

S.A.A.R: Design and Implementation of a 12-DOF Quadruped Robot Using ROS 2 Humble, Raspberry Pi 4B, and Inverse Kinematics-Based Crawl Gait with Stand, Sit, and Walk Behaviours

Rohit Sonawane¹, Vaibhav Suryawanshi², Asad Sayyed³, Shrikant Taware⁴, Rohan Kharat⁵,
Dhananjay Poul⁶

¹²³⁴⁵Students, Department of Electronics and Telecommunication Engineering

⁶Assistant Professor, Dhole Patil College of Engineering, Savitribai Phule Pune University, Pune,
Maharashtra, India

Abstract—This paper presents the design, development, and experimental evaluation of S.A.A.R. (Semi-Autonomous Articulated Robot), a 12 Degree-of-Freedom (DOF) quadruped robot inspired by canine locomotion. The mechanical structure, adapted from an open-source GrabCAD CAD model, features four three-DOF legs each comprising a shoulder abductor link, a 100 mm upper leg link, and a curved lower shank link. Twelve 25 kg/cm digital servo motors at 7.4 V provide joint actuation, powered by an 11.4 V 2200 mAh 3S Li-Po battery through a 20 A 330 W DC-DC buck converter. The robot is controlled by a Raspberry Pi 4B running Ubuntu 22.04 LTS with ROS 2 Humble as the middleware. The S.A.A.R Driver implements four motion behaviours: Stand Up, Sit Down, Crawl/Walk (Single Leg Pattern gait), and Rest. A closed-form analytical IK solver maps desired foot positions (X, Y, Z) and shoulder offset (a) to joint angles (θ_1 , θ_2 , θ_3). Remote operation is achieved via Bluetooth keyboard. IK instability during crawl locomotion is investigated and attributed to link length assembly tolerances, body frame compliance, and open-loop control architecture.

Keywords—Quadruped Robot; Inverse Kinematics; ROS 2 Humble; Raspberry Pi 4B; Single Leg Pattern Gait; Crawl Gait; Legged Locomotion

I. INTRODUCTION

Legged robots have attracted sustained research interest due to their ability to traverse discontinuous and uneven terrain inaccessible to wheeled platforms [1]. Among legged morphologies, the quadruped configuration — four legs, three DOF each — offers a compelling balance between mechanical complexity, static stability, and bio-mechanical relevance to mammalian locomotion [2].

S.A.A.R. (Semi-Autonomous Articulated Robot) is a 12-DOF quadruped platform developed as a final-year engineering project at Dhole Patil College of Engineering, SPPU. It assigns three rotational joints per leg — shoulder abduction, hip flexion, and knee flexion — following the standard three-link serial chain morphology.

The control stack runs on a Raspberry Pi 4B under Ubuntu 22.04 LTS using ROS 2 Humble [4]. The S.A.A.R Driver implements four motion behaviours: Stand Up, Sit Down, Crawl/Walk (Single Leg Pattern gait), and Rest. Servo actuation is managed via a PCA9685 PWM driver over I²C. Power uses an 11.4 V Li-Po battery regulated to 7.4 V through a 20 A DC-DC buck converter. Wireless teleoperation uses a Bluetooth HID keyboard.

IK instability causing body lean and joint saturation was observed during crawl locomotion. This paper investigates root causes and proposes remediation. Structure: Section II — Literature Review; Section III — System Design; Section IV — Motion Behaviours and Kinematics; Section V — Results; Section VI — Conclusion.

II. LITERATURE REVIEW

Legged robot research spans from McGhee and Frank's foundational gait stability analysis [1] to Boston Dynamics' Spot platform [6]. For low-cost educational platforms, the open-source Spot Micro quadruped [7] established the viability of commodity servo motors and 3D-printed structures — a design philosophy directly applicable to S.A.A.R.

Stand-up and sit-down behaviours are typically implemented as interpolated joint-space trajectories between rest and standing postures, reducing mechanical stress compared to abrupt servo initialisation [8].

The Single Leg Pattern (SLP) crawl gait is the most statically stable quadruped gait, guaranteeing a positive static stability margin provided the centre of mass lies within the support polygon of the three stance feet [5].

Analytical IK for a three-DOF leg decomposes into a lateral abduction angle and a planar two-link chain [9]. Accuracy is sensitive to link lengths: $\pm 3\text{--}5$ mm discrepancies produce 8–12 mm foot position errors and visible body sway in open-loop systems [10]. ROS 2 has been validated for embedded servo control on Raspberry Pi at gait frequencies below 2 Hz [11].

III. METHODOLOGY / SYSTEM DESIGN

A. Mechanical Structure

The S.A.A.R mechanical platform is based on the GrabCAD open-source quadruped CAD design [3], adapted for local fabrication. The body frame (Fig. 1, Fig. 2) houses the computing stack and battery. Four hip servo assemblies at the body corners initiate each three-link leg chain.

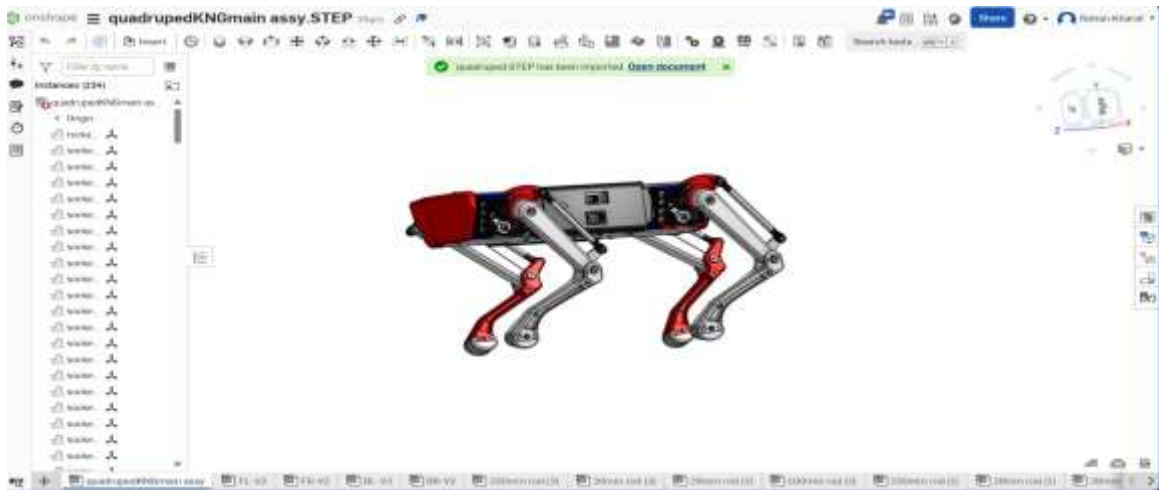


Fig. 1. S.A.A.R full CAD assembly (Onshape — quadrupedKNGmain assy). Four three-DOF legs with shoulder, upper leg, and shank segments.

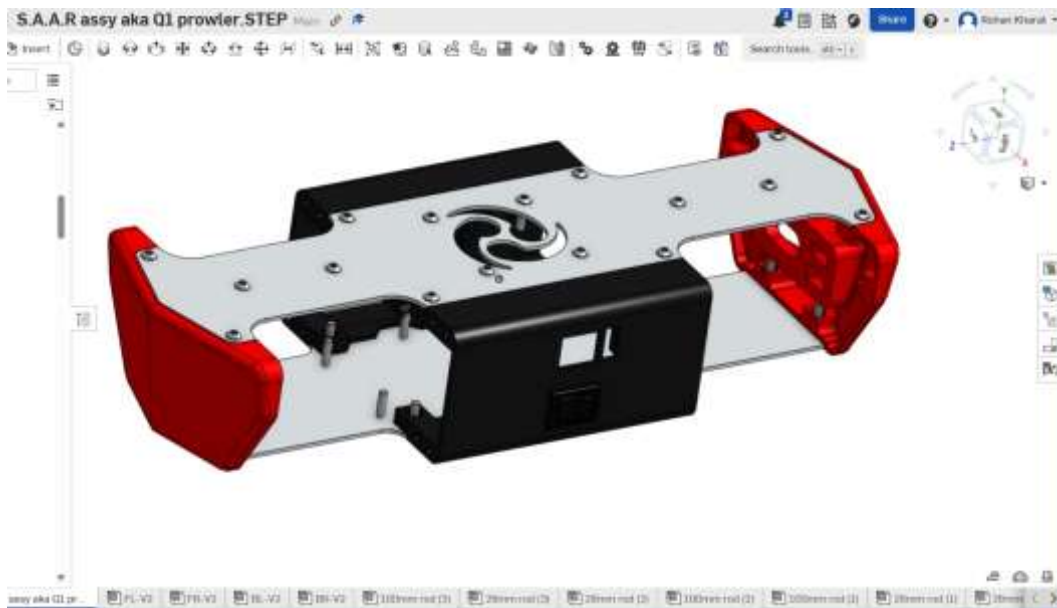


Fig. 2. S.A.A.R body frame CAD (S.A.A.R assy / Q1 Prowler). Hip servo mount slots at all four corners; central bay for computing stack.

B. Leg Link Geometry

Each leg has three links. The upper leg link (Fig. 3) has an effective kinematic length of 100 mm. The lower shank link (Fig. 4) is a curved member with bounding box ~ 129.5 mm $\times$ 102.9 mm; effective kinematic length estimated at ~ 115 mm.

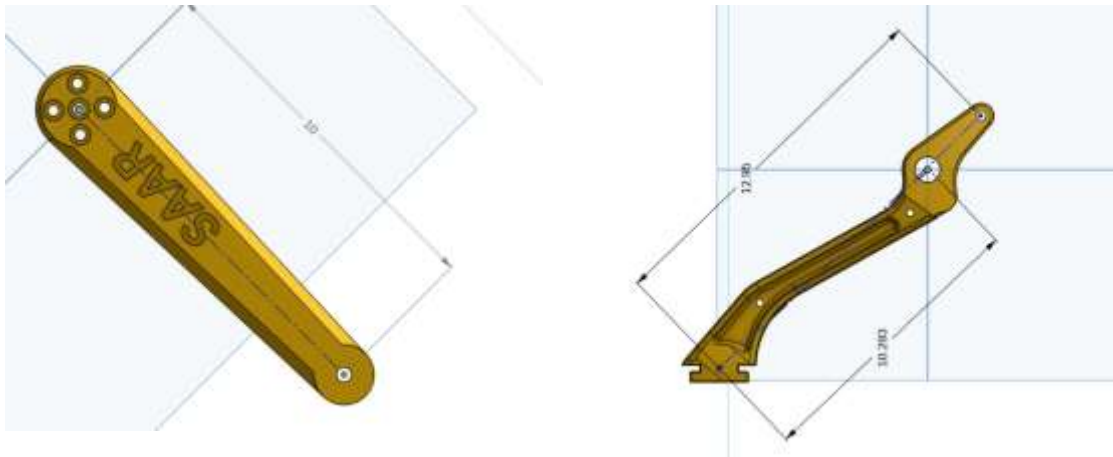


Fig. 3 (left). Upper leg link — 100 mm effective length, SAAR-branded. Fig. 4 (right). Lower shank link — $\sim 129.5 \times 102.9$ mm bounding box.

C. Electronics and Power Architecture

Table I lists all major system components. The Raspberry Pi 4B runs the full ROS 2 stack. Twelve 25 kg/cm digital servo motors provide joint actuation at 7.4 V across a 180° range. The PCA9685 PWM driver generates independent 50 Hz signals for all 12 servos over I²C — necessary because the Raspberry Pi lacks sufficient native hardware PWM outputs.

An 11.4 V 2200 mAh 3S Li-Po battery powers the system. A 20 A 330 W DC-DC buck converter steps this to 7.4 V for the servo rail, handling up to 20 A continuous current — adequate for simultaneous multi-servo operation. The Raspberry Pi is powered independently via USB-C to prevent servo voltage transients from causing resets.

Component	Specification	Role
Raspberry Pi 4B	4 GB RAM, quad-core 1.8 GHz, Ubuntu 22.04	Main controller / ROS 2 host
Digital Servo (×12)	25 kg/cm, 180°, 7.4 V	Joint actuation (3 per leg)
PCA9685	16-ch PWM, I ² C (0x40), 50 Hz	Servo PWM driver
Li-Po Battery	11.4 V, 2200 mAh, 3S	Main power source
DC-DC Buck Converter	20 A, 330 W, 11.4 V → 7.4 V	Regulated servo power rail
Bluetooth Keyboard	HID profile, onboard Pi BT module	Wireless teleoperation

TABLE I. S.A.A.R System Components

D. Software Architecture — S.A.A.R Driver

The S.A.A.R Driver runs entirely on the Raspberry Pi within the ROS 2 Humble framework. Fig. 5 shows the complete software architecture. The pipeline is organised into three layers: the Input Interface, the Motion Controller, and the Hardware Interface.

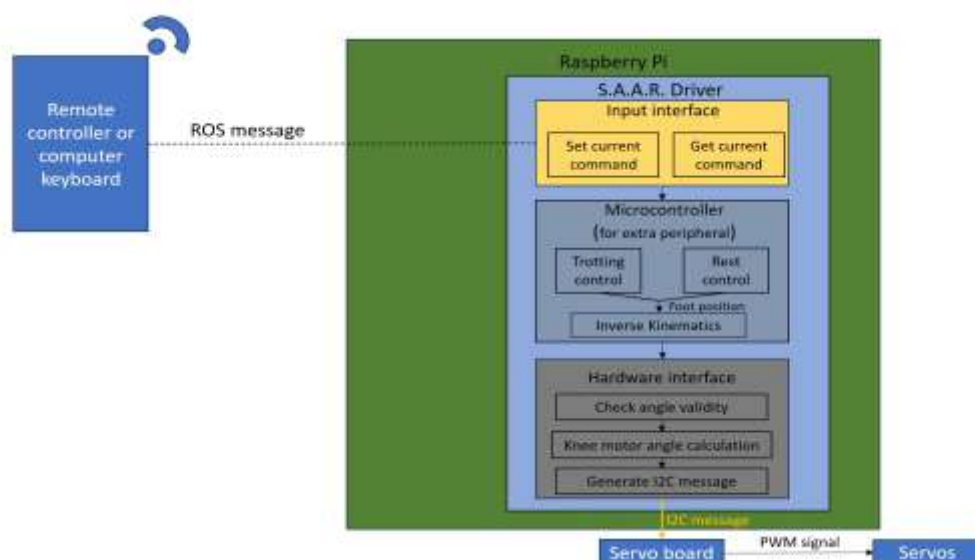


Fig. 5. S.A.A.R software architecture. Remote controller sends ROS messages to the Input Interface; Motion Controller generates foot positions; Hardware Interface validates angles, computes knee offsets, and generates I²C commands to the servo board.

Input Interface: Receives ROS messages from the Bluetooth keyboard node. Exposes Set Current Command (write new motion command) and Get Current Command (read active state) services.

Motion Controller: Implements four motion behaviours — (1) **Stand Up** — interpolates all 12 joints from rest to standing posture; (2) **Sit Down** — reverses trajectory to low crouch; (3) **Crawl/Walk** — executes Single Leg Pattern gait; (4) **Rest** — holds servos at relaxed default position.

Hardware Interface: Receives IK joint angles, performs: (i) **Check Angle Validity** — clips to servo mechanical limits; (ii) **Knee Motor Angle Calculation** — four-bar linkage correction; (iii) **Generate I²C Message** — encodes PWM and transmits to PCA9685 servo board.

Layer	Component	Function
Input Interface	Set / Get Current Command	Receives and stores active motion command from BT keyboard
Motion Controller	Stand Up	Interpolates joints to nominal standing posture
Motion Controller	Sit Down	Interpolates joints to low rest / crouch posture
Motion Controller	Crawl / Walk (SLP)	Sequential single-leg swing gait; generates foot positions
Motion Controller	Rest	Holds servos at relaxed default angles
Hardware Interface	Check Angle Validity	Clips joint angles to servo mechanical limits
Hardware Interface	Knee Motor Angle Calc.	Applies four-bar linkage correction to knee joint
Hardware Interface	Generate I ² C Message	Encodes angles as PWM; sends to PCA9685 servo board

TABLE II. S.A.A.R Driver — Layer-by-Layer Software Summary

IV. MOTION BEHAVIOURS AND INVERSE KINEMATICS

A. Stand Up and Sit Down

The Stand Up behaviour interpolates all 12 servo angles linearly from rest configuration to a predefined standing posture over ~1.5 s. This prevents mechanical shock and ensures symmetric body rise. Sit Down executes the reverse trajectory to the low-crouch rest configuration under the same interpolation scheme. Both are triggered by keyboard command.

B. Crawl / Walk Gait — Single Leg Pattern

The Crawl gait uses the SLP, cycling legs in the fixed sequence: Left-Front (LF) → Right-Hind (RH) → Right-Front (RF) → Left-Hind (LH). The swing leg follows a trapezoidal trajectory — rise to clearance height $h_c = 30$ mm, advance by step length $\Delta x = 40$ mm, descend to ground — while the other three legs remain in stance. SLP guarantees a positive static stability margin at all times [5].

Parameter	Value
Gait type	Single Leg Pattern (SLP) — Crawl/Walk
Leg swing sequence	LF → RH → RF → LH
Step frequency	0.5 Hz
Step length (Δx)	40 mm
Foot clearance (h_c)	30 mm
Stance duration	1.5 s per leg
Trajectory profile	Trapezoidal (rise – advance – descend)

TABLE III. S.A.A.R Crawl Gait Parameters

C. Inverse Kinematics Formulation

The IK for each three-DOF leg is solved analytically using geometric decomposition (Fig. 6, Fig. 7). Given desired foot position (X, Y, Z) and shoulder offset a:

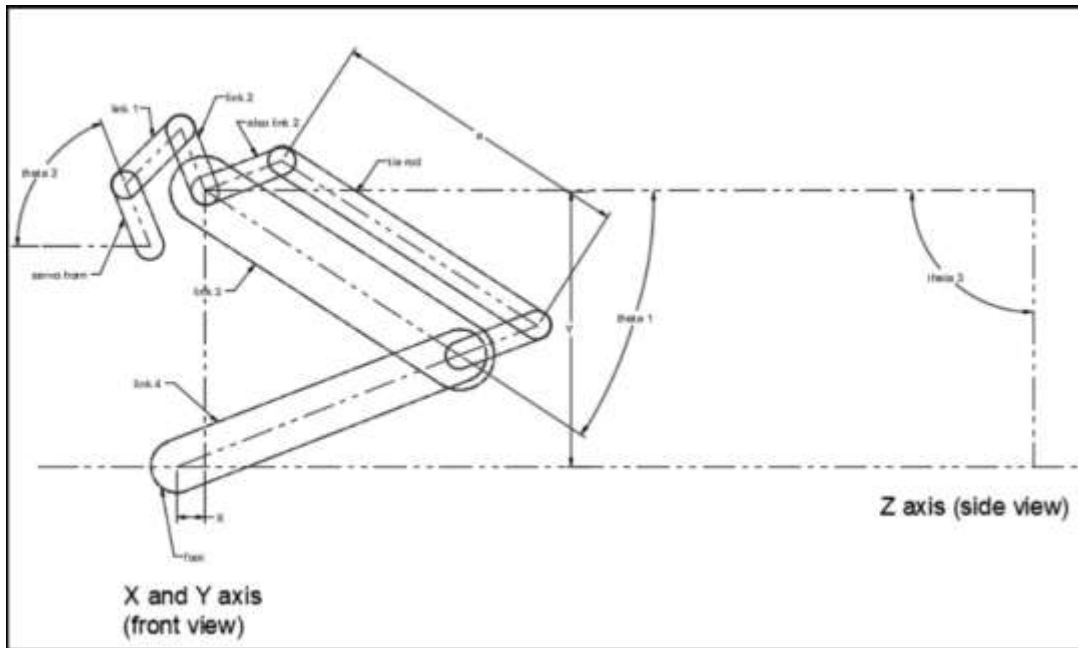


Fig. 6. Kinematic diagram — front view (X-Y plane) and side view (Z axis). Links 1–4, servo horn, tie rod, and foot labelled.

Joint 1 — Shoulder Abduction (θ_1): $\theta_1 = 90^\circ - \tan^{-1}(-X / \sqrt{Y^2 + Z^2}) - \cos^{-1}(\sqrt{X^2 + Y^2 + Z^2} / 2a)$

Joint 2 — Hip Flexion (θ_2): $\theta_2 = 90^\circ - \cos^{-1}(1 - (X^2 + Y^2 + Z^2) / 2a^2) + \theta_1$

Joint 3 — Knee Flexion (θ_3): $\theta_3 = \sin^{-1}(Z / \sqrt{Y^2 + Z^2}) + 90^\circ$

After angle computation, the Hardware Interface applies a knee motor angle correction for the four-bar linkage, clips all angles to servo limits (0° – 180°), and maps to PWM: pulse (μs) = $1500 + (\theta / 90^\circ) \times 500$.

V. RESULTS AND DISCUSSION

A. Motion Behaviour Validation

All four motion behaviours were successfully executed. Stand Up and Sit Down transitions complete smoothly within ~ 1.5 s without mechanical shock. Rest correctly holds servo positions. Crawl/Walk produces forward progression of approximately 15–20 mm per four-leg cycle on flat surfaces with no leg interference.

[PHOTO PLACEHOLDER — Insert physical robot photograph here]

B. IK Instability — Observations and Root Cause Analysis

During crawl locomotion the robot exhibits: (1) persistent lateral body lean of 8 – 12° ; (2) occasional joint angle saturation; and (3) inconsistent step lengths across legs. Three root causes were identified:

1. Link Length Parameter Mismatch: IK uses nominal CAD link lengths. Assembled hardware exhibits ± 3 – 5 mm discrepancies per link due to servo horn offsets and bracket thickness. Errors accumulate through the chain, producing foot position errors of 8–12 mm.

2. Body Frame Compliance: The body frame exhibits torsional flex under combined servo loads, shifting actual hip joint origins relative to the assumed body-fixed frame and corrupting IK workspace for all legs simultaneously.

3. Open-Loop Architecture: No IMU tilt feedback, no foot force sensing, and no servo encoder feedback means accumulated errors from (1) and (2) cannot be corrected at runtime.

Parameter	Target / Expected	Observed
Stand Up / Sit Down	Smooth, < 2 s	Achieved, ~ 1.5 s
Step length / cycle	40 mm	15–20 mm (floor slip)
Body lateral lean	$< 2^\circ$	8 – 12°
Gait frequency	0.5 Hz	0.4–0.5 Hz
Joint angle error	$< 2^\circ$	5 – 15° (accumulated)
IK feasibility	Always valid	Occasional saturation

TABLE IV. Design Targets vs. Observed Experimental Performance

VI. CONCLUSION AND FUTURE SCOPE

This paper presented S.A.A.R, a 12-DOF quadruped robot controlled by a Raspberry Pi 4B running ROS 2 Humble. The S.A.A.R Driver implements four validated motion behaviours — Stand Up, Sit Down, Crawl/Walk (SLP gait), and Rest — through a layered software architecture. Power is provided by an 11.4 V 2200 mAh Li-Po battery through a 20 A buck converter to twelve 25 kg/cm servo motors.

IK instability during crawl locomotion was systematically attributed to link length parameter mismatch, body frame torsional compliance, and absence of closed-loop feedback. Stand Up and Sit Down transitions were successful, confirming correctness of the joint-space interpolation approach.

Future work: (1) Kinematic Calibration — measure actual link lengths, update IK parameters; (2) IMU Feedback — MPU-6050 for body-tilt compensation; (3) Trajectory Smoothing — cubic Bézier curves for foot swing paths; (4) Advanced Gaits — trot gait via Central Pattern Generator (CPG); (5) Reinforcement Learning — sim-to-real policy in Gazebo/PyBullet for terrain-adaptive locomotion.

ACKNOWLEDGEMENT

The authors sincerely thank their internal guide Mr. Dhananjay Paul, Assistant Professor, Department of Electronics and Telecommunication Engineering, Dhole Patil College of Engineering, for his continuous guidance throughout this project. The authors also acknowledge [External Guide Name], [Designation], [Organisation], for external mentorship and support. Special thanks to the project's core technical contributor for foundational work in kinematic design, ROS 2 architecture, and system integration.

VII. REFERENCES

- [1] R. B. McGhee and A. A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences*, vol. 3, no. 2, pp. 331–351, 1968.
- [2] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822–10825, 2008.
- [3] GrabCAD Community, "DIY Quadruped Robot," *GrabCAD Library*, 2020. [Online]. Available: <https://grabcad.com/library/diy-quadruped-robot-1> [Accessed: Mar. 2025].
- [4] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, eabm6074, May 2022.
- [5] L. Maes, A. Schootemeijer, and A. Aerts, "Static stability and the gait transition in quadrupeds," *Journal of Experimental Biology*, vol. 211, pp. 3196–3206, 2008.
- [6] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot," in *Proc. IEEE/RSJ IROS*, Madrid, Spain, Oct. 2018, pp. 2245–2252.
- [7] M. Kamrad, "Spot Micro Quadruped Project," *GitHub*. [Online]. Available: <https://github.com/mike4192/spotMicro> [Accessed: 2025].
- [8] P. Fankhauser et al., "Robust rough-terrain locomotion with a quadrupedal robot," in *Proc. IEEE ICRA*, Brisbane, Australia, 2018, pp. 5761–5768.
- [9] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London: Springer, 2009, ch. 2, pp. 40–62.
- [10] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, Sep. 2019.
- [11] A. Soragna et al., "Active Benchmark of ROS 2 Communication Middlewares," in *Proc. IEEE ICRA*, Xi'an, China, 2021, pp. 1–7.