Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 15, Issue. 5, May 2026, pg.163 – 170

Programming Quadruped Robots: A Survey of Locomotion, Control, and Perception

Asia Harris; Yujian Fu

Department of Electrical Engineering and Computer Science, College of Engineering, Technology, and Physical Sciences, Alabama A&M University, USA, 35762
[1] asiamar.harris@gmail.com; [2] yujian.fu@aamu.edu

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i05.018

Abstract: Quadrupedal robots are sophisticated autonomous systems inspired by animal locomotion, evolving from mechanical curiosities. These machines can traverse rough and spontaneous environments. This survey will provide an integrated overview of the various aspects that go into developing a marveling system, including the programming frameworks, locomotion mechanisms, and perception architectures. Developments will be categorized into six topics, drawing on over 50 sources from 2010 to 2025. This study synthesizes major trends, including dynamic gait optimization, reinforcement learning-based control, and LiDAR-vision integration, for enhanced terrain adaptability. Furthermore, it highlights the challenges engineers still face. Energy efficiency and environmental generalization, to name a couple. This review concludes with a discussion of ethics and the role of robots in society, particularly considering the integration of AI.

Keywords: Quadruped robots, locomotion, gait programming, reinforcement learning, perception, robotics software

---

I. INTRODUCTION

The field of quadruped robotics has evolved through a dynamic interplay of mechanical innovation, biological inspiration, and computational control. Tracing its origins to 19th-century mechanical linkages, the foundational work of Pafnuty Chebyshev laid the mathematical groundwork for approximating rectilinear motion using multi-bar mechanisms such as the "Chebyshev straightener" and "Chebyshev's horse", inventions that would later influence the structure of robotic limbs [1]. As discussed in [2], this early mechanical curiosity within a broader historical arc, highlighting Chebyshev's 1870 walking machine as the world's first quadruped robot prototype, albeit with limited terrain adaptability. Over the next century, quadrupedal locomotion matured into a viable robotic paradigm due to its balance of stability, terrain adaptability, and simplified control relative to bipeds. Early prototypes were entirely mechanical until the development of the “Phony Pony” at USC in 1968 [3]. Raibert’s pioneering contributions in the 1980s established a dynamic model for legged locomotion, where gait stability is maintained through real-time control of vertical support, body attitude, and foot placement [4]. This control philosophy was later brought to life in Boston Dynamics' BigDog, which demonstrated dynamic trotting and real-time terrain response using hydraulic actuation, onboard force sensors, and gyroscopic feedback [5]. While mechanical stability forms the backbone of quadruped mobility, bioinspired design now drives advancements in agility and energy efficiency. Fukuhara et al. [6] emphasizes the critical differences between robots and animals, notably in spinal flexibility, passive dynamics, and actuator compliance, that still separate engineered systems from their biological counterparts. Majithia et al. [7] consolidate these perspectives into a taxonomy of quadruped designs and gaits, noting that articulated legs enable better adaptability to complex terrains, while discontinuous gaits improve energy performance. The convergence of these innovations suggests a future where quadruped robots can blend mechanical precision, biological adaptability, and autonomous programming to meet the demands of extreme, unpredictable environments.

II. LOCOMOTION & GAIT PROGRAMMING

Quadruped locomotion programming concerns the synthesis, transition, and execution of distinct gait patterns that allow four-legged robots to move dynamically and stably across varying terrain. This differs from broader control or sensing architectures by focusing specifically on how gait cycles are constructed, sequenced, and parameterized for real-time execution.

A primary thread in current research is the use of data-driven learning to automate gait generation. Hwangbo et al. [8] trained neural policies to produce dynamic gaits such as trotting and jumping on the ANYmal platform, where locomotion behaviors emerged from reinforcement learning conditioned on velocity and phase variables. Similar efforts in hierarchical control bye Xie et al. [9] and Peng et al. [10] decomposed gait generation into layered policies, where low-level networks handled motion primitives, and high-level controllers governed transitions between gaits. Together, these methods allow learned gait controllers to generalize across dynamic motion styles.

Gait programming has also shifted toward robust terrain adaptation. Kumar et al. [11] proposed Rapid Motor Adaptation (RMA), where policy outputs are modulated by latent embeddings of terrain inferred in real-time. Miki et al. [12] extended this direction by integrating visual terrain perception into locomotion control, enabling quadrupeds to traverse complex environments without explicit pre-programmed gait transitions. This theme is further supported by Kim et al. [13], who evaluated sim-to-real transfer of gait behavior on deformable terrain and demonstrated that adaptive behaviors can be preserved despite contact uncertainty, underscoring the need for gait policies to account for real-world surface inconsistencies.

Specialized gait logic also plays a role in unique mobility tasks. Zhu et al. demonstrated a terrain-perception-free spinning gait using phase-synchronized foot trajectories designed for spherical-foot quadrupeds [14]. Similarly, Miki et al. [12] trained gaits for spatially constrained environments, where traditional gaits may fail due to collision risks. These approaches demonstrate how locomotion programming can extend beyond forward movement to address various physical constraints.

Beyond learned methods, model-based optimization remains integral to the execution of gait. Di Carlo et al. [15] applied convex model predictive control (MPC) to predict foot contacts and forces for bounding gaits on the MIT Cheetah 3. Bellicoso et al. [16] and Raiola et al. [17] expanded this by optimizing entire motion trajectories online with nonlinear solvers, allowing smooth transitions across gait phases and tasks. While model-free policies excel in adaptation, MPC-based programming offers formal guarantees on motion feasibility and stability.

Training infrastructure has also become a programming concern. Rudin et al. [18] leveraged massively parallel simulation to evolve walking behaviors in minutes. This work suggests that gait learning pipelines, like compilers, can be tuned for runtime and performance. Kim et al. [19] contributed to this notion by demonstrating that whole-body impulse control enables rapid gait modulation at high frequencies, thereby eliminating the need for distinct transitions between stance and flight phases.

Finally, curriculum learning has emerged as a gait programming tool. Gurram et al. [20] surveyed curriculum-based reinforcement learning (RL) pipelines in which gait attributes, such as step frequency, foot swing, and stance timing, were embedded into the training process. This enables precise behavioral shaping throughout training, contrasting with fixed reward schemes used in earlier gait learning systems.

Comparing these approaches reveals a spectrum in quadruped gait programming, ranging from classical model-based planning, which prioritizes predictability and real-time feasibility, to adaptive and perceptual learning models that enable flexibility across various terrains. While the former ensures robustness in tightly modelled environments, the latter allows generalization and emergent behaviors. Increasingly, hybrid methods that fuse perception, learning, and control are defining the future of gait programming. A future where locomotion behaviors are not only reactive but anticipatory, terrain-aware, and architecture-independent.

III. CONTROL & LEARNING ARCHITECTURES

The programming of quadruped robots has evolved from rule-based controllers to advanced whole-body architectures that unify balance, locomotion, and manipulation. Central to this shift is the adoption of Model Predictive Control (MPC) and its integration with learning-based modules, enabling quadrupeds to adapt robustly to dynamic terrain and complex tasks.

Kim et al. [19] introduced a Whole-Body Impulse Controller (WBIC) combined with MPC for the MIT Cheetah 3, enabling precise torque-level control and real-time trajectory correction in high-speed locomotion. This established the foundation for contact-aware control, crucial for agile quadruped programming. Building on this, Raiola et al. [17] developed the WoLF framework, a modular control stack that separates centroidal dynamics planning from inverse kinematics execution. Unlike earlier systems, WoLF supports multiple morphologies and is open-source, making it valuable for researchers building reusable quadruped control code across platforms.

Katayama and Ohtsuka [21] advanced MPC by incorporating contact-switch timing optimization, a feature especially vital for quadrupeds navigating unstable terrain where contact sequences are not fixed. Elobaid et al. [22] further emphasized safety by introducing nonlinear centroidal MPC formulations that guarantee input-to-state stability, which is essential in unpredictable outdoor environments. A complementary contribution by Zhang et al. [23] introduced a hybrid architecture that integrates iterative Linear Quadratic Regulation (iLQR) with full-body MPC, striking a balance between precision and computational efficiency in MuJoCo simulation environments. This offers developers a lightweight and responsive control option for simulated training pipelines or onboard deployment.

Meanwhile, the CENTAURO platform, developed by Dadiotis et al. [24], expands quadruped programming to include the integration of manipulation and mobility, where the robot’s arms and legs must coordinate in real-time. Arm et al. [25] extended this concept with "Pedipulation," a programming paradigm that merges stepping and manipulation behaviors through shared kinematic optimization, allowing a robot's legs to double as tools for environmental interaction. Finally, Shafiee et al. [26] proposed ManyQuadrupeds, a single policy capable of operating across morphologically distinct quadrupeds, showcasing how generalization across robot forms is becoming more feasible through shared neural control.

Taken together, these works demonstrate that modern quadruped control architectures are trending toward modularity, formal robustness, and morphological generalization. MPC is no longer used in isolation but as a backbone within hybrid systems that combine classical optimization with adaptive learning. The result is a flexible control framework capable of scaling across tasks, environments, and robot designs.

IV. PROGRAMMING LANGUAGES & TECHNOLOGIES

The software backbone of quadruped robotics is defined by a hybrid landscape of real-time middleware, control-oriented programming frameworks, and learning-optimized simulation environments, each enabling key capabilities such as perception, motion generation, and policy execution. At the heart of most modern deployments is ROS 2, a modular, distributed robotics middleware designed for deterministic, real-time communication. Built on DDS (Data Distribution Service), ROS 2 enables low-latency message passing between sensors, estimators, and control loops. This is essential for four-legged robots traversing unstable terrain, where sub-second response times impact stability [27].

To support environment awareness and terrain handling, ANYbotics’ open-source elevation mapping package integrates seamlessly with ROS 2 to produce real-time elevation maps using sensor fusion [28]. This enables quadruped platforms, such as ANYmal, to detect and plan safe footholds in rough terrain environments [29], serving as a critical bridge between perception and motion generation.

For programming movement behaviors through learning, Isaac Gym, a GPU-accelerated simulator by NVIDIA, enables massive parallelization of reinforcement learning for quadrupeds. Makoviychuk et al. [30] demonstrated how Isaac Gym was used to train walking policies in minutes using thousands of parallel agents, avoiding the physical degradation associated with hardware training. When combined with Legged Gym, a benchmark designed explicitly for quadrupeds, these environments enable researchers to program dynamic behaviors, such as trotting or recovering from disturbances, using neural policies optimized in silico.

Complementing this are frameworks that target real-time execution and trajectory optimization. Crocoddyl, introduced by Mastalli et al. [31], provides a fast optimal control library for multi-contact motion planning. Built in C++ with Python bindings, Crocoddyl serves as a powerful backend for whole-body controllers, fitting naturally into ROS-based quadruped stacks. Its strength lies in generating low-latency motion trajectories that account for contact dynamics, making it ideal for robots operating in cluttered or multi-contact environments.

While ROS 2 excels in modularity, its asynchronous nature can hinder deterministic timing. To address this, Ferigo et al. [32] proposed a synchronous dataflow architecture based on the YARP middleware, allowing tightly coupled scheduling across perception, control, and planning modules. Applied in systems like WALK-MAN and CENTAURO, this approach offers microsecond-level timing precision, an advantage when programming legged robots where delays in sensor-motor coupling can cause instability.

A broader trend in quadruped software design is the use of modular reinforcement learning systems. Cai et al. [33] proposed a modular RL framework that decomposes behavior into policy units, each trained for a distinct function such as terrain traversal or balance recovery. This allows developers to program quadrupeds for adaptive behavior without retraining from scratch, aligning with the need for composability and scalability in legged robotics.

In comparison, ROS 2 offers community-backed extensibility but introduces complexity in ensuring real-time constraints. Isaac Gym and Legged Gym are optimized for learning efficiency, excelling in simulation but requiring adaptation for deployment. Crocoddyl and YARP architectures trade off generality for low-level timing precision and deterministic control, making them ideal for hardware-focused applications where millisecond decisions matter. Each framework thus fills a distinct niche in the quadruped programming stack, from neural training in virtual environments to real-time execution on robotic platforms.

Ultimately, quadruped programming necessitates a multilayered technology stack that combines high-level behavioral abstractions with low-level, real-time execution. Developers must be fluent in navigating Python, C++, and CUDA, while orchestrating modules through ROS, YARP, or custom middleware. The selection of technologies should reflect the target deployment scenario, whether for simulation-heavy learning, deterministic control. There must be a balance between abstraction and hardware precision in pursuit of agile, adaptive, and robust quadruped locomotion.

V. PERCEPTION & SENSING

Quadruped robots require highly sophisticated perception systems to traverse unpredictable, unstructured environments. Unlike wheeled or aerial robots, legged robots must respond in real-time to complex terrain geometries, shifting compliance, and occluded surfaces. This section explores how perception-specific programming, from sensor fusion to terrain-aware motion planning, enables quadrupeds to interact adaptively with their surroundings.

While exteroceptive sensing (e.g., vision, LiDAR) is essential for mapping and obstacle detection, proprioceptive sensing and contact detection are equally critical for maintaining balance and executing adaptive locomotion in legged robots [34]. As Chen et al. emphasizes [35], single-sensor SLAM systems (using LiDAR or vision alone) are inherently limited; LiDAR lacks rich visual information, while vision systems are sensitive to lighting. This necessitates multi-sensor fusion for stability and reliability. Furthermore, Olufade et al. [36] shows that integrating inputs from visual, LiDAR, radar, inertial, and tactile sensors significantly improves scene understanding, obstacle detection, and real-time decision-making. Yu et al. [37] demonstrated that sensor fusion pipelines integrating inertial and terrain classification inputs significantly enhance locomotion adaptability on visually degraded surfaces. Xu et al. [38] underscore the importance of adaptive data coupling strategies and system-level architecture choices, which directly affect computational efficiency and SLAM accuracy across various operational conditions. Camurri et al. [39] introduced Pronto. This real-time estimator fuses encoder, IMU, and contact data to infer body pose without the need for GPS, which is crucial in GPS-denied environments. Han et al. [40] further extended this by combining GPS and VINS in a tightly coupled estimator, specifically tailored to the state estimation needs of legged robots navigating long-range outdoor domains.

Quadrupeds face terrain challenges that go beyond localization. They must also assess terrain shape, slope, and navigability. To support this, Fankhauser et al. [41] introduced the widely adopted elevation_mapping ROS package, which generates continuous terrain heightmaps from LiDAR and IMU data. This was enhanced in Legged Robotics [42] with GPU-accelerated elevation mapping, enabling fine-grained terrain reconstruction at real-time speeds, facilitating safe footstep placement in rocky or cluttered environments. Xu et al. [38] confirmed that LiDAR-vision fusion greatly improves edge detection and terrain segmentation, which is vital for planning precise footholds in mixed-visibility conditions.

Beyond geometry, terrain classification has become critical in gait adaptation. Wellhausen et al. [43] developed a vision-based foothold assessment system utilizing a CNN, enabling the robot to identify safe stepping regions visually. Shafiee et al. [26] expanded this to terrain-type classification (e.g., mud, gravel, grass), allowing for real-time modulation of foot trajectories based on the terrain label. These learning-based modules reduce reliance on dense elevation data, opening new pathways for semantic-aware locomotion programming.

Robust state estimation is also a significant challenge in quadruped locomotion, as intermittent foot contact, slippage, and rapid body orientation shifts can lead to drift. Hartley et al. [44] proposed an invariant EKF that incorporates foot contact constraints to improve pose estimation in disturbed scenarios. Bloesch et al. [45] introduced terrain-adaptive Kalman filtering, which enables estimator parameters to adjust according to terrain quality. Another approach dynamically integrates foot-end position and velocity observations, modulating their influence through a contact probability model and adaptive covariance adjustment, which yields lower RMSEs in experiments across multiple terrains [46]. Xu et al. [38] extended this to a LiDAR-vision SLAM framework tailored for long-horizon terrain transitions, such as slopes or tunnels, where visual or inertial odometry alone proves insufficient.

Emerging sensor technologies offer alternative modalities when vision or LiDAR fails. Jung et al. [47] introduced Co-RaL, a radar–leg odometry, which is effective in low-visibility environments such as fog or dust. Haddeler et al. [48] developed a probing-based terrain interaction strategy, enabling quadrupeds to assess surface compliance (e.g., ice, mud) through brief toe contact before complete weight transfer. Yang et al. [49] explored the use of only onboard sensors to classify contact conditions, enabling terrain-aware gait adjustments even in sensor-degraded conditions.

For large-scale outdoor navigation, perception is extended through SLAM. One experiment integrates visual, inertial, LiDAR, and leg odometry to provide centimeter-scale tracking in complex natural environments. Experimental results showed a 62% improvement in translational and 51% in rotational error compared to state-of-the-art approaches, demonstrating that high-fidelity multi-sensor fusion directly enhances both motion tracking and terrain adaptation [50].

These SLAM outputs often feed into motion planners or learning agents, enabling memory-aware foot placement and route replanning, particularly useful in forests, construction sites, or disaster zones.

In quadruped programming, perception must be integrated not just as raw sensor input, but as a real-time feedback loop that modifies gait cycles and trajectory plans. This includes terrain-informed step timing (e.g., more extended stance phase on mud), adaptive swing height for obstacles, and failure recovery triggered by state estimators. Programming APIs must accommodate this feedback through sensor fusion modules, terrain classification services, and tightly coupled elevation data, which are used directly by locomotion and balance controllers.

In summary, perception in quadruped robotics is not passive. It is a programmable, sensor-driven control module that feeds directly into locomotion. Through tightly integrated estimation pipelines, semantic terrain classification, and modular SLAM frameworks, modern quadrupeds are becoming increasingly autonomous in their interactions with their environment. The success of these systems relies on the seamless incorporation of perception into the robot’s control and planning software, reflecting the convergence of sensing and programming as a unified discipline in legged robotics.

VI. APPLICATIONS, ETHICS, & EMERGING CHALLENGES

Quadruped robots have rapidly evolved from experimental machines to functional agents deployed across a wide range of industries and public domains. Their legged morphology grants superior mobility across irregular terrain compared to wheeled robots, making them especially suited for applications in hazardous, complex, or dynamic environments. In industrial contexts, robots like ANYmal and Spot have proven highly effective for inspection, monitoring, and data collection in various sectors, including oil and gas, manufacturing, and energy infrastructure. Equipped with advanced sensing packages, including LiDAR, thermal cameras, and inertial measurement units, these robots autonomously navigate sites, collect actionable data, and reduce the need for human workers in hazardous areas [51]. These deployments are enabled by real-time middleware platforms, such as ROS 2, which support modularity and reliable data transmission in time-sensitive control loops [27]. Furthermore, robust mapping systems, such as the elevation_mapping package developed by ANYbotics [28], allow the robot to dynamically adjust gait and foot placement based on detected surface topology [29]. A recent study by Go et al. [52] found that stakeholder acceptance in industrial environments is closely tied to robots’ perceived safety and ability to operate in proximity to workers, emphasizing the importance of predictive safety logic in programming frameworks.

In high-risk environments such as energy plants or offshore rigs, quadrupeds are being tested for autonomous inspection under strict environmental constraints. A field deployment study by Pencelli et al. [53] showed that quadrupeds performed reliably in rough, unstructured terrains while maintaining communication with control centers. The ability to carry out non-contact inspections autonomously was particularly valued in environments with explosive atmospheres, where human presence is highly restricted. These trials highlight the importance of physical robustness, real-time autonomy, and integration with industrial sensor networks for successful deployment. From a programming perspective, this requires tightly coupled modules that ensure localization, anomaly detection, and failure recovery under uncertainty. These are core challenges for software architectures supporting long-term autonomy in field robotics.

Military and law enforcement deployments raise separate ethical and technical issues. Spot, manufactured by Boston Dynamics, has been equipped with payloads ranging from surveillance cameras to communication modules and has even been trialed in tactical scenarios [51]. However, the use of robotic dogs for policing and military surveillance has been heavily criticized. The ACLU argues that opaque deployment policies and the potential for excessive surveillance raise significant concerns about civil liberties and constitutional protections [54]. Similarly, an opinion from Medium advocates for the development of ethical frameworks and community-informed protocols before these machines are deployed in public spaces [55]. The growing militarization of robotic technologies has sparked debate over accountability, transparency, and escalation of force, especially in the absence of legal frameworks governing autonomous systems. Critics argue that such technologies risk normalizing dehumanized law enforcement, raising concerns about excessive data collection and the misuse of facial recognition [55]. The ethical concerns surrounding robotic systems in warfare have significant parallels in domestic law enforcement, where the delegation of potentially coercive power to machines prompts questions about accountability, proportionality, and control [56]. These concerns suggest that ethical programming is not merely about safety protocols, but must also address issues of consent, monitoring, and social impact through embedded policy modules.

In disaster response and search-and-rescue (SAR) missions, quadrupeds excel in scenarios where wheeled robots or drones may fail. Their ability to traverse collapsed structures, stairs, rubble, and even flooded environments has made them indispensable in urban search and rescue simulations [57]-[58]. These applications require high fault tolerance, rapid terrain adaptation, and robust SLAM capabilities in environments where GPS may be unavailable. Hamrani et al. [57] highlights that perception-driven locomotion is essential for operating in low-visibility environments and recommend integrating terrain classification with fail-safe motion primitives for critical deployments. From a programming perspective, this necessitates the fusion of perception and control pipelines to maintain mission continuity even during partial sensor failure.

Quadruped robots are also being explored for logistics and indoor automation, particularly in warehouses and last-mile delivery contexts. Their ability to navigate cluttered and uneven floors makes them suitable for environments without infrastructure designed for traditional AGVs (automated guided vehicles). Li et al. [58] demonstrate how adaptive foot trajectory planning combined with lightweight object recognition systems can support object transportation, door traversal, and shelf-level inspection. These systems rely heavily on modular control software and perception stacks that can be reconfigured according to the operational context. In this area, ROS 2 and Crocoddyl [31] have shown strong interoperability. These examples demonstrate that success in logistics is contingent upon dynamic task allocation and the real-time integration of sensor data for reactive planning and decision-making.

Ethics becomes even more critical when quadrupeds are introduced into public and social domains. Studies by Go et al. [52] note that public trust in robotic systems depends not only on performance but on the robot’s perceived intent, transparency, and behavior. In human-facing deployments, quadrupeds must be programmed to act predictably and convey non-threatening intent through body language, audio cues, and motion smoothness. This introduces a new layer of complexity in programming, where gait style, response time, and proximity regulation must be designed with psychological and social considerations in mind. For instance, pausing before approach, adjusting head orientation, and using soft gait transitions can significantly influence public acceptance.

Across all sectors, a key insight is the trade-off between autonomy and human oversight. Industrial and logistics deployments often prioritize predictability and safety, with clearly defined operational parameters. In contrast, SAR and military applications demand reactive autonomy and minimal latency, pushing control architectures toward decentralized decision-making. The convergence of these requirements challenges developers to create systems that are simultaneously safe, responsive, and ethically constrained. Emerging trends in programming reflect this need: from embedded ethics modules to self-diagnostic subsystems that halt operations in the face of ambiguity.

In summary, quadruped robots are reshaping the operational landscape across industrial, civil, and emergency domains. However, the challenges are no longer just mechanical or algorithmic. They are also legal, ethical, and societal. Programming these machines demands a multidisciplinary approach that blends real-time control, perception fusion, adaptive learning, and embedded ethical protocols. The shift toward responsible autonomy in quadruped robotics mirrors the broader trajectory of robotics itself: one that must align technological power with public trust and societal value.

VII. CONCLUSION

Quadruped robotics represents a uniquely complex and promising frontier at the intersection of mechanical design, algorithmic control, software engineering, and human-centered ethics. Unlike their wheeled or aerial counterparts, quadrupeds must contend with the challenges of biologically inspired locomotion, dynamic terrain negotiation, and tightly integrated sensorimotor control. Requirements that have driven the development of whole-body motion planning, robust perception pipelines, and reinforcement learning-based motor adaptation. The maturation of middleware, such as ROS 2, and simulation frameworks, like Isaac Gym, has accelerated the prototyping and programming of these systems, enabling reproducible and scalable development across research and industry. Meanwhile, advances in perception technologies, from multi-modal state estimation to terrain classification, have broadened the scope of deployment from structured industrial spaces to unstructured natural and urban environments. As quadrupeds transition from experimental labs to public streets, power plants, and disaster zones, the programming paradigms that govern them must increasingly encode not just control laws and locomotion strategies, but also principles of safety, explainability, and ethical alignment. Applications in defense, inspection, logistics, and planetary exploration all carry distinct performance constraints and societal expectations, requiring tailored approaches to autonomy and trust calibration. The future of quadruped robotics programming thus lies in unifying these diverse into coherent, modular systems that can adapt, collaborate, and earn trust across the full spectrum of human-robot interaction.

ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation under Award Nos. 2426293 and 2402017. The authors gratefully acknowledge the constructive feedback and insightful comments provided by the reviewers, which significantly contributed to improving the quality and clarity of this manuscript.

REFERENCES

[1]. A. Zhukova, “Math and motion: A look at Chebyshev’s works on linkages,” Bhāvanā, 2021. [Online]. Available: https://bhavana.org.in/math-and-motion-a-look-at-chebyshevs-works-on-linkages
[2]. D. Wu, “Overview of the development history and current design status of Quadrupedal Animal Robots,” Journal of Artificial Intelligence Practice, vol. 7, no. 2, pp. 161–165, 2024.
[3]. P. Biswal and P. K. Mohanty, “Development of quadruped walking robots: A review,” Ain Shams Engineering Journal, vol. 12, no. 2, pp. 2017–2031, 2021.
[4]. M. H. Raibert, “Running on four legs as though they were one,” IEEE Journal of Robotics and Automation, vol. 2, no. 2, pp. 70–82, 1986.
[5]. M. Raibert et al., “BigDog, the rough-terrain quadruped robot,” in Proc. 17th IFAC World Congress, 2008, pp. 10822–10825.
[6]. A. Fukuhara, M. Gunji, and Y. Masuda, “Comparative anatomy of quadruped robots and animals: A review,” Advanced Robotics, vol. 36, pp. 1–19, 2022.
[7]. A. Majithia et al., “Design, motions, capabilities, and applications of quadruped robots: A comprehensive review,” Frontiers in Mechanical Engineering, vol. 10, 1448681, 2024.
[8]. J. Hwangbo et al., “Learning agile and dynamic motor skills for legged robots,” arXiv:1901.08652, 2019.
[9]. Z. Xie et al., “Feedback control for Cassie with deep reinforcement learning,” arXiv:1803.05580, 2018.
[10]. X. B. Peng, G. Berseth, K. Yin, and M. van de Panne, “DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning,” ACM Trans. Graph., vol. 36, no. 4, Art. 41, 2017.
[11]. A. Kumar, Z. Fu, D. Pathak, and J. Malik, “RMA: Rapid motor adaptation for legged robots,” arXiv:2107.04034, 2021.
[12]. T. Miki et al., “Learning robust perceptive locomotion for quadrupedal robots in the wild,” arXiv:2201.08117, 2022.
[13]. T. Kim and S.-H. Lee, “Quadruped locomotion on non-rigid terrain using reinforcement learning,” arXiv:2107.02955, 2021.
[14]. H. Zhu et al., “Terrain-perception-free quadrupedal spinning locomotion,” Frontiers in Robotics and AI, vol. 8, 724138, 2021.
[15]. J. Di Carlo, “Software and Control Design for the MIT Cheetah Quadruped Robots,” Master’s thesis, MIT, 2020.
[16]. C. D. Bellicoso et al., “Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots,” IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 2261–2268, 2018.
[17]. G. Raiola, M. Focchi, and E. Mingo Hoffman, “WoLF: The Whole-body Locomotion Framework for quadruped robots,” arXiv:2205.06526, 2022.
[18]. N. Rudin et al., “Learning to walk in minutes using massively parallel deep reinforcement learning,” arXiv:2109.11978, 2021.
[19]. D. Kim et al., “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” arXiv:1909.06586, 2019.
[20]. M. Gurram, P. K. Uttam, and S. S. Ohol, “Reinforcement learning for quadrupedal locomotion,” arXiv:2410.10438, 2024.
[21]. S. Katayama and T. Ohtsuka, “Whole-body model predictive control with rigid contacts via online switching time optimization,” arXiv:2203.00997, 2022.
[22]. M. Elobaid et al., “Adaptive non-linear centroidal MPC with stability guarantees,” arXiv:2409.01144, 2024.
[23]. J. Z. Zhang et al., “Whole-body model-predictive control of legged robots with MuJoCo,” arXiv:2503.04613, 2025.
[24]. I. Dadiotis et al., “Whole-body MPC for highly redundant legged manipulators,” arXiv:2310.02907, 2023.
[25]. P. Arm et al., “Pedipulate: Enabling manipulation skills using a quadruped robot’s leg,” arXiv:2402.10837, 2024.
[26]. M. Shafiee, G. Bellegarda, and A. Ijspeert, “ManyQuadrupeds: Learning a single locomotion policy for diverse quadruped robots,” arXiv:2310.10486, 2023.
[27]. Open Source Robotics Foundation, “ROS 2 documentation: Foxy Fitzroy (LTS),” n.d. [Online]. Available: https://docs.ros.org/en/foxy/
[28]. ANYbotics, “Elevation Mapping,” GitHub repository, n.d. [Online]. Available: https://github.com/ANYbotics/elevation_mapping

[29]. ANYbotics, “ANYmal: Autonomous legged robot for industrial inspection,” n.d. [Online]. Available: https://www.anybotics.com/robotics/anymal/

[30]. V. Makoviychuk et al., “Isaac Gym: High performance GPU-based physics simulation for robot learning,” arXiv:2108.10470, 2021.

[31]. C. Mastalli et al., “Crocoddyl: An efficient and versatile framework for multi-contact optimal control,” arXiv:1909.04947, 2020.

[32]. M. Ferigo, S. Traversaro, H. Dallali, and L. Natale, “A generic synchronous dataflow architecture to rapidly prototype and deploy robot controllers,” Int. J. Adv. Robotic Systems, vol. 17, no. 3, 2020.

[33]. M. Cai, M. Hasanbeig, S. Xiao, A. Abate, and Z. Kan, “Modular deep reinforcement learning for continuous motion planning with temporal logic,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 7973–7980, 2021.

[34]. M. Menner and K. Berntorp, “Simultaneous state estimation and contact detection for legged robots,” arXiv:2404.03444, 2024.

[35]. G. Chen and L. Hong, “Research on environment perception system of quadruped robots based on LiDAR and vision,” Drones, vol. 7, no. 5, 329, 2023.

[36]. M. A. Olufade et al., “Multimodal sensor fusion for autonomous systems,” Journal of Basic and Applied Research International, vol. 31, no. 4, pp. 37–54, 2025.

[37]. R. Yu et al., “Walking with terrain reconstruction,” arXiv:2409.15692, 2024.

[38]. X. Xu et al., “A review of multi-sensor fusion SLAM systems based on 3D LiDAR,” Remote Sensing, vol. 14, no. 12, 2835, 2022.

[39]. M. Camurri et al., “Pronto: A multi-sensor state estimator for legged robots,” Frontiers in Robotics and AI, vol. 7, 68, 2020.

[40]. S. Han et al., “Tightly coupled optimization-based GPS-visual-inertial odometry,” arXiv:2203.02677, 2022.

[41]. P. Fankhauser, M. Bloesch, and M. Hutter, “Probabilistic terrain mapping for mobile robots with uncertain localization,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[42]. Legged Robotics, “Elevation Mapping CuPy: GPU-accelerated elevation mapping for legged robots,” n.d. [Online]. Available: https://leggedrobotics.github.io/elevation_mapping_cupy/

[43]. L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, “Where should I walk? Predicting terrain properties from images via self-supervised learning,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.

[44]. R. Hartley et al., “Contact-aided invariant extended Kalman filtering,” arXiv:1805.10410, 2018.

[45]. M. Bloesch et al., “State estimation for legged robots on unstable and slippery terrain,” in Proc. IEEE/RSJ IROS, 2013, pp. 6058–6064.

[46]. M. Wan et al., “State estimation for quadruped robots on non-stationary terrain,” Sensors, vol. 24, no. 22, 7290, 2024.

[47]. S. Jung, W. Yang, and A. Kim, “Co-RaL: Complementary radar–leg odometry,” arXiv:2407.05820, 2024.

[48]. G. Haddeler et al., “Traversability analysis with vision and terrain probing,” Frontiers in Robotics and AI, vol. 9, 887910, 2022.

[49]. S. Yang et al., “Multi-IMU sensor fusion for legged robots,” arXiv:2507.11447, 2025.

[50]. D. Wisth, M. Camurri, and M. Fallon, “VILENS: Visual, inertial, lidar, and leg odometry,” IEEE Trans. Robotics, vol. 39, no. 1, pp. 309–326, 2023.

[51]. Boston Dynamics, “Spot robot specifications,” 2020. [Online]. Available: https://bostondynamics.com/wp-content/uploads/2020/10/spot-specifications.pdf

[52]. A. Go, S. Lee, J. Park, and H. Kim, “Stakeholder perspectives on safety issues in collaborative mobile robots: A case study of quadruped robot applications in a smart factory,” Applied Sciences, vol. 14, no. 22, p. 10232, Nov. 2024, doi: 10.3390/app142210232.

[53]. M. Pencelli et al., “Testing the robustness of quadruped robots for unmanned inspection activities,” IPTC, 2024.

[54]. J. Stanley, “Robot police dogs are here. Should we be worried?” ACLU, 2021.

[55]. J. Luca, “The rise of robotic police dogs,” Medium, Nov. 2023.

[56]. I. M. Shaughnessey, “The ethics of robots in war,” NCO Journal, Feb. 2024.

[57]. A. Hamrani et al., “Smart quadruped robotics: A systematic review,” Advanced Robotics, vol. 39, no. 4, pp. 1–27, 2024.

[58]. Q. Li et al., “Quadruped robots: Bridging mechanical design, control, and applications,” Robotics, vol. 14, no.5,57,2025.