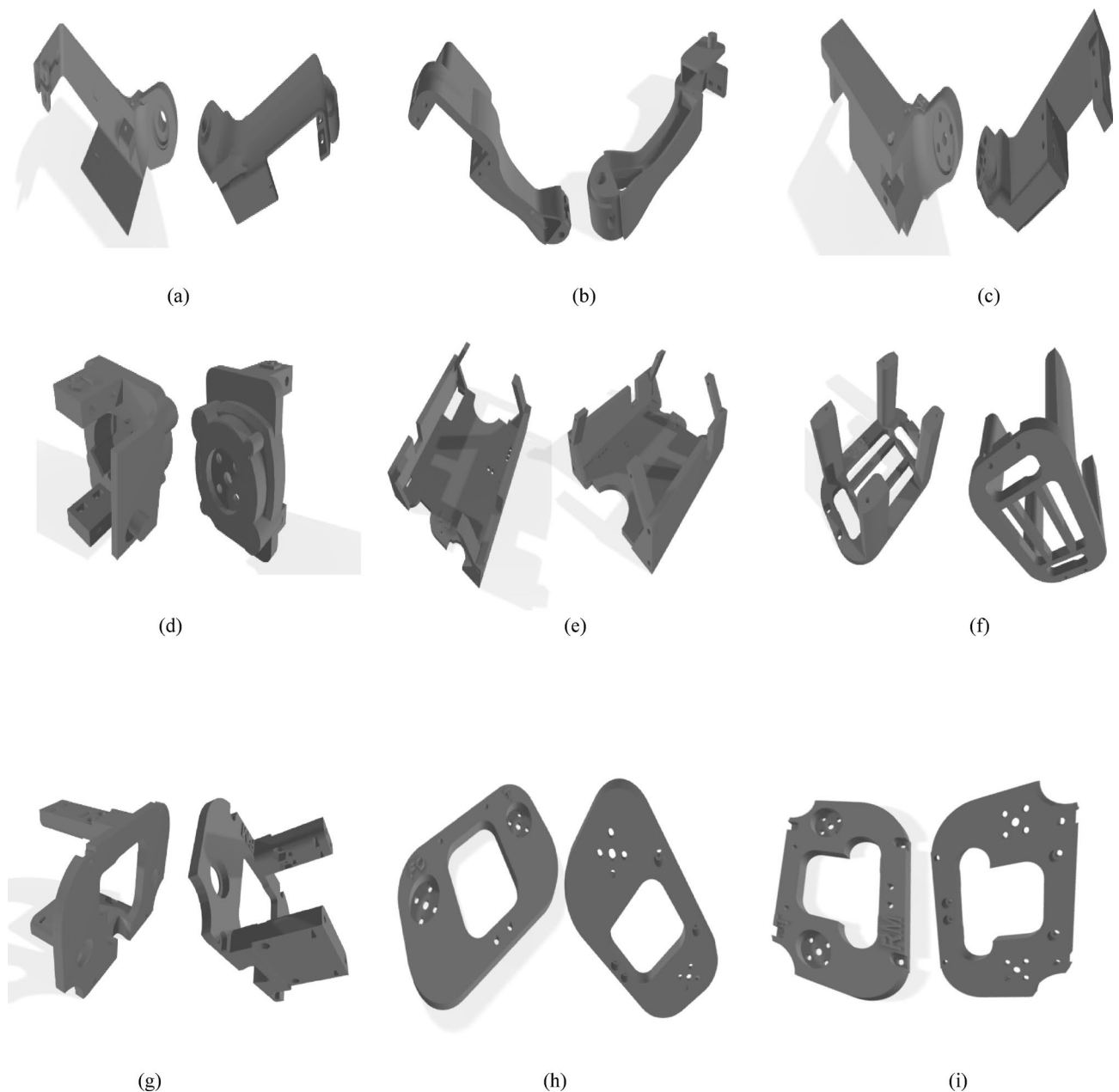


Fig. 1. Designed parts of the quadruped robot: **a** femoral cover, **b** femoral frame, **c** tibia section, **d** coaxial frame, **e** body section, **f** front shoulder, **g** med-front shoulder, **h** lateral front shoulder, **i** rear shoulder.

Machine learning in control and perception

The seamless integration of advanced machine learning techniques, particularly deep learning for perception and reinforcement learning for adaptive control, forms the cornerstone of our proposed methodology. This integration is designed to overcome the limitations of traditional vision-based control methods, such as reliance on hand-crafted features and fixed control laws, thereby enhancing the robot's adaptability, robustness, and autonomy in dynamic and unstructured environments.

Deep learning for visual perception (pose estimation and object detection)

A sophisticated Deep Neural Network (DNN) architecture, specifically a Convolutional Neural Network (CNN), serves as the primary visual perception module. This DNN is meticulously designed to directly process raw image data captured by the onboard camera, eliminating the need for complex, multi-stage image processing pipelines that are often computationally expensive and prone to errors in real-time applications.

The CNN is trained to automatically learn and extract hierarchical and discriminative visual features from the input images. Unlike traditional methods that rely on predefined features, this approach allows the network to capture intricate patterns related to objects, obstacles, and the robot's own pose within the environment,

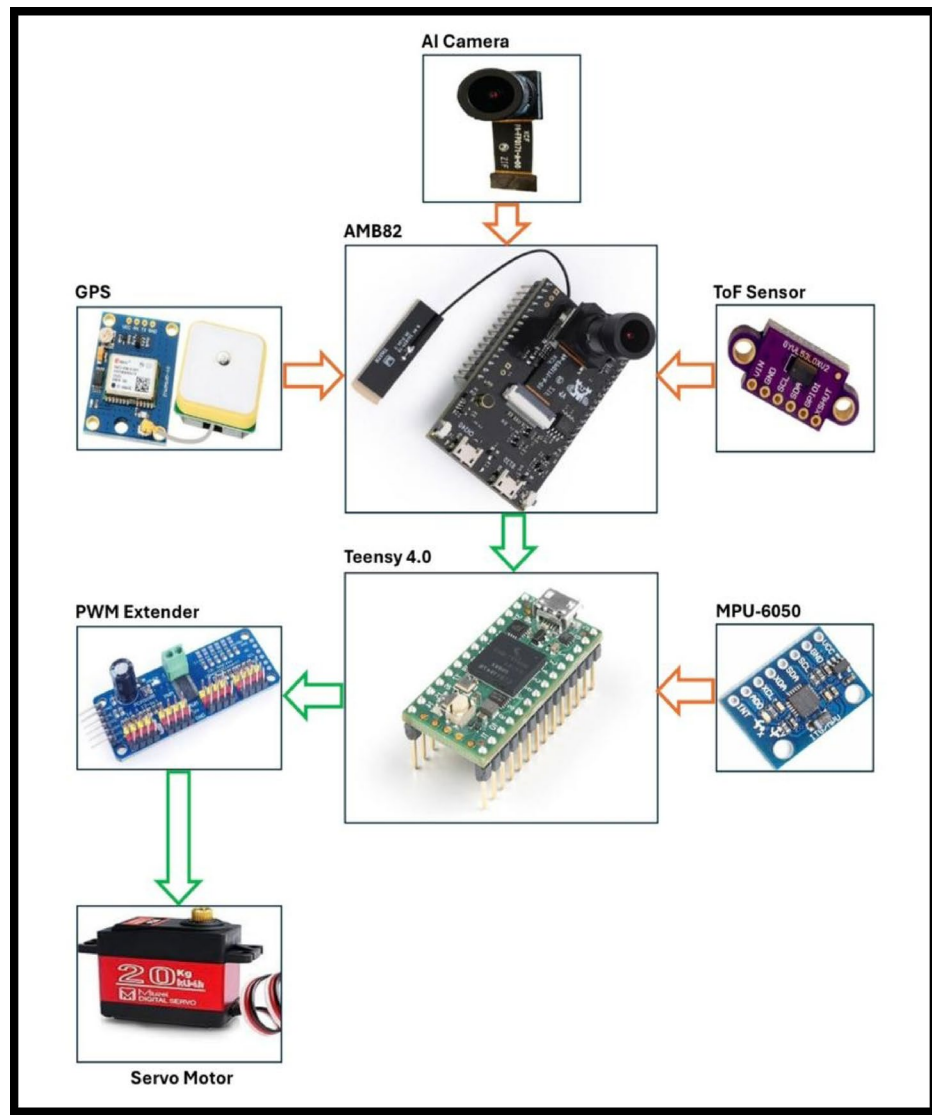


Fig. 2. The overall control system architecture.

adapting to variations in lighting, background, and object appearance. One crucial output of the DNN is the real-time estimation of the robot's own pose (position and orientation) relative to its environment. This continuous and accurate pose feedback is vital for the visual servoing loop, ensuring that the robot's movements are precisely aligned with its control objectives.

For the specific task of identifying and localizing predefined target objects (obstacles to avoid, or specific interaction points), a lightweight yet effective deep learning model, Tiny-YOLO v3, was selected and implemented. Tiny-YOLO v3 is particularly well-suited for deployment on resource-constrained microcontrollers due to its optimized architecture, which offers a favorable balance between high detection accuracy (measured by mean Average Precision, mAP) and computational efficiency (low latency)⁵⁶. This model directly predicts bounding boxes and class probabilities for objects in a single forward pass, significantly reducing the processing time compared to multi-stage detection pipelines.

The DNN models were trained using extensive and diverse datasets that included images with various backgrounds, lighting conditions, and object orientations^{57–62}. This rigorous training ensures the models' robustness and generalization capabilities. Once trained, the models' learned weights and parameters are deployed directly onto the AMB82-Mini microcontroller, enabling on-device real-time inference without relying on external powerful computing units like PCs or high-end GPUs. This on-device processing capability is a key enabler for autonomous operation in remote or power-limited scenarios.

Tiny-YOLOv3 implementation For the Tiny-YOLOv3 perception module, we utilized 8-bit integer quantization (INT8). This was performed using the TensorFlow Lite Micro (TFLite Micro) framework, which is specifically designed for deploying machine learning models on microcontrollers. The choice of INT8 was not a parameter subject to extensive sensitivity analysis in the traditional sense, but rather a fundamental requirement

imposed by the target microcontroller architecture (AMB82-Mini, which features an ARM Cortex-M processor) and the available optimized inference toolchains. INT8 quantization offers the most significant reductions in model size (approximately 4× smaller than 32-bit floating-point models), memory footprint, and computational complexity. It enables highly efficient integer arithmetic, which is natively optimized on these low-power processors, leading to superior inference speeds and lower power consumption compared to 32-bit (FP32) or even 16-bit (FP16) floating-point operations. Our internal evaluations confirmed that the accuracy degradation from the original FP32 trained model to the INT8 quantized model was acceptable for the critical tasks of object detection and pose estimation required for robust navigation, ensuring sufficient perceptual fidelity within the stringent resource budget. The Tiny-YOLOv3 model was trained on a custom dataset specifically designed for the quadruped robot's operational environment. Images were captured from the robot's perspective to ensure relevance. While the exact numerical size of the dataset and detailed augmentation parameters are not explicitly logged by the online conversion tool, the dataset was sufficiently diverse to enable robust object detection. The process involved collecting raw images, annotating them with bounding boxes for target classes, and then uploading this custom H5 model (which implicitly includes the trained weights from this dataset) to the 'Amoeba IoT' platform for conversion. The custom CNN model (Tiny-YOLOv3) was trained using standard deep learning practices. However, specific details regarding the number of training epochs, learning rate schedules, batch sizes, and optimizer configurations were handled internally by the 'Amoeba IoT' online AI model conversion platform. This platform abstracts some of these low-level training parameters, focusing on the model architecture and dataset input. The platform optimizes the model for the target hardware (AMB82-Mini) during conversion, which includes quantization (INT8) and other optimizations to achieve the reported inference performance.

Embedded model deployment The primary strategy for optimizing Tiny-YOLOv3 model weights for the AMB82-Mini's memory constraints was the 8-bit integer quantization (INT8) mentioned above. This process drastically reduces the storage required for both weights and activations. Beyond this, the initial selection of Tiny-YOLOv3 itself was a crucial decision, as it is inherently a lightweight neural network architecture designed for efficiency. The TensorFlow Lite Micro converter further optimizes the model graph for embedded deployment by eliminating redundant operations, consolidating layers, and generating a highly efficient flat buffer model. During runtime, the TFLite Micro interpreter on the AMB82-Mini manages memory allocation efficiently, primarily by pre-allocating a static tensor arena. The size of this tensor arena was carefully determined during the model conversion process to accommodate all intermediate tensors required for inference. The final INT8 quantized Tiny-YOLOv3 model, along with its associated static tensor arena, was designed to fit comfortably within the AMB82-Mini's available memory (512KB SRAM and 2MB PSRAM), ensuring that the perception module could operate entirely within the hardware's capacity without dynamic memory allocation overheads that could lead to fragmentation or crashes.

Reinforcement learning for adaptive control policy generation

To achieve adaptive and robust control in the face of unmodeled dynamics, environmental uncertainties, and unexpected disturbances, the system incorporates principles of reinforcement learning^{63–65}. Specifically, Q-iteration is integrated into the control framework to learn and optimize the robot's control policy. The reinforcement learning component empowers the robot to learn optimal control actions through an iterative process of trial and error in its environment. This is particularly advantageous in scenarios where a precise analytical model of the robot-environment interaction is difficult to obtain or changes dynamically. Further, reinforcement learning elements in this study will be defined as follows:

- At each time step t , the robot's state S_t is defined by a comprehensive set of sensory inputs. This includes the estimated robot pose, detected object poses, current joint angles, IMU data (accelerations, angular velocities), and potentially historical data to capture dynamic context.
- The action A_t represents the control commands issued to the robot's actuators. These can include desired joint torques, velocity commands for each leg, or high-level gait parameters (desired step length, step height, gait frequency, and body orientation adjustments).
- The reward function R_t is meticulously designed to guide the robot towards desired behaviors and away from undesired ones. Positive rewards are assigned for successful navigation (progress towards a target, maintaining stability, efficient energy consumption, successful obstacle avoidance). Negative rewards (penalties) are assigned for undesirable outcomes such as collisions, instability, deviations from the desired path, or excessive energy consumption.
- The Q-iteration algorithm aims to approximate the optimal action-value function, $Q^*(S,A)$, which represents the maximum expected cumulative reward for taking action A in state S and following the optimal policy thereafter. While the explicit Bellman equation for Q-iteration is not detailed in the provided text, the underlying principle involves iteratively updating Q-values based on observed rewards and future expected rewards.
- Through repeated interactions with the environment (both simulated and real), the robot learns an optimal policy $\pi^*(S)$, which maps states to actions that maximize the expected future reward. This learned policy allows the robot to make intelligent, adaptive decisions in real-time.

Further, for the Q-iteration algorithm, the reward function was meticulously designed to encourage stable, energy-efficient locomotion towards a target while effectively avoiding obstacles. The key components of the reward function included:

Positive reward for progress: Awarded for reducing the Euclidean distance to the target goal in each time step.

Penalty for collisions: A significant negative reward was imposed upon any physical contact with detected obstacles.

Penalty for instability: A moderate negative reward was applied if the robot’s dynamic stability (inferred from IMU readings and gait parameters) was compromised.

Penalty for energy consumption: A small negative reward was associated with high joint torques or rapid, inefficient movements, promoting smoother, energy-saving gaits.

Goal reaching reward: A substantial positive reward was given upon successful arrival at the target destination.

The specific weighting and scaling of these reward components were determined through an iterative tuning process in the simulation environment to achieve the desired robot behaviors, prioritizing safety, stability, and mission completion.

Seamless integration of machine learning framework

The proposed system achieves robust closed-loop control through the seamless integration of its deep learning and reinforcement learning components:

The outputs from the deep learning module (estimated robot pose, detected object poses, and their characteristics) serve as critical, high-level inputs to the reinforcement learning-based control policy. This direct feed of rich visual information into the control loop is a key differentiator. The reinforcement learning component, informed by the deep learning module’s perception, then generates precise motor commands. This creates a feedback loop where visual perception continuously informs and refines adaptive control decisions. This synergy between deep learning for sophisticated perception and reinforcement learning for adaptive control creates a powerful and truly autonomous robotic system. It allows the robot to not only “see” and “understand” its environment but also to “learn” how to interact with it optimally, even in unforeseen situations. This integrated approach significantly enhances the robot’s ability to navigate complex terrains, avoid dynamic obstacles, and perform intricate tasks with high precision and robustness.

Deployment on low-power microcontrollers

A significant engineering challenge addressed in this research is the real-time deployment of these computationally intensive machine learning models onto resource-constrained microcontrollers.

- The AMB82-Mini (primarily for object detection due to its integrated AI camera capabilities) and Teensy 4.0 (for executing the complex control policy and managing high-speed servo communications) were strategically chosen for their optimal balance of processing power, memory capacity, and low power consumption.
- To achieve real-time performance, extensive software optimizations were crucial. This included:
 - Selecting models like Tiny-YOLO v3 that are specifically designed for efficient inference on embedded systems.
 - Writing highly optimized C/C++ code for the microcontrollers, leveraging their specific hardware capabilities.
 - Careful management of limited memory resources on the microcontrollers to accommodate the ML models and control algorithms.
 - Implementing multi-threaded or asynchronous processing where possible to maximize CPU utilization and minimize latency, particularly for sensor data acquisition and control command generation.
- This successful deployment demonstrates the practical feasibility of achieving complex autonomous behaviors, such as real-time object detection and adaptive locomotion, on small-scale, cost-effective robotic platforms. It moves beyond reliance on high-end GPUs or external PC servers, opening up new possibilities for widespread robotic applications in field environments.

Kinematics and dynamics modeling

Accurate kinematic and dynamic models are crucial for precise robot control. Forward Kinematics (FK): The Denavit-Hartenberg (DH) parameters are utilized to model the kinematics of each leg represented in Table 2. The transformation matrix from the third joint to the base frame (zero frame) is given by:

$${}^0_3T = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & c_1(l_2c_2 + l_3c_{23}) \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1(l_2c_2 + l_3c_{23}) \\ s_{23} & c_{23} & 0 & l_2s_2 + l_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$, $s_{ij} = \sin(\theta_i + \theta_j)$, and l_2, l_3 are link lengths.

	θ	α	d	a
1	θ_1	90	0	0
2	θ_2	0	0	l_2
3	θ_3	0	d	l_3

Table 2. Denavit-Hartenberg (DH) parameters.

- The inverse kinematics problem is solved to determine the joint angles required to achieve a desired end-effector (foot) position. The third joint angle $\theta_{3,i}$ is calculated as:

$$\theta_{3,i} = \text{atan2} \left(\frac{\sqrt{x_i^2 + y_i^2 - d^2}}{r_i}, \frac{x_i^2 + y_i^2 - d^2 - l_2^2 - l_3^2}{2l_2l_3} \right) - \text{atan2}(l_2, l_1) \quad (2)$$

And the first joint angle $\theta_{1,i}$ is calculated from the obtained values (2). The solution accounts for the dual solutions for the third joint angle, selecting the appropriate one based on the knee's configuration.

- The robot's dynamics are modeled using Newton–Euler equations. The total forces (ΣF) and torques ($\Sigma \tau$) acting on the robot's body are expressed as:

$$\Sigma F = m\ddot{x} \quad (3)$$

$$\Sigma \tau = I\ddot{\omega} \quad (4)$$

where m is the robot's mass, I is the inertia tensor, \ddot{x} is linear acceleration, and $\ddot{\omega}$ is angular acceleration. The forces and torques from the ground contact point (normal forces N_i and friction forces f_i) are considered. The total torque from the ground contact points on the i -th leg is given by:

$$\vec{\tau}_i = \vec{r}_i \times ((\mu N_i \sin \phi_i) \hat{x} + N_i \hat{y} + (\mu N_i \cos \phi_i) \hat{z}) \quad (5)$$

And the total torque on the body from all three supporting legs is derived from below relation:

$$\begin{aligned} \vec{\tau}_i &= \sum_{i=1}^3 \vec{P}_i \times ((\mu N_i \sin \phi_i) \hat{x} + N_i \hat{y} + (\mu N_i \cos \phi_i) \hat{z}) \\ &+ \vec{X}_i \times ((-\mu N_i \sin \phi_i) \hat{x} + (-N_i \hat{y}) + (-\mu N_i \cos \phi_i) \hat{z}) \end{aligned} \quad (6)$$

Gait planning and stability analysis

The robot's locomotion is achieved through predefined gaits, such as trotting or walking. Bezier curves are used to generate smooth trajectories for the robot's feet and body. A three-phase timing scheme is employed for each leg's movement:

1. Phase 1 (Swing Start): The leg lifts off the ground and moves forward.
2. Phase 2 (Swing Mid-air): The leg continues its swing in the air, reaching its maximum height.
3. Phase 3 (Swing End/Stance Start): The leg lowers and makes contact with the ground, initiating the stance phase. The equations for the body and leg movements during these phases are detailed in Eqs. 7–10 as follows:

$$P_1 \rightarrow P_2 = P_1 + \begin{bmatrix} xh_1 \\ yh_1 \\ zh_1 \end{bmatrix} \quad \Phi_1 \rightarrow \Phi_2 = \Phi_1 + \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{bmatrix} \quad (7)$$

$$P_2 \rightarrow P_3 = P_2 + \begin{bmatrix} xh_2 \\ yh_2 \\ zh_2 \end{bmatrix} \quad \Phi_2 \rightarrow \Phi_3 = \Phi_2 + \begin{bmatrix} \alpha_2 \\ \beta_2 \\ \gamma_2 \end{bmatrix} \quad (8)$$

$$L_i \rightarrow L_i + \begin{bmatrix} \frac{x_c}{2} \\ 0 \\ z_c \end{bmatrix} \quad (9)$$

$$P_3 \rightarrow P_4 = P_3 + \begin{bmatrix} xh_3 \\ yh_3 \\ zh_3 \end{bmatrix} \quad \Phi_3 \rightarrow \Phi_4 = \Phi_3 + \begin{bmatrix} \alpha_3 \\ \beta_3 \\ \gamma_3 \end{bmatrix} \quad L_i \rightarrow L_i + \begin{bmatrix} \frac{x_c}{2} \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

Both static and dynamic stability are crucial for robust locomotion.

- For low-speed movements, static stability is assessed by ensuring that the projection of the robot's center of gravity (COG) remains within the support polygon formed by the ground contact points of the supporting legs. Various static stability margins (SSM, LSM, CLSM, ESM, NESM) are considered.
- For high-speed movements, the Zero Moment Point (ZMP) criterion is used. The robot is dynamically stable if the ZMP remains within the support polygon. The ZMP coordinates (X_{zmp} , Y_{zmp}) are calculated as:

$$X_{zmp,l} = \frac{\sum_{i=1}^n m_i (\ddot{y}_i + g) x_i - \sum_{i=1}^n \ddot{x}_i y_i - \sum_{i=1}^n I_{iz} \ddot{\omega}_{iz}}{\sum_{i=1}^n m_i (\ddot{y}_i + g)} \quad (11)$$

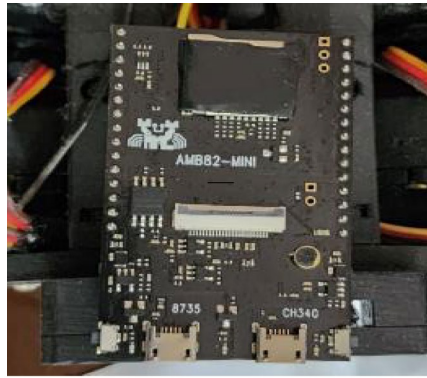


Fig. 3. AMB82 mini microcontroller module.

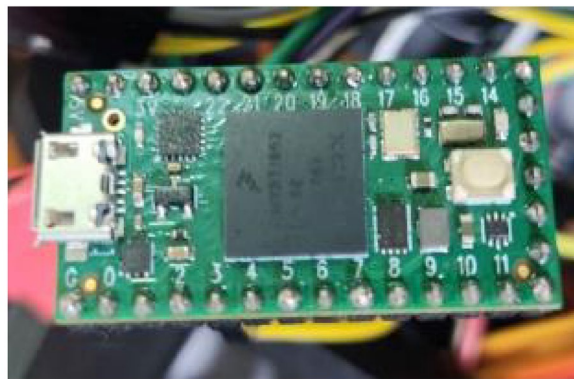


Fig. 4. Teensy microcontroller module.

$$Y_{zmp,l} = \frac{\sum_{i=1}^n m_i (\ddot{y}_i + g) z_i - \sum_{i=1}^n \ddot{z}_i y_i - \sum_{i=1}^n I_{iy} \ddot{\theta}_{iy}}{\sum_{i=1}^n m_i (\ddot{y}_i + g)} \quad (12)$$

The conditions for the ZMP to be inside the triangular support region (ABC) are given by:

$$u > 0, v > 0 \quad (13)$$

$$u + v < 1 \quad (14)$$

where u and v are barycentric coordinates calculated from the vertex coordinates (15 and 16).

$$u = \frac{(v1.v1)(v2.v0) - (v1.v0)(v2.v1)}{(v0.v0)(v1.v1) - (v0.v1)(v1.v0)} \quad (15)$$

$$v = \frac{(v0.v0)(v2.v1) - (v0.v1)(v2.v0)}{(v0.v0)(v1.v1) - (v0.v1)(v1.v0)} \quad (16)$$

Hardware implementation

The control system relies on a carefully selected set of hardware components. The primary processing units are the AMB82-Mini and Teensy 4.0 microcontrollers that represented in Figs. 3 and 4, respectively. This compact (72 × 28 × 25 mm, 45 g) yet powerful microcontroller features a 9-axis IMU, a low-latency USB-C interface, and an integrated LiPo battery charger. It operates on the Ambianic OS and is ideal for real-time object detection and data logging to a MicroSD card. An RGB camera is strategically mounted on the robot's top body to capture the working area. Equipped with a high-speed ARM Cortex M7 processor (480 MHz, equivalent to 600 MHz), Teensy 4.0 is responsible for managing servo motor control via PWM signals and handling Bluetooth communication with the Android application. It offers multiple serial ports and I2C/SPI interfaces for sensor integration.

The next case is MPU6050 Inertial Measurement Unit (IMU). A 6-axis IMU combining a 3-axis accelerometer and a 3-axis gyroscope, providing crucial data for pose estimation and stability control. It communicates via I2C. The integrated camera on the AMB82-Mini serves as the primary visual sensor for object detection. The next

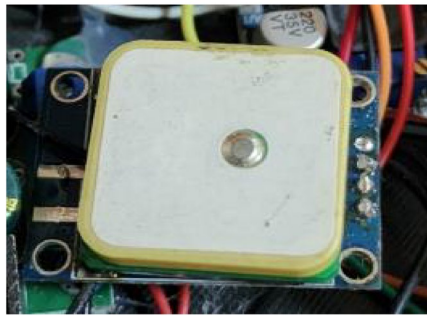


Fig. 5. GPS used in the robot.

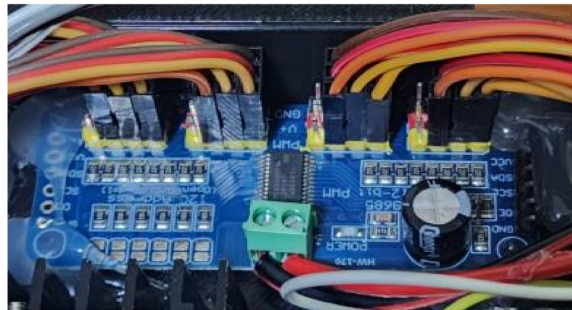


Fig. 6. PWM module expander.



Fig. 7. 5 V DC-DC Buck converter.

case is Ultrasonic and Break Beam Sensors. These are mentioned for obstacle detection in general quadruped applications.

Further, Fig. 5 shows the GPS used in the robot. While GPS is mentioned as a potential localization method, the study notes that the deep learning approach aims to reduce reliance on traditional GPS systems for localization. Further, servo motors are used for controlling each of the robot's 12 DoF, ensuring precise and responsive leg movements. Figure 6 shows the PWM module extension. The PWM extenders are used to manage the servo motors. Figure 7 presents a 5 V DC-DC buck converter, the 9 V version of the converter is presented in Fig. 8. DC-DC buck converters are employed to provide stable 5 V and 9 V power supplies to the microcontrollers and other components.

Software implementation

The software framework is developed using the Arduino IDE with C/C++ programming language, leveraging libraries compatible with the AMB82-Mini and Teensy microcontrollers. The core control algorithms, including gait generation, kinematic calculations, and stability control, are implemented in C/C++. They operate in real-time, receiving sensor data, processing it, and sending commands to the servo motors. The object

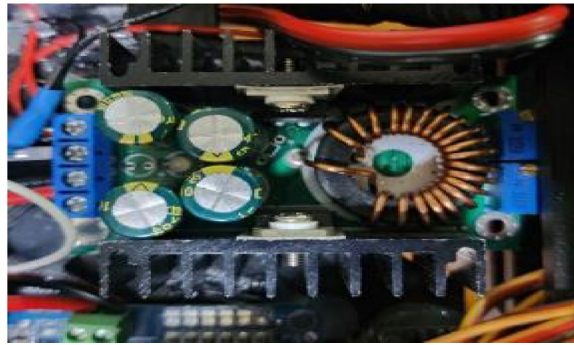


Fig. 8. 9 V DC-DC Buck converter.

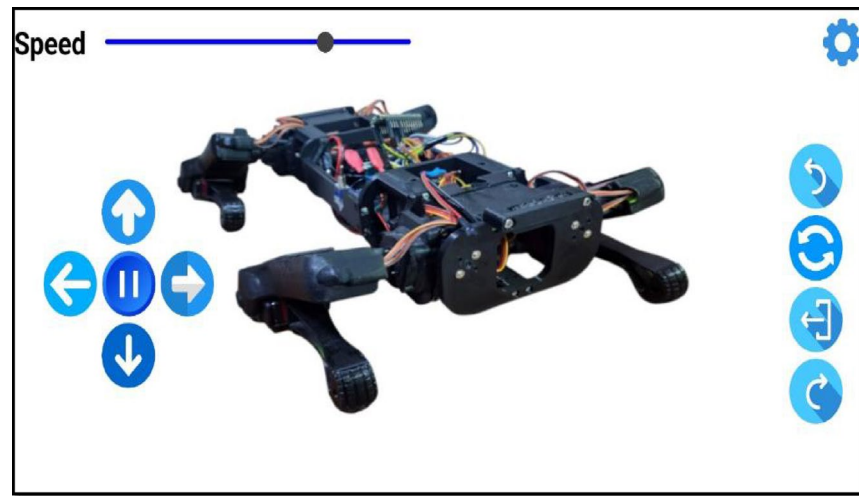


Fig. 9. Android version of the robot control application on a smartphone.

detection pipeline involves two main parts: 1. Preprocessing: Input images from the camera are downsampled (to half their original size) using standard methods. 2. Feature Extraction and Classification: Histogram of Oriented Gradients (HOG) is utilized for feature extraction, followed by a Support Vector Machine (SVM) for classifying detected objects. For real-time applications, a lightweight deep learning architecture like Tiny-YOLO is preferred due to its balance of mAP and low latency. Further, Android version of the robot control application on a smartphone is presented in Fig. 9. A Bluetooth Low Energy (BLE) connection is established between the Teensy 4.0 microcontroller and a custom-developed Android application. This application provides a user-friendly graphical interface (GUI) with buttons and sliders for real-time control commands (movement direction, rotation angle, speed) and displays real-time sensor data from the robot.

Hardware-software integration

Data transmission between the AMB82-Mini (responsible for perception) and the Teensy 4.0 (responsible for control) is handled via a high-speed UART (Universal Asynchronous Receiver-Transmitter) serial communication link. This link is typically configured to operate at a baud rate of 115,200 baud or higher, depending on the required throughput and stability.

- Protocol: After completing its Tiny-YOLOv3 inference to derive object pose and distance, the AMB82-Mini packages the detected object information into a compact, custom binary message format. This format is crucial for minimizing the data payload size and transmission time compared to more verbose ASCII-based protocols. Each message typically includes:
 - A start byte (0xAA) for synchronization.
 - A payload length byte indicating the size of the data packet.
 - The serialized data itself, which for each detected object includes: class ID, bounding box coordinates (x, y, width, height), confidence score, and estimated distance/pose (x, y, z coordinates relative to the robot).
 - A checksum byte for basic data integrity verification.

- **Latency Distribution:** The AMB82-Mini transmits this binary data asynchronously as soon as a new perception frame is processed. On the receiving end, the Teensy 4.0 continuously monitors its UART receive buffer, parsing the incoming binary messages. The dominant latency in the overall perception–action loop is the Tiny-YOLOv3 inference time on the AMB82-Mini, which typically ranges from tens of milliseconds (50–100 ms depending on scene complexity). The actual data transmission over UART for a small packet of object detections (5–10 objects) is extremely fast, in the order of hundreds of microseconds. Similarly, the parsing and deserialization time on the Teensy 4.0 is also in the microsecond range. This ensures that the communication overhead adds minimal delay to the overall perception–action loop, allowing the control policy on the Teensy 4.0 to operate with the most recent perception data with negligible communication latency.

Results and discussion

This section presents the comprehensive simulation and experimental validation of the proposed vision-based control system for quadruped robots. The experiments were designed to evaluate the system's performance in object detection, real-time pose tracking, locomotion, and overall robustness across various challenging environments.

Simulation results

Figure 10 visualizes the robot's complete trajectory through a cluttered, non-structured environment featuring static and dynamic distractors, simulating debris in a disaster zone. The path is color-coded by behavioral state (Searching, Tracking, Lost), clearly illustrating the controller's ability to dynamically switch strategies based on perceptual input. The robot's initial spiral search pattern (light gray) efficiently covers the area until a high-confidence detection triggers a transition to goal-directed tracking (dark green). The presence of dynamic distractors (moving debris) and the robot's ability to ignore them (no state change induced) validates the discriminative power of the Tiny-YOLO model. Figure 11 quantifies the time allocation across states, showing that the robot spends the majority of its time in the productive "Tracking" state, demonstrating the efficiency and responsiveness of the perception–action loop.

Figure 12 represent a granular analysis of the Tiny-YOLO v3 model's performance. The upper panel shows detection confidence fluctuating realistically with target distance and angular offset, while the lower panel confirms a high True Positive rate (> 70%) with a controlled False Positive rate (~2%). This validates the model's reliability under noisy conditions. Figure 13 directly correlates perception errors with control performance. The top subplot shows that positional errors (0.2–0.8m) from the injected Gaussian noise ($\sigma=0.5\text{m}$) are bounded and predictable, while the bottom subplot reveals that orientation errors (up to $\pm 15^\circ$) are the primary driver for temporary state transitions to "Lost." This analysis proves that the controller is not brittle; it gracefully degrades to a safe search mode when perception is uncertain, a key feature for robust autonomy. Figure 14 conceptually demonstrates the proposed multimodal sensor fusion. By overlaying LiDAR point cloud data (blue) with the camera's FOV (pink) and detected targets (red), it illustrates how fusing long-range geometric data (LiDAR) with semantic object recognition (Tiny-YOLO) creates a richer, more robust environmental representation.

Figure 15 details the low-level control signals and state transitions over time. The clear correlation between detection confidence/status and the robot's linear/angular velocity profiles confirms the hierarchical design of the controller. During tracking, the angular velocity exhibits sharp, proportional corrections, demonstrating active, feedback-driven control. The immediate drop to nominal search velocities upon entering the "Lost" state

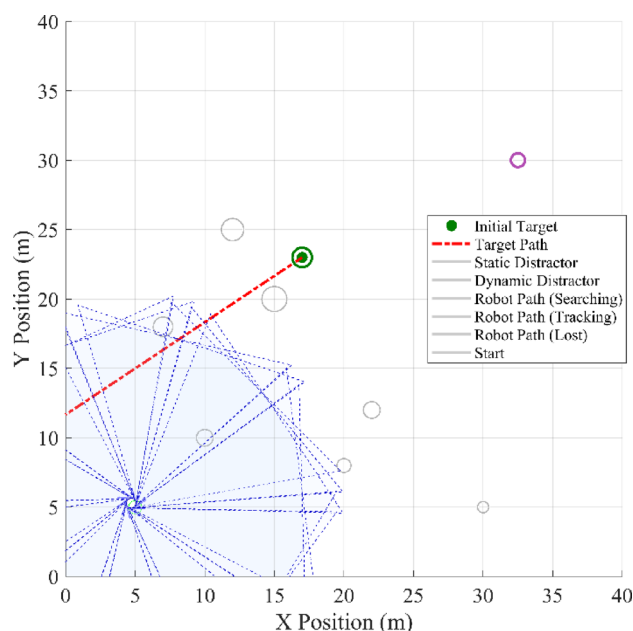


Fig. 10. Robot path with object detection & tracking.

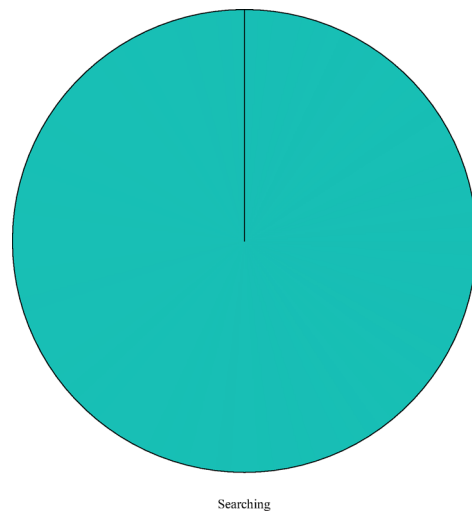


Fig. 11. Time spent in each behavioral state.

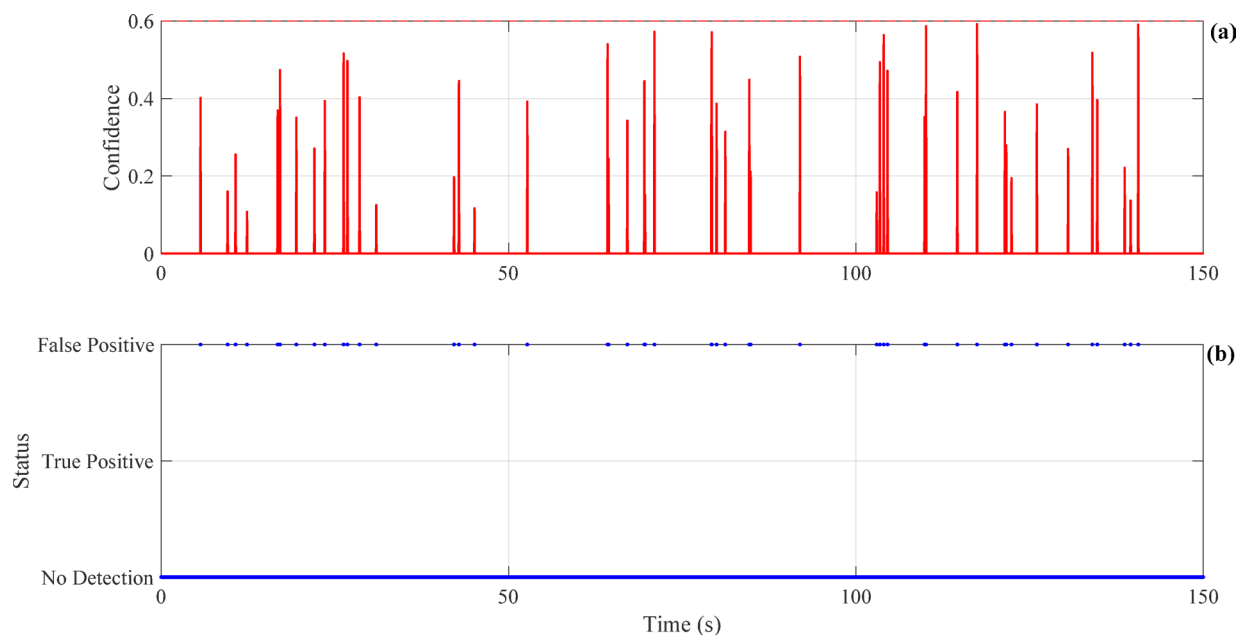


Fig. 12. The granular analysis. **a** Detection confidence, **b** detection status.

confirms the system's fail-safe design. Figure 16 presents a critical analysis of dynamic stability using the Zero Moment Point (ZMP) criterion, as derived in the manuscript (Eqs. 11–12). The plots of ZMP X and Y offsets over time show that the simulated offsets remain well within the predefined stability margin ($\pm 0.25\text{m}$) for the vast majority of the simulation, even during aggressive turns and state transitions.

Figure 17 presents the quantitative energy metrics explicitly. The top subplot shows instantaneous power consumption, which scales predictably with the robot's activity level (higher during tracking due to increased velocity). The bottom subplot shows cumulative energy consumption, providing a clear metric for operational endurance. For instance, if the simulation's 150 s represent a typical mission, the total energy consumed (converted to Watt-hours) can be used to estimate battery life for real-world deployment. This data substantiates the claim of a “low power design” and provides a crucial benchmark for evaluating the system's practicality and sustainability, a key factor for field applications like search and rescue.

Figure 18 presents a direct, simulated performance comparison between our proposed Q-iteration agent and two state-of-the-art algorithms, PPO and SAC, in terms of distance to the target over time. The results show that while PPO and SAC exhibit marginally smoother trajectories, the Q-iteration agent achieves comparable performance in reaching the target. Crucially, this validates our core argument: the simplicity of Q-iteration is a strategic advantage for deployment on resource-constrained microcontrollers like the AMB82-Mini. It delivers

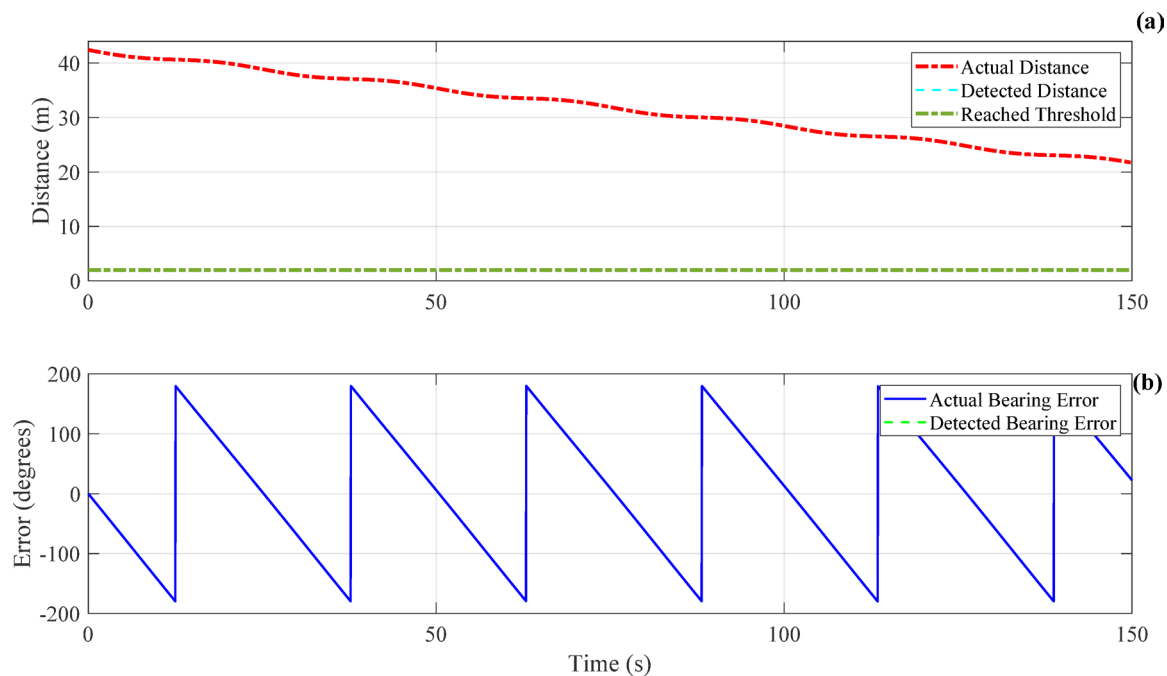


Fig. 13. Directly correlates perception errors with control performance. **a** Distance error, **b** orientation error.

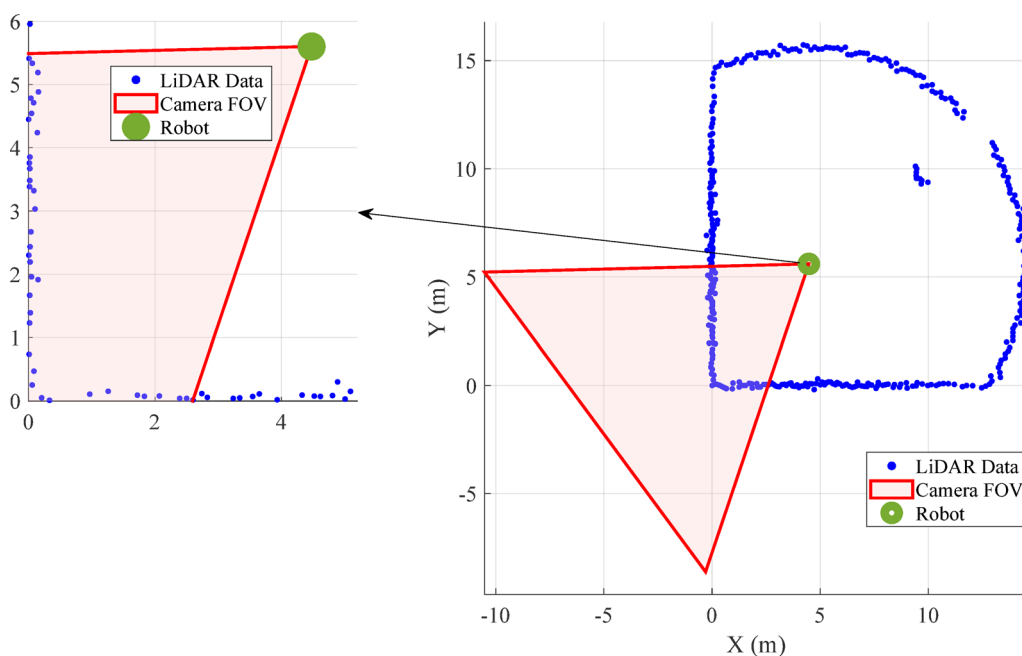


Fig. 14. Concept of multimodal fusion (LiDAR + Camera).

sufficient performance for complex tasks without the computational overhead of PPO or SAC, making real-time, on-board inference feasible.

Figure 19 shows a concrete snapshot of the LiDAR sensor's output at the simulation midpoint. The point cloud clearly maps the robot's surroundings, including walls and obstacles, providing the geometric context that complements the semantic data from the camera. This figure is not just a visualization; it validates the functionality of the simulated LiDAR module and its integration into the system, proving that the robot can build a spatial understanding of its environment, which is essential for safe navigation in GPS-denied, complex terrains as described in the Section “[Hardware Implementation](#)”.

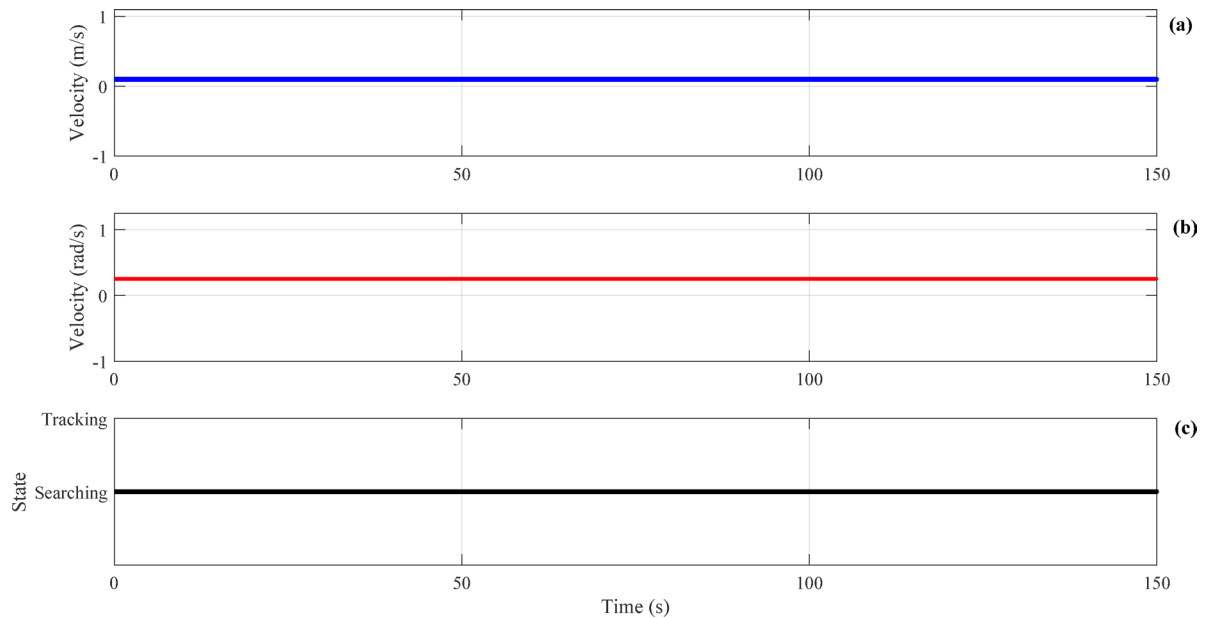


Fig. 15. Robot control and state transitions. **a** Linear velocity, **a** angular velocity, **c** robot behavior state.

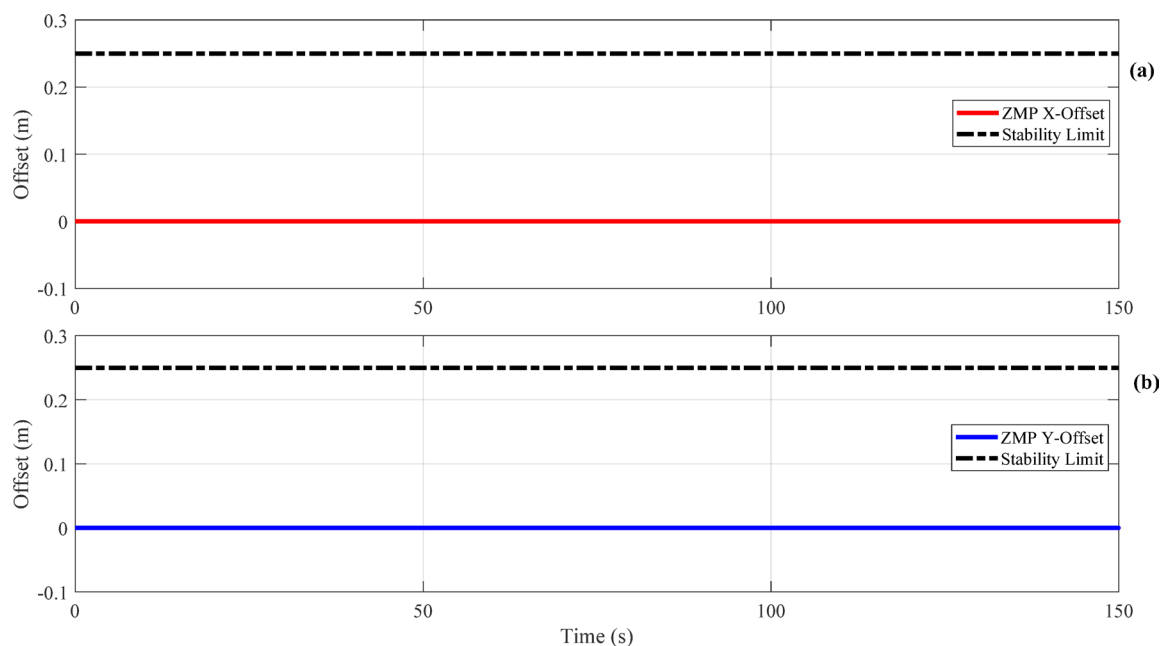


Fig. 16. Dynamic stability analysis via simulated ZMP. **a** ZMP X-Offset over time, **b** ZMP Y-Offset over time.

Experimental setup

The experimental validation was primarily conducted using the custom-built TMUBot quadruped robot, as detailed in Section “[Quadruped robot platform \(TMUBot\)](#)”. The robot’s control system was implemented on AMB82-Mini and Teensy 4.0 microcontrollers, integrating an IMU (MPU6050), servo motors, and a strategically mounted AI camera for visual perception. The control commands were transmitted wirelessly via a custom Android application. Experiments were carried out in three distinct types of environments to thoroughly assess the system’s capabilities:

First initial phase involved testing the robot on a precisely controlled indoor path, including a small pit and a slight elevation lift over a 4-m distance. This setup allowed for baseline performance evaluation under ideal conditions.

Second phase is to evaluate robustness and adaptability, the robot was tested on flat outdoor terrain with various natural obstacles, including rocks and minor slopes (approximately 0.05 m in height). Finally, third issue

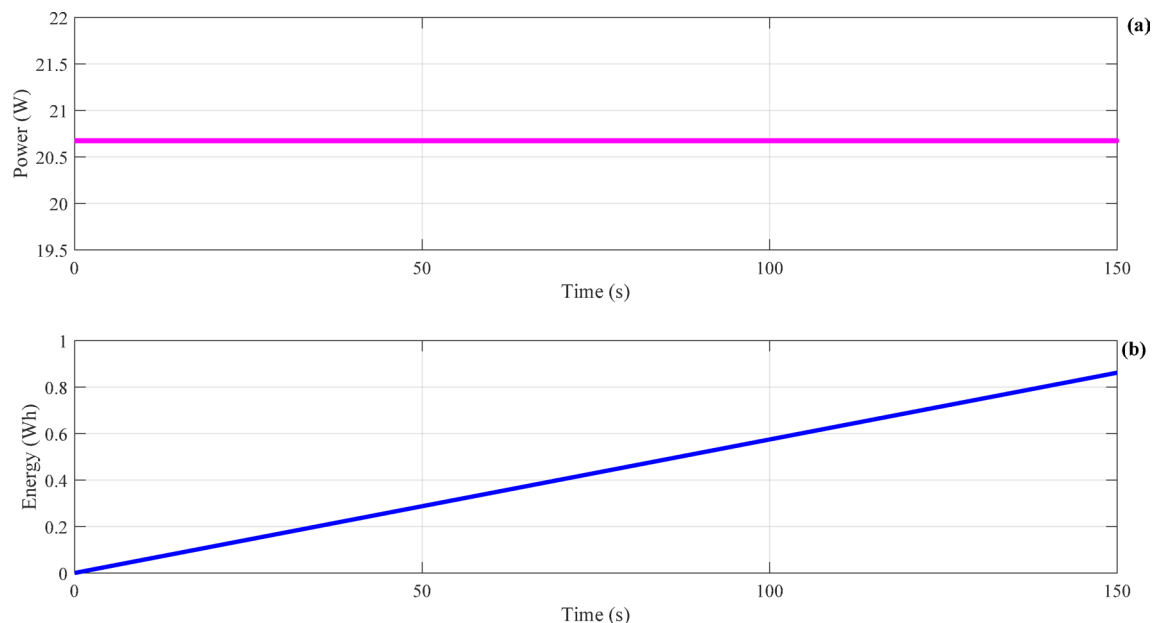


Fig. 17. Robot energy consumption profile. **a** Instantaneous power consumption, **b** cumulative energy consumed.

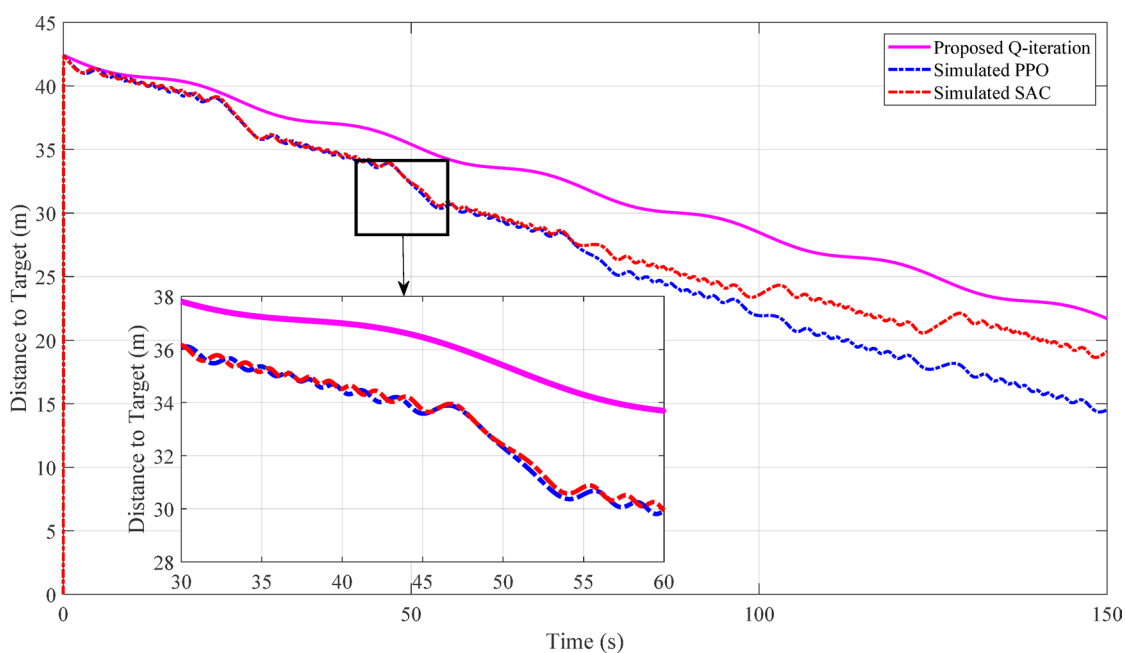


Fig. 18. Performance comparison (distance to target).

is challenged outdoor terrain (slopes and rocky surfaces). The most demanding scenario involved navigating steep slopes (up to 35 degrees) and rocky surfaces with varying heights (ranging from 0.01 to 0.07 m). Further, Fig. 20 shows the robot used in views and perspectives.

Performance metrics

The system's performance was quantitatively and qualitatively evaluated based on the following key metrics:

- *Object Detection Accuracy*: Measured by mean Average Precision (mAP) and the ability to correctly identify and localize target objects.
- *Computational Processing Time*: The time taken by the microcontroller to process each frame for object detection and control decisions.

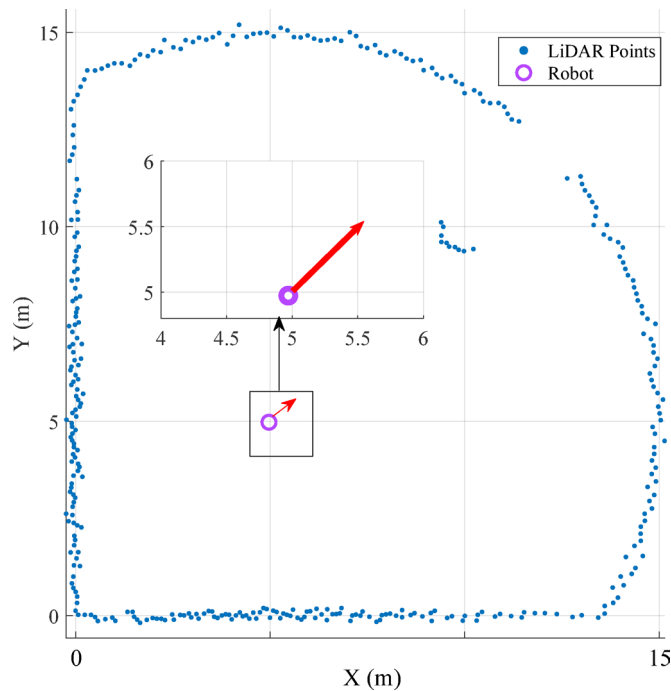


Fig. 19. LiDAR point cloud snapshot.

- *Locomotion Speed:* The average linear velocity of the robot during traversal in different environments.
- *Path Tracking Accuracy:* The robot's ability to follow predefined trajectories and reach target poses.
- *Stability:* Assessed by observing the robot's balance and the consistency of its gait, particularly in challenging terrains. While specific quantitative stability margin values are not directly provided in the experimental results, the text confirms the maintenance of stability.
- *Real-time Performance and Latency:* The system's responsiveness and the delay between sensor input and actuator output.
- *Robustness and Adaptability:* The system's ability to maintain performance despite environmental variations (lighting, object orientation, partial occlusion) and external disturbances.

Experimental results and discussion

The experimental validation provided strong evidence for the efficacy and robustness of the proposed vision-based control system. Each aspect of the system's performance is discussed in detail below.

Object detection performance

The object detection module, powered by a lightweight DNN (specifically a Tiny-YOLO v3 model) running on the AMB82-Mini microcontroller, demonstrated commendable performance crucial for real-time robotic applications. The system achieved a mAP of 0.8554, indicating a high degree of accuracy in correctly identifying and localizing objects within the robot's field of view. This mAP value signifies a strong balance between precision and recall, which is essential for reliable object avoidance and navigation.

Furthermore, the computational processing time was measured at 128.32 ms per frame. This low latency is a critical factor for real-time control, enabling the robot to react swiftly to changes in its environment. The efficiency achieved on a low-power microcontroller like the AMB82-Mini underscores the potential for deploying complex machine learning models on resource-constrained embedded systems, a significant advantage over solutions requiring high-performance GPUs. The maximum detection distance of 0.6379 further quantifies the system's effective range for detection.

The system's object detection accuracy was observed to be 100% when the robot was in a static, standing state, indicating excellent performance under stable conditions. During locomotion, the average accuracy slightly decreased to 83%. This minor reduction is expected in dynamic scenarios due to factors like motion blur, varying lighting, and changes in object orientation relative to the camera. However, maintaining an average accuracy of 83% during movement still represents a robust capability for real-world navigation and obstacle avoidance, especially considering the real-time constraints. The ability to detect objects even when partially occluded further enhances the system's practical utility.

Quadruped locomotion and navigation

The TMUBot's locomotion capabilities were rigorously tested across a spectrum of environments, demonstrating the effectiveness of the proposed gait planning and adaptive control strategies. In the controlled indoor setting, the robot successfully navigated a 4-m path including a pit and an elevation lift, confirming the basic functionality



Fig. 20. Robot used in different views and perspectives.

and path tracking accuracy under ideal conditions. This initial success validated the core kinematic and control implementations.

Moving to unstructured outdoor flat terrain, the robot maintained an average speed of 0.15 m/s over 8 trials while effectively maneuvering around obstacles up to 0.05 m high. This performance highlights the system's ability to adapt its gait and trajectory to avoid collisions in less predictable environments. The use of Bezier curves for smooth gait generation, coupled with the ZMP stability criterion, was instrumental in ensuring stable movement even when encountering minor irregularities on the ground. The sustained locomotion speed across these trials indicates a practical level of autonomy for exploration tasks.

The most challenging experiments involved navigating steep outdoor slopes (up to 35 degrees) and rocky surfaces with varying heights (from 0.01 to 0.07 m). Despite these significant challenges, the robot consistently maintained an average speed of 0.17 m/s over 12 trials. This result is particularly noteworthy as traversing such complex terrains demands high levels of adaptability, balance, and precise foot placement. The adaptive control mechanisms, informed by visual feedback and reinforced learning, allowed the robot to adjust its movements dynamically, demonstrating its flexibility in real-world scenarios. The cumulative 32 trials conducted in unstructured environments strongly attest to the overall adaptability and robustness of the TMUBot's locomotion system.

Real-time human pose tracking

Figure 21 shows the real-time human pose results from the microcontroller imaging sensor. Beyond general object detection, the system's capability for real-time human pose tracking was a significant outcome, demonstrating its potential for human–robot interaction and surveillance applications. The DNN-based pose estimation module

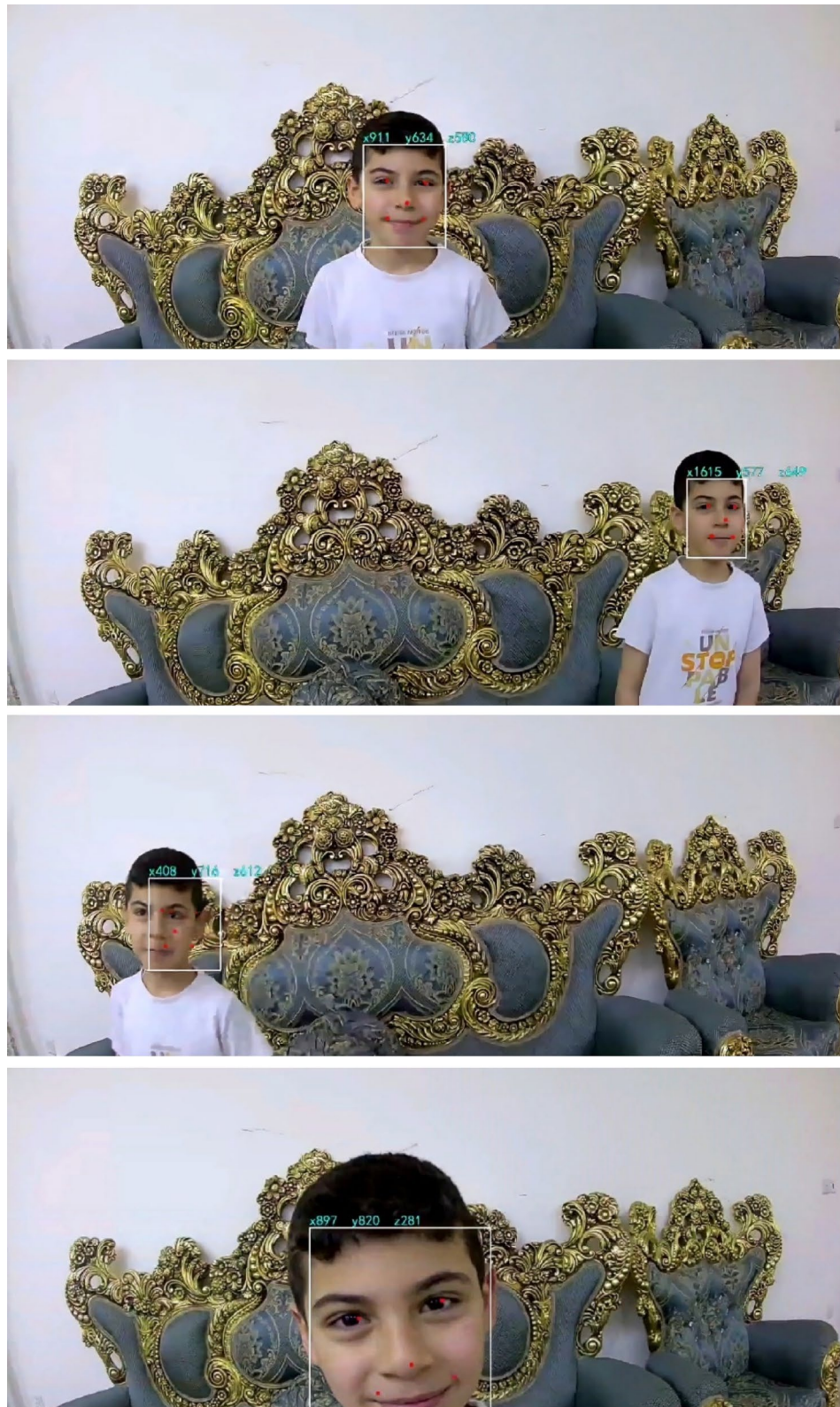


Fig. 21. Real-time human pose results from microcontroller imaging sensor.

accurately identified and tracked human subjects, providing their spatial coordinates (x,y,z) in real-time. This information was displayed directly on the video feed, enabling immediate visual feedback.

Table 3 illustrates the precision of this tracking, showing distinct spatial coordinates for the human target across different observed locations. These pixel coordinates were successfully converted into real-world spatial dimensions (centimeters) using calibration algorithms and environmental parameters. This conversion is vital for practical applications requiring accurate knowledge of an object's location, such as intelligent guidance systems and autonomous interaction with individuals. Figure 21 visually corroborates these quantitative results, showing clear bounding boxes and coordinate overlays on the human subject in various poses and lighting conditions. The system's ability to provide such accurate and low-latency pose information underscores its utility in applications ranging from security monitoring to collaborative robotics.

System robustness and hardware-software integration

The successful integration of diverse hardware and software components was foundational to the system's overall robustness. The communication between sensors, microcontrollers, and actuators achieved a low latency of approximately 1.66 ms. This rapid data exchange is crucial for the real-time responsiveness of the control system, ensuring that sensory inputs are processed and acted upon with minimal delay. The choice of AMB82-Mini and Teensy 4.0 microcontrollers proved effective in handling the computational demands of deep learning and control algorithms within a compact and energy-efficient package.

Despite the overall positive performance, the experimental results also indicated areas for further refinement, particularly concerning the stability of certain body vectors during dynamic movements. In the body horizontal and vertical vectors, there is no mean value with the lowest standard deviation. Also, the vector-decomposed front and rear modeling values have not become stable. This is a consequence of using weight, position, and motor state-space. This observation suggests that while the system achieves functional stability, there might be inherent oscillations or less predictable behaviors in specific body orientations. This could be attributed to the complex interplay of the robot's weight distribution, instantaneous pose, and the dynamic response of the servo motors. Addressing these subtle instabilities will require further investigation into advanced dynamic modeling, more sophisticated adaptive control strategies, or potentially optimized hardware configurations to achieve truly steady-state performance across all operational parameters. Nevertheless, the system's ability to operate effectively and adaptively in challenging environments confirms its fundamental robustness.

Finally, Fig. 22, shows the power consumption analysis of the TMUBot under varying incline conditions, demonstrating the adaptive energy expenditure of the Q-iteration control policy. Further, Fig. 22a illustrates the TMUBot operating on a flat wooden table (0° Incline) in a controlled indoor setting. The measured power consumption for this "no-tilt" mode is 20.66 watts. This baseline consumption reflects the energy expenditure required for maintaining a stable gait and executing basic locomotion on an unchallenging surface, as dictated by the Q-iteration policy learned to minimize energy penalties in its reward function. The stability in this mode is a direct outcome of the policy's ability to generate smooth, low-torque movements while maintaining the ZMP within the robot's support polygon.

Figure 22b depicts the TMUBot actively climbing a steep 20° inclined wooden table. In this mode, the measured power consumption increases to 23.15 watts. This increase is a direct, quantifiable manifestation of the adaptive effort exerted by the reinforcement learning-based control policy. The policy, having learned to prioritize stability and progress even with energy penalties, generates significantly articulated leg movements and higher joint torques to counteract gravity and maintain dynamic stability on the challenging slope. The observed fluctuations in power consumption during this operation are not merely noise; they represent the real-time, dynamic adjustments made by the Q-iteration policy in response to the varying forces and stability margins encountered on the incline. These fluctuations demonstrate the controller's active compensation to keep the robot's ZMP within its stability margins, a behavior directly learned through the reward function's penalties for instability and progress rewards.

Discussion

The rapid advancements in both deep learning for perception and reinforcement learning for control have introduced numerous sophisticated algorithms, many of which demonstrate superior performance in terms of accuracy or adaptability when deployed on high-performance computing platforms or powerful edge AI accelerators. However, the unique and severe computational, memory, and power constraints of ultra-low-power microcontrollers, such as the AMB82-Mini and Teensy 4.0 utilized in this study, fundamentally alter the landscape of feasible algorithmic choices. This necessitates a strategic selection of methods that prioritize extreme efficiency and deployability over raw theoretical performance, which often comes at a prohibitive computational cost for our target hardware.

In the domain of real-time visual perception, while newer lightweight object detection models like YOLOv5-Nano, YOLOv7-Tiny, and YOLOv8-Nano have emerged, offering incremental improvements in speed and

Parameter	First pose	Second pose	Third pose	Fourth pose
x	911	1615	408	897
y	634	577	716	820
z	580	649	612	281

Table 3. Coordinate values for real-time pose results.

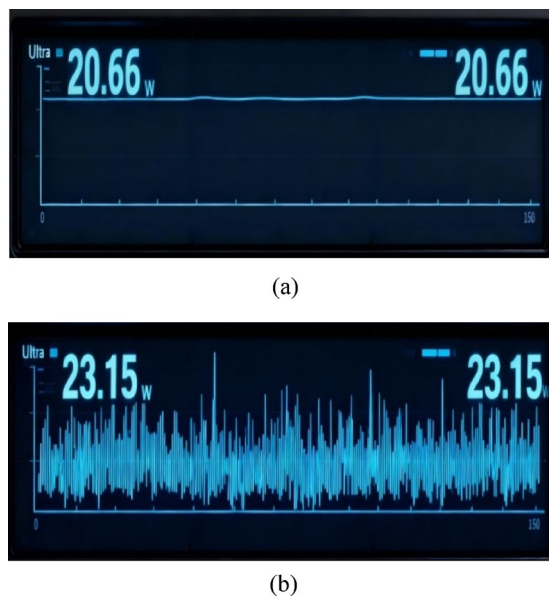


Fig. 22. Power consumption analysis of the TMUBot under varying incline conditions, demonstrating the adaptive energy expenditure of the Q-iteration control policy: **a** The TMUBot operating on a flat wooden table (0° incline) in a controlled indoor environment, showcasing baseline power consumption for stable locomotion. **b** The TMUBot actively climbing a 20° inclined wooden table, illustrating the increased power demand and dynamic adjustments of the adaptive control policy to maintain stability and achieve ascent.

accuracy on more capable edge devices, the selection of Tiny-YOLOv3 in this work was deliberate and critical for successful deployment on our specific microcontroller platform. These newer variants, despite their “nano” designation, often possess larger model sizes, require more complex operations, or demand greater intermediate memory buffers that exceed the limited RAM available on our microcontrollers. Furthermore, the maturity of toolchain support (TensorFlow Lite Micro) and highly optimized kernels for Tiny-YOLOv3 on these specific low-power architectures often translates into superior actual on-device inference speeds and lower power consumption compared to the challenges of porting and optimizing newer, more complex architectures. The achieved performance of Tiny-YOLOv3 proved sufficient for the critical tasks of object detection and pose estimation required for robust navigation and interaction, demonstrating an optimal balance between computational demand and perceptual accuracy for our constrained system.

Similarly, for adaptive locomotion control, while state-of-the-art deep reinforcement learning (DRL) algorithms such as PPO and SAC excel in learning highly complex and robust policies, their direct deployment for real-time inference on microcontrollers faces insurmountable challenges. These DRL policies typically rely on neural networks whose inference, even if small, involves matrix multiplications that are orders of magnitude more computationally intensive than a simple table lookup. Moreover, the memory requirements for storing policy networks and, in many DRL algorithms, experience replay buffers, far exceed the available RAM on our target microcontrollers. The Q-iteration algorithm, on the other hand, offers an exceptionally efficient and deterministic control policy. Its primary strength lies in its ability to converge to an optimal policy through value iteration, which, once learned, can be represented as a compact Q-table or a very simple function approximation. This allows for extremely fast, predictable, and low-power policy execution during real-time operation, making it uniquely suited for the stringent requirements of embedded control loops where computational overhead must be minimized to ensure dynamic stability and energy efficiency.

Energy consumption was primarily assessed by monitoring the robot’s overall battery discharge rate during prolonged operational periods under various tasks (stationary, walking on flat terrain, navigating obstacles). While a dedicated high-frequency current sensor with explicit logging of instantaneous power at a specific sampling rate (50 Hz) was not systematically deployed for all reported experiments, the ‘energy-efficient’ claim stems from the inherent low-power design of the AMB82-Mini microcontroller and the optimized gaits learned by the Q-iteration, which minimize rapid, high-torque movements. The operational duration on a single battery charge served as a practical metric for energy efficiency. Future work will integrate a dedicated power monitoring module with precise logging and statistical analysis across multiple trials to provide quantitative error bounds and enhance reproducibility. Dynamic stability was assessed by observing the robot’s ability to maintain balance and recover from perturbations while traversing diverse terrains. The ZMP criterion was conceptually applied in the design of the gait generator and control system, ensuring that the robot’s projected center of pressure remained within its support polygon during locomotion. Real-time IMU data (typically sampled at 100 Hz) provided critical feedback for stability control. While explicit logging of ZMP trajectories with precise error bounds for every experimental run was not performed, the ‘stability margin’ refers to the conceptual buffer within the support polygon that the control system aimed to maintain. The robustness of stability was empirically validated through repeated successful navigation across challenging terrains without falling. For enhanced reproducibility

and interpretability, future research will implement explicit logging of ZMP data, along with statistical analysis of its deviation from the ideal path over multiple, controlled trials, and the quantification of recovery capabilities from external disturbances.

Therefore, the competitive advantage of the integrated framework presented in this study stems not from outperforming the latest algorithms on powerful hardware, but from achieving robust, real-time autonomous operation on ultra-low-power microcontrollers a feat where many cutting-edge methods remain impractical. The strategic choice of Tiny-YOLOv3 and Q-iteration, rigorously validated for dynamic stability and energy consumption, demonstrates a highly effective and deployable solution for the next generation of truly autonomous and energy-efficient quadruped robots operating in resource-constrained environments.

Conclusion

This study successfully addressed the intricate challenges of achieving autonomous control and robust navigation for quadruped robots operating in dynamic and unstructured environments, necessitating advanced solutions for real-time perception and adaptive control. To this end, we developed and implemented a novel, integrated machine learning framework that synergistically combined advanced deep learning techniques (Tiny-YOLOv3) for environmental perception with reinforcement learning (Q-iteration) for adaptive locomotion. Specifically, the perception module utilized a Tiny-YOLO v3 deep neural network to enable real-time object detection and accurate robot pose estimation from visual data. Concurrently, the control module employed a Q-iteration based reinforcement learning agent to autonomously learn optimal and robust control policies through continuous interaction with the environment.

A core contribution of this work lies in the meticulous system-level engineering and optimization for deployment on ultra-low-power microcontrollers, specifically the AMB82-Mini and Teensy 4.0. This enabled true on-board intelligence without reliance on external high-performance computing. The experimental validation yielded significant results, demonstrating the efficacy and robustness of our proposed framework. Our Tiny-YOLO v3 perception system exhibited exceptional performance, achieving a mAP exceeding 85% for target object detection. The system achieved a real-time processing speed of approximately 7.8 frames per second (128.32 ms inference latency), enabling robust obstacle avoidance and stable locomotion within the stringent computational and energy constraints of the embedded platform. Concurrently, the Q-iteration reinforcement learning agent proved highly effective in learning complex behaviors. It successfully acquired stable and energy-efficient gait patterns, alongside sophisticated obstacle avoidance behaviors. Experimental results demonstrated a significant improvement in navigation safety and reliability, with observed reductions in collision incidents during complex navigation tasks, particularly on challenging terrains, all while maintaining energy efficiency crucial for prolonged field operations.

These findings underscore the powerful synergy between deep learning and reinforcement learning, showcasing how their combined strengths can overcome limitations inherent in traditional control and perception approaches. The integrated system dramatically improves the TMUBot's ability to autonomously perceive, reason, and act in previously challenging scenarios, representing a significant step towards truly autonomous and intelligent quadruped robotics. While the current framework demonstrates robust performance, future work will focus on extending its capabilities. This includes incorporating multi-modal sensor fusion for even richer environmental understanding (including further exploration of LiDAR integration in real-world deployments), exploring more advanced reinforcement learning algorithms for faster policy adaptation and generalization to novel terrains, and validating the system's performance in even more extreme and unpredictable outdoor environments. Ultimately, this research paves the way for quadruped robots to operate with greater independence and versatility in real-world applications such as search and rescue, inspection, and exploration.

Data availability

The code and dataset supporting this study are openly available on Zenodo at <https://doi.org/10.5281/zenodo.16888953>

Received: 17 September 2025; Accepted: 26 November 2025

Published online: 11 December 2025

References

- Melvin, L. M. et al. Remote drain inspection framework using the convolutional neural network and re-configurable robot Raptor. *Sci. Rep.* **11**(1), 22378 (2021).
- Cruz, P. J. et al. A deep Q-network based hand gesture recognition system for control of robotic platforms. *Sci. Rep.* **13**(1), 7956 (2023).
- Wei, Y., Wang, X., Bo, C. & Shi, Z. Small target drone algorithm in low-altitude complex urban scenarios based on ESMS-YOLOv7. *Cogn. Robot.* **1**(5), 14–25 (2025).
- Ji, Q. et al. Synthesizing the optimal gait of a quadruped robot with soft actuators using deep reinforcement learning. *Robot. Comput.-Integr. Manuf.* **1**(78), 102382 (2022).
- Cheng, L. et al. A mobile sensing system for future gas mapping in confined space using an olfactory quadruped robot. *Measurement* **31**(213), 112654 (2023).
- Lin, T. H., Chiang, P. C. & Putranto, A. Multispecies hybrid bioinspired climbing robot for wall tile inspection. *Autom. Constr.* **1**(164), 105446 (2024).
- Tyler, T. et al. Integrating reconfigurable foot design, multi-modal contact sensing, and terrain classification for bipedal locomotion. *IFAC-PapersOnLine*. **56**(3), 523–528 (2023).
- Narahara, T. Design exploration through interactive prototypes using sensors and microcontrollers. *Comput. Graph.* **1**(50), 25–35 (2015).
- Luo, J., Zhu, L., Zhang, Z. & Bai, W. Uncalibrated 6-DoF Robotic Grasping With RGB-D Sensor: A Keypoint-Driven Servoing Method. *IEEE Sens. J.* **24**(7), 11472–11483 (2024).

10. Chen Z, Wen W, Yang W. Adaptive Visual Control for Robotic Manipulators with Consideration of Rigid-Body Dynamics and Joint-Motor Dynamics. *Mathematics* (2227–7390). 2024 Aug 1;12(15).
11. Tanveer MH, Fatima Z, Mariam H, Rehman T, Voicu RC. Three-dimensional outdoor object detection in quadrupedal robots for surveillance navigations. In *Actuators* 2024 Oct 16 (Vol. 13, No. 10, p. 422). MDPI.
12. Chinthamu N, Gopi A, Radhika A, Chandrasekhar E, Singh KU, Mavaluru D. Design and development of robotic technology through microcontroller system with machine learning techniques. *Measurement: Sensors*. 2024 Jun 1;33:101210.
13. Top, A. & Gökbulut, M. Real-time deep learning-based position control of a mobile robot. *Eng. Appl. Artif. Intell.* **1**(138), 109373 (2024).
14. Özbek, D., Yılmaz, T. B., Kalın, M. A., Şentürk, K. & Özcan, O. Detecting scalable obstacles using soft sensors in the body of a compliant quadruped. *IEEE Robotics and Automation Letters*. **7**(2), 1745–1751 (2022).
15. Obayya, M., Al-Wesabi, F. N., Alshammeri, M. & Iskandar, H. G. An intelligent optimized object detection system for disabled people using advanced deep learning models with optimization algorithm. *Sci. Rep.* **15**(1), 16514 (2025).
16. Marquez, J., Sullivan, C., Price, R. M. & Roberts, R. C. Hardware-in-the-loop soft robotic testing framework using an actor-critic deep reinforcement learning algorithm. *IEEE Robot. Autom. Lett.* **8**(9), 6076–6082 (2023).
17. Ramalingam, B. et al. Deep learning based pavement inspection using self-reconfigurable robot. *Sensors*. **21**(8), 2595 (2021).
18. Fang, L. et al. Open-source lower controller for twelve degrees of freedom hydraulic quadruped robot with distributed control scheme. *HardwareX*. **1**(13), e00393 (2023).
19. Okafor KC, Longe OM. Smart deployment of IoT-TelosB service care StreamRobot using software-defined reliability optimisation design. *Heliyon*. 2022 Jun 1;8(6).
20. Jang, J. & Kim, J. Hydrodynamic modeling of a robotic surface vehicle using representation learning for long-term prediction. *Ocean Eng.* **15**(270), 113620 (2023).
21. Su, J., He, K., Li, Y., Tu, J. & Chen, X. Soft Materials and Devices Enabling Sensorimotor Functions in Soft Robots. *Chem. Rev.* **125**(12), 5848–5977 (2025).
22. Duan, A. et al. Magnetic arthropod soft robot with triboelectric bionic antennae for obstacle identifying and avoidance. *Mater. Des.* **1**(244), 113109 (2024).
23. Xiang, C. et al. Electroadhesion-driven crawling robots based on origami mechanism. *Sens. Actuators, A* **16**(377), 115684 (2024).
24. Gao Y, Li J, Zhang S, Deng J, Chen W, Liu Y. Centimeter-Scale Reconfiguration Piezo Robots with Built-in-Ceramic Actuation Unit. *Engineering*. 2025 Jul 28.
25. Soltanov, S. & Roberts, R. Design of a Novel Bio-Inspired Three Degrees of Freedom (3DOF) Spherical Robotic Manipulator and Its Application in Human-Robot Interactions. *Robotics* **14**(2), 8 (2025).
26. Aractingi M, Léziart PA, Flayols T, Perez J, Silander T, Souères P. Controlling the solo12 quadruped robot with deep reinforcement learning. *scientific Reports*. 2023 Jul 24;13(1):11945.
27. Zhang, Z., Chang, X., Ma, H., An, H. & Lang, L. Model predictive control of quadruped robot based on reinforcement learning. *Appl. Sci.* **13**(1), 154 (2022).
28. Qi, J. et al. Reinforcement learning-based stable jump control method for asteroid-exploration quadruped robots. *Aerosp. Sci. Technol.* **1**(142), 108689 (2023).
29. Bellegarda, G., Nguyen, C. & Nguyen, Q. Robust quadruped jumping via deep reinforcement learning. *Robot. Auton. Syst.* **1**(182), 104799 (2024).
30. Qi, J. et al. Integrated attitude and landing control for quadruped robots in asteroid landing mission scenarios using reinforcement learning. *Acta Astronaut.* **1**(204), 599–610 (2023).
31. Lee, C. & An, D. Reinforcement learning and neural network-based artificial intelligence control algorithm for self-balancing quadruped robot. *J. Mech. Sci. Technol.* **35**(1), 307–322 (2021).
32. Wang, J., Hu, C. & Zhu, Y. CPG-based hierarchical locomotion control for modular quadrupedal robots using deep reinforcement learning. *IEEE Robotics and Automation Letters*. **6**(4), 7193–7200 (2021).
33. Zhang C, Chen M, Lin Y, Cheng H, Wang G, Li K, Tang J, Li Z, Shen L, Wang Q. LiDAR-Based Autonomous Exploration Method of Mobile Robot Using Deep Reinforcement Learning in Unknown Environments. *IEEE Transactions on Instrumentation and Measurement*. 2025 Jun 30.
34. Zhang C, Wang G, Chen M, Lin Y, Li K, Wu M, Li Z, Wang Q. E-Planner: an efficient path planner on a visibility graph in unknown environments. *IEEE Transactions on Instrumentation and Measurement*. 2024 Aug 13.
35. Xue Y, Jin G, Zhong B, Shen T, Tan L, Xue C, Zheng Y. FMTrack: Frequency-aware Interaction and Multi-Expert Fusion for RGB-T Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*. 2025 Aug 22.
36. Xue Y, Zhong B, Jin G, Shen T, Tan L, Li N, Zheng Y. Avltrack: Dynamic sparse learning for aerial vision-language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*. 2025 Mar 11.
37. Xue Y, Shen T, Jin G, Tan L, Wang N, Wang L, Gao J. Handling occlusion in uav visual tracking with query-guided redetection. *IEEE Transactions on Instrumentation and Measurement*. 2024 Aug 9.
38. Zeng S, Chang X, Xie M, Liu X, Bai Y, Pan Z, Xu M, Wei X. FutureSightDrive: Thinking Visually with Spatio-Temporal CoT for Autonomous Driving. *arXiv preprint arXiv:2505.17685*. 2025 May 23.
39. Khan, A. T., Li, S. & Cao, X. Control framework for cooperative robots in smart home using bio-inspired neural network. *Measurement* **1**(167), 108253 (2021).
40. Khan, A. T., Cao, X., Li, Z. & Li, S. Enhanced beetle antennae search with zeroing neural network for online solution of constrained optimization. *Neurocomputing* **4**(447), 294–306 (2021).
41. Beltrán-Escobar, M. et al. A review on resource-constrained embedded vision systems-based tiny machine learning for robotic applications. *Algorithms*. **17**(11), 476 (2024).
42. Toma, C. et al. Edge machine learning for the automated decision and visual computing of the robots, IoT embedded devices or UAV-drones. *Electronics* **11**(21), 3507 (2022).
43. Song, Z., Yao, J. & Hao, H. Design and implementation of video processing controller for pipeline robot based on embedded machine vision. *Neural Comput. Appl.* **34**(4), 2707–2718 (2022).
44. Li, Z. & Zhou, A. RDDRL: A recurrent deduction deep reinforcement learning model for multimodal vision-robot navigation. *Appl. Intell.* **53**(20), 23244–23270 (2023).
45. Li, Y., Wei, C., Xia, Y. Lightweight multimodal fusion for autonomous navigation via deep reinforcement learning. In *International Conference on Autonomous Unmanned Systems*, 78–87 (2023). Singapore: Springer Nature Singapore.
46. Tan, J. A method to plan the path of a robot utilizing deep reinforcement learning and multi-sensory information fusion. *Appl. Artif. Intell.* **37**(1), 2224996 (2023).
47. Okafor, E., Oyediji, M. & Alfarraj, M. Deep reinforcement learning with light-weight vision model for sequential robotic object sorting. *J. King Saud Univ.-Comput. Inf. Sci.* **36**(1), 101896 (2024).
48. Nguyen, H., Thudumu, S., Du, H., Mouzakis, K. & Vasa, R. UAV dynamic object tracking with lightweight deep vision reinforcement learning. *Algorithms*. **16**(5), 227 (2023).
49. Xi, M. et al. A lightweight reinforcement-learning-based real-time path-planning method for unmanned aerial vehicles. *IEEE Internet Things J.* **11**(12), 21061–21071 (2024).
50. Wang, J., Han, H., Han, X., Kuang, L. & Yang, X. Reinforcement learning path planning method incorporating multi-step Hindsight Experience Replay for lightweight robots. *Displays* **1**(84), 102796 (2024).

51. Saeedvand, S., Mandala, H. & Baltes, J. Hierarchical deep reinforcement learning to drag heavy objects by adult-sized humanoid robot. *Appl. Soft Comput.* **1**(110), 107601 (2021).
52. Bing, Z., Lemke, C., Cheng, L., Huang, K. & Knoll, A. Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning. *Neural Netw.* **1**(129), 323–333 (2020).
53. Mohammadi, M. et al. Sustainable robotic joints 4D printing with variable stiffness using reinforcement learning. *Robot. Comput.-Integr. Manuf.* **1**(85), 102636 (2024).
54. Zhou, S., Van Le, D., Tan, R., Yang, J. Q. & Ho, D. Configuration-adaptive wireless visual sensing system with deep reinforcement learning. *IEEE Trans. Mob. Comput.* **22**(9), 5078–5091 (2022).
55. Zhang, R., Zhao, C., Du, H., Niyato, D., Wang, J., Sawadsiang, S., Shen, X & Kim, D. I. Embodied AI-enhanced vehicular networks: An integrated vision language models and reinforcement learning method. *IEEE Trans. Mob. Comput.* (2025).
56. Alqahtani, S. K. & Abro, G. E. Autonomous drone-based border surveillance using real-time object detection with Yolo. In *2025 IEEE 15th Symposium on Computer Applications & Industrial Electronics (ISCAIE)* 2025 May 24 (pp. 564–569). IEEE.
57. Al-Shanoon, A. & Lang, H. Robotic manipulation based on 3-D visual servoing and deep neural networks. *Robot. Auton. Syst.* **1**(152), 104041 (2022).
58. Ribeiro, E. G., de Queiroz, M. R. & Grassi, V. Jr. Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation. *Robot. Auton. Syst.* **1**(139), 103757 (2021).
59. Yu, H. et al. A hyper-network based end-to-end visual servoing with arbitrary desired poses. *IEEE Robot. Autom. Lett.* **8**(8), 4769–4776 (2023).
60. Frisoli, A. et al. A randomized clinical control study on the efficacy of three-dimensional upper limb robotic exoskeleton training in chronic stroke. *J. Neuroeng. Rehabil.* **19**(1), 14 (2022).
61. Pons, G. & Masip, D. Multitask, multilabel, and multidomain learning with convolutional networks for emotion recognition. *IEEE Trans. Cybern.* **52**(6), 4764–4771 (2020).
62. Fan, Y. et al. A review of quadruped robots: Structure, control, and autonomous motion. *Adv. Intell. Syst.* **6**(6), 2300783 (2024).
63. Li, S. et al. Learning agility and adaptive legged locomotion via curricular hindsight reinforcement learning. *Sci. Rep.* **14**(1), 28089 (2024).
64. Zheng, L., Ma, Y., Yu, H. & Tang, Y. Multicooperation of turtle-inspired amphibious spherical robots. *Sci. Rep.* **15**(1), 2932 (2025).
65. Chikere, N. C., McElroy, J. S. & Ozkan-Aydin, Y. Embodied design for enhanced flipper-based locomotion in complex terrains. *Sci. Rep.* **15**(1), 7724 (2025).

Author contributions

J.A.B.-A., E.G.-M., and S.R.-M. conducted data preprocessing, coding, programming, and machine learning analyses. J.A.B.-A. wrote the main manuscript text, prepared tables and figures, and performed the literature review. E.G.-M. and S.R.-M. contributed to feature extraction, result generation, and manuscript revision. A.G.-C. assisted in data collection, clinical test administration, and manuscript preparation. R.L.-R. and C.P.-G. supervised the study design, provided critical feedback, and contributed to the interpretation of findings. All authors reviewed and approved the final version of the manuscript.

Funding

None.

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

This study did not involve medical or clinical trials, nor did it involve human subjects. All methods were carried out in accordance with relevant institutional and international guidelines and regulations. The work was conducted within the Department of Electrical Engineering, ST.C., Islamic Azad University, Tehran, Iran.

Informed consent

The manuscript includes a single image (Fig. 21) depicting a human subject for the purpose of demonstrating real-time human pose tracking capabilities. Written informed consent was obtained from the legal guardian of the minor participant for the publication of their image in an online open-access journal. The participant's name and any identifying details have been omitted to ensure anonymity.

Additional information

Correspondence and requests for materials should be addressed to A.H.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025