FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Development of a Lightweight Hybrid Legged Robot

Gonçalo Paulino



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Vítor Hugo Pinto

Second Supervisor: Prof. Paulo Gomes da Costa

July 30, 2026

© Gonçalo Paulino, 2026

Resumo

Robôs híbridos com pernas e rodas combinam a adaptabilidade da locomoção por pernas com a velocidade e eficiência energética das rodas, permitindo uma maior versatilidade em diferentes tipos de terreno. Esta dissertação apresenta o desenvolvimento de um robô quadrúpede híbrido, leve e de baixo custo, destinado à operação em terrenos regulares e irregulares, com potenciais aplicações em busca e salvamento, reconhecimento e inspeção industrial.

O trabalho parte da plataforma quadrúpede open-source Dingo, que foi modificada para integrar quatro rodas e motores nas extremidades das pernas. Foram redesenhados: uma nova estrutura corporal, pernas inferiores adaptadas, um sistema de alimentação e eletrónica atualizado, bem como software de controlo para integrar a locomoção com rodas e pernas. A plataforma utiliza uma Raspberry Pi 5 como computador principal, uma Raspberry Pi Pico W para o controlo dos motores das rodas e uma arquitetura baseada em ROS. Foi também implementado um mecanismo de interbloqueio mecânico que permite manter uma configuração agachada durante a locomoção com rodas sem exigir binário dos servomotores das pernas, reduzindo significativamente o consumo energético.

A plataforma final apresenta uma massa de 3,54 kg e um custo aproximado de 580 €, sendo a sua estrutura quase totalmente fabricada por impressão 3D. Os ensaios experimentais demonstraram uma velocidade de 0,070 m/s em locomoção com rodas, 0,043 m/s em trote e 0,107 m/s combinando ambos. O modo de condução de baixo consumo reduziu a potência média de 29,2 W para 18,0 W e aumentou a autonomia projetada de 65 para 105 minutos. A locomoção com rodas apresentou ainda um consumo energético por unidade de distância substancialmente inferior ao da locomoção por pernas. Os testes em diferentes superfícies demonstraram a capacidade da plataforma para combinar os dois modos de locomoção, incluindo a utilização simultânea de rodas e pernas em terreno irregular.

Os resultados demonstram a viabilidade de uma plataforma híbrida leve, acessível e maioritariamente impressa em 3D, constituindo uma base para o desenvolvimento futuro de sistemas de locomoção híbrida de maior velocidade, autonomia e robustez.

Palavras-chave: robô híbrido com rodas e pernas, robô quadrúpede, robótica de baixo custo, impressão 3D, locomoção híbrida, eficiência energética, ROS

Abstract

Hybrid wheeled-legged robots combine the adaptability of legged locomotion with the speed and energy efficiency of wheels, allowing greater versatility across different types of terrain. This dissertation presents the development of a lightweight, low-cost hybrid quadruped robot intended to operate on both regular and irregular terrain, with potential applications in search and rescue, reconnaissance, and industrial inspection.

The work builds on the open-source Dingo quadruped platform, which was modified to integrate four motorized wheels at the tips of the legs. A new structural body, adapted lower legs, an updated power and electronics system, and control software integrating wheeled and legged locomotion were developed. The platform uses a Raspberry Pi 5 as its main computer, a Raspberry Pi Pico W to control the wheel motors, and a ROS based architecture. A mechanical interlock was also implemented, which holds a crouched configuration during wheeled locomotion without requiring torque from the leg servos, significantly reducing energy consumption.

The final platform has a mass of 3.54 kg and a cost of approximately €580, with its structure almost entirely manufactured by 3D printing. Experimental tests showed a speed of 0.070 m/s in wheeled locomotion, 0.043 m/s when trotting, and 0.107 m/s when both were combined. The low-power driving mode reduced the average power from 29.2 W to 18.0 W and increased the projected endurance from 65 to 105 minutes. Wheeled locomotion also consumed substantially less energy per unit of distance than legged locomotion. Tests on different surfaces demonstrated the platform's ability to combine both locomotion modes, including the simultaneous use of wheels and legs on irregular terrain.

The results demonstrate the feasibility of a lightweight, affordable, and mostly 3D-printed hybrid platform, which provides a foundation for the future development of hybrid locomotion systems with greater speed, endurance, and robustness.

Keywords: wheeled-legged robot, quadruped robot, low-cost robotics, 3D printing, hybrid locomotion, energy efficiency, ROS

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all, and include 17 goals[1] to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation develops a lightweight, low-cost hybrid wheeled-legged quadruped robot intended for search and rescue, industrial inspection, and reconnaissance. Its relationship to the SDGs follows from two distinct properties of the work, what the robot is for, and how it is built. The intended applications, surveying hazardous areas and inspecting infrastructure that is hazardous for people to enter, connect the platform to SDG 11. The way it is built, by extending an open-source design into an almost entirely 3D-printable machine assembled from commodity components, connects it to SDG 9, both as a contribution to accessible research capability and as an exercise in resource efficiency, since the low-power driving mode developed here exists precisely to reduce the energy cost of moving the robot.

The robot produced in this work is a laboratory prototype and has not been deployed in a real rescue or inspection operation. The contributions listed below are therefore expressed as the capabilities the platform enables and the barriers to access it lowers, together with the indicators by which that impact would be measured, rather than as impact already realized.

The specific Sustainable Development Goals addressed are the following.

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation. The targets addressed are 9.4 resource-use efficiency and environmentally sound technologies, and 9.5 scientific research and technological capability.

SDG 11 Make cities and human settlements inclusive, safe, resilient and sustainable. The target addressed is 11.5 reduce deaths, people affected, and economic losses caused by disasters.

[1] https://sdgs.un.org/goals

SDG	Target	Contribution	Performance Indicators and Metrics
9	9.4	Rolls instead of walking on flat ground, and turns a mechanical design decision, the interlocking crouch, into an energy saving by releasing the leg servo torque while driving.	Driving power reduced by 38.4%, from 29.2 to 18.0 W, endurance from 65 to 105 min, and cost of transport from 11.9 to 7.4.
	9.5	A reproducible, documented research testbed extending an open-source quadruped, which can be replicated with a desktop 3D printer and hobby-grade electronics.	Bill of materials of €577.70, against €792.42 for the original legged-only platform, with almost the entire structure printed from 1278 g of PLA.
11	11.5	Surveys hazardous areas in place of a first responder, reducing human exposure, and can be carried and deployed by one person.	Mass of 3.54 kg, operational from power-on without an external computer, and a 130.5 g camera carried in the field over gravel, grass and dirt.

Agradecimentos

Quero começar por agradecer ao meu orientador, Professor Vítor Hugo Pinto, por todos os conselhos e ideias que partilhou ao longo do desenvolvimento deste trabalho, e por toda a disponibilidade que demonstrou sempre que precisei de ajuda. Agradeço-lhe também por me ter dado a oportunidade de trabalhar num projeto de que gosto e que me permitiu aprender imenso sobre robótica e engenharia, tornando esta dissertação numa experiência que vou certamente levar comigo.

A toda a minha família, por todo o apoio que me deram ao longo deste percurso. Um agradecimento especial aos meus pais e avós, por acreditarem sempre em mim, por me apoiarem nas minhas escolhas e por me incentivarem a seguir aquilo de que gosto. Sem eles, não teria tido as oportunidades que me trouxeram até aqui. À minha irmã, Sofia, por me aturar todos os dias e por toda a ajuda que me deu durante este trabalho, especialmente na parte mais repetitiva dos testes.

Aos meus amigos de infância, por terem estado sempre presentes, por puxarem por mim e por me lembrarem que a vida é muito mais do que trabalho e estudos.

Aos amigos que fizeram estes cinco anos de faculdade comigo, desde o primeiro dia até ao último, por terem tornado esta caminhada muito mais divertida. Foram cinco anos de aulas, trabalhos, noites mal dormidas e, sobretudo, muitos bons momentos que não teriam sido os mesmos sem vocês. Levo comigo não só o que aprendi na faculdade, mas também todas as memórias que fizemos juntos.

À Joana e à Rita, por estarem sempre disponíveis para me ouvir quando precisei e por todo o apoio que me deram. Obrigado por estarem sempre lá.

Aos amigos que conheci em Poznań, apesar de vos ter conhecido há relativamente pouco tempo, obrigado por tudo. É bom saber que, mesmo em pouco tempo, se podem criar boas amizades.

Ao Paulo Cx, por estes três anos em que me ensinou muito mais do que fazer pulseiras. Obrigado por me motivar sempre a seguir aquilo de que gosto e por todas as experiências que levo comigo destes anos.

Por último, um obrigado muito especial à Maria. Obrigado por todos os dias estares lá para mim, por todo o apoio, paciência e carinho incondicional ao longo deste percurso. Tenho a certeza de que, sem ti, não teria sido o mesmo.

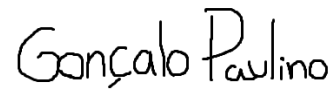
A todos os que, de uma forma ou de outra, fizeram parte destes cinco anos e contribuíram para que chegasse até aqui, o meu sincero obrigado.

Gonçalo Paulino

Declaration of honor

I declare that this dissertation is my original work and has not previously been submitted or assessed in any other course or curricular unit, whether at this or any other institution. References to other authors (statements, ideas, thoughts), as well as the use of published works fully complies with the rules of attribution and are appropriately cited in the text and bibliography, in accordance with reference standards. I am aware that plagiarism and self-plagiarism constitute academic misconduct.
I further declare that I have not used Artificial Intelligence tools to produce any part(s) of this dissertation or, when such tools were used, their use and results are clearly indicated and were carefully reviewed for the detection of errors, inaccuracies, unfounded attributions, or other forms of content and argument bias, as well as for the determination of authorship.
I further declare that I have included studies carried out with multiple co-authors, having specified, with transparency and rigor, the contribution of each co-author, provided information on the possible existence of any other dissertation in which the same article has been included, reproduced the work in full with the permission of the journal in which it was published, and obtained the explicit written consent of all co-authors for the inclusion of the article.

Gonçalo Paulino



Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Problem Definition . . . 2
- 1.3 Literature Perspective . . . 2
- 1.4 Scope and Objectives . . . 3
- 1.5 Dissertation Structure . . . 4

2 Background and Fundamental Aspects **5**
- 2.1 Mobile Robots and Locomotion Basics . . . 5
  - 2.1.1 Wheeled Robots . . . 5
  - 2.1.2 Legged Robots . . . 5
  - 2.1.3 Wheeled-Legged Robots . . . 6
- 2.2 Wheeled-Legged Locomotion and Control Principles . . . 7

3 State of the Art **9**
- 3.1 Taxonomy of Hybrid Wheeled-Legged Robots . . . 9
- 3.2 Mechanical Design and Lightweight Approaches . . . 11
  - 3.2.1 Design Considerations . . . 11
  - 3.2.2 Low-Cost Solutions . . . 13
- 3.3 Control and Planning . . . 13
  - 3.3.1 Model Predictive Control . . . 14
  - 3.3.2 Terrain Perception and Mapping . . . 14
  - 3.3.3 Reinforcement Learning Approaches . . . 14
- 3.4 Positioning of This Work . . . 14

4 System Architecture **17**
- 4.1 Overview . . . 17
- 4.2 System Architecture Overview . . . 18
- 4.3 Base Platform and Modifications . . . 19
- 4.4 Hardware Components . . . 19
  - 4.4.1 Raspberry Pi 5 Main Computer . . . 19
  - 4.4.2 Raspberry Pi Pico W Microcontroller . . . 21
  - 4.4.3 Leg Servo Motors and BusLinker V2.5 . . . 21
  - 4.4.4 Wheel Motors and IBT-2 Drivers . . . 22
  - 4.4.5 MPU-6050 Inertial Measurement Unit . . . 22
  - 4.4.6 3D-Printed Chassis . . . 22
- 4.5 Power System . . . 23
- 4.6 Communication Architecture . . . 23

4.7 Software Architecture . . . 24
4.8 Design Rationale . . . 25
4.9 Hardware Component Summary . . . 26
4.10 Summary . . . 26

5 Mechanical Design **29**
5.1 Leg Mechanism . . . 29
5.1.1 Servo Torque Requirements . . . 32
5.2 Body Structure . . . 33
5.2.1 From Laser-Cut Acrylic to a Printed Chassis . . . 33
5.2.2 Interior Redesign . . . 33
5.3 Wheel Integration . . . 34
5.4 Manufacturing and Assembly . . . 37
5.5 Printed Parts . . . 38

6 Electronics **41**
6.1 Electronic Components . . . 41
6.1.1 Raspberry Pi 5 . . . 41
6.1.2 Raspberry Pi Pico W and I/O Shield . . . 41
6.1.3 BusLinker V2.5 . . . 41
6.1.4 IBT-2 Motor Drivers . . . 41
6.1.5 Inertial Measurement Unit . . . 42
6.1.6 DC-DC Converters . . . 42
6.2 Wiring and Connections . . . 42
6.2.1 Power Distribution . . . 42
6.2.2 Signal Wiring . . . 44
6.3 Power Budget . . . 44
6.3.1 Component Consumption and Measured System Power . . . 44
6.3.2 Servo Rail Sizing . . . 46
6.4 Component Layout . . . 47

7 Software and Control **49**
7.1 Software Architecture Overview . . . 49
7.2 Docker Deployment and Automatic Startup . . . 49
7.2.1 Containerized ROS Environment . . . 49
7.2.2 Power On and Launch Sequence . . . 50
7.3 Inverse Kinematics . . . 51
7.4 Gait Generation . . . 52
7.5 Operating Modes and State Machine . . . 53
7.6 Body Stabilization with IMU . . . 55
7.7 Wheel Control . . . 55
7.7.1 Firmware . . . 55
7.7.2 ROS Interface and Steering . . . 56
7.8 Input and Command Interface . . . 56

8 Experimental Evaluation **59**
8.1 Test Setup and Methodology . . . 59
8.1.1 Configuration Under Test . . . 59
8.1.2 Test Environments . . . 59

8.1.3 Energy Measurement . . . 59
8.1.4 Speed and Thermal Measurement . . . 60
8.2 Flat Terrain Locomotion . . . 60
8.2.1 Speed . . . 60
8.2.2 Energy Consumption and Endurance . . . 61
8.3 Uneven Terrain Locomotion . . . 63
8.3.1 Behavior by Surface . . . 63
8.3.2 Crawl Gait . . . 63
8.3.3 Simultaneous Wheeled and Legged Locomotion . . . 64
8.3.4 Body Attitude Compensation . . . 64
8.4 Load Capacity . . . 64
8.5 Field Demonstration with a Sensor Payload . . . 65
8.6 Discussion . . . 67
8.6.1 Comparison with the Platform It Extends . . . 67
8.6.2 Position Relative to the State of the Art . . . 68
8.6.3 What the Hybrid Arrangement Delivered . . . 69
8.6.4 Observed Limitations . . . 69

9 Conclusions and Future Work **71**
9.1 Summary of Contributions . . . 71
9.2 Limitations . . . 72
9.3 Future Work . . . 73
9.4 Final Remarks . . . 74

References **75**

A Bill of Materials **81**

B Knee Linkage Solution **83**
B.1 Intersection of Two Circles . . . 83
B.2 From the Servo Angle to the Knee Angle . . . 84
B.3 From the Knee Angle to the Servo Angle . . . 84

C Inverse Kinematics Derivation **87**
C.1 Leg Frame and Symmetry . . . 87
C.2 Hip Abduction Angle . . . 87
C.3 Reduction to the Leg Plane . . . 89
C.4 Planar Two-Link Solution . . . 89
C.5 Angle Conventions . . . 90

List of Figures

1.1 MAMMOTH rover navigating Mars-analog terrain [18] . . . 3
1.2 RIVR ONE robot in an urban delivery scenario [19] . . . 3

2.1 Static stability during stair climbing on Momaro. Left, the center of mass (red) lies inside the support polygon (green), so the front right leg can be lifted. Right, it stands on the next step. From [11]. . . . 7

3.1 Mammal like leg structures: (a) all-elbow, (b) all-knee, (c) front-knee-back-elbow, (d) front-elbow-back-knee. From [30] . . . 11
3.2 3 DOF Leg with a parallelogram linkage. From [32] . . . 12
3.3 Mini Cheetah leg, with the ab/ad actuator in green, the hip in purple and the knee in red. From [33] . . . 12
3.4 PADWQ robot [32] . . . 13
3.5 Dingo robot [21] . . . 13
3.6 Wheely robot [41] . . . 13

4.1 Parts created or modified in this work (highlighted) against those inherited from the original Dingo. . . . 18
4.2 High-level system architecture block diagram showing the computation, actuation, and sensing layers. . . . 19
4.3 Original Dingo quadruped, with fixed compliant feet [20], [21]. . . . 20
4.4 Wheeled-legged platform developed in this work. . . . 20
4.5 CAD model of the complete robot. . . . 23
4.6 Power distribution across the motor, servo and logic rails. . . . 24

5.1 Line diagram of the leg mechanism over the CAD model. In green the linkage members a to h, in blue the leg links L_2 and L_3, and in orange the wheel radius below the axle where L_3 ends. Redrawn from [20]. . . . 30
5.2 Exploded view of the three-level interior, with the Raspberry Pi 5, battery and servo converter on the base plate, the Pico W, IBT-2 drivers, BusLinker and perf-board on the middle plate, and the fans on top. . . . 34
5.3 Final design of the 3D-printed wheel rim, with the D-shaped bore, the boss for the locking screw, and the channel that retains the tire. . . . 35
5.4 The redesigned lower leg and wheel (left) against the original lower leg and foot (right), which meet the ground in nearly the same place. . . . 36
5.5 The interlocking profile between the leg links, which seats as the knee closes and gives the crouched pose a repeatable geometry. . . . 36
5.6 Redesigned lower leg with the JGY-371 motor and the printed rim and tire. . . . 37
5.7 Assembled robot viewed from above. . . . 38

6.1 Main electronic components of the robot. 43
6.2 Complete wiring of the robot. Solid arrows carry power, colored by rail voltage, and dashed arrows carry data. 45
6.3 Top view with the top plate removed, showing the four IBT-2 drivers (red) and the perfboard with the DFR1015 and 5 V bus (blue), the IMU (gray) and the 14.8 V bus (red). 48

7.1 Runtime software architecture, with the `joy` node feeding the driver node that holds the control pipeline and the hardware interfaces. 50
7.2 Kinematic model of one leg. Following [20]. 52
7.3 Behavior state machine. 54
7.4 Entry into the low-power driving mode, with the body ramped down until the leg links seat into the interlocking cut and the leg servos are relaxed. 54
7.5 The PS4 controller with each control named, and Table 7.1 the function bound to each. 56

8.1 The robot operating outdoors with the 360 camera on the top plate, the locomotion mode chosen to suit the ground. 66
8.2 Two frames from the carried camera. 67

C.1 Nomenclature for the hip abduction angle θ_1 in the frontal plane, with A from the abduction axis to the wheel axle. Following [20]. 88
C.2 Nomenclature for the hip flexion angle θ_2 and the knee angle θ_3 in the leg plane, with B the distance to the wheel axle. Following [20]. 90

List of Tables

2.1 Comparison between wheeled, legged, and wheeled-legged robots, the first two columns from [23] and the third added here (✓ satisfied, **X** not). 6

3.1 Comparison of Hybrid Legged-Wheeled Robots. 10

4.1 Comparison between the original Dingo design and the modified platform. 20
4.2 Communication interfaces summary. 24
4.3 ROS packages in the `dingo_ws` workspace. 25
4.4 Hardware component summary for the wheeled-legged robot. 26

5.1 Leg dimensions, keyed to Figure 5.1. Above the rule, the linkage members in the original design [20] and on this robot, of which a, b, f and j differ. 31
5.2 3D-printed parts of the robot. 39

6.1 Datasheet consumption of the main components. 45
6.2 Measured system power before and after the servo rail converter replacement, with the servos (LX-225) and the rail voltage (7.8 V) unchanged. Endurance is to 80% of the battery. 46

7.1 Function bound to each control of Figure 7.5. 57
7.2 Light bar color feedback. 57

8.1 Measured power, projected endurance to a 20% battery reserve, energy per meter and CoT by condition, normalized with each mode's own measured speed. 61
8.2 Measured load capacity by condition, with the mechanism that ended each test. . 65
8.3 This work compared with the original legged-only Dingo. Original figures from [20]. 67

A.1 Complete bill of materials with unit and total costs. 81

List of Acronyms

ASCII	American Standard Code for Information Interchange
AUD	Australian Dollar
CAD	Computer-Aided Design
CoT	Cost of Transport
CPU	Central Processing Unit
DC	Direct Current
DOF	Degrees of Freedom
DRL	Deep Reinforcement Learning
EUR	Euro
FDM	Fused Deposition Modeling
GPIO	General-Purpose Input/Output
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LiDAR	Light Detection and Ranging
LiPo	Lithium Polymer
MPC	Model Predictive Control
PCB	Printed Circuit Board
PID	Proportional-Integral-Derivative
PLA	Polylactic Acid
PWM	Pulse-Width Modulation
RAM	Random-Access Memory
RGB-D	Red-Green-Blue-Depth
ROS	Robot Operating System
RPM	Revolutions Per Minute
SDG	Sustainable Development Goal
SPI	Serial Peripheral Interface
TPU	Thermoplastic Polyurethane
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver-Transmitter
UBEC	Universal Battery Eliminator Circuit
UN	United Nations
URDF	Unified Robot Description Format
USB	Universal Serial Bus
WBC	Whole-Body Control
ZMP	Zero-Moment Point

Chapter 1

Introduction

1.1 Context and Motivation

Robots are becoming increasingly important in many areas, as they allow tasks to be carried out more efficiently and safely, especially in environments that are dangerous or difficult for humans to access. Fields such as search and rescue, industrial inspection, and reconnaissance rely heavily on robotic systems to operate in uncertain and hazardous conditions, where human intervention may be risky or impractical [1].

Most mobile robots are typically designed using either wheels or legs. Wheeled robots are widely adopted due to their mechanical simplicity, high achievable speeds, and good energy efficiency. However, their performance degrades significantly when dealing with obstacles, uneven terrain, or debris. In contrast, legged robots are well suited for rough and unstructured environments, as they can step over obstacles and adapt their posture to the terrain. Despite this flexibility, legged systems are generally slower, mechanically more complex, and more energy-consuming than wheeled platforms.

Hybrid wheeled-legged robots aim to take advantage of both locomotion principles, addressing the lack of speed and efficiency of legged robots on flat ground by enabling fast rolling locomotion combined with agile stepping when required [2]. By combining the adaptability of legs with the efficiency and speed of wheels, these robots can operate effectively across a wide range of terrains, making them particularly appealing for real-world missions that demand both endurance and mobility.

Despite the significant progress achieved in this field, many existing hybrid robotic platforms remain heavy, expensive, and mechanically complex. These factors limit their portability and reduce their suitability for widespread deployment outside specialized research or industrial contexts. Consequently, lightweight and cost-effective hybrid solutions remain relatively uncommon.

1.2 Problem Definition

Although recent advances in hybrid wheeled-legged robotics have produced highly capable research and commercial platforms, many of these systems are characterized by high cost, significant mass, and design complexity [3]. This creates a gap between the potential benefits of hybrid locomotion and its practical use in applications such as search and rescue or inspection tasks.

The motivation for this dissertation is to help reduce this gap by developing a lightweight and cost-effective wheeled-legged robot that preserves the core advantages of hybrid locomotion. The objective is to design a practical platform that is easy to transport and deploy, while remaining capable of operating in scenarios such as search and rescue, industrial inspection, and reconnaissance. By prioritizing simplicity and reduced weight, this work seeks to contribute toward more accessible hybrid robotic systems that can reliably combine efficient motion on flat terrain with adaptability to rough environments.

1.3 Literature Perspective

Mobile legged robotics has gained increasing attention in recent years, with quadruped robots in particular offering an effective compromise between stability and mobility. Fully legged systems such as ANYmal [4], [5], Unitree B2 [6], and Boston Dynamics Spot [7] have demonstrated impressive performance in terms of agility, robustness, and terrain adaptability.

Quadruped wheeled-legged robots typically adopt architectures similar to purely legged robots, with wheels integrated into different parts of the leg structure. Examples include robots with wheels mounted at the feet, such as ANYmal on Wheels [8], [9], Centauro [10], Momaro [11], Unitree Go2-W [12], and GQR0440-W [13]. Other designs adopt convertible configurations, such as Max [14] and robots where wheels are integrated at the knee joints [15], enabling transitions between walking and rolling locomotion modes. While these platforms demonstrate the feasibility and performance benefits of hybrid locomotion, most emphasize high mobility, speed, and robustness rather than reduced cost, low weight, or mechanical simplicity.

Hybrid wheeled-legged robots have proven valuable across a wide range of applications that require mobility in complex and unpredictable environments, including post-disaster rescue operations, military reconnaissance, and industrial inspection tasks [16]. In planetary exploration, platforms such as the Sherpa [17] and MAMMOTH [18] (Figure 1.1) rovers have been developed to investigate locomotion strategies suitable for extraterrestrial terrain. Equipped with actively articulated suspensions, these robots can adapt to rocks, slopes, and uneven surfaces in Mars-analog environments.

A notable contemporary example of real-world deployment is the RIVR ONE robot [19], developed by RIVR AG, a spin-off from ETH Zurich's Robotic Systems Lab where ANYmal on Wheels was developed. Although no formal research papers on it have been published yet, RIVR ONE, shown in Figure 1.2, represents a highly optimized, commercially oriented hybrid

robot designed for last-mile autonomous delivery tasks, demonstrating the growing maturity of wheeled-legged technology in practical urban environments.



Figure 1.1: MAMMOTH rover navigating Mars-analog terrain [18]



Figure 1.2: RIVR ONE robot in an urban delivery scenario [19]

1.4 Scope and Objectives

This dissertation focuses on the design, construction, control, and experimental evaluation of a lightweight and cost-effective hybrid wheeled-legged quadruped robot intended for operation on uneven and off-road terrain. The work emphasizes mechanical simplicity and practical design choices while aiming for sufficient mobility and robustness for real-world environments.

Rather than designing a platform from scratch, the work starts from an existing open-source legged quadruped, the Dingo [20], [21], and extends it into a hybrid machine. This choice concentrates the effort on the hybrid capability itself, and it is stated here because it defines the boundary between what was inherited and what was contributed. The leg kinematics, the trot gait generator, the IMU body stabilization, and the structure of the control software originate in that platform, while the wheel subsystem, the redesigned body, the electronics, and the control software built on top of them are the object of this work. Each of the following chapters identifies which of its content falls on either side of that boundary.

This dissertation aims to develop a hybrid wheeled-legged robot that is both low in cost and light in weight. The intention is to join two complementary forms of mobility, the ability of legs to cope with irregular ground while keeping the body stable, and the efficiency with which wheels cover flat surfaces. Keeping the mass low is treated as a goal in its own right, since it benefits mobility and energy consumption and makes the robot easier to put to use where conditions are difficult.

Combining the two locomotion modes should allow the robot to step over obstacles when the terrain demands it and to roll quickly and steadily when it does not. This flexibility is what makes such a platform attractive for missions in which both maneuverability and endurance matter, such as search and rescue, reconnaissance and defense, and industrial inspection. A light robot is also

simpler to carry to the site and put into operation, an advantage that becomes more important in remote or hazardous locations.

The work is deliberately bounded in two respects, autonomy and simulation. Autonomy is not addressed, since the robot carries no perception of the terrain ahead, and the choice between walking and rolling is made by the operator rather than by the robot. Simulation is likewise outside the scope, since the inherited simulation model describes the legged-only platform and updating it was not necessary to meet the objectives above.

1.5 Dissertation Structure

This dissertation is structured into nine chapters, of which Chapter 1 introduces the context and motivation for hybrid wheeled-legged robots, defines the problem addressed, and presents the scope and objectives of the work. Chapter 2 provides fundamental background on mobile robot locomotion, including wheeled, legged, and hybrid systems, as well as key principles of stability and control. Chapter 3 reviews the state of the art, presenting a taxonomy of existing hybrid platforms, lightweight and low-cost mechanical design approaches, and current control and planning strategies.

Chapter 4 presents the overall architecture of the developed robot, identifying its main subsystems and the modifications made relative to the open-source platform on which it is based. The three chapters that follow detail each engineering domain. Chapter 5 covers the mechanical design, including the leg mechanism, the 3D-printed body, and the integration of the drive wheels. Chapter 6 describes the electronics, including the wiring, power distribution, and power budget, and Chapter 7 presents the software and control architecture, from the Robot Operating System (ROS) control stack to the wheel control firmware.

Chapter 8 reports the experimental evaluation of the robot, characterizing its locomotion performance, endurance, load capacity, and cost against the goals of the work and comparable platforms. Finally, Chapter 9 summarizes the main contributions, discusses the limitations of the platform, and outlines directions for future work.

Chapter 2

Background and Fundamental Aspects

2.1 Mobile Robots and Locomotion Basics

The wheel, one of humanity's earliest and most impactful inventions, enables fast and energy efficient movement, particularly on smooth and structured surfaces. However, its efficiency and usability decrease significantly on uneven or unstructured terrain. In such environments, legged locomotion, as naturally adopted by many living organisms, provides superior adaptability and obstacle negotiation capabilities.

To leverage the advantages of both locomotion principles, hybrid wheeled-legged robots have been developed, combining the energy efficiency and speed of wheels with the terrain adaptability of legs, enabling rapid traversal on flat ground while maintaining stable mobility over rough or cluttered environments. As a result, wheeled-legged robots are well suited for applications such as exploration, inspection, and search and rescue missions.

2.1.1 Wheeled Robots

Wheeled robots exist in a variety of forms and shapes and are highly popular due to their simplicity, mechanical efficiency, and energy efficiency on smooth terrain. They are typically stable, cost-effective, and easier to control compared to legged systems. However, their performance degrades significantly on irregular or discontinuous surfaces, which limits their use in natural or unstructured environments [2].

2.1.2 Legged Robots

Advances in robotics and control have enabled legged robots to traverse rough terrain in a manner similar to humans and animals, which gives them a clear advantage over wheeled and tracked robots in complex or unstable environments. Compared with aerial systems of similar scale, legged robots can carry larger payloads and operate for longer durations, making them suitable for tasks requiring heavy sensors or manipulators [2], [22].

Among various legged forms, quadrupedal robots are particularly popular in research and industry due to their balance between structural simplicity, stability, and agility. Consequently, this work focuses primarily on the study of quadrupedal configurations.

As summarized in Table 2.1, legged robots provide enhanced traversability and obstacle negotiation in unstructured environments compared to wheeled platforms, but these benefits come at the cost of higher mechanical and control complexity, and of reduced stability on flat ground [23].

Table 2.1: Comparison between wheeled, legged, and wheeled-legged robots, the first two columns from [23] and the third added here (✓ satisfied, **X** not).

Technical Criteria	**Wheeled Robot**	**Legged Robot**	**Wheeled-Legged Robot**
Maneuverability	**X**	✓	✓
Traversability	**X**	✓	✓
Controllability	✓	**X**	✓
Terrain adaptability	**X**	✓	✓
Efficiency	**X**	✓	✓
Stability	✓	**X**	✓
Cost effective	✓	**X**	✓
Navigation over obstacles	**X**	✓	✓

Two aspects of Table 2.1 require clarification. The first is efficiency, where the source classifies wheeled platforms as inefficient and legged platforms as efficient, which refers to the effectiveness of each locomotion principle in negotiating the rough terrain that the review addresses, and not to energetic cost. On flat ground the opposite holds, since rolling is considerably more economical than stepping, as was also measured on the robot built in this work (Section 8.2.2), and it is precisely this asymmetry that motivates the hybrid designs discussed in Section 2.1.3. The second is the wheeled-legged column, which describes the potential of the concept rather than the state of existing implementations, which, as noted in Chapter 1, remain for the most part expensive, mechanically complex, and demanding to control.

2.1.3 Wheeled-Legged Robots

Hybrid wheeled-legged robots unify the strengths of both locomotion principles, operating in wheeled mode in structured environments to achieve higher speeds and improved energy efficiency, and transitioning to legged motion in unstructured terrain to overcome obstacles and irregular surfaces [16].

As reported in [9], such hybrid systems can reduce the cost of transport (CoT) by up to 83% compared to purely legged systems while maintaining comparable rough terrain adaptability. Wheeled-legged locomotion nevertheless remains an open research problem, since the absence of direct biological analogs and the need for tight coordination between wheel and leg motions impose significant challenges on motion planning, control architectures, and real-time decision making [2].

Another clear example of the energy saved by adding wheels to a legged robot is Max, whose convertible wheel-leg mechanisms make it a good platform for comparing both locomotion modes.

In wheeled mode, Max consumes around 50% less energy than the quadruped mode at the same speed, underscoring the advantages of hybrid locomotion [14]. Although such platforms introduce additional mechanical and control complexity, the resulting gains in versatility, mobility, and operational efficiency make wheeled-legged robots a compelling solution for versatile locomotion across diverse environments.

2.2 Wheeled-Legged Locomotion and Control Principles

A wheeled-legged robot combines the motion of a conventional legged robot, walking, trotting or running, with the motion of a simple wheeled robot, driving. To save energy, the locomotion logic should remain in pure driving mode for most of the time and, when necessary, combine walking to turn, bypass obstacles, or climb stairs, for example. In pure driving mode, most of these robots operate like a conventional vehicle, with the legs functioning as a suspension system to maintain a stable center of mass. However, the majority of these robots do not have steerable wheels, so turning is achieved through differential wheel speeds, leg motion, or a combination of both.

During walking mode (one foot at a time), similarly to a quadruped robot without wheels, the robot weight must be shifted away from a leg before that leg can be lifted. Under the static stability criterion, the projection of the center of mass onto the ground must remain inside the support polygon, defined as the convex hull of the contact points of the legs currently touching the ground. For a quadruped this polygon is a quadrilateral while all four legs are in contact, and reduces to a triangle when one leg is lifted, which is the limiting case that constrains a statically stable gait such as the crawl. Legs with three or more degrees of freedom (DOF) make this condition easier to satisfy, since the additional lateral freedom allows the body, and with it the center of mass, to be shifted over the remaining support polygon while the feet stay in place. This case is illustrated in Figure 2.1 [11].

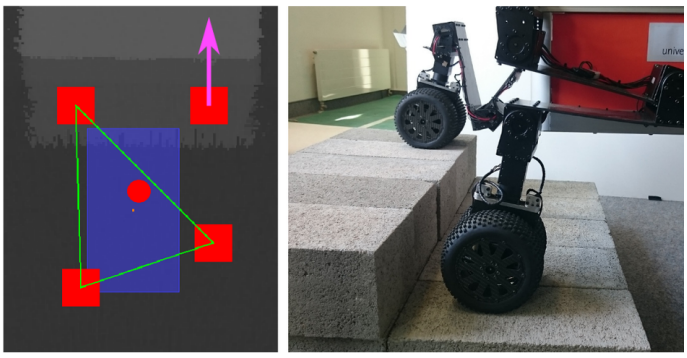


Figure 2.1: Static stability during stair climbing on Momaro. Left, the center of mass (red) lies inside the support polygon (green), so the front right leg can be lifted. Right, it stands on the next step. From [11].

Dynamic gaits such as the trot violate this criterion by design, since during each swing phase only two diagonally opposite legs are in contact, so the support polygon degenerates to a line and the projected center of mass cannot lie inside it. Stability is then maintained by the motion of the

body itself rather than by the geometry of the contacts, which is why dynamic gaits require faster control and continuous state feedback.

Chapter 3

State of the Art

3.1 Taxonomy of Hybrid Wheeled-Legged Robots

A wide range of design solutions to address hybrid wheeled-legged locomotion is available in the literature. To better understand these different solutions, Table 3.1 presents a comparative overview of representative wheeled-legged robots reported in previous works and developed commercially, ordered chronologically to highlight the evolution of design choices over time.

The comparison between the different solutions presented in Table 3.1 reveals substantial diversity in mechanical configuration, size, and performance across platforms. Most of the analyzed robots are relatively heavy, with total mass typically ranging from approximately 14 kg to 92 kg. Notable exceptions include OmniQuad [24] and the robot presented in [25], which represent some of the few lightweight hybrid platforms reported in the literature.

From a kinematic perspective, the majority of the surveyed robots employ three DOF per leg and are equipped with four wheels, reflecting a common trade-off between mechanical complexity and locomotion capability. In terms of endurance, most platforms report an operational autonomy of at least two hours, making them suitable for extended missions.

Regarding sensing capabilities, most systems integrate some form of environment perception, such as LiDAR sensors or depth cameras, enabling terrain awareness and supporting autonomous or semi-autonomous operation. The systems normally integrate some level of internal state perception like joint/wheel encoders and IMU sensors. The maximum velocities reported vary significantly between platforms, typically ranging from 0.8 m/s to 6 m/s, with the highest speeds generally reached in pure wheeled mode.

Overall, this taxonomy highlights a clear trend in existing hybrid wheeled-legged robots toward robust, high-performance platforms, often at the expense of increased weight, cost, and mechanical complexity. These observations motivate the need for alternative design approaches that prioritize lightweight construction and simplicity, while still retaining the key advantages of hybrid locomotion.

Table 3.1: Comparison of Hybrid Legged-Wheeled Robots.

Robot	**Year**	**Mass (kg)**	**Dimensions L x W x H (cm)**	**Number of Wheels**	**DOF (per leg)**	**Autonomy (hours)**	**Max Payload (kg)**	**Max Speed (m/s)**	**Main Feature**
Momaro [11]	2015	57	200x80x70	8	4 + wheels	2	-	1.1	Omnidirectional driving
Centauro [10]	2018	92	171x61.5x61	4	5 + wheels	2	60	1.6	High strength payload, impact resilience, hybrid mobility, human size compatible
ANYmal on Wheels [8], [9], [26]	2020	50	89x93x54	4	3 + wheels	-	10	4	High speed, strong rough terrain, seamless mode switching
Robot from [27]	2021	18	50.5x85x46.5	8	3 + wheels	-	-	0.8	Additional omnidirectional wheels, cost-effective
Robot from [15]	2022	28	75x69x58	4	3 + wheels	-	15	2.5	High payload, convertible
Max [14]	2024	14	60x12x-	4	3 + wheels	-	-	6.0	High speed, convertible, smooth mode switching, energy efficient
Pegasus [3]	2024	20	65x70x50.5	4	4 + wheels	3-4	10	2.13	Bio-inspired ankle, cost-effective
OmniQuad [24]	2025	7	46x54.2x49.8	4	2 + wheels	1.5	2.5	2	Lightweight hybrid design, omnidirectional motion, low complexity, low-cost
Lynx M20 [28]	2025	33	82x57x43	4	3 + wheels	2.5-3	50	5	High speed, overcome high obstacles, swappable batteries, harsh-environment use, high payload
Unitree Go2-W [12]	2025	18	70x50x43	4	3 + wheels	1.5-3	12	2.5	All-terrain
RIVR ONE [19]	2025	-	-	4	3 + wheels	3	30	3.9	High speed, prepared for urban terrains, high payload
Robot from [25]	2025	3.8	-	2	3	-	-	1.2	Low cost, lightweight, clear control architecture, terrain adaptive mode switching
GQR0440-W [13]	2025	85	45x95x55	4	3 + wheels	4-6	40	4.17	Heavy-duty platform, obstacle avoidance, high payload, multimodal locomotion
PUDU D5 [29]	2025	61	57x90x54	4	3 + wheels	2	30	5	All-terrain mobility, IP67, NVIDIA-powered, modular interaction modes, high speed

3.2 Mechanical Design and Lightweight Approaches

3.2.1 Design Considerations

Wheels on legs mechanisms, implemented in either active or passive configuration, significantly improve the mobility and energy efficiency of hybrid robots. However, these systems also increase the number of DOF, the mechanical complexity, and the hardware costs. Achieving an efficient yet lightweight structure therefore remains a major design challenge, primarily due to the additional actuators required.

3.2.1.1 Leg Configurations

Most of the studied robots follow a mammal like leg structure, which provides improved stability, a larger workspace, and enhanced obstacle avoidance capabilities. However, maintaining balance with this configuration requires high joint torques. Leg joints can be classified as knee joints, which point in the direction of motion, or elbow joints, which point opposite to the direction of motion. Different combinations of these joint types lead to distinct leg configurations robots: all-elbow, all-knee, front-knee-back-elbow, and front-elbow-back-knee, as illustrated in Figure 3.1 [30].

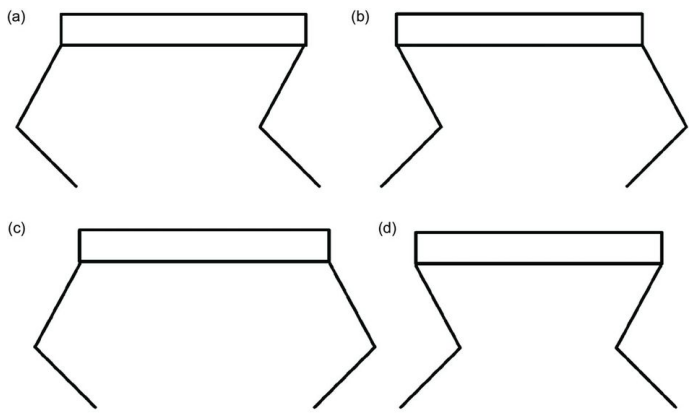


Figure 3.1: Mammal like leg structures: (a) all-elbow, (b) all-knee, (c) front-knee-back-elbow, (d) front-elbow-back-knee. From [30]

The two configurations in which the front legs have forward-facing knees (Figure 3.1b and c) tend to experience greater difficulty when traversing steep slopes or stairs [30]. Consequently, these configurations are rarely used in practice. Among the 21 legged and hybrid robots analyzed in this work, the most common configuration was the front-elbow-back-knee structure (Figure 3.1 d), adopted by 11 robots with symmetrical inward pointing legs, which helps minimize oscillations of the torso center of mass [30]. The remaining 10 robots used an all-elbow (Figure 3.1 a) configuration with parallel leg arrangements. This suggests that these two configurations are the most suitable for obstacle negotiation and locomotion in quadruped robots.

Some platforms feature legs that can be converted between configurations, such as MiniHyQ, which uses the front-elbow-back-knee configuration (Figure 3.1d) by default but can be converted into all the other leg structures [31].

3.2.1.2 Motor Placement

To ensure a structurally robust yet lightweight leg, it is best to position the main mass of the leg (the actuators) at the hip, as proximally as possible to the robot's torso. This design principle enhances stability and reduces the inertial load on distal joints. In such legs, the actuators account for the majority of the leg's mass, so concentrating all three on the proximal segment offers substantial advantages. As illustrated in Figure 3.2 [32], this arrangement ensures that the upper actuators only bear the weight of the distal links rather than the additional mass of a knee-mounted motor, thereby improving both mechanical efficiency and payload capacity. In the case of the Mini Cheetah (Figure 3.3), this minimizes limb inertia and allows a large range of motion, which supports both high-speed gait generation and the ability to operate in knee-forward, knee-backward, or even upside-down configurations [33]. However, this design choice also introduces more mechanical complexity, because the movement of one motor has to be transmitted to the lower joint whether through a linkage (Figure 3.2) or a belt (Figure 3.3).

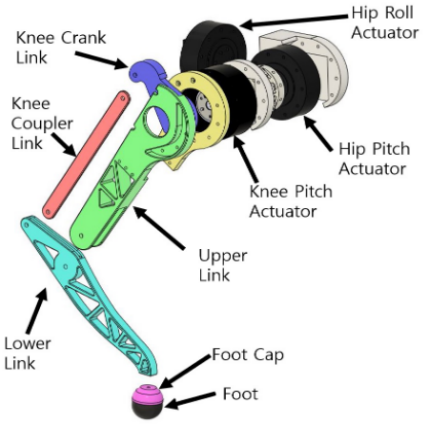


Figure 3.2: 3 DOF Leg with a parallelogram linkage. From [32]

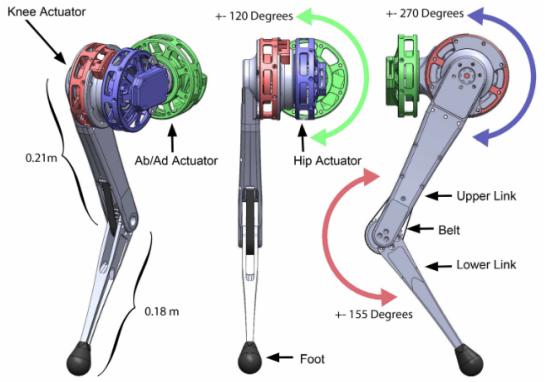


Figure 3.3: Mini Cheetah leg, with the ab/ad actuator in green, the hip in purple and the knee in red. From [33]

3.2.1.3 Wheel Placement

Hybrid wheeled-legged robots adopt a variety of wheel placement strategies to balance terrain adaptability, mechanical simplicity, and drivetrain efficiency. Some arrangements place a wheel at each foot [9], two wheels at each foot [34], wheels at the feet and heels [35], wheels at the knees [14], [15] or shoulders [36], or wheels directly on the body frame [25]. Among these, the most prevalent approach in contemporary designs involves positioning the wheels at the distal ends

of the legs, effectively replacing the traditional foot. This configuration simplifies ground interaction, provides efficient rolling on flat surfaces, and maintains stepping capability in unstructured terrain.

3.2.2 Low-Cost Solutions

Although most of the hybrid robots studied previously are neither cost-effective nor lightweight, several legged robot examples provide useful insight into how these objectives can be achieved.

Some of these robots include PADWQ (Figure 3.4) [32], Solo12 [37], Pupper [38], [39], Mini Pupper 2 [40], and Dingo (Figure 3.5) [20], [21]. All of these examples rely on 3D printing for their main structural components, which makes the designs lightweight and cost-effective. For example, the Stanford Pupper was reported at about $900 to $1,000 in its first version, weighing 1.36 kg, and at under $2,000 in its second version, weighing 2.1 kg, the difference coming from the change from servo motors to brushless motors [20], [38], [39]. Mini Pupper 2 costs around $500 and weighs less than 0.5 kg [40], while Dingo was built for $1,290.69 AUD (approximately €792.42 at the average 2023 exchange rate used in Appendix A) and weighs around 3 kg [20].

Wheeled-legged platforms in this cost and weight class are considerably rarer. The closest counterpart identified with a published price was Wheely (Figure 3.6) [41], which costs approximately $530 and weighs roughly 3 kg. This robot is a small scale platform, measuring 31 cm in height, 39 cm in length, and 44 cm in width, and exhibits low locomotion performance, with a maximum speed of 0.093 m/s and limited agility.

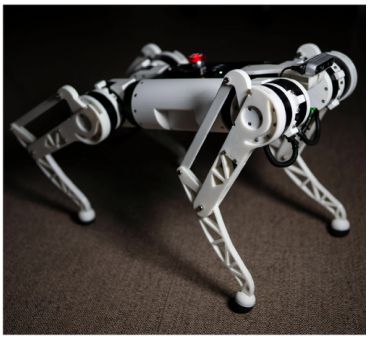


Figure 3.4: PADWQ robot [32]



Figure 3.5: Dingo robot [21]

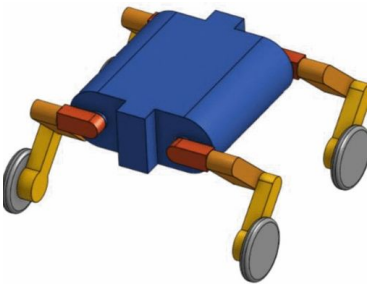


Figure 3.6: Wheely robot [41]

3.3 Control and Planning

Hybrid wheeled-legged robots pose unique challenges in motion planning and control due to nonholonomic wheel constraints, mode switching between locomotion types, and stability maintenance. Wheels impose rolling constraints that must be explicitly incorporated into planning and control [8], [42]. Transitions between wheeled, legged, and hybrid modes require careful coordination, while stability is often enforced using Zero-Moment Point (ZMP) criteria [43]. Efficient distribution of contact forces across legs and wheels is also critical for effective locomotion.

3.3.1 Model Predictive Control

Model Predictive Control (MPC) has become a dominant strategy for wheeled-legged robots due to its ability to handle multicontact dynamics, actuator constraints, and online trajectory optimization [3], [9]. MPC optimizes robot motion over a receding horizon using whole-body variables, enabling feedback based trajectory tracking and balance control.

In hybrid robots, MPC enables online gait switching and blind obstacle negotiation by incorporating rolling and nonholonomic constraints. The effectiveness of MPC has been demonstrated on legged platforms such as Mini Cheetah [33], where it produces dynamic locomotion with real-time performance, and on hybrid platforms such as ANYmal [9] and Pegasus [3], where it also provides smooth transitions between locomotion modes.

Whole-Body Control (WBC) frameworks [44] are often combined with an optimization based motion planner, allowing optimal computation of generalized accelerations and contact forces while enforcing physical constraints. On wheeled-legged robots this takes the form of a hierarchical structure, in which the planned motion is tracked by a whole-body controller that also accounts for the nonholonomic rolling constraints, separating high-level planning from low-level torque control [42].

3.3.2 Terrain Perception and Mapping

Almost every robot reviewed has some form of environmental perception, such as LiDAR or RGB-D cameras, and Momaro, for example, employs 3D laser scanning to generate height maps that support semi-autonomous stepping and weight shifting behaviors [11]. Due to the cost and power demands of LiDAR, recent low-cost platforms increasingly rely on vision-based perception, which offers sufficient environmental awareness with reduced hardware complexity [45].

3.3.3 Reinforcement Learning Approaches

Deep reinforcement learning (DRL) has emerged as a promising approach for locomotion and navigation in hybrid robots, as shown on a low-cost wheeled quadruped whose learned policies achieve agile motion with low-torque actuators and recover autonomously from falls [45]. The WaLTER robot combines an intuitive teleoperation scheme with a deep reinforcement learning policy that tracks a commanded body velocity, and demonstrates stair climbing and tumbling over broken terrain [34]. Learning based methods therefore offer robust control even with limited actuation, although both of these results were obtained in simulation.

3.4 Positioning of This Work

The reviewed literature shows that most hybrid wheeled-legged robots prioritize high-performance locomotion and robustness, often resulting in heavy, expensive, and mechanically complex platforms, while few works address lightweight and low-cost hybrid solutions.

This dissertation positions itself at the intersection of hybrid locomotion and lightweight, cost-effective robot design. By focusing on a compact quadrupedal platform with simplified actuation, accessible sensing, and computationally manageable control strategies, this work aims to bridge the gap between high-end research platforms and practical hybrid robots suitable for real world deployment in inspection, reconnaissance, or search and rescue applications.

Chapter 4

System Architecture

4.1 Overview

This chapter describes the overall architecture of the wheeled-legged robot developed in this dissertation. The system is based on the Dingo quadruped [20], [21], an open-source, low-cost four-legged robot originally conceived for research and educational purposes. The Dingo was selected as the base platform because it is well documented, ROS-integrated, well designed and almost entirely 3D-printed, making it straightforward to modify mechanically. These properties made it the most suitable starting point for extending a known working system with new capabilities rather than designing a robot from scratch.

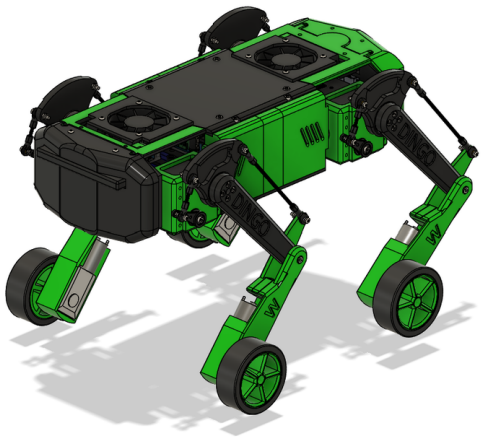
The principal contribution of this work is the adaptation of the Dingo into a hybrid locomotion robot. Each leg extremity was redesigned to carry a DC motor-driven wheel, enabling the robot to walk on unstructured terrain using its legs and to roll efficiently on flat surfaces using its wheels. This dual capability is achieved at low cost by retaining the original kinematic structure of the Dingo and adding the wheel subsystem to it. The goal is to build a 3D-printable and affordable platform that can serve as a basis for research into wheeled-legged locomotion and control.

Relative to the original Dingo design, the following modifications were introduced.

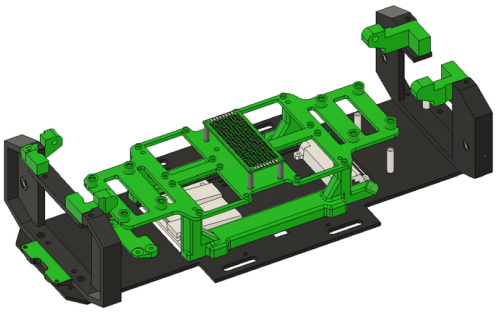
- The twelve JX CLS6336HV PWM servo motors (35 kg·cm) were replaced with Hiwonder HTD-45H bus-linked serial servos (45 kg·cm), and the PCA9685 servo driver board was replaced with a BusLinker V2.5. The higher torque rating was needed because of the weight added by the wheel subsystem.
- The Arduino Nano was replaced with a Raspberry Pi Pico W.
- The Raspberry Pi 4B was upgraded to a Raspberry Pi 5.
- Four DC wheel motors and their respective IBT-2 H-bridge drivers were added.
- The LCD display and the emergency-stop safety button were removed, simplifying the design.

- The structural acrylic plates of the original chassis were replaced with 3D-printed parts, so that almost the entire structure can be manufactured with a standard fused deposition modeling (FDM) printer.

Figure 4.1 shows the extent of these mechanical changes directly on the CAD model, with every part that was created or modified in this work highlighted. Externally, the changes concentrate in the lower legs and the wheel assemblies, together with the body panels that had to be reprinted, while internally they consist mainly of the redesigned middle support plate that carries the new electronics.



(a) External view.



(b) Internal structure.

Figure 4.1: Parts created or modified in this work (highlighted) against those inherited from the original Dingo.

4.2 System Architecture Overview

Figure 4.2 presents a high-level block diagram of the complete system, organized into three layers, namely the computation layer, the actuation layer, and the sensing layer.

The computation layer is centered on a Raspberry Pi 5, which runs Ubuntu 24.04 and hosts a Docker container running ROS Noetic on Ubuntu 20.04. The Raspberry Pi 5 is responsible for all high-level control, ROS node execution, and communication coordination. A Raspberry Pi Pico W handles low-level real-time tasks, namely generating PWM signals for the wheel motor drivers, and communicating with the main computer over a serial UART link.

The actuation layer comprises the twelve HTD-45H bus-linked servo motors for leg joint control, driven through the BusLinker V2.5 interface board, and four DC wheel motors each controlled by a dedicated IBT-2 H-bridge driver.

The sensing layer consists of the MPU-6050 IMU, which provides orientation data used by the control algorithm to maintain body stability, and wheel encoders for speed feedback.

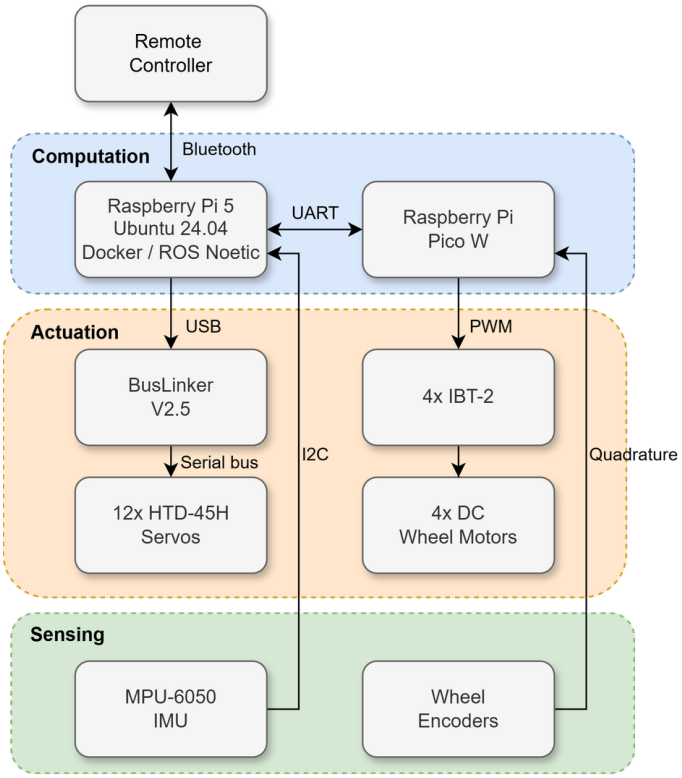


Figure 4.2: High-level system architecture block diagram showing the computation, actuation, and sensing layers.

4.3 Base Platform and Modifications

The Dingo quadruped [21] is a lightweight, open-source legged robot designed with 3D-printed structural components, twelve servo motors providing three DOF per leg, and a Raspberry Pi 4B running ROS Noetic for control. Its modular and well-documented design makes it an accessible platform for modification.

Table 4.1 details the changes introduced in this work relative to the original Dingo design. The modifications fall into three categories, component upgrades to improve performance and simplify wiring, additions to enable wheeled locomotion, and structural changes to make the robot almost fully 3D-printable and suitable for the new components.

The visual result of these changes is shown in Figure 4.3 and Figure 4.4. The body proportions and the leg linkage are the same as the original platform, while the lower leg links now terminate in driven wheels instead of fixed feet, and the acrylic plates have been replaced by printed ones.

4.4 Hardware Components

4.4.1 Raspberry Pi 5 Main Computer

The Raspberry Pi 5 serves as the central computing unit of the robot, and was chosen as an upgrade over the Raspberry Pi 4B used in the original Dingo due to its significantly improved processing

Table 4.1: Comparison between the original Dingo design and the modified platform.

Subsystem	**Original Dingo**	**This Work**
Main computer	Raspberry Pi 4B	Raspberry Pi 5
Microcontroller	Arduino Nano	Raspberry Pi Pico W
Servo motors	12× CLS6336HV (PWM)	12× HTD-45H (serial bus)
Servo driver	PCA9685	BusLinker V2.5
Wheel actuation	—	4× (DC motors + IBT-2 drivers)
DC-DC (servo rail)	ABC-H10 20 A	300 W adjustable 9A buck
DC-DC (logic rail)	Buck 7–24V to 5V 4A	DFRobot DFR1015 5V 5A
DC-DC (fans)	UBEC 5V @ 3A	(included in DFR1015)
IMU	BNO055	MPU-6050
Chassis plates	Laser-cut acrylic	3D-printed PLA
LCD display	Present	Removed
E-stop button	Present	Removed
Pi Breakout Board	Present	Removed
Power Distribution	Manufactured PCB	Perfboard
Battery	LiPo 4000mAh 4S	LiPo 2400mAh 4S
Feet	Fixed contact points	Wheels



Figure 4.3: Original Dingo quadruped, with fixed compliant feet [20], [21].



Figure 4.4: Wheeled-legged platform developed in this work.

performance, since the Pi 5 features a CPU that is two to three times faster than the Pi 4B. This additional processing headroom ensures sufficient performance for running the complete control pipeline at the 50 Hz control rate required by the gait controller.

The Raspberry Pi 5 runs Ubuntu 24.04 Server as its host operating system. Since ROS Noetic is only officially supported on Ubuntu 20.04, the entire ROS environment is deployed inside a Docker container. The container runs in privileged mode with the host's `/dev` directory mounted, providing full access to USB, UART, I^2C, and GPIO peripherals. This approach preserves compatibility with the original Dingo ROS packages without requiring a downgrade of the host operating system.

4.4.2 Raspberry Pi Pico W Microcontroller

The Raspberry Pi Pico W replaces the Arduino Nano found in the original Dingo design, and was chosen to remain within the Raspberry Pi ecosystem. In the original platform, the Nano was used for analog voltage reading and digital I/O. In this work, the Pico W runs a closed-loop speed controller for the four DC wheel motors, a function that did not exist in the original design.

The Pico W reads quadrature encoders via GPIO interrupts (two channels per motor, 14 200 counts per revolution) and computes the actual wheel speed 20 times per second. A PID controller with feedforward correction compares the measured speed against the target RPM received from the Raspberry Pi 5 over UART, and outputs the appropriate PWM signals to the IBT-2 drivers. The PWM duty cycle is capped at 80% to limit the effective voltage applied to the 12 V rated motors from the 14.8 V battery. The Pico W also reports motor status back to the Raspberry Pi 5 over the same UART link, and supports runtime tuning of PID and feedforward gains.

Offloading the real-time motor control loop to a dedicated microcontroller avoids burdening the Raspberry Pi 5, which is already occupied with ROS node execution, servo communication, IMU reading, and joystick input processing.

The UART link between the Raspberry Pi 5 and the Pico W is an improvement over the original Dingo. The original design intended a serial connection between the Raspberry Pi 4B and the Arduino Nano to carry battery voltage readings and other status data, but the original authors were unable to get this serial link working due to an unresolved issue with the Raspberry Pi's hardware serial initialization, and had to fall back to a 4-bit bit-banged digital connection carrying only an emergency-stop flag and a coarse battery level [20]. Serial communication between the two boards was explicitly left as future work. This issue was solved in this work, the Pi 5 to Pico W UART link operates reliably and carries the full closed-loop wheel control traffic described above.

4.4.3 Leg Servo Motors and BusLinker V2.5

The twelve servo motors that actuate the robot's leg joints (three per leg) are Hiwonder HTD-45H high-voltage serial bus servos, rated for 45 kg·cm of torque. They replace the JX CLS6336HV digital coreless servos (35 kg·cm) used in the original Dingo. The torque upgrade was made

necessary by the wheel subsystem, as the four DC motors, wheels, and IBT-2 drivers added to the platform increase the overall mass that the legs must support and move.

Beyond the torque increase, the change also replaces conventional PWM servos with serial bus servos. The HTD-45H communicates using a half-duplex single-wire UART protocol at 115 200 bps, and each unit provides three identical connectors, allowing all twelve motors to be daisy-chained on a single communication line. Each servo is addressed individually by its unique ID, and position commands are sent as serial packets rather than individual PWM signals. This significantly reduces wiring complexity compared to the original design, which required twelve individual PWM signal wires routed from the PCA9685 driver board. The serial bus is also bidirectional, so each servo can report its current position, input voltage, and temperature back to the controller, providing feedback that PWM servos cannot offer.

The BusLinker V2.5 interface board acts as the bridge between the Raspberry Pi 5 and the servo bus, receiving serial commands from the Raspberry Pi 5 over USB and translating them into the half-duplex bus protocol required by the HTD-45H servos.

4.4.4 Wheel Motors and IBT-2 Drivers

Four JGY-371 brushed DC geared motors (12 V, 20 RPM) are mounted at the distal end of each leg to drive the wheels. Each motor is controlled by a dedicated IBT-2 H-bridge motor driver, which supports bidirectional control and PWM speed regulation. The IBT-2 was selected for its current handling capacity, its compatibility with 3.3 V logic signals, and its low cost.

The four IBT-2 drivers receive PWM signals from the Raspberry Pi Pico W, which receives velocity commands from the Raspberry Pi 5 over a UART serial link. Since the motors are rated for 12 V but are powered directly from the 14.8 V lithium polymer (LiPo) battery, the software limits the PWM duty cycle to 80%, ensuring that the effective voltage applied to the motors does not exceed their rated value.

4.4.5 MPU-6050 Inertial Measurement Unit

An MPU-6050 6-axis IMU is mounted on the robot core to provide real-time orientation feedback. The MPU-6050 integrates a 3-axis accelerometer and a 3-axis gyroscope into a single package and communicates over I^2C. Its data is read by a dedicated ROS node and used by the control algorithm to compensate for body inclination in all four operating modes, rest, trot, crawl and wheeled, improving stability on uneven terrain and compensating for the motion of the gait.

4.4.6 3D-Printed Chassis

One of the key differences from the original Dingo is that the base and top plates, previously laser cut from acrylic, are now 3D-printed, so the entire chassis is printed. The robot therefore requires no laser-cutting or custom sheet materials, and almost its entire structure can be manufactured on a standard FDM 3D printer. The leg links, body frame, and the new wheel mounts at each leg extremity were all designed in CAD and printed in PLA, as shown in the CAD model of Figure 4.5.

This design choice reduces manufacturing cost and barrier to entry, making the platform more accessible for replication. The interior layout was also redesigned to accommodate the four IBT-2 motor drivers and the perfboard, which were not present in the original platform.

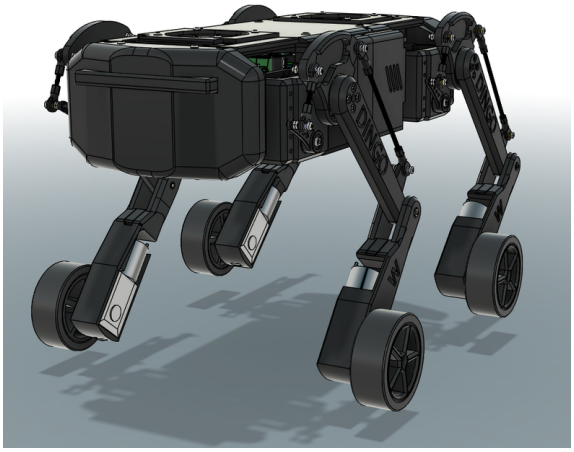


Figure 4.5: CAD model of the complete robot.

4.5 Power System

The robot is powered by a 4S LiPo battery with a capacity of 2400 mAh, a nominal voltage of 14.8 V, an energy capacity of 35.5 Wh, and a 25C continuous discharge rate. Power flows from the battery through a master switch and is then distributed across three rails.

1. On the servo rail, a 300 W-class adjustable DC-DC step-down converter regulates the battery voltage to 11.1 V, the HTD-45H's rated operating voltage (within its 9–12.6 V range), at which its 45 kg·cm torque rating applies. This rail is fed to the BusLinker V2.5 screw terminal and distributed to the twelve servos. The sizing of this converter is discussed in Chapter 6.

2. On the 5 V logic rail, a DFRobot DFR1015 multioutput DC-DC converter provides regulated 5 V power to the Raspberry Pi 5, the Raspberry Pi Pico W, and the cooling fans.

3. On the motor rail, the four IBT-2 H-bridge drivers are connected directly to the battery voltage (≈14.8 V). The 12 V rated wheel motors are protected by a software imposed limit of 80% on the PWM duty cycle, which caps the effective voltage at approximately 12 V.

Figure 4.6 illustrates the power distribution across the three rails.

4.6 Communication Architecture

Table 4.2 summarizes the communication interfaces used in the system, in which four distinct buses connect the computation layer to the actuation and sensing layers.

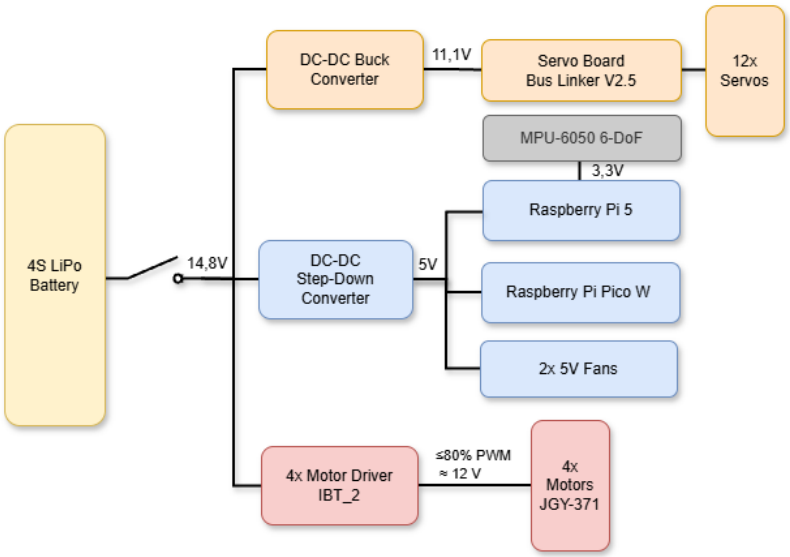


Figure 4.6: Power distribution across the motor, servo and logic rails.

The leg servos are controlled over a USB serial link from the Raspberry Pi 5 to the BusLinker V2.5, which converts the commands into the half-duplex single-wire protocol used by the HTD-45H bus. The wheel motors are controlled indirectly, as the Raspberry Pi 5 sends velocity commands to the Pico W over a UART serial link, and the Pico W generates the corresponding PWM signals for the four IBT-2 drivers. The MPU-6050 IMU is connected to the Raspberry Pi 5 via the I^2C bus, and the controller communicates with the Raspberry Pi 5 over Bluetooth.

Table 4.2: Communication interfaces summary.

Bus	**From**	**To**	**Purpose**
USB serial	RPi 5	BusLinker / HTD-45H	Leg joint commands
UART	RPi 5	Pico W / IBT-2	Wheel speed commands
I^2C	RPi 5	MPU-6050	Orientation data
Bluetooth	Controller	RPi 5	User input

4.7 Software Architecture

The software stack is built on ROS Noetic, running inside a Docker container on Ubuntu 20.04. ROS was retained from the original Dingo design because the inherited control stack was already built on it, and its topic-based communication framework leaves clear attachment points for future extensions.

The workspace structure was inherited from the original Dingo project and is organized into eight packages, summarized in Table 4.3. Five of these packages were modified or extended in

this work, covering the launch configuration, the gaits and operating modes, the PS4 input mappings, the IMU driver, and the serial bus servo and wheel interfaces. The remaining three packages (`dingo_description`, `dingo_gazebo`, and `dingo_utilities`) were carried over unchanged from the original project. A detailed description of the software developed in this work, including the control algorithms, gait generation, and kinematics, is provided in Chapter 7.

Table 4.3: ROS packages in the `dingo_ws` workspace.

Package	Responsibility	Status
`dingo`	Main launch files, configuring hardware/simulation mode, input source, and IMU usage	Modified
`dingo_control`	Inverse kinematics, gait controller, state machine, and command processing	Modified
`dingo_description`	URDF robot model and mesh files for visualization and simulation	Unchanged
`dingo_gazebo`	Gazebo simulation configuration, world files, and launch scripts	Unchanged
`dingo_input_interfacing`	Joystick and keyboard input parsing into velocity, pose, and mode commands	Modified
`dingo_peripheral_interfacing`	IMU driver, providing orientation data	Modified
`dingo_servo_interfacing`	Joint angle commands to the servos via BusLinker V2.5, and wheel commands to the Pico W	Modified
`dingo_utilities`	Utility functions (deadband filtering, first-order filters) and test scripts	Unchanged

4.8 Design Rationale

The design choices made in this work were guided by three primary criteria: low cost, ease of fabrication and accessibility, and compatibility with existing open-source software.

The Dingo was selected as the base platform because it already satisfied these criteria for the legged subsystem, and its modular ROS architecture made it practical to extend with wheel actuation. The switch from the original PWM servos to HTD-45H bus-linked serial servos was motivated by two factors, namely the higher torque rating (45 kg·cm against the original 35 kg·cm) compensates for the additional mass of the wheel subsystem, and the serial bus replaces twelve individual PWM signal wires with a single daisy-chained line, which simplifies assembly and reduces the risk of wiring errors. The replacement of the Arduino Nano with the Raspberry Pi Pico W provides more processing headroom and built-in wireless connectivity for future sensor additions. The upgrade from Raspberry Pi 4B to Raspberry Pi 5 ensures sufficient CPU performance for running the complete control pipeline, including the new wheel control interface, within the control period.

The use of Docker to deploy ROS Noetic was a practical necessity rather than a design preference. Since the Raspberry Pi 5 requires Ubuntu 24.04 as its minimum supported operating system and ROS Noetic is only compatible with Ubuntu 20.04, containerization was the most reliable way to bridge this gap without sacrificing access to the hardware peripherals.

The decision to make the full chassis 3D-printable removes the dependency on laser-cutting services, which are not always accessible in low-resource settings. This aligns with the broader goal of making the platform as reproducible and affordable as possible.

4.9 Hardware Component Summary

Table 4.4 presents the main components of the robot, while the complete bill of materials, including unit prices and total cost, is provided in Appendix A.

Table 4.4: Hardware component summary for the wheeled-legged robot.

Component	Model	Qty	Role
Main computer	Raspberry Pi 5	1	ROS execution, high-level control
Microcontroller	Raspberry Pi Pico W	1	Wheel PWM generation, serial comms
Leg servo motors	Hiwonder HTD-45H	12	Leg joint actuation (3 per leg)
Servo interface	BusLinker V2.5	1	USB-to-servo bus bridge
Wheel motors	JGY-371 (20 RPM, 12 V)	4	Wheel drive at leg extremity
Motor drivers	IBT-2 H-bridge	4	Bidirectional PWM motor control
IMU	MPU-6050	1	Orientation sensing (6-DOF)
DC-DC (servo rail)	300 W adjustable buck	1	Step-down to 11.1 V for servos
DC-DC (logic rail)	DFRobot DFR1015	1	Multioutput 5 V for Pi, Pico, fans
Battery	LiPo 4S (14.8 V, 2400 mAh)	1	Main power source (35.5 Wh, 25C)
Wheels	70 mm diameter	4	Mounted at leg extremities
3D-printed body	Custom 3D-printed (PLA)	—	Full structural body

4.10 Summary

This chapter presented the overall architecture of the wheeled-legged robot developed in this dissertation. The system is an adaptation of the open-source Dingo quadruped, extended with four wheel motors and an almost entirely 3D-printed body. The key hardware changes relative to the original design are the use of HTD-45H bus-linked serial servos driven by a BusLinker V2.5, four IBT-2-controlled wheel motors commanded by a Raspberry Pi Pico W, and a Raspberry Pi 5 running ROS Noetic inside a Docker container. Power is distributed across three rails from a 4S LiPo

battery through two DC-DC converters, and the system communicates over USB serial, UART, I^2C, and Bluetooth interfaces. The following chapters detail the mechanical design, electronics, and software implementation in turn.

Chapter 5

Mechanical Design

5.1 Leg Mechanism

The leg mechanism was retained from the original Dingo design [20], since its kinematic structure had already been validated and the goal of this work was to extend the platform with wheeled locomotion rather than redesign the legs. Each of the four legs provides three DOF. Hip abduction/adduction tilts the leg plane sideways relative to the body, hip flexion/extension swings the leg forwards and backwards, and knee flexion/extension extends or retracts the foot.

Each leg is composed of a hip assembly and two links, in which the hip abduction axis is offset from the leg plane by 51.6 mm, the upper leg link measures 130 mm between the hip and knee joints, and the lower leg link now measures 121 mm from the knee joint to the wheel axle. The four hip assemblies are mounted 111.7 mm in front of and behind the body center and 61 mm to each side of the body center line.

The lower link length deserves a note, because the wheel conversion changed it. The original design measured this link to the tip of its compliant foot, giving 138.1 mm. A wheel is not a longer link, since its contact patch sits directly below the axle at any leg angle, whereas a rigid link extends along the leg's own direction. The correct kinematic length is therefore measured to the wheel axle, with the wheel radius entering separately as a constant vertical offset between axle and ground, as shown in Figure 5.1.

A key feature of the design is that the knee joint is not actuated directly. Instead, the knee servo is mounted at the hip, alongside the other two servos, and its motion is transmitted to the lower leg through a linkage mechanism, shown schematically in Figure 5.1. This arrangement concentrates the mass of all three actuators at the hip, keeping the moving portion of the leg lighter and reducing the inertia that the servos must overcome during the swing phase of the gait, a benefit the wheel conversion partly gives up, since each lower leg now carries a motor and a wheel at its tip (Section 8.2.1). The trade-off is that the servo angle and the knee angle it produces are not the same quantity, so the control software must convert between them. Establishing the correct form of that conversion turned out to be a substantial part of this work.

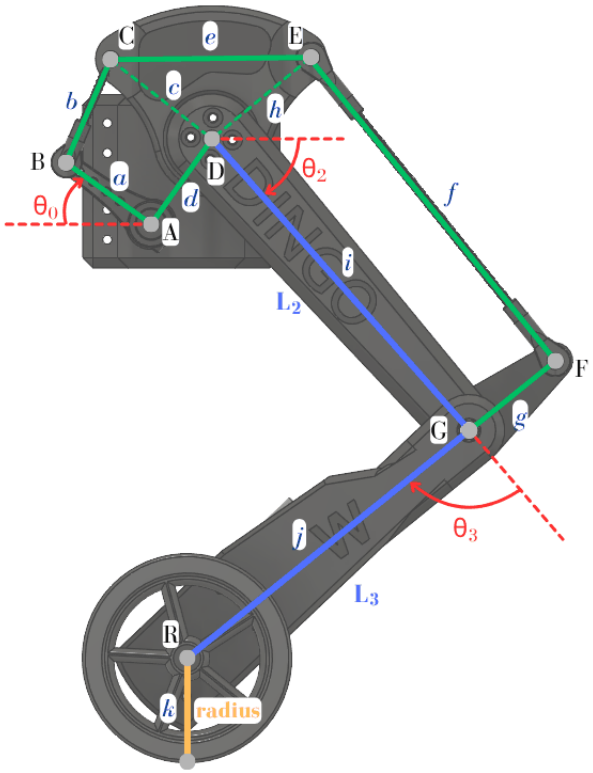


Figure 5.1: Line diagram of the leg mechanism over the CAD model. In green the linkage members a to h, in blue the leg links L_2 and L_3, and in orange the wheel radius below the axle where L_3 ends. Redrawn from [20].

The original design describes this mechanism as two four-bar linkages, $ABCD$ and $DEFG$, sharing the triangular substructure CDE, with joint A as the servo output axis and joint D as the hip pivot. In that model the servo rotation propagates through the first four-bar, then the triangle, then the second four-bar to joint G, the knee pivot, and recovering the knee angle from the servo angle requires Freudenstein's method applied to each four-bar in turn [20]. The member dimensions of that model are listed in Table 5.1.

Table 5.1: Leg dimensions, keyed to Figure 5.1. Above the rule, the linkage members in the original design [20] and on this robot, of which a, b, f and j differ.

Parameter	Segment	Original (mm)	This work (mm)
a	A–B	35.12	25.12
b	B–C	37.60	43.00
c	C–D	43.00	43.00
d	A–D	35.23	35.23
e	C–E	67.10	67.10
f	E–F	130.00	144.00
g	F–G	37.00	37.00
h	D–E	43.00	43.00
i / L_2	D–G (upper leg)	130.00	130.00
j / L_3	G–R (lower leg, to wheel axle)	138.10	121.00
k	R–contact patch (wheel radius)	—	35.00

This formulation worked on the original robot, but it did not describe the one built here. Commanded foot positions were persistently not reached, and repeated recalibration never corrected the error, which pointed at the conversion itself rather than at the calibration. The relationship was therefore measured directly, by holding the upper joint fixed, stepping the lower servo across its range, and recording where the wheel axle actually went at each step. The measured relationship differs from the predicted one proportionally rather than by a constant, since the inherited conversion calls for substantially too little servo movement for a given change in joint angle. No offset adjustment can absorb an error of that kind, which is why recalibration had never helped.

Measuring the linkage members pivot to pivot showed that three of them differ from the dimensions specified by the original design, a from 35.12 to 25.12 mm, b from 37.60 to 43.00 mm, and f from 130.00 to 144.00 mm (Table 5.1). These are the servo horn (a) and the two metal links (b and f), the only members of the linkage that are not 3D-printed, so their dimensions depend on the parts sourced rather than on the printed geometry. Substituting the measured lengths into the inherited routine did not, however, make it usable, because over roughly half of the leg's working range the loop it attempts to close has no solution, and the inverse cosine at the base of the routine then returns an undefined value that propagates as far as the servo command.

The difficulty lies in the direction in which the linkage is solved rather than in its dimensions. The inherited routine poses the problem backwards, taking the knee angle as given and treating the upper leg as a crank turning about a fixed frame, and with the measured lengths that question has no answer over much of the range. In reality the mechanism is driven from the servo, whose

horn sets the position of joint B, while the hip flexion joint sets that of the knee pivot G. Every remaining joint then sits at a fixed distance from two joints that are already known, which places it at the intersection of two circles. Joint C lies at distance b from B and c from D, closing the first four-bar, the rigid triangle CDE carries it into joint E, and joint F lies at distance f from E and g from G, closing the second. The knee angle follows from the direction of the segment G to F. Built this way, the linkage as measured describes the leg correctly, and the mismatched parts did not have to be replaced.

The conversion adopted in this work is therefore the forward solution of the linkage as built,

$$\theta_3 = \mathscr{F}(\theta_0, \theta_2), \tag{5.1}$$

where θ_0 is the servo angle at joint A, θ_2 the hip flexion angle, θ_3 the knee angle, and $\mathscr{F}$ the geometric closure of the two loops described above. The full construction is given in Appendix B.

One structural property of this arrangement was characterized only later, during the load tests of Section 8.4. Carrying weight in the crouched driving pose, the legs began to splay outward at approximately 5 kg, the upper and lower links opening away from the body while the hip abduction servos went on holding their commanded angles. The limit is therefore the stiffness with which the leg joints are constrained, rather than the strength of the links or the torque available at the joints, and constraining those joints more firmly is the change that would raise the platform's carrying capacity beyond the figure reported there.

5.1.1 Servo Torque Requirements

The original Dingo report sized its servos analytically, computing the joint torques required to support the robot's weight in the worst case of only two legs in contact with the ground (as occurs mid-trot), with all legs at their most extreme angles [20]. Assuming a total mass of 3 kg, this analysis found a maximum required torque of 32.92 kg·cm at the abduction joint, against which a 35 kg·cm servo was selected, a margin of approximately 6%.

The robot developed in this work has a measured mass of 3.54 kg, an increase of 18% over the original 3 kg assumption, due to the four wheel motors, wheels, and IBT-2 drivers added to each leg and to the chassis. Scaling the original worst-case torque requirement linearly with total mass gives an estimated requirement of

$$\tau_{\text{req}} \approx 32.92\ \text{kg}\cdot\text{cm} \times \frac{3.540\ \text{kg}}{3.0\ \text{kg}} \approx 38.8\ \text{kg}\cdot\text{cm}, \tag{5.2}$$

which already exceeds the 35 kg·cm rating of the original servos. The 45 kg·cm HTD-45H servos selected for this work restore a comfortable margin of approximately 16% over this estimated requirement, compared to the original design's 6%.

The need for more torque was also seen in practice, since before settling on the HTD-45H, a Hiwonder LX-225 bus servo (25 kg·cm), already on hand from earlier leg development testing, was tried on the robot. The LX-225 was able to support the robot's weight while the legs remained

static, including while driving on the wheels, but could not sustain the joint torques required for walking, and when the legs were commanded to move, the servos were unable to hold the required angles and the legs buckled under the robot's weight. Because the LX-225 is rated well below even the original 35 kg·cm servos, this test does not measure the requirement directly, but it is consistent with the estimate above and motivated the move to the higher-torque HTD-45H. Beyond the torque increase, the HTD-45H also draws a lower stall current than the LX-225, 3 A against 4 A, reducing the peak current demand on the servo power rail despite the higher torque rating.

5.2 Body Structure

5.2.1 From Laser-Cut Acrylic to a Printed Chassis

In the original Dingo, the two main structural plates of the torso (base and top plates) were laser cut from 3 mm acrylic sheet, while the remaining body parts were 3D-printed [20]. In this work, the entire chassis is 3D-printed in PLA. The change was primarily motivated by manufacturing access, since a 3D printer was available while a laser cutter was not, and printing the plates as well makes the robot easier and cheaper to replicate, since a single manufacturing process then covers almost the whole structure. It should be noted that acrylic sheet likely offers higher stiffness than a printed plate of the same thickness, so this change trades some structural strength for accessibility.

To compensate, the plates were printed with reinforced settings, keeping the original CAD geometry, including the 3 mm plate thickness, but using four perimeter walls and 15% infill, increasing their strength relative to default print settings. In practice the printed chassis has held up well, and no chassis part has cracked or required reprinting during development and testing. The heat-set threaded inserts used in the printed parts, a point of concern raised in the original report regarding long-term loosening in printed materials [20], have also shown no problems.

The two large plates exceed the build area of common printers, so they were printed diagonally on a 300×300 mm print bed, with only small corner sections split off and glued in place afterwards. All remaining body parts fit a standard 220×220 mm bed, keeping the design accessible to typical desktop printers.

5.2.2 Interior Redesign

The interior of the torso was reorganized to make room for the components added in this work. Several nonessential components of the original design were removed, namely the LCD display, the emergency-stop button, the Raspberry Pi breakout board, and the custom power-distribution PCB, which occupied most of the central volume of the body. Their removal freed the space needed for the wheel motor drivers and the new power converters.

The interior is organized in three levels, shown in Figure 5.2, the base plate, a middle support plate, and the top plate. The base plate carries the Raspberry Pi 5, in the same location as in the original design, and the servo rail DC-DC converter. The Raspberry Pi is now mounted with its ports facing the inside of the body, since the BusLinker connects over USB from within the

robot and a mobile platform rarely needs external wired connections, inward-facing ports keep the cabling internal, and any future sensor can still be connected from the inside. The middle support plate was redesigned to hold most of the electronics, namely the four IBT-2 drivers on top, the Raspberry Pi Pico W on its expansion board, the BusLinker, the 5 V converter, and the IMU, which remains in its original position.

The battery remains centrally placed, retaining the original's access flap mechanism for quick removal. However, the battery used in this work is smaller than the original (2400 mAh against 4000 mAh), a choice enabled by the expectation of better energy efficiency in wheeled locomotion. The smaller battery allowed the middle plate to be lowered, providing more room for the components and cabling above it.

The two 60 mm cooling fans remain in their original positions and provide active cooling for the 12 servos, the Raspberry Pi 5 and the power converters. The rear panel was reprinted as a flat part, with the cutouts for the LCD and the Raspberry Pi ports removed, the emergency-stop opening covered, and only the power button remaining. All of the original design's external mounting holes were kept, preserving the platform's modularity for future sensor additions.

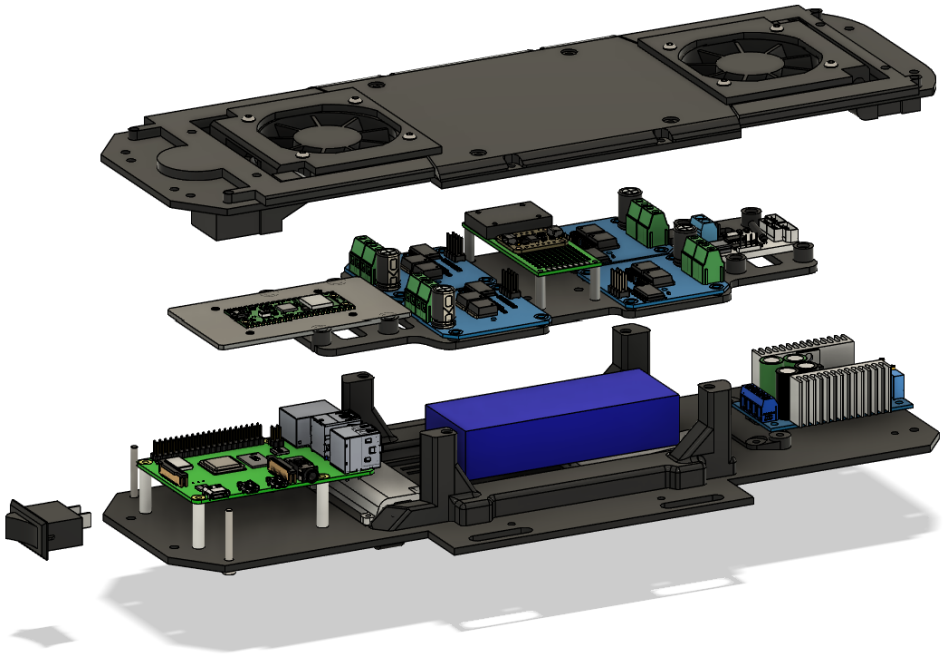


Figure 5.2: Exploded view of the three-level interior, with the Raspberry Pi 5, battery and servo converter on the base plate, the Pico W, IBT-2 drivers, BusLinker and perfboard on the middle plate, and the fans on top.

5.3 Wheel Integration

The integration of the drive wheels is the main mechanical contribution of this work, for which the lower leg link of the original Dingo was redesigned so that a JGY-371 geared motor can be bolted

directly onto it, with the wheel mounted on the outer side of the leg. The four wheels are oriented like the wheels of a car, with their rotation axes parallel to the lateral axis of the body, so the robot drives forwards and backwards in wheeled mode and steers by imposing a speed differential between the left-side and right-side wheels.

The wheel itself consists of a custom-designed 3D-printed rim, shown in Figure 5.3, fitted with a standard rubber tire and giving a total diameter of 70 mm. The rim is coupled to the motor in two ways, with a screw threaded axially into the end of the motor shaft to hold the rim against it, and a second screw entering radially through the rim to press against the flat of the motor's D-shaped shaft, locking the rim rotationally. This double fixation resulted from a design iteration, since the first version of the rim relied only on the axial screw, which did not prevent the rim from slipping on the shaft under load, and the second version added the radial screw to take advantage of the D-shaft profile.



Figure 5.3: Final design of the 3D-printed wheel rim, with the D-shaped bore, the boss for the locking screw, and the channel that retains the tire.

The wheel was positioned so that its contact point with the ground falls close to where the original foot met the ground, as compared in Figure 5.4. Keeping ground contact in approximately the same place means that the stance geometry and the gait parameters inherited from the original platform remain usable, rather than having to be rederived for a leg of different effective length. The kinematic model itself did change, as described above, since the lower link now terminates at the wheel axle, with the wheel radius entering as a separate vertical offset, so the distance from the knee to the ground contact varies slightly with leg pose rather than being a fixed link length. From the perspective of the gait controller, however, the wheel is simply the new foot.

This is possible because the JGY-371 uses a worm gearbox with a 1:380 reduction ratio, which makes the drive self-locking, meaning that the output shaft cannot be back-driven by external loads when the motor is unpowered. The wheels therefore lock in place without consuming any current and behave as static feet during legged locomotion, making them the only part of the robot that touches the ground in both locomotion modes.

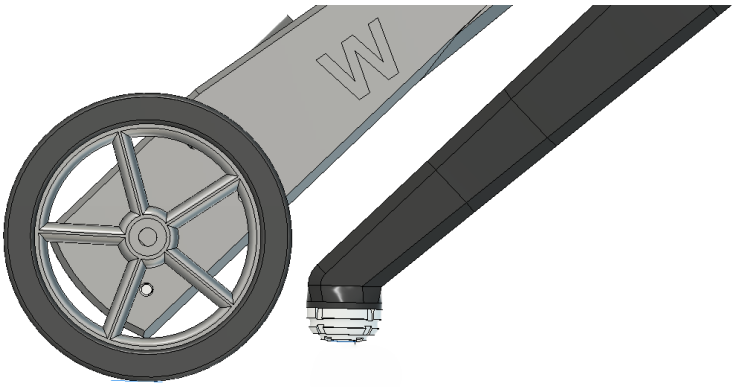
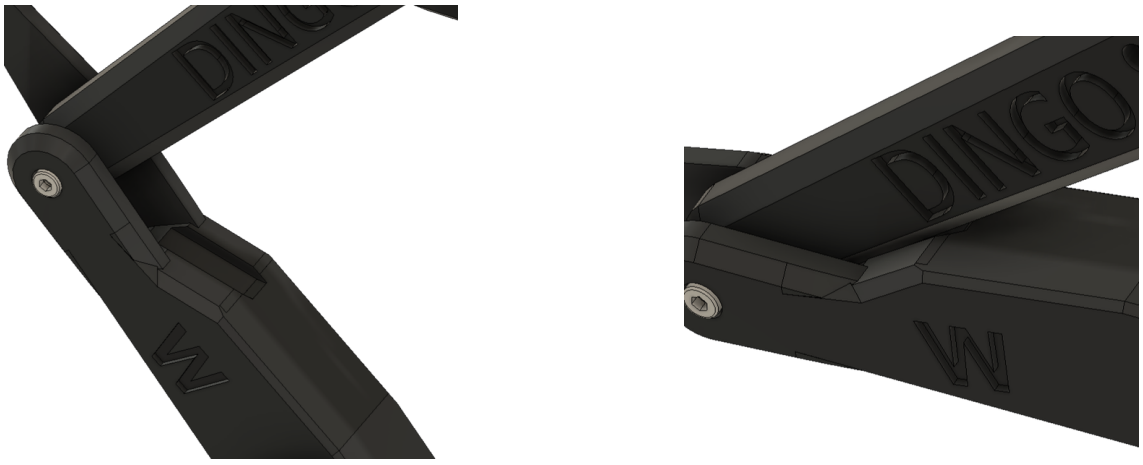


Figure 5.4: The redesigned lower leg and wheel (left) against the original lower leg and foot (right), which meet the ground in nearly the same place.

Beyond the motor mounting, the geometry of the lower leg itself was iterated in two ways. First, material was removed near the foot end of the link, allowing the knee joint to reach a larger flexion angle before the printed part touches the ground. Second, a cutout was added at the point where the lower leg previously pressed against the upper leg, so that when the two links reach their minimum relative angle, the cut engages and the links lock against each other in a defined position, as shown in Figure 5.5. This interlock was designed specifically for the crouched driving pose used in wheeled mode, where the legs are fully flexed. Instead of the joint resting against an arbitrary contact between the two parts, the links seat into the cut, giving the crouch a repeatable geometry and additional stability. It is this designed lock, combined with gravity, that makes it safe to release the leg servos torque in the low-power driving mode described in Chapter 7.



(a) Disengaged, at an intermediate knee angle.

(b) Engaged, at the minimum knee angle.

Figure 5.5: The interlocking profile between the leg links, which seats as the knee closes and gives the crouched pose a repeatable geometry.

This design does, however, trade away a property of the original foot. The original Dingo's feet were made from a flexible filament (TPU), a choice motivated by the fact that servo motors, unlike back-drivable brushless DC motors, cannot provide passive compliance at the joint level [20], and the soft foot material was used to compensate, absorbing shock and improving ground contact. The wheel assembly used in this work is rigid, with no equivalent compliant element, since the rubber tire only provides a small amount of local cushioning.

Figure 5.6 shows the assembled leg, with the motor bolted to the outer face of the redesigned lower leg link, the printed rim and its rubber tire mounted on the output shaft.



Figure 5.6: Redesigned lower leg with the JGY-371 motor and the printed rim and tire.

5.4 Manufacturing and Assembly

All parts were produced on consumer-grade FDM printers, with the base and top plates, which exceed a standard build volume, printed on a Creality Ender 3 Max Neo (300×300 mm bed), and every other part printed on a Creality Ender 3 V2 Neo with a standard 220×220 mm bed. As described in Section 5.2, the structural plates were printed with four perimeter walls and 15% infill, while the remaining parts, which carry lighter loads, used two to three walls and 8–10% infill. The complete robot consumed 1278 g of PLA filament (428.5 m), at a material cost of roughly €25, with a total printing time of about 143 hours. This reinforces the accessibility goal of the platform, since almost the entire structure can be manufactured on entry-level printers for the cost of little more than a single 1 kg spool of filament.

The original design specifies ABS for every part in contact with the servos, since PLA may soften under the heat the motors produce [46]. In this work every printed part was made in PLA, and no softening or deformation was observed during testing. Printing all of these parts in a

single, easily printed material keeps it within reach of entry-level printers, without the enclosed build chamber that ABS usually requires, and PLA is usually also slightly cheaper than ABS.

The nonprinted mechanical hardware of the leg linkages (bearings, metal pushrods, rod ends, and servo horns) was purchased as specified in the original Dingo design [20]. Threaded connections in the printed parts use heat-set brass inserts, installed with a soldering iron, and the assembled robot contains approximately 35 to 40 inserts.

The most delicate step of the assembly is the installation and calibration of the servos. Since the twelve servos share a single communication bus, each one must first be assigned a unique ID (0 to 11) before being mounted. After mounting, the servos were calibrated following the procedure of the original project, in which a Python script drives each servo to its nominal zero position, the angular offset between the commanded and actual position of each joint is measured, and the offsets are stored and applied in software, correcting the joint zeros virtually rather than by mechanical adjustment of the servo horns.

Figure 5.7 shows the assembled robot from above with the top plate in place. The printed structure, the fan supports on their flap mechanism, and the top view of the four leg assemblies with their servos.

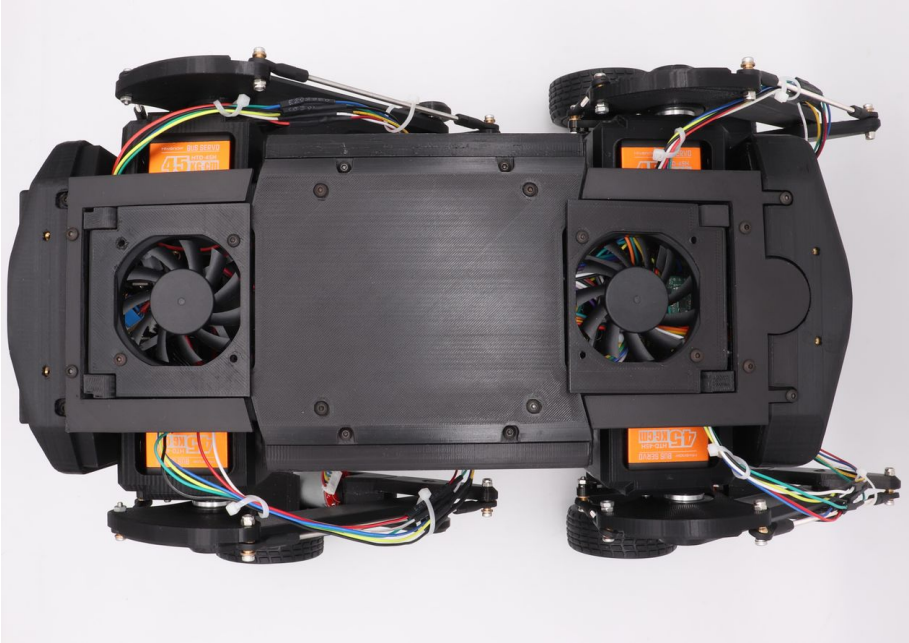


Figure 5.7: Assembled robot viewed from above.

5.5 Printed Parts

Table 5.2 lists the 3D-printed parts of the robot, grouped by assembly. Parts marked as modified were redesigned in this work relative to the original Dingo, while parts marked as new did not exist in the original design. In total, printing all parts required 5 days and 23 hours of machine time and 1278 g of PLA.

Table 5.2: 3D-printed parts of the robot.

Assembly	**Part**	**Qty**	**Status**
Torso	Base plate	1	Original
	Top plate	1	Original
	Top cover	1	Original
	Middle support plate	1	Modified
	Middle platform support	2	Modified
	Front plate	1	Original
	Rear panel	1	Modified
	Side panels	2	Modified
	BusLinker support	1	New
	Fan mounts	2	Original
	Battery flap base	1	Original
	Battery latch top cover	1	Original
	Battery flap rod support	1	Original
	Battery holder hinge rod	1	Original
Legs	Hip servo assembly	4	Modified
(mirrored pairs)	Motor hub	4	Modified
	Upper leg assembly	4	Original
	Lower leg	4	Modified
	Upper leg bearing washer	4	Original
Wheels	Wheel rim	4	New

Chapter 6

Electronics

6.1 Electronic Components

6.1.1 Raspberry Pi 5

The Raspberry Pi 5 (4 GB RAM variant) is the main compute platform, running Ubuntu 24.04 and hosting the Docker container that runs ROS. The board officially requires a stable 5 V supply capable of delivering up to 5 A, and compared to earlier Raspberry Pi models, the Pi 5 is known to be more sensitive to voltage drop and noise on its power input, particularly when powered through its GPIO pins rather than its USB-C port. This characteristic directly shaped the power wiring approach described in Section 6.2.

6.1.2 Raspberry Pi Pico W and I/O Shield

The Raspberry Pi Pico W runs the closed-loop wheel control firmware described in Chapter 7. It is mounted on a solderless I/O expansion shield, which breaks out the Pico's pins to male and female pin headers and exposes two I^2C, two UART, and two SPI ports, three analog inputs, and thirteen digital I/O pins. The shield's headers and mounting holes considerably simplify wiring the four IBT-2 drivers and the four motor encoders to the Pico, compared to soldering directly to its pins.

6.1.3 BusLinker V2.5

The BusLinker V2.5 bridges the Raspberry Pi 5 to the twelve-servo bus, translating USB serial commands into the half-duplex bus protocol used by the HTD-45H servos. The board also exposes a TTL UART header as an alternative to USB, but this header operates at 5 V logic, while the Raspberry Pi 5's GPIO is 3.3 V only, so USB was used instead to avoid the need for a level-shifting circuit between the two boards.

6.1.4 IBT-2 Motor Drivers

Each of the four wheel motors is driven by a dedicated IBT-2 H-bridge module, which exposes two PWM control inputs (RPWM and LPWM, one per rotation direction) and two enable inputs

(R_EN and L_EN), and accepts 3.3–5 V logic signals, compatible with the Pico's 3.3 V GPIO. In this design, R_EN and L_EN are wired permanently high, so only the two PWM inputs are actively driven by the Pico, reducing the number of microcontroller pins required per driver from four to two.

6.1.5 Inertial Measurement Unit

The IMU is an MPU-6050 breakout board, which provides a 3-axis accelerometer and a 3-axis gyroscope over an I^2C interface to the Raspberry Pi 5.

6.1.6 DC-DC Converters

Two DC-DC converters distribute power from the battery, replacing the three separate converters (a 20 A servo buck, a 3 A fan buck, and a 4 A logic buck) used in the original design. The servo rail is supplied by a 300 W-class adjustable buck converter (7–40 V input, 1.2–35 V output), set to 11.1 V, the HTD-45H's rated operating voltage, at which its 45 kg·cm torque specification applies. This converter replaced the VHM-101 module (5 A, 180 W) used initially, whose current limiting proved inadequate for the servo bus under load, and the diagnosis that led to this replacement is described in Section 6.3.2. A DFRobot DFR1015 multioutput converter (3.3 V/5 V/9 V/12 V, 5 A) supplies a single 5 V rail that in this design powers the Raspberry Pi 5, the Pico W, and both cooling fans, consolidating what were three separate rails in the original design into two.

Figure 6.1 shows the main electronic components described in this section.

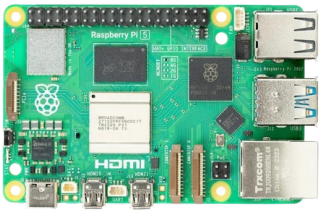
6.2 Wiring and Connections

6.2.1 Power Distribution

Power distribution is built around a custom perfboard, which replaces the custom PCB of the original Dingo [20]. The battery connects through an XT60 connector, and its positive wire passes through the main on/off switch on the rear panel, which cuts power to the entire robot, while the ground lead runs directly to the perfboard. Both leads are soldered to the perfboard, forming the main battery bus (14.8 V nominal).

The battery bus feeds three loads, namely the four IBT-2 motor drivers, connected directly so that the wheel motors receive the full battery voltage, the servo rail buck converter, mounted at the front of the base plate, and the DFR1015 5 V converter, mounted on the perfboard itself. The output of the DFR1015 forms a 5 V bus on the perfboard, which powers the Raspberry Pi 5, the Pico W (through the power input of its I/O shield), and the two cooling fans, which run whenever the robot is switched on.

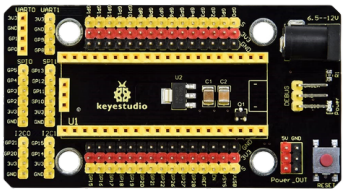
The Raspberry Pi 5 is powered through its GPIO header rather than its USB-C port, and since the Pi 5 is sensitive to voltage drop when powered this way, its power wiring uses 2 mm diameter wire, and the current is split across two wires feeding the two 5 V GPIO pins, with a third wire to ground. With this arrangement the Pi 5 operates reliably without undervoltage warnings.



(a) Raspberry Pi 5



(b) Raspberry Pi Pico W



(c) Pico I/O shield



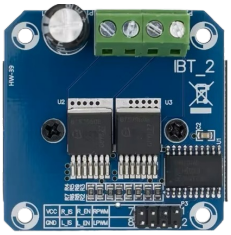
(d) Hiwonder HTD-45H



(e) BusLinker V2.5



(f) JGY-371 wheel motor and built-in encoder



(g) IBT-2 motor driver



(h) MPU-6050 IMU



(i) Servo rail buck (300 W)



(j) DFRobot DFR1015



(k) LiPo 4S battery

Figure 6.1: Main electronic components of the robot.

The servo rail runs from the servo buck converter, set to 11.1 V, to the screw terminal of the BusLinker. The BusLinker provides four servo ports, but only two are compatible with the HTD-45H connectors, so the twelve servos are wired as two daisy chains of six, one per side of the body. This split also keeps each chain shorter than a single twelve-servo chain.

It should be noted that the power path contains no fuse or reverse-polarity protection, whereas the original design included an inline fuse between the battery and the logic converter. Protection circuitry was omitted for simplicity and is identified as a point to improve in the future.

6.2.2 Signal Wiring

All control signals operate at 3.3 V logic. The Raspberry Pi 5 communicates with the BusLinker over USB, and with the Pico W over UART (the Pi's UART2, exposed on GPIO4/GPIO5 as `/dev/ttyAMA2`), with the Pi's GPIO4 (TX) connected to the Pico's GP9 (RX) and the Pi's GPIO5 (RX) to the Pico's GP8 (TX). Since both devices use 3.3 V logic, no level shifter is required. The MPU-6050 IMU, physically mounted on the perfboard near the center of the torso, connects to the Pi's I^2C pins (GPIO2/3) and is powered from the Pi's 3.3 V rail, both over jumper wires.

The Pico W drives each IBT-2 with two PWM signals (RPWM and LPWM), while the enable inputs are tied permanently high and the logic side of each driver is powered with 3.3 V from the Pico shield. In the opposite direction, each motor's quadrature encoder feeds its two channels to the Pico's GPIO pins, with the Hall sensors also powered at 3.3 V. On the motor side these connections use the original JST connectors and short cables supplied with the motors, which were soldered to Dupont jumper wires so they could reach the Pico.

In summary, power connections use 2 mm diameter wire throughout (except the low-current Pico and IMU feeds), signal connections use Dupont jumper wires plugged onto the pin headers of the Pico I/O shield, and JST connectors appear only at the motor/encoder side. Figure 6.2 shows the complete wiring of the robot.

6.3 Power Budget

6.3.1 Component Consumption and Measured System Power

Table 6.1 summarizes the electrical consumption of the main components, from their datasheets. The stall figures represent absolute worst-case ceilings that do not occur simultaneously in practice, since the gait never stalls all twelve servos at once, and the 80% PWM cap combined with closed-loop speed control keeps the wheel motors far from stall in normal driving.

The system level power consumption was measured directly on the robot, by recharging the battery after each session and recording the charge restored by the charger, cross checked against the resting pack voltage (the two measures always agreed). The measurements reported in this section were taken at an earlier stage of the build, with the LX-225 development servos (25 kg·cm) on a 7.8 V rail, before the final HTD-45H servos were fitted. The two campaigns below differ only in which converter feeds the servo rail, which makes them the cleanest evidence available

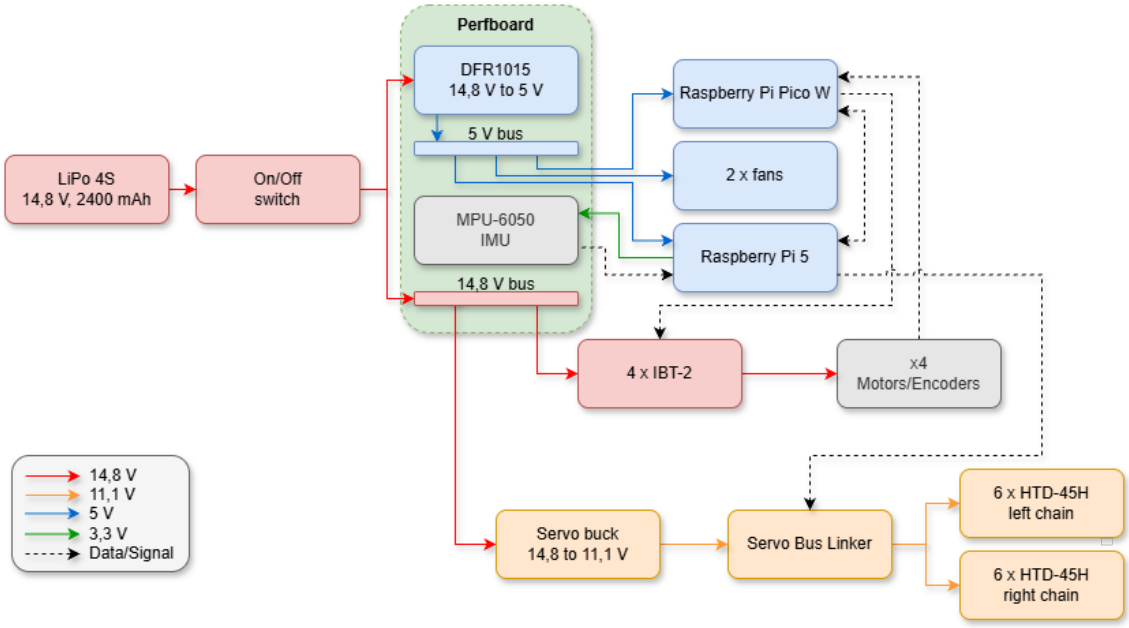


Figure 6.2: Complete wiring of the robot. Solid arrows carry power, colored by rail voltage, and dashed arrows carry data.

Table 6.1: Datasheet consumption of the main components.

Component	Rail	Typical	Stall / max
12× HTD-45H servo	11.1 V	0.1 A each (no load)	3 A each
4× JGY-371 motor	14.8 V	0.17 A each (rated)	2.1 A each
Raspberry Pi 5	5 V	3–8 W (est.)	25 W (5 A)
Raspberry Pi Pico W	5 V	<0.5 W	—
Fans, IMU, BusLinker	5 V / 3.3 V	<1.5 W combined	—

for how the rail affects consumption. Measurements of the robot in its final configuration, and the performance conclusions drawn from them, are reported in Chapter 8.

Two measurement campaigns were carried out, separated by the replacement of the servo rail converter described in Section 6.3.2. Campaign A used the original VHM-101 converter and Campaign B the replacement 300 W buck, with the servos and rail voltage otherwise unchanged, isolating the effect of the converter alone. Table 6.2 presents both.

Table 6.2: Measured system power before and after the servo rail converter replacement, with the servos (LX-225) and the rail voltage (7.8 V) unchanged. Endurance is to 80% of the battery.

	A: VHM-101			**B: 300 W buck**		
Condition	I (A)	P (W)	**Endur.**	I (A)	P (W)	**Endur.**
Idle, servos off	0.40	6.6	4.8 h	0.40	6.6	4.8 h
Standing	1.48	24.0	78 min	0.76	12.5	150 min
Driving, full speed	1.67	26.0	69 min	1.49	24.2	77 min
Driving, low-power	1.11	18.0	103 min	1.03	16.8	112 min

Three conclusions follow. First, consumption is dominated by the servos and not by the wheel motors, which together draw only 2–3 W, so most of the power drawn whenever the servos are powered goes to holding and correcting the legs, whether the robot is standing or driving. Second, replacing the servo rail converter alone, with no other change, roughly halved the standing current (1.48 A to 0.76 A). Since the servos and rail voltage were identical between campaigns, this drop is attributable entirely to the converter, as the VHM-101's current limiting caused the servos to hunt around their holding position, wasting power correcting for a rail that could not quite supply what they asked for. Third, the low-power crouch mode saves about 31% of driving power in both campaigns, confirming that the saving comes from the mode itself and not from any property of the converter.

Component temperatures were measured in the same configuration, after a 30 minute driving session at an ambient temperature of 28–30 °C, the servos reached 33 °C, the Raspberry Pi 5 34 °C, and the wheel motors 35 °C. None of the components showed thermal stress, which established that the limit on servo performance under load was torque rather than heat, and that the cooling arrangement described in Section 6.4 is adequate.

6.3.2 Servo Rail Sizing

The servo rail proved to be the critical element of the power system, and the converter was identified as the cause of the servo bus disconnections by substituting a bench power supply for it, feeding the servo bus directly at 7.8 V with the Raspberry Pi powered separately. On that supply the bus drew 1.5 to 3 A continuously, but when trotting the peak drawn at the onset of each movement could not be measured, because the bench supply was itself limited to 5 A and saturated at that value, so the demand is known only to exceed 5 A, with an absolute worst case of twelve servos at their 4 A stall current. The VHM-101 converter initially used for this rail (5 A, 180 W, with constant current limiting) sits at the bottom of that range, and its current limiting caused the

rail voltage to collapse at exactly the moments of highest demand. Beyond stopping the servos, the voltage sag caused the BusLinker to drop its USB connection to the Raspberry Pi 5, interrupting all servo control. Automatic reconnection of the serial port was added in software, which recovers control in approximately one second, but the disconnections still occurred occasionally since the root cause was electrical.

The VHM-101 was therefore replaced with a higher capacity adjustable buck converter (300 W class, 9 A, 7–40 V input, 1.2–35 V output). Retesting on this converter, still with the LX-225 servos, showed a clear improvement, standing power consumption roughly halved and walking was visibly better supported, but the rail was not entirely free of disconnections under hard movements. This indicates that the problem is not purely one of converter sizing, the LX-225 servos themselves stall at 4 A each, so even a handful of servos approaching stall simultaneously can exceed what any reasonably sized rail can supply. Fitting the final HTD-45H servos and raising the rail to 11.1 V improved it further, since those servos stall at a lower 3 A while providing more torque margin (Section 5.1.1), so an individual servo is less likely to approach stall current in normal operation. Together with the higher capacity converter this removed the disconnections in most operation, but not entirely. The demanding case is the trot, where nearly all twelve servos move at the same time, and the current they draw together during those transitions can still momentarily exceed what the rail delivers, and a disconnection follows. The solution is straightforward in principle, namely a converter with substantially more current headroom, of the order of 20 A, but one could not be sourced and validated within the time available for this work. It is identified as the outstanding hardware improvement for the platform.

A software mitigation was used during development, staggering the servo commands so that only half the legs are updated in any one control cycle, which lowers the peak current drawn. With the present converter it is normally left disabled, because the gait is visibly smoother when every leg is updated on the same cycle, and the disconnections it was introduced to prevent still happen but are now infrequent enough that the trade is not worth making.

6.4 Component Layout

The physical placement of the electronics, introduced in Chapter 5, resulted from a compromise between the space left available by the mechanical design and the optimization of cable runs. The perfboard sits at the center of the body, which allows it to deliver power over short, symmetric runs, directly to the adjacent IBT-2 drivers, rearwards to the Raspberry Pi 5 and Pico W, and forwards to the servo buck converter. The four IBT-2 drivers occupy the top of the middle support plate, the only space that could accommodate them, but a convenient one, since it places each driver close to the leg whose motor it drives, keeping the motor power cables short. The IMU is mounted at the center of the perfboard, which is also the preferred location for an orientation sensor, close to the geometric center of the body. Signal wiring uses thin jumper wires and power wiring uses 2 mm cable, and no interference problems between the two were observed in operation.

Regarding cooling, the two fans blow air into the body. The original Dingo project tested all combinations of fan directions (both blowing in, both out, and mixed) and found similar cooling performance in all of them [20], which is consistent with the body not being completely sealed, so the airflow direction was not optimized further. The Raspberry Pi 5 is fitted with self-adhesive copper heatsinks on the main processor and on the RP1 I/O controller, and with the fan airflow above it, it operates without thermal throttling. The IBT-2 drivers operate far below their current rating and do not heat appreciably. This is confirmed by the temperature measurements reported in Section 6.3.1, which showed no component under thermal stress after 30 minutes of driving.

The layout also favors maintenance, since without any tools the battery can be connected and disconnected through its access flap, and the fan supports open on their own flap mechanism, exposing the components beneath them, namely the BusLinker and the servo buck at the front, the Pico W at the rear, and enough of the IBT-2 drivers to probe their input and output voltages. Removing four screws frees the top plate and gives access to every component except the Raspberry Pi 5, which requires removing four additional screws and the rear panel. In practice, most inspection and testing tasks can be performed without disassembling anything.

Figure 6.3 shows the central bay with the top plate removed, where the four IBT-2 drivers surround the perfboard carrying the IMU, the logic converter and the two power buses.

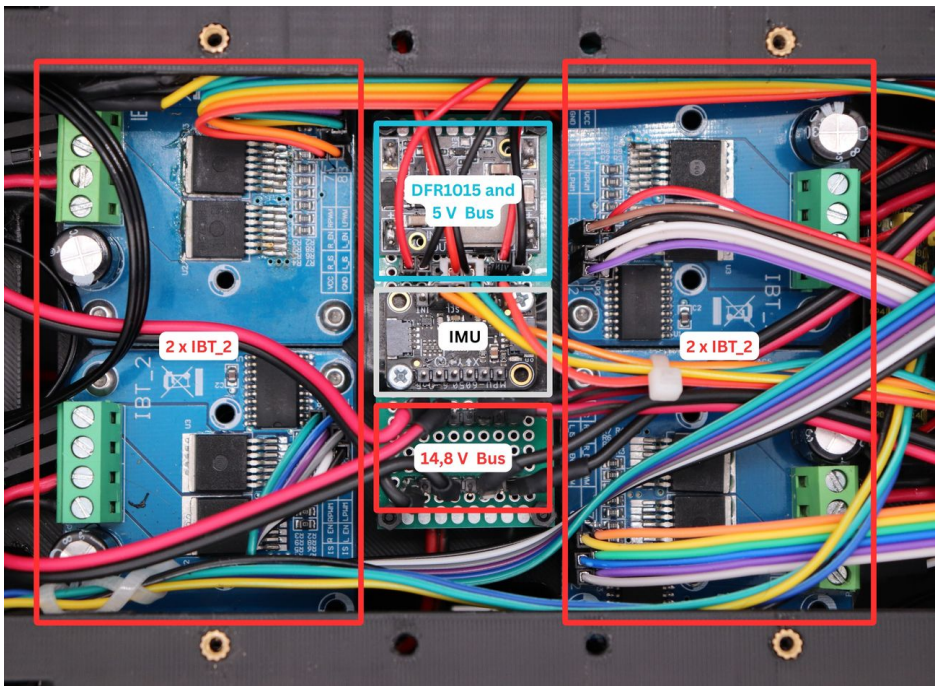


Figure 6.3: Top view with the top plate removed, showing the four IBT-2 drivers (red) and the perfboard with the DFR1015 and 5 V bus (blue), the IMU (gray) and the 14.8 V bus (red).

Chapter 7

Software and Control

7.1 Software Architecture Overview

The software of the original Dingo was adapted from the open-source Stanford Quadruped (Pupper) codebase and integrated into ROS Noetic [20]. This work inherits that structure and extends it with wheel control, new operating modes, and a revised input interface. The workspace organization into packages was presented in Chapter 4 (Table 4.3), this chapter describes how the software operates.

At runtime, the system consists of only two ROS nodes. The `joy` node reads the PS4 controller through the Linux joystick driver and publishes its state on the `/joy` topic at 30 Hz. The main driver node (`dingo_driver.py`) contains everything else, it runs the control loop at 50 Hz, and internally instantiates the input interface, the controller, the IMU reader, and the two hardware interfaces (servo bus and wheel interface). Concentrating the control pipeline in a single node avoids inter node message latency inside the 20 ms control period, ROS topics are used at the system boundaries, where their decoupling is valuable, rather than between every module.

Each control cycle follows the same pipeline, illustrated in Figure 7.1. The input interface converts the latest `/joy` message into a `Command` object (velocities, pose targets, and mode change events). The controller updates the behavior state machine, computes the desired foot positions for the active mode, applies IMU based tilt compensation when enabled, and solves the inverse kinematics to obtain the twelve joint angles. Finally, the hardware interfaces translate these into servo bus commands and wheel speed targets.

7.2 Docker Deployment and Automatic Startup

7.2.1 Containerized ROS Environment

ROS Noetic requires Ubuntu 20.04 while the Raspberry Pi 5 requires Ubuntu 24.04, so the entire ROS stack runs in a Docker container built from the repository's Dockerfile. The container is configured in a `docker-compose` file with three notable properties. First, it uses `network_mode: host`

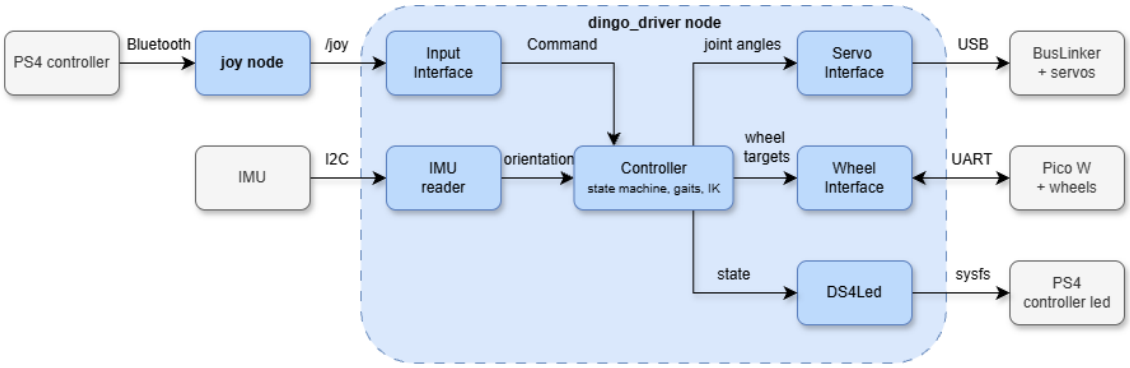


Figure 7.1: Runtime software architecture, with the `joy` node feeding the driver node that holds the control pipeline and the hardware interfaces.

and runs privileged, so ROS networking and hardware access behave as if running natively. Second, the host's `/dev` directory is bind mounted as a volume rather than passing individual devices, so that devices that appear after the container starts or reenumerate during operation (the joystick device once the controller pairs, or the BusLinker's USB port when it reconnects after a brownout) remain visible to the container without a restart. And third, the workspace source directory is bind mounted from the host, so code can be edited without rebuilding the image.

The container itself starts with an idle command and `restart: unless-stopped`, meaning it is always up and waiting, the robot software is executed inside it on demand by the launcher described next. A single `roslaunch` file starts both runtime nodes, with arguments selecting physical or simulated operation and enabling the IMU and joystick.

7.2.2 Power On and Launch Sequence

The robot is designed to go from flipping the power switch to being drivable without connecting a keyboard or display. This required solving a hardware detail first, the Raspberry Pi 5 does not power on by itself when power is applied to its pins, as it waits for its power button. The Pico W, which boots its MicroPython firmware immediately on power up, was given this responsibility, a wire connects the Pico's GPIO22 to pin 1 (`global_en`) of the Pi 5's J2 header, and the firmware pulses this line low for 200 ms shortly after boot, which the Pi's power management circuit interprets as a power button press. The Pico therefore acts as the robot's power sequencer in addition to its motor control role.

Once the Pi 5 boots, a systemd service on the host starts the launcher script (`start_robot.py`) and restarts it if it ever exits. The launcher waits for the PS4 controller to connect over Bluetooth, identifying the correct input device by checking for the PS button capability. The controller lights its bar blue by itself once it pairs, so blue confirms only that the Bluetooth link is up and then when the system is ready to start on the next button press the controller LED turns purple.

Pressing the PS button starts the robot, the launcher executes the `roslaunch` command inside the container, the driver node initializes, and the robot enters its resting state, indicated by the light bar turning white. Pressing the PS button again stops it, the launcher sends an interrupt signal to

the `roslaunch` process inside the container, allowing all nodes to shut down cleanly, and returns the light bar to purple. Signaling the process inside the container explicitly is necessary because terminating the local `docker exec` client does not propagate signals to the remote process, without it, a stopped robot would keep running invisibly and a subsequent start would launch a second, conflicting instance. The light bar colors give the operator feedback throughout, purple means the launcher is ready or the robot has stopped, and white means the robot is live in its resting state.

7.3 Inverse Kinematics

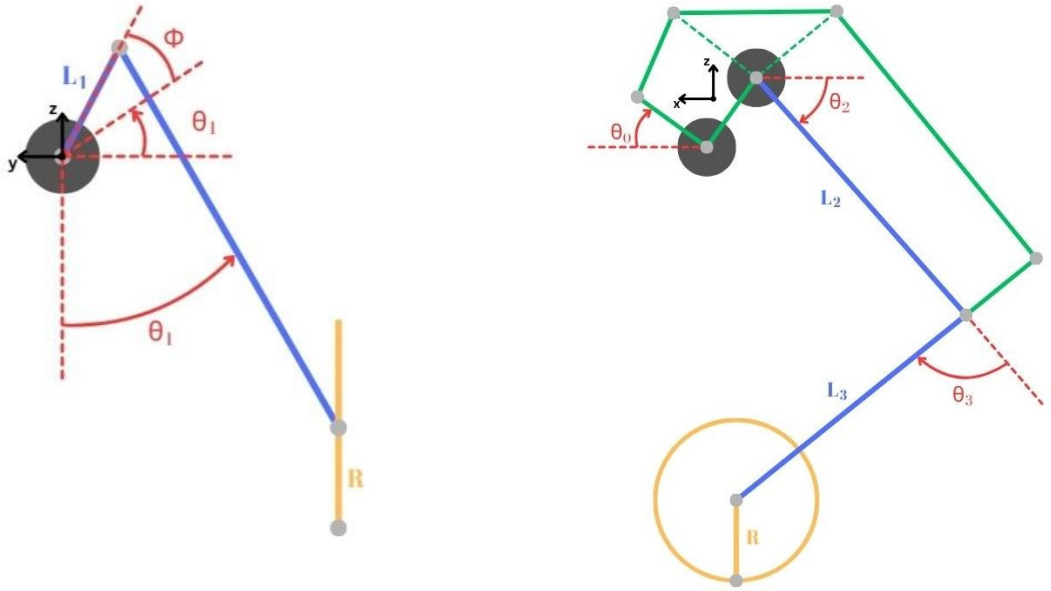
The controller expresses all motion in task space, as desired foot positions relative to the body, and converts them to joint angles at every control cycle. The inverse kinematics formulation is inherited from the original Dingo [20], this section summarizes it, and the complete derivation is reproduced in Appendix C.

Each leg is solved independently, in the two geometric steps shown in Figure 7.2. Given a desired foot position expressed in the leg's own frame, obtained by subtracting the leg's mounting offset from the body frame target, the three joint angles follow. The first step solves the hip abduction angle θ_1 in the frontal (YZ) plane. Because the leg plane is laterally offset from the abduction axis by the hip link L_1, mounted at a fixed structural angle $\phi = 73.9^\circ$, the abduction angle is recovered from the projection of the foot position onto this plane using the sine rule. The second step transforms the target into the now known leg plane, reducing the problem to a planar two link mechanism with link lengths L_2 (upper leg) and L_3 (lower leg), the hip flexion angle θ_2 and knee angle θ_3 then follow from the law of cosines. Finally, constant offsets are applied to match the servos' zero conventions, and all angles are wrapped to $[-\pi, \pi]$.

Two safeguards protect this solution, both clamping an out of range intermediate value rather than letting an inverse trigonometric function be evaluated outside its domain. In the first step, a foot target lying too close to the hip abduction axis drives the argument of the arcsine above unity. In the second, a target beyond the leg's maximum reach ($L_2 + L_3$) does the same to the argument of the arccosine. In both cases the offending quantity is clamped to the edge of its valid range, the reach limit stopping just inside full extension, and a warning is logged.

The consequence of omitting either clamp is not a distorted posture but an exception, the call raises immediately, and since nothing downstream can recover from it, the driver node terminates and the robot goes limp. This happened during a trot before the first of the two clamps was added. Reaching as far as possible toward an unreachable target is a far less disruptive failure than losing the control loop altogether.

The joint angles produced by the inverse kinematics are not yet servo commands. The hip abduction and hip flexion angles map onto their servos directly, through a per joint offset and direction, but the knee is driven remotely through a linkage, as described in Section 5.1. The knee servo angle therefore depends on both θ_2 and θ_3, and follows from the linkage solution of Equation 5.1, inverted for the servo angle with a crank zero measured for each leg. This conversion



(a) Frontal (YZ) plane.

(b) Leg plane.

Figure 7.2: Kinematic model of one leg. Following [20].

is performed inside the servo hardware interface, which keeps the controller itself independent of the linkage.

7.4 Gait Generation

Legged locomotion is produced by a phase based gait engine inherited from the original platform [20] and extended in this work with a second gait. Time is discretized into control ticks, and a gait is defined by a contact matrix, each column is a gait phase, each row a leg, and each entry states whether that leg is in stance (supporting) or swing (stepping) during that phase. At every tick, the gait controller determines the current phase from the elapsed ticks and the configured phase durations, and each leg is delegated to one of two subcontrollers accordingly.

The stance controller moves grounded feet opposite to the commanded body velocity, rotates them opposite to the commanded yaw rate, and regulates the body height with a first-order response, so that the body glides over the planted feet. The swing controller lifts the stepping foot along a triangular height profile and steers it toward a touchdown point computed with a Raibert style heuristic, the touchdown location leads the leg's neutral position by half of the distance the body will travel during the upcoming stance period, which centers the foot's stroke under the body and keeps the gait symmetric.

Two gaits are implemented:

- Trot, inherited from the original platform, moves diagonal leg pairs together. A cycle has four phases, an overlap phase with all four feet planted, the swing of one diagonal pair, a

second overlap, and the swing of the opposite pair. The trot is fast but requires the two stance legs to carry the whole robot during each swing phase, which produces the servo current peaks discussed in Chapter 6.

- Crawl, added in this work, is a wave gait intended for statically stable walking, exactly one leg swings at a time while the other three remain planted, so the support polygon is always a triangle rather than degenerating to a line. The swing order alternates between diagonally opposite corners (front-right, back-left, front-left, back-right), the standard maximum stability creep order. The crawl is slower than the trot but draws far lower peak current, since only one leg accelerates at a time.

Keeping three feet on the ground is a necessary condition for static stability, but not a sufficient one, the center of mass must also lie inside the triangle those three feet form. Simply lifting one leg of the default stance does not satisfy this, because the stance is roughly rectangular and its center does not coincide with the centroid of any of the four possible support triangles. The crawl therefore includes an explicit body weight shift, which moves the body a configurable fraction of the way toward the centroid of the triangle that is about to support it. The shift is applied ahead of the lift, so that the body is already supported when the foot leaves the ground, and it holds position while the foot is up rather than interpolating continuously across the phase, interpolating was found to move the body away from the support triangle at exactly the moment the foot was highest.

The trot, by contrast, cannot be statically stable at all. Two diagonal feet define a line, a support polygon of zero area, so the center of mass is by construction always outside it. The trot is stable only dynamically, relying on the next diagonal pair landing before the body has time to fall. This is a property of the gait itself and not something that can be corrected by tuning the foot trajectories.

The cadence of the active gait can be adjusted during operation. A lower cadence is gentler on the servo power rail, while a higher one produces faster, snappier stepping at the cost of higher peak current.

7.5 Operating Modes and State Machine

The robot's behavior is organized as a finite state machine, updated once per control cycle from the input events. The states used in this work are REST (standing at the default pose), TROT and CRAWL (the two gaits described above), and WHEELED (legs held in a fixed pose while the wheels drive). Figure 7.3 shows the states and their transitions.

The transition rules are restrictive, WHEELED and CRAWL can only be entered from REST, and each toggles back to REST with the same button. The trot toggle is more permissive, pressed in any locomotion state it switches to TROT, and pressed in TROT it returns to REST. Restricting mode entries to REST guarantees that every gait starts from a known, stable stance rather than from an arbitrary mid-motion posture.

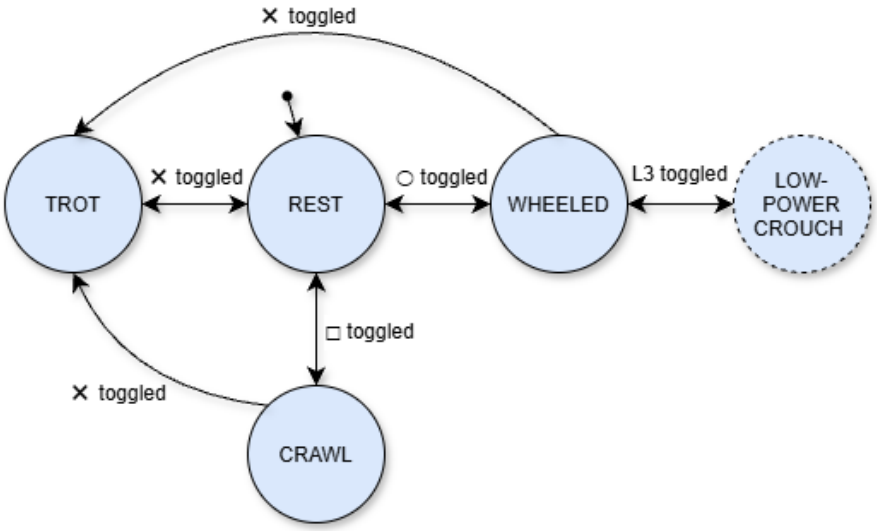


Figure 7.3: Behavior state machine.

Within WHEELED, a further submode implements low-power driving. When toggled, the controller first ramps the body down into a crouch at a configured height, chosen so that the fully flexed leg links seat into the interlocking cut designed into the lower leg for this purpose (Chapter 5), settling into a repeatable, mechanically stable pose under the robot's own weight. Once in the crouch, the torque of the four upper and four lower leg servos is released, while the four hip abduction servos remain powered, since the hip joint has no equivalent mechanical lock and provides the lateral stability of the crouched pose. Because the crouch is stable under gravity, the robot remains fully drivable in this condition at roughly 60% of the normal driving power, functioning essentially as a wheeled vehicle. Figure 7.4 shows the entry into this pose.

Two safeguards protect this submode. Leaving WHEELED for any reason automatically ends low-power mode and re-engages all servos, and the trot toggle is ignored while torque is released, since entering a gait with relaxed servos would produce a violent snap to the commanded pose. An explicit exit from low-power mode is therefore always required before walking again.



(a) Normal driving pose, all twelve servos powered.

(b) Descending toward the crouch height.

(c) Low-power crouch, leg servo torque released.

Figure 7.4: Entry into the low-power driving mode, with the body ramped down until the leg links seat into the interlocking cut and the leg servos are relaxed.

7.6 Body Stabilization with IMU

In every operating mode except the low-power crouch, the MPU-6050 provides orientation feedback used to keep the body level. Each control cycle, the measured roll and pitch angles are clipped to a maximum of 0.4 rad and scaled by a correction factor of 0.8, and the resulting compensation rotation is applied inversely to the commanded foot positions, if the body tilts forward, the feet are rotated backward by a proportional amount, driving the body back toward level. The correction factor below unity makes the compensation proportional, avoiding overcorrection and oscillation, while the clipping bounds the reaction to large disturbances such as a foot slipping off an obstacle.

The stabilization can be toggled at runtime with a controller button. The controller's light bar serves as output, each behavior state has a bright color variant when stabilization is on and a dark variant when it is off.

7.7 Wheel Control

7.7.1 Firmware

The Pico W runs a MicroPython firmware, flashed to the board as its main program, that implements the complete low-level wheel control loop. Each motor's quadrature encoder is decoded in interrupt handlers, every edge on either channel indexes a lookup table with the previous and current two bit channel state, adding ± 1 to the position count, which realizes full $4\times$ quadrature decoding. The wheel resolution of 14 200 counts per output revolution was calibrated manually by commanding a full wheel revolution and reading the accumulated count.

The control loop runs at 20 Hz. Each cycle, the speed of each wheel is estimated from the encoder count delta, and the PWM command is formed by a feedforward term proportional to the target speed, which provides the bulk of the drive voltage immediately, plus a PID correction that removes the remaining error under varying load. The output is clamped to 80% duty cycle, limiting the effective voltage to the motors' 12 V rating from the higher battery voltage. Two protections handle direction changes, the PID state is reset whenever the target changes sign, and a commanded reversal first stops the motor briefly before applying opposite drive, avoiding current spikes through the H-bridges.

The firmware communicates with the Raspberry Pi 5 through a simple line based ASCII protocol over UART. The Pi sends target speeds, stop commands, encoder resets, and runtime tuning of the PID and feedforward gains, the firmware acknowledges each command and streams a status line at 20 Hz containing, for each motor, the target speed, measured speed, applied PWM, and encoder position. The runtime tuning commands were essential during development, allowing the control gains to be adjusted iteratively without reflashing the firmware.

7.7.2 ROS Interface and Steering

On the Raspberry Pi side, the wheel interface translates driving commands into per wheel speed targets. The two left-side motors are mounted mirrored to the right side, so their commanded speeds are sign flipped to produce the same rolling direction. In wheeled mode, steering is differential, from the commanded forward speed and turning rate, the interface computes different speed targets for the left and right sides using the measured track width of 0.265 m between the wheel contact lines. The interface also parses the firmware's status stream and republishes it as ROS topics (`/wheel_rpms`, `/wheel_positions`), providing wheel telemetry to any other node.

7.8 Input and Command Interface

All operator input comes from the PS4 controller, processed by the input interface into a command object each control cycle. Figure 7.5 names the control buttons and Table 7.1 lists the functions associated with used ones to operate the robot. The buttons switch modes, with cross toggling the trot, square the crawl, and circle the wheeled mode, while triangle toggles IMU stabilization. The shoulder buttons R1 and L1 step the gait cadence up and down, and pressing the left stick toggles the low-power crouch in wheeled mode. The PS button is reserved by the launcher for starting and stopping the whole software stack.

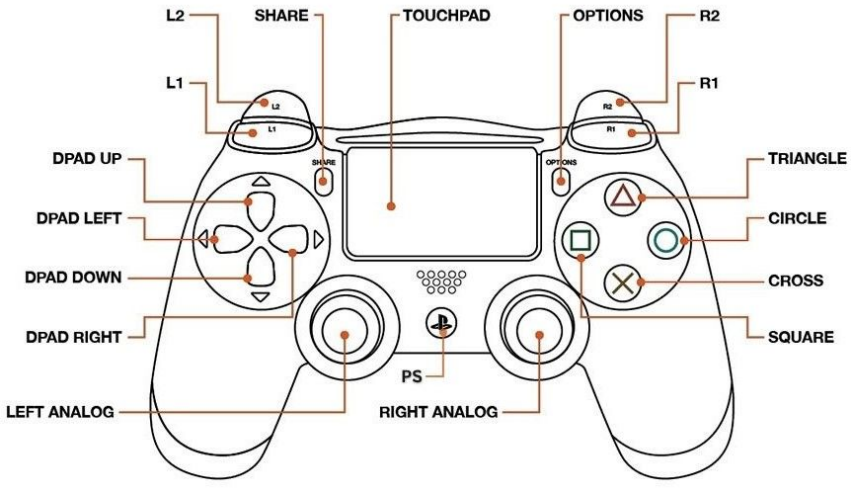


Figure 7.5: The PS4 controller with each control named, and Table 7.1 the function bound to each.

Table 7.1: Function bound to each control of Figure 7.5.

Control	TROT / CRAWL	WHEELED
Left analog	Body linear velocity	Body yaw and pitch
Left analog, pressed	—	Toggle the low-power crouch
Right analog	Body yaw rate and pitch	Wheel steering and wheel throttle
D-pad up / down	Raise / lower the body	
D-pad left / right	Roll the body left / right	
Cross	Toggle the trot gait	
Square	Toggle the crawl gait	
Circle	Toggle wheeled mode	
Triangle	Toggle IMU stabilization	
R1 / L1	Step the gait cadence up / down	
R2	Drive the wheels forward, proportionally to the pull	
L2	Drive the wheels backward, proportionally to the pull	
PS	Start and stop the software stack (launcher)	

The two joysticks change meaning with the mode, during the gaits the left stick commands the body's linear velocity and the right stick commands yaw rate and body pitch, whereas in wheeled mode the two exchange roles, the right stick becomes throttle and steering, and the left stick takes over body motion. The wheels can also be driven from the analog triggers while a gait is running, which is what allows both drive systems to be used at the same time, an option that proved useful on loose ground and is also the fastest way to cover flat ground (Section 8.3.3). In pure wheeled mode the triggers are ignored, since the right stick already carries the throttle.

Since the operator has no screen, all feedback is given through the controller's light bar, driven via the Linux LED subsystem. Each behavior state has a distinct color, and each color has a dark variant shown when IMU stabilization is disabled, so both the mode and the stabilization status are visible at the same time. Table 7.2 lists the base colors.

Table 7.2: Light bar color feedback.

Condition	LED color
Controller connected	Blue
Launcher idle / stopped	Purple
REST	White
TROT	Orange
CRAWL	Cyan
WHEELED	Green

Chapter 8

Experimental Evaluation

8.1 Test Setup and Methodology

8.1.1 Configuration Under Test

All results reported in this chapter were obtained with the final hardware configuration described in Chapters 5 and 6, HTD-45H servos supplied at 11.1 V through the 300 W buck converter, the redesigned lower legs with the interlocking crouch feature, and the four JGY-371 driven wheels. The robot's measured mass in this configuration is 3.54 kg, including the battery.

Earlier measurements taken during development, with the lower torque LX-225 servos on a 7.8 V rail, are not reported here as performance results, they appear in Section 6.3 only where they document the behavior of the power supply itself.

8.1.2 Test Environments

Testing was carried out both indoors and outdoors, with a different purpose in each. The quantitative measurements were made indoors on a flat, smooth floor, so that the characteristics of the robot itself could be measured without the terrain interfering. The outdoor tests served to verify that the robot moves across the surfaces it is intended for, from smooth pavements such as concrete and tiles to rough ground including dirt, grass, gravel and loose rocks.

8.1.3 Energy Measurement

Electrical energy consumption was measured by recharging the battery after each run and recording the charge restored by the charger, which gives the charge actually consumed by the run. Each run lasted 20 minutes of continuous operation in a single condition, and the resting pack voltage was recorded before and after as an independent cross check, the two measures agreed throughout.

From the restored charge Q over a run of duration t, the average current and power follow directly, and the endurance is projected as the time to consume 80% of the pack's 2400 mAh capacity, leaving a 20% reserve as is standard practice for LiPo batteries. For locomotion conditions,

two normalized metrics are also derived. The energy per unit distance,

$$E_d = \frac{P}{v}, \tag{8.1}$$

expresses consumption independently of duration, and the dimensionless CoT,

$$\mathrm{CoT} = \frac{P}{mgv} = \frac{E_d}{mg}, \tag{8.2}$$

normalizes it further by the robot's weight, allowing comparison with platforms of different mass. Both are computed with the measured mass of 3.54 kg and the measured speed of the corresponding mode.

8.1.4 Speed and Thermal Measurement

The forward speed was obtained by timing the robot across a 1.2 m straight track with a stopwatch at full commanded velocity, and each speed reported is the average of six runs. Component temperatures were measured with an infrared thermometer immediately after the continuous operation test, with the ambient temperature recorded alongside them.

8.2 Flat Terrain Locomotion

8.2.1 Speed

In wheeled mode the robot traverses the 1.2 m track in an average of 17.0 s, giving a forward speed of 0.070 m/s. This is 96% of the theoretical maximum imposed by the drivetrain, since the JGY-371's 20 RPM output on a 35 mm wheel radius corresponds to 0.073 m/s, confirming that the wheels run near their commanded speed under the robot's own weight rather than being dragged down by load. The same speed was measured in the crouched pose, so the pose does not affect wheeled travel.

In trot, the robot covers the same track in an average of 27.8 s, giving 0.043 m/s. This is far below the 0.3 m/s reported for the original legged-only Dingo [20], the dominant cause is cadence, the gait is run at a reduced speed factor, adopted to keep the peak servo current within what the power rail can supply, whereas the original figure was obtained at the platform's full design cadence. The redesigned legs contribute as well, and by more than their added mass alone, each wheel and its gearmotor sit at the distal end of the lower link, the point furthest from the joints that have to swing it, whereas the original Dingo swung a lower leg ending in a light printed part and a compliant foot. Because the contribution of a distal mass to the leg's rotational inertia increases with the square of its distance from the joint, placing the motor and wheel at the foot substantially increases the inertia that must be accelerated during swing.

Using both drives at once is faster than either of them. With the wheels driven through the analog triggers while the trot gait runs, the robot crosses the track in an average of 11.2 s, or 0.107 m/s, which is 1.5 times the wheeled speed, 2.5 times the trotting speed, and close to the sum

of the two. The two drives therefore add almost fully rather than interfering, since the powered wheels roll each planted foot forward while the gait carries the legs through their stride. This combined mode is examined further in Section 8.3.3.

Comparing the platform's two locomotion modes against each other, wheeled driving is the faster of the two, at 0.070 m/s against 0.043 m/s. Both figures are low in absolute terms, and each is limited by a different and independent cause. The wheeled speed is capped by the motors, the JGY-371 gearmotors were selected for their self-locking worm drive, which is what allows the wheels to double as static feet without consuming current, and their 1:380 reduction limits the output to 20 RPM. For a future upgrade, faster gearmotors of the same physical form factor would raise it directly, with no change to the mechanical or control design. The trotting speed, by contrast, is not limited by the actuators rated speed but by the cadence at which the gait can be run, and two things set that cadence, the current the servo rail can supply, and the swing inertia of legs that carry a motor and a wheel at their tips. Neither is a property of the gait itself, and both follow from the hybrid design, so both admit a hardware solution. A servo rail converter with considerably more current headroom would allow a higher cadence to be commanded without the rail collapsing, and lighter wheel motors would cut the inertia each leg has to swing.

8.2.2 Energy Consumption and Endurance

Table 8.1 reports the measured consumption in each operating condition, together with the projected endurance and, for the locomotion conditions, the normalized energy metrics defined in Section 8.1.3.

Table 8.1: Measured power, projected endurance to a 20% battery reserve, energy per meter and CoT by condition, normalized with each mode's own measured speed.

Condition	Current	Power	Endurance	E_d	CoT
Powered, servos relaxed	0.40 A	6.6 W	≈4.8 h	—	—
Standing, servos holding	0.91 A	15.0 W	127 min	—	—
Trotting	2.32 A	37.7 W	50 min	872 J/m	25.1
Trotting, wheels driven	2.52 A	40.9 W	46 min	381 J/m	11.0
Driving, standing pose	1.79 A	29.2 W	65 min	414 J/m	11.9
Driving, crouched pose	1.43 A	23.5 W	80 min	334 J/m	9.6
Driving, low-power crouch	1.09 A	18.0 W	105 min	256 J/m	7.4

Consumption is dominated by the leg servos, not by locomotion, as expected. Holding the twelve servos in a standing pose costs 8.4 W above the powered but relaxed baseline, and driving adds a further 14.2 W over standing at the same body height. Since the four wheel motors together account for only 2 to 3 W of that increase, the remaining 11 to 12 W is work done by the leg servos while the robot rolls. The servos are not merely holding a static pose, they are continuously correcting against the vibration and small disturbances transmitted through the rigid wheels and holding the body at a constant height.

The low-power driving mode saves 38.4% of driving power, reducing consumption from 29.2 W to 18.0 W and extending projected endurance from 65 to 105 minutes. Lowering the body into the crouch accounts for 5.7 W purely through the improved leg geometry, and releasing the servo torque once the legs rest in the mechanical interlock accounts for the remaining 5.5 W, so the saving is split almost exactly in half between the two.

Rolling is substantially cheaper than walking per unit distance. Trotting draws 37.7 W against 29.2 W for driving in the same standing pose, which is only 29% more instantaneous power. Because the robot trots more slowly than it drives, however, that power is spent covering less ground, and the energy required per meter diverges sharply, 872 J/m trotting against 414 J/m driving, and 256 J/m in the low-power crouch. Walking therefore costs 2.1 times as much as ordinary driving per meter, and 3.4 times as much as driving in the low-power crouch. The same ordering appears in the CoT, which falls from 25.1 for trotting to 11.9 for driving and 7.4 for the low-power crouch.

Wheeled locomotion is generally more economical than legged locomotion on flat ground, and what is worth noting is that it holds here despite two things working against the wheels. They are driven through a self-locking worm gearbox, a transmission chosen for its holding property rather than for its efficiency, and they are only around 60% faster than the trot, so the advantage does not come from a large speed margin. It comes instead from the fixed cost identified above, most of the power in either mode is spent holding the robot up rather than moving it, and that cost is charged per second, so the slower mode pays it over fewer meters.

One caveat bounds the size of this gap, though not its direction, standing with the servos holding already costs 15.0 W, which at the measured trotting speed works out to 347 J/m, about 40% of the 872 J/m total, before the robot has advanced at all. That component is paid per second rather than per meter, so running the trot at a higher cadence would spread it over more ground and lower the energy per meter even if the power drawn rose. The 872 J/m therefore reflects the low cadence imposed by the servo rail as much as it reflects any intrinsic cost of walking. A systematic sweep of cadence against both speed and power is the single most valuable experiment, although closing the gap entirely would require roughly doubling the trotting speed at unchanged power.

Driving the wheels during the trot demonstrates the same effect from another direction. Doing so raised the power drawn by only 3.2 W over trotting alone, from 37.7 to 40.9 W, roughly what the wheel motors themselves add, while the speed rose 2.5 times, so the energy per meter fell from 872 to 381 J/m, a reduction of 56%. The combined mode is therefore cheaper per meter than driving in the standing pose, at 414 J/m, although not than the two crouched poses, and it is the fastest way the robot has of covering ground. It was also the most demanding run for the servo power rail, and over its 20 minutes the servo bus disconnected several times, each time pausing the gait until the connection was restored, so the measured charge may slightly understate the consumption of uninterrupted operation.

The practical consequence for the platform is clear, and it is the same conclusion the terrain testing reaches independently in Section 8.3, the wheels are the efficient way to cover ground, and the legs exist to reach ground the wheels cannot. The hybrid arrangement is justified by capability,

not by economy.

8.3 Uneven Terrain Locomotion

The purpose of a hybrid platform is that it should handle surfaces on which a purely wheeled or purely legged machine of the same size would struggle. This section reports how each locomotion mode behaved across the surfaces tested. These tests are qualitative, and describe which modes remain functional rather than measuring traversal performance.

8.3.1 Behavior by Surface

On smooth pavement, both modes performed as they did indoors, wheeled driving and trotting were each fully functional, and the flat-ground results of Section 8.2 carry over.

On gravel with a stone size of roughly 2 to 5 cm, the two modes behaved differently. Trotting was effective, the legs step over individual stones and the loose surface is negotiated much as a flat one. Wheeled driving remained functional but degraded, as the small wheels must climb the stones rather than step over them, and the loose surface allows slip. Grass produced very similar behavior to gravel, with the wheels performing slightly better.

On large rocks, substantially bigger than the gravel, the robot failed to make progress. The legs could not lift high enough to place a foot on top of a rock and the feet slipped off the sloped surfaces, so the robot became stuck. This is a limit imposed by scale, the achievable step height of a machine of this size is small relative to bigger obstacles.

Regarding steps and thresholds, the negotiable obstacle height was measured directly and depends strongly on the commanded body height. At the normal standing pose the robot clears 1.0 cm, raised to its maximum body height it clears 3.5 cm. This is a useful result, because it shows that body height should be treated as an operational control rather than a fixed configuration value, the same crouch that saves 5.7 W while driving (Section 8.2.2) is also the pose that gives up the most obstacle clearance, so the two must be traded against each other according to the terrain.

8.3.2 Crawl Gait

The crawl gait is the least developed of the three locomotion modes. In its first tests it walked on smooth indoor floors and paved surfaces, but only for short distances before the body drifted out of a stable posture and the gait had to be restarted, and on rough ground it made no progress. By that point the inherited trot was working on exactly the surfaces the crawl had been added for, so the time that remained went into the trot rather than into making the crawl reliable, and what is reported here is the gait as first implemented rather than a tuned one.

A likely cause was identified but not yet tested. The crawl shifts the body toward the centroid of the three planted feet before each lift, and that shift is computed from the geometry of the stance, which places the center of mass at the geometric center of the body. The robot is slightly

rear heavy, so the mass sits behind that assumed point and the shift leaves a smaller stability margin than intended, an error a flat floor tolerates for a few cycles and uneven ground does not.

8.3.3 Simultaneous Wheeled and Legged Locomotion

The most significant result of the terrain testing was obtained on gravel, by driving the wheels while the trot gait was running. The control interface allows the wheel throttle to be commanded during the trot state, so the operator can walk the robot forward with the joystick and simultaneously apply wheel drive with the analog triggers.

Used this way, the robot negotiated the gravel noticeably better than in either mode alone. The interpretation is that the two subsystems contribute different things at the same time, the legs provide the ability to place each foot deliberately and to accommodate the irregularity of the surface, while the powered wheels add tractive effort at every ground contact, so that a foot that would otherwise slip on a loose stone instead drives the robot forward. The wheels are in contact with the ground throughout the stance phase in any case, since they are the feet, so powering them adds propulsion without requiring any change to the gait.

The same combination was then measured on flat ground, where it proved to be the fastest mode of the robot. With the wheels driven during the trot, the robot crosses the 1.2 m track at 0.107 m/s, against 0.070 m/s on the wheels alone and 0.043 m/s trotting, and it does so for 381 J/m, less than half the energy per meter of trotting alone (Section 8.2.2).

8.3.4 Body Attitude Compensation

The IMU based body stabilization described in Section 7.6 was verified separately, by placing the robot in a standing pose on a board and tilting the board while observing the body attitude. With compensation enabled, the legs adjusted to keep the body approximately level as the supporting surface was inclined, confirming that it behaves as intended. Disabling the stabilization during a trot caused the body to swing and the gait to degrade, so this stabilization is essential for the robot to properly move with its legs.

8.4 Load Capacity

The load capacity was measured in four different conditions, and the results vary a lot, more informative than the spread itself is that the limiting mechanism is not the same in each case, and that it changes as the leg servos are progressively unloaded. Table 8.2 summarizes the results.

While trotting, carried payload is limited to approximately 200 g before the servo bus browns out, far below the 500 g the original platform was specified for and below the 250 g it was reported to manage reliably [20]. The cause is electrical, not a matter of torque, and it is the transient peaks that count, since the average draw sits comfortably inside what the rail can supply. Nearly all twelve servos accelerate at once at touchdown and their combined draw already approaches the

Table 8.2: Measured load capacity by condition, with the mechanism that ended each test.

Condition	Load	What ended the test
Carrying, trotting	200 g	Servo bus brownout at foot touchdown
Carrying, driving with servos powered	2 kg	Servo bus brownout under steady holding load
Carrying, driving in low-power crouch	5 kg	Legs beginning to splay outward under load
Pushing, either mode	3.3 kg	Could not push a heavier load

converter's limit with no payload at all, so a small added mass is enough to push the peak beyond what the rail can deliver.

Driving on the wheels removes those transients, with the servos still powered, holding the body at a constant height and running the IMU compensation, the robot carried 2 kg, ten times the trotting figure, before the same brownout appeared. The servos now do steady work in place of repeatedly accelerating the legs, so the rail sees a continuous load instead of a run of spikes, and far more of its capacity becomes usable.

In the low-power crouch the limit stops being electrical altogether, with the leg links folded into the interlocking cut and the eight upper and lower leg servos relaxed (Section 7.5), the load passes through the mechanical lock and no longer through servo torque, and the robot drove with 5 kg aboard, well over its own mass. The test ended not in a brownout but with the legs beginning to splay outward under the load, the upper and lower links opening away from the body together. The hip abduction servos continued to hold their commanded angles throughout, so it is the leg joint structure that yields while the actuators hold. The binding constraint in this pose is therefore structural rather than electrical, and it concerns how firmly the leg joints are constrained, not the strength of the links themselves. This is the clearest payoff of the interlocking feature introduced in Chapter 5, designed to save power while driving, it is also what makes the platform usable as a carrier.

The robot was also able to push a load of 3.3 kg along the ground, almost its own mass, and did so just as effectively using its legs alone as with the wheels driving, so the pushing capability draws on both. The gap between what the robot can push and what it can carry while trotting follows from how each load reaches the actuators. A pushed object resists only through friction, so the reaction arrives horizontally and steadily, and is carried largely as compression along the leg links rather than as torque at the joints. Carried mass instead adds to what the servos hold through the whole gait cycle, acts on a longer moment arm above the hips, and peaks at touchdown, which is why the wheeled figures are an order of magnitude higher.

8.5 Field Demonstration with a Sensor Payload

The tests reported so far each isolate one property of the platform. A final test exercised them together, by fitting the robot with a payload and operating it outdoors over mixed natural ground. The payload was a consumer 360 camera (Insta360 One RS) of 130.5 g, roughly two thirds of

the 200 g the robot carries while trotting and a small fraction of what it carries on the wheels (Section 8.4). Mounting it required only a hole added to the top plate to pass the screw of the camera's own bracket.

The ground covered gravel, grass, and bare dirt, and the locomotion mode was selected to suit it, the wheels on flat stretches, the gait on broken ground, and both together where traction was poor, as described in Section 8.3.3. This is the operating pattern the platform was built around, and it is the first test in which all three were used in sequence rather than one at a time. Figure 8.1 shows the robot on the surfaces covered.

The robot carried the camera and covered the ground for most of the run, but the servo bus brownouts still occurred occasionally, which is consistent with the payload results above. At 130.5 g the robot is inside its measured budget but not comfortably so, and the added mass raises the touchdown current peaks that were already close to what the rail can deliver.

Figure 8.2 shows two frames from the recorded video, and confirms that the footage is usable despite the rigid wheels and the ground contact compliance given up by the wheel conversion. This is the closest the work comes to the reconnaissance and inspection applications that motivate hybrid platforms, carried out with the robot as built and a payload costing a small fraction of the platform.



(a) Walking over loose gravel.



(b) Crossing from paving onto gravel.



(c) On bare dirt and dry grass.



(d) Driving on flat ground.

Figure 8.1: The robot operating outdoors with the 360 camera on the top plate, the locomotion mode chosen to suit the ground.



(a) Approaching gravel.



(b) On a dirt track.

Figure 8.2: Two frames from the carried camera.

8.6 Discussion

8.6.1 Comparison with the Platform It Extends

The most direct comparison available is with the original Dingo, since this work is an extension of it and the original report documents its delivered performance [20]. Table 8.3 places the two side by side.

Table 8.3: This work compared with the original legged-only Dingo. Original figures from [20].

Metric	Original Dingo	This work
Locomotion modes	Legged	Legged and wheeled
Trotting speed	0.3 m/s	0.043 m/s
Wheeled speed	—	0.070 m/s
Trot with wheels speed	—	0.107 m/s
Mass (with battery)	3.15 kg	3.54 kg
Battery capacity	4000 mAh	2400 mAh
Endurance	≥25 min (walking)	50 min (walking) 65–105 min (driving) 46 min (trot with wheels)
Range per charge	≥450 m (walking)	130 m (walking) 275–444 m (driving) 295 m (trot with wheels)
Carried payload	500 g spec. (250 g reported reliable)	200 g (walking) 2 kg (driving) 5 kg (low-power crouch)
Approximate cost	≈€790	≈€580
Structure	Printed + acrylic	Printed

Three observations follow, the platform gained a complete second locomotion mode while becoming cheaper than the machine it extends, which is the result the low-cost objective was aiming at, although the comparison should be read with the currency and inflation caveats given in Appendix A, and with the note that the original's bill of materials includes components removed here.

Two figures are markedly worse than the original's, trotting speed fell from 0.3 to 0.043 m/s, and carried payload from the specified 500 g (250 g reported reliable) to 200 g. The speed figure is dominated by the reduced gait cadence adopted to keep peak servo current within the rail's capability, as discussed in Section 8.2.1, rather than by the added mass alone, and it is the clearest cost the wheel conversion has imposed, while the cadence sweep identified is the route to recovering part of it. The payload figure reflects the same underlying constraint, since it is transient current at touchdown that limits it, and like the endurance rows it applies to walking alone. On the wheels this robot carries 2 kg with the servos powered and 5 kg in the low-power crouch, where the load passes through the mechanical interlock rather than through servo torque (Section 8.4), so it exceeds the original's specified payload by a wide margin in the mode it is most likely to be used for carrying. What the platform gained in exchange is a second locomotion mode that covers flat ground faster than it walks, for less than half the energy per meter, and that carries an order of magnitude more.

The endurance rows need reading with care, because the longer walking figure is not an efficiency gain, this robot trots for twice as long as the original on 60% of the battery, but it does so at roughly a seventh of the speed, so it covers about 130 m against the original's 450 m or more. Normalizing for the smaller pack does not close that gap, the original travels some 0.11 m per mAh against 0.05 m here. The extra runtime is bought by moving slowly, and it is the same effect described in Section 8.2.2, where a large fixed cost of holding the body up is paid per second and therefore penalizes whichever mode covers the least ground in that time.

Driving changes the picture, though it does not reverse it, rolling for 65 to 105 minutes at 0.070 m/s gives a range of 275 to 444 m, two to three times what this robot manages on its legs and almost matching the original's walking range on a smaller battery. Trotting with the wheels driven reaches about 295 m, well beyond the range of walking alone. Rolling is cheap once the body is supported, which is the property the hybrid design set out to exploit, but on range per charge the platform does not yet overtake the machine it extends in any mode.

8.6.2 Position Relative to the State of the Art

Against the platforms reviewed in Chapter 3, this robot sits firmly at the accessible end of the cost spectrum, roughly two orders of magnitude below the commercial and research quadrupeds and comparable to the low-cost servo driven designs. Its distinguishing feature within that group is that it offers hybrid locomotion at all, the low-cost category is otherwise almost entirely legged only, while the wheeled legged platforms in the review are research or commercial machines at far higher cost and complexity.

The trade-offs that buy this are visible in the results. The platform is slow in absolute terms, its obstacle capability is bounded by its size, and it uses hobby grade serial servos whose torque and speed limits shape what the gaits can do. It is not competitive with the high-performance platforms on any capability metric, and it is not intended to be. What it demonstrates is that hybrid wheeled-legged locomotion can be implemented at a price point that had previously been occupied only by legged-only designs, using entirely commodity components and desktop manufacturing.

8.6.3 What the Hybrid Arrangement Delivered

Three results bear on whether the hybrid approach is justified on this platform.

The first is the low-power driving mode, because the leg links lock into the interlocking crouch, the robot can rest its own weight on a mechanically stable pose and switch off most of its actuators while still driving on the wheels. This reduced driving power by more than a third and extended endurance from 65 to 105 minutes, with no loss of speed, and the measurement showed the saving is split evenly between the crouch geometry and the torque release. This is a capability the legged platform did not have and could not have had, since a legged-only robot must actively hold its legs whenever it is moving.

The second is the behavior on gravel reported in Section 8.3.3, where driving the wheels during the trot gait produced better progress than either mode alone, and on flat ground made the robot faster than either mode alone, at 0.107 m/s, for less than half the energy per meter of trotting. This is the most significant of the three, because it demonstrates cooperation rather than merely a choice between modes. It also suggests that the value of a hybrid platform of this kind is not only in selecting the appropriate mode for the terrain, but in combining them on terrain where neither is sufficient.

The third result is the energy comparison between the two modes, and it settles what the hybrid arrangement is actually for. Rolling costs 414 J/m in the standing pose and 256 J/m in the low-power crouch, against 872 J/m for walking, so covering flat ground on the wheels takes less than half the energy per meter of covering it on the legs. Combined with the wheels also being faster than the legs, this means that, under the tested conditions, there was no measured advantage to walking alone on flat ground, and where the legs are needed, driving the wheels at the same time is both faster and cheaper per meter.

The justification for carrying both subsystems is therefore capability rather than economy. The legs are not a more efficient alternative to the wheels, they are what allows the robot to reach ground the wheels cannot handle, as the gravel and rock testing showed, and what allows the wheels to be used as feet in the first place. Read this way, the two locomotion modes are complementary rather than competing, and the energy result is not a down side but a statement of when each should be used, roll wherever rolling is possible, walk only where it is not.

8.6.4 Observed Limitations

Three limitations were apparent during testing, in addition to those discussed in Chapter 9.

Obstacle capability is bounded by scale. The robot handles surface irregularity of a few centimeters well, but obstacles approaching a significant fraction of its body height defeat it, the legs cannot lift high enough to step onto them and the wheels slip off their sloped faces. The achievable step height is limited by the swing clearance the leg geometry allows, which is small on a machine of this size.

Intermittent servo bus disconnections persist. Although the upgraded converter and actuators substantially reduced their frequency, disconnections still occur during the most demanding movements, when nearly all twelve servos move at once and their combined current draw momentarily exceeds what the rail can deliver. A converter with more current headroom is the remedy, and is identified as the main outstanding hardware improvement.

The trot does not hold a straight line. Over a straight run the robot drifts consistently to the left, so a heading correction is needed to keep it on course. The cause was not isolated, leg calibration asymmetry and the slightly uneven mass distribution are both plausible. Beyond being a limitation in its own right, it also means the measured trotting speed is slightly pessimistic, since the path actually traveled is longer than the straight line track.

Chapter 9

Conclusions and Future Work

9.1 Summary of Contributions

This dissertation set out to design, build, and test a lightweight, low-cost hybrid wheeled-legged quadruped robot. The result is a working platform, built by extending the open-source Dingo quadruped [20] with a wheel subsystem, a redesigned body, upgraded electronics, and new control software. The main contributions are the following.

The robot itself is a hybrid locomotion platform built on an existing legged design rather than one designed from scratch, which allowed the effort to concentrate on the hybrid capability itself. Each leg extremity was redesigned to carry a geared DC motor driving a wheel, so the robot can walk using its legs and roll using its wheels. The wheel was positioned so that its ground contact point falls close to that of the original foot, which allowed the inherited inverse kinematics and gait controllers to be reused, with the lower link length measured to the wheel axle and the conversion from joint angles to knee servo commands adapted to the legs as built (Chapter 5).

Its wheel design doubles as a foot, the drive motors use worm gearboxes, which are self-locking, so when unpowered the output shaft cannot be back-driven. The wheels therefore hold position without drawing any current and act as static feet during legged locomotion, so no additional locking mechanism was needed to switch between locomotion modes. The lower leg was further modified with an interlocking cut that seats the leg links against each other at full flexion, giving the crouched driving pose a repeatable, mechanically stable geometry.

This mechanical property was then exploited in software, after the measured power budget showed that the platform's endurance is limited by the servos holding the body and not by locomotion, since standing consumes several times more power than the wheel motors add when driving. This motivated a low-power driving mode that lowers the body into the mechanically locking crouch and then releases the torque of the leg servos, leaving only the hip abduction servos engaged for lateral stability. The mode reduced driving power by 38.4%, from 29.2 to 18.0 W, and extended projected wheeled endurance from 65 to 105 minutes, at no cost in driving speed. Measuring the crouched pose with the servos still powered showed that the saving divides almost evenly between the improved leg geometry of the crouch and the release of servo torque, so both

halves of the mechanism contribute. This is a direct example of a mechanical design decision being converted into an energy saving through software. The same mode turned out to pay a second time in load capacity, because the crouch carries weight through the interlock rather than through servo torque, the robot drove with 5 kg aboard in this pose, against 200 g while trotting, and the limit there was the legs splaying outward and not the power supply (Section 8.4).

The implementation is also low cost and almost entirely 3D-printed, the structural acrylic plates of the original design were replaced with printed parts, so that the structure can be manufactured on entry-level desktop printers from 1278 g of PLA, with only a few linkage parts not printed. The complete robot was built for approximately €580 in components, less than the original legged-only platform, despite adding the wheel subsystem and upgrading the computing and actuation hardware.

The control stack was also modernized and extended, the ROS control software was migrated to a Raspberry Pi 5 running the ROS environment inside a Docker container, resolving the incompatibility between the operating system required by the hardware and the one required by the ROS distribution. The software was extended with closed-loop wheel speed control running on a dedicated microcontroller, a crawl gait intended for statically stable walking (which was not developed far enough to be reliable), runtime gait cadence control, a runtime toggle with light bar feedback for the inherited IMU body stabilization, and an autonomous startup sequence that brings the robot from power-on to operational without any external computer, using the game controller as the sole interface.

9.2 Limitations

The platform has a number of limitations, several of which follow directly from the design decisions that make it inexpensive and simple.

Both locomotion modes are slow, the wheel drives are limited to 0.070 m/s because the selected gearmotors deliver only 20 RPM at the output, and the trot is slower still at 0.043 m/s, well below the 0.3 m/s of the platform this work extends. Even the fastest combination, trotting with the wheels driven, reaches only 0.107 m/s. The two limits have different causes and different remedies. The wheeled ceiling follows from the specific motor chosen rather than from the hybrid architecture, gearmotors of the same physical form factor are available with higher output speeds, and fitting them would raise the rolling speed directly, with no change to the mechanical design or the control software. The trotting speed, in contrast, is limited by the cadence at which the gait can be run, which is set both by the current the servo power rail can supply and by the swing inertia of legs carrying a motor and a wheel at their tips, and not by the actuators rated speed.

The servo power rail remains a constraint, peak servo current during walking exceeded what the original converter could supply, causing the servo bus to disconnect intermittently. Replacing the converter and upgrading the actuators both improved the situation substantially, but neither eliminated it. In a trot nearly all twelve servos move at once, and the current they draw together

during those transitions can still momentarily exceed what the rail delivers, so occasional disconnections remain. The fix is a converter with considerably more current headroom, of the order of 20 A. Sourcing and validating one did not fit within the time available for this work, and it is the clearest hardware improvement for the platform.

Carried payload while walking is severely limited, the robot can carry only about 200 g while trotting before transient servo current at foot touchdown browns out the servo bus. Carried mass loads the actuators throughout the gait cycle and on a longer moment arm than a pushed load, and the platform is slightly rear heavy, which aggravates the case. This is below the original platform's reported figure and is the clearest functional regression introduced by the added mass of the wheel subsystem. It is a walking limit rather than a platform limit, on the wheels the robot carries 2 kg with the servos powered and 5 kg in the low-power crouch, where the load passes through the mechanical interlock instead of through servo torque and the constraint becomes structural rather than electrical.

Ground compliance was also reduced, the original design used flexible printed feet to provide passive compliance, compensating for the fact that servo actuators cannot offer it at the joint. The wheel assembly is rigid apart from the tire, so the robot has less shock absorption at ground contact than the platform it is based on.

The simulation model does not represent the current robot, the Unified Robot Description Format (URDF) description and the Gazebo simulation package were inherited unchanged and still describe the legged-only platform, without wheels. Simulation was therefore not used in this work, and any future use of the simulation environment requires the model to be updated first.

The crawl gait is not reliable, it walks only short distances on smooth ground and does not function on rough ground, where the inherited trot remains the more capable gait. It was implemented and tested early, and once the trot was working on rough terrain the remaining time went into the trot rather than into tuning the crawl, so it was left as first implemented. The suspected cause is the body weight shift the crawl depends on, which is computed from the geometry of the stance while the robot is slightly rear heavy, so the shift leaves a smaller stability margin than intended.

Sensing is minimal, the robot's only environmental awareness comes from the IMU and the wheel encoders. It has no perception of the terrain ahead, so the choice between walking and rolling is made by the operator rather than by the robot.

9.3 Future Work

The platform was built to be extended, and several directions follow naturally from the limitations above.

The most valuable extension for a hybrid robot is automatic locomotion mode selection, letting the robot itself decide when to roll and when to walk. This requires terrain perception, and since the top plate already accepts a camera mount, as used in the field demonstration of Section 8.5, a depth camera or a small LiDAR could be fitted to let the robot detect obstacles that exceed the

wheels capability and switch to a gait autonomously, closing the loop on the hybrid concept that this work implements manually.

An improved wheel drive would remove the current rolling speed ceiling. The same gearmotor is sold in the same body and mounting at higher output speeds, 40 and 60 RPM against the 20 RPM fitted here, so the rolling speed can be raised by a component swap alone. The change would widen the margin by which rolling already outperforms walking on flat ground and make the robot more practical to use.

The simulation model can also be updated, extending the URDF description to include the wheels and their actuation would allow the hybrid platform to be simulated. This is a prerequisite for any learning-based control work, and would allow controllers to be developed and tested without risking the physical robot.

With an updated simulation model, learned locomotion policies become possible, reinforcement learning being a well established approach to gait generation for legged robots and of particular interest for a hybrid platform, where the controller must additionally decide how to coordinate wheeled and legged motion over irregular terrain.

Migrating to ROS 2 is also needed, since the current stack depends on a ROS distribution that has reached end of life, which is the reason it must run inside a container on the current hardware. A migration would remove that constraint and allow the software to run natively on the operating system supported by the computing platform.

Finally, stiffer leg joints would raise the carrying capacity further, since the load tests ended with the legs splaying outward rather than with any part breaking or any electrical limit being reached. Constraining the joints more firmly would also improve stability in general, as the same compliance is present, in smaller measure, during ordinary walking and driving.

9.4 Final Remarks

The robot developed in this work is a prototype, and it should be read as one. On its legs it is slower and less capable than the platform it extends, mainly because each leg now carries a motor and a wheel at its tip, which makes the legs considerably heavier to swing, and because the servo power rail limits how fast the gait can be run. Neither is a limit of the hybrid approach itself, and both would improve with more development time and better components, in particular a servo supply with more current headroom. What the platform does demonstrate is that wheels and legs can be combined on a low-cost, almost entirely 3D-printed quadruped, and that the combination offers something neither mode provides alone, as the gravel and flat-ground tests of Section 8.3.3, where driving the wheels during the trot made the robot faster than either mode alone, and the load capacity results of Section 8.4 showed. It is therefore best seen as a test platform and a starting point for further upgrades, on which the directions listed above can be explored.

References

[1] D. Bellicoso et al., "Advances in real-world applications for legged robots," *Journal of Field Robotics*, Oct. 2018. DOI: 10.1002/rob.21839

[2] M. Bjelonic, V. Klemm, J. Lee, and M. Hutter, "A survey of wheeled-legged robots," in *Robotics in Natural Settings*, J. M. Cascalho, M. O. Tokhi, M. F. Silva, A. Mendes, K. Goher, and M. Funk, Eds., vol. 530, Cham: Springer International Publishing, 2023, pp. 83–94, ISBN: 978-3-031-15225-2 978-3-031-15226-9. DOI: 10.1007/978-3-031-15226-9_11 [Online]. Available: https://link.springer.com/10.1007/978-3-031-15226-9_11

[3] Y. Pan, R. A. I. Khan, C. Zhang, A. Zhang, and H. Shang, "Pegasus: A novel bio-inspired quadruped robot with underactuated wheeled-legged mechanism*," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan: IEEE, May 13, 2024, pp. 1464–1469, ISBN: 979-8-3503-8457-4. DOI: 10.1109/ICRA57147.2024.10611633 [Online]. Available: https://ieeexplore.ieee.org/document/10611633/

[4] M. Hutter et al., "ANYmal - a highly mobile and dynamic quadrupedal robot," 2016. DOI: 10.3929/ETHZ-A-010686165 Accessed: Jan. 4, 2026. [Online]. Available: http://hdl.handle.net/20.500.11850/118642

[5] ANYbotics AG, *ANYmal Generation D: Technical specifications*, Version 1, ANYbotics AG, Zurich, Switzerland, May 2022.

[6] Unitree Robotics, *Unitree B2 quadruped robot: Technical specifications*, Manufacturer website and technical brochure. [Online]. Available: https://www.unitree.com/b2

[7] Boston Dynamics, *Spot robot specifications*, PDF technical document, 2020. [Online]. Available: https://bostondynamics.com/wp-content/uploads/2020/10/spot-specifications.pdf

[8] M. Bjelonic, R. Grandia, O. Harley, C. Galliard, S. Zimmermann, and M. Hutter, "Whole-body MPC and online gait sequence generation for wheeled-legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic: IEEE, Sep. 27, 2021, pp. 8388–8395, ISBN: 978-1-6654-1714-3. DOI: 10.1109/IROS51168.2021.9636371 Accessed: Jan. 6, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9636371/

[9] M. Bjelonic, "Planning and control for hybrid locomotion of wheeled-legged robots," Ph.D. dissertation, ETH Zurich, 2021. DOI: 10.3929/ETHZ-B-000515694 Accessed: Jan. 6, 2026. [Online]. Available: http://hdl.handle.net/20.500.11850/515694

[10] N. Kashiri et al., "CENTAURO: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1595–1602, Apr. 2019, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2019.2896758 Accessed: Jan. 7, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8630605/

[11] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke, "Hybrid driving-stepping locomotion with the wheeled-legged robot Momaro," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm: IEEE, May 2016, pp. 5589–5595, ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487776 Accessed: Jan. 7, 2026. [Online]. Available: http://ieeexplore.ieee.org/document/7487776/

[12] Unitree Robotics. "Go2-W developer – about Go2-W," Accessed: Jan. 2026. [Online]. Available: https://support.unitree.com/home/en/Go2-W_developer/about_Go2-W

[13] Norsat International Inc., *GQR0440-W wheel version intelligent quadruped robot - industrial model*, https://www.norsat.com/products/intelligent-quadruped-robot-industrial-model-wheel-version?utm_source=litpdp_pdf_link&variant=51943441826159, 2025. Accessed: Jan. 11, 2026. [Online]. Available: https://www.norsat.com/products/intelligent-quadruped-robot-industrial-model-wheel-version?utm_source=litpdp_pdf_link&variant=51943441826159

[14] Q. Zhou et al., "Max: A wheeled-legged quadruped robot for multimodal agile locomotion," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 7562–7582, Oct. 2024, ISSN: 1545-5955, 1558-3783. DOI: 10.1109/TASE.2023.3345876 Accessed: Jan. 4, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10375960/

[15] W. Li, L. Wei, and X. Zhang, "A wheels-on-knees quadruped assistive robot to carry loads," *Applied Sciences*, vol. 12, no. 18, p. 9239, Sep. 15, 2022, ISSN: 2076-3417. DOI: 10.3390/app12189239 Accessed: Jan. 7, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/12/18/9239

[16] L. Liang et al., "Design and simulation of wheeled-legged robot with variable configurations," in *Proceedings of the First International Conference on Advanced Robotics, Control, and Artificial Intelligence*, H. Wang, Ed., vol. 1376, Series Title: Lecture Notes in Networks and Systems, Singapore: Springer Nature Singapore, 2025, pp. 765–776, ISBN: 978-981-96-5372-0 978-981-96-5373-7. DOI: 10.1007/978-981-96-5373-7_62 Accessed: Jan. 7, 2026. [Online]. Available: https://link.springer.com/10.1007/978-981-96-5373-7_62

[17] G. Reina and M. Foglia, "An adaptive suspension system for planetary rovers," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 199–204, 2010, 7th IFAC Symposium on Intelligent Autonomous Vehicles, ISSN: 1474-6670. DOI: https://doi.org/10.3182/20100906-3-IT-2019.00036 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S147466701635056X

[18] W. Reid, R. Fitch, A. Göktoğan, and S. Sukkarieh, "Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover," *Journal of Field Robotics*, vol. 37, pp. 786–811, Oct. 2019. DOI: 10.1002/rob.21894

[19] RIVR Technologies. "RIVR ONE," Accessed: Jan. 2026. [Online]. Available: https://www.rivr.ai/product

[20] A. Calvert and N. Ferguson, "Design and construction of a low-cost robot quadruped," Supervisor: Dr Michael Burke, Final Year Project – Final Report (ENG4702), Monash University, Faculty of Engineering, Melbourne, Australia, May 2023.

[21] Yerbert, *DingoQuadruped*, https://github.com/Yerbert/DingoQuadruped. Accessed: Jan. 11, 2026.

[22] G. Valsecchi, N. Rudin, L. Nachtigall, K. Mayer, F. Tischhauser, and M. Hutter, "Barry: A high-payload and agile quadruped robot," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 6939–6946, Nov. 2023, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2023.3313923 Accessed: Jan. 7, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10246325/

[23] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2017–2031, Jun. 2021, ISSN: 20904479. DOI: 10.1016/j.asej.2020.11.005 Accessed: Jan. 7, 2026. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2090447920302501

[24] F. Iotti, "OmniQuad: A wheeled-legged hybrid robot with omnidirectional wheels," *Mechanism and Machine Theory*, 2025. DOI: 10.1016/J.MECHMACHTHEORY.2025.106125

[25] A. G. Joseph and S. P. Menon, "Development of a terrain adaptive quadruped-wheeled robot," in *2025 9th International Conference on Robotics and Automation Sciences (ICRAS)*, Osaka, Japan: IEEE, Jun. 27, 2025, pp. 94–99, ISBN: 979-8-3503-9249-4. DOI: 10.1109/ICRAS65818.2025.11108851 Accessed: Jan. 7, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/11108851/

[26] M. Bjelonic et al., "Offline motion libraries and online MPC for advanced mobility skills," *The International Journal of Robotics Research*, vol. 41, no. 9, pp. 903–924, Aug. 2022, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/02783649221102473 Accessed: Jan. 6, 2026. [Online]. Available: https://journals.sagepub.com/doi/10.1177/02783649221102473

[27] V. H. Pinto, "Hybrid legged-wheeled locomotion system for robotic vehicles," 2021.

[28] DEEP Robotics, *LYNX M20 industrial wheeled-legged robot*, https://www.deeprobotics.cn/en/index/lynx.html, 2025. Accessed: Jan. 11, 2026. [Online]. Available: https://www.deeprobotics.cn/en/index/lynx.html

[29] Pudu Robotics, *PUDU D5 industry-grade autonomous quadruped robot*, https://www.pudurobotics.com/en/products/d5, 2025. Accessed: Jan. 11, 2026. [Online]. Available: https://www.pudurobotics.com/en/products/d5

[30] Y. Fan, Z. Pei, C. Wang, M. Li, Z. Tang, and Q. Liu, "A review of quadruped robots: Structure, control, and autonomous motion," *Advanced Intelligent Systems*, 2024. DOI: 10.1002/AISY.202300783

[31] H. Khan et al., "Development of the lightweight hydraulic quadruped robot — MiniHyQ," *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2015. DOI: 10.1109/TEPRA.2015.7219671

[32] J. Kim, T. Kang, D. Song, and S.-J. Yi, "Design and control of a open-source, low cost, 3D printed dynamic quadruped robot," *Applied Sciences*, vol. 11, no. 9, p. 3762, Apr. 22, 2021, ISSN: 2076-3417. DOI: 10.3390/app11093762 Accessed: Jan. 7, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/11/9/3762

[33] B. Katz, J. D. Carlo, and S. Kim, "Mini Cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, May 2019, pp. 6295–6301, ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793865 Accessed: Jan. 7, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8793865/

[34] D. Jay, J. Hackett, P. Bosscher, C. Hubicki, and J. E. Clark, "WaLTER: A wheel and leg tumbling expedition robot," *IEEE International Conference on Robotics and Automation*, 2025. DOI: 10.1109/ICRA55743.2025.11128221

[35] V. H. Pinto, I. N. Soares, M. Rocha, J. Lima, J. Gonçalves, and P. Costa, "Design, modeling, and control of an autonomous legged-wheeled hybrid robotic vehicle with non-rigid joints," *Applied Sciences*, vol. 11, no. 13, p. 6116, Jun. 30, 2021, ISSN: 2076-3417. DOI: 10.3390/app11136116 Accessed: Jan. 2, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/11/13/6116

[36] Z. Wei, J. Zhang, Y. Yang, X. Shen, H. Sun, and A. Song, "Design, control and simulation of a leg-wheel robot: STransleg," *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, 2023. DOI: 10.1109/ICCCR56747.2023.10193990

[37] P.-A. Leziart, T. Flayols, F. Grimminger, N. Mansard, and P. Soueres, "Implementation of a reactive walking controller for the new open-hardware quadruped Solo-12," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China: IEEE, May 30, 2021, pp. 5007–5013, ISBN: 978-1-7281-9077-8. DOI: 10.1109/ICRA48506.2021.9561559 Accessed: Jan. 8, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9561559/

[38] N. Kau, "Stanford Pupper: A low-cost agile quadruped robot for benchmarking and education," *arXiv (Cornell University)*, 2021. DOI: 10.48550/ARXIV.2110.00736

[39] Stanford Pupper, *Pupper's documentation*, https://pupper.readthedocs.io/en/latest/. Accessed: Jan. 11, 2026.

[40] MangDang Technology Co., Limited, *Mini Pupper user manual*, https://minipupperdocs.readthedocs.io/en/latest/index.html. Accessed: Jan. 11, 2026.

[41] W. Zhang and K. Wang, "Wheely: An accessible platform for experimenting with wheeled-legged quadrupedal robots," *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, 2023. DOI: 10.1109/ICCCR56747.2023.10193993

[42] M. Bjelonic et al., "Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters*, 2019. DOI: 10.1109/LRA.2019.2899750

[43] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized ZMP constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1633–1640, 2019. DOI: 10.1109/LRA.2019.2896721

[44] S. S. Kotha et al., "Next generation legged robot locomotion: A review on control techniques," *Heliyon*, vol. 10, no. 18, e37237, Sep. 2024, ISSN: 24058440. DOI: 10.1016/j.heliyon.2024.e37237 Accessed: Jan. 10, 2026. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405844024132689

[45] J. Solano and D. Quiroz, *Stand, walk, navigate: Recovery-aware visual navigation on a low-cost wheeled quadruped*, 2025. arXiv: 2510.23902 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2510.23902

[46] M. Burke, *CAD files for Dingo robot*, Dataset, Monash University Bridges, Jun. 2023. DOI: 10.26180/23499384.v1

[47] engineerm-jp. "Inverse kinematics YouTube, four-legged robot implementation," Accessed: Sep. 24, 2026. [Online]. Available: https://github.com/engineerm-jp/Inverse_Kinematics_YouTube/tree/main/4Legs

Appendix A

Bill of Materials

Table A.1 lists all the components used to build the robot, together with their unit prices and the total cost of the platform.

Table A.1: Complete bill of materials with unit and total costs.

Component	Model	Qty	Unit (€)	Total (€)
Main computer	Raspberry Pi 5 (4 GB RAM)	1	119.88	119.88
Microcontroller	Raspberry Pi Pico W	1	3.12	3.12
Pico expansion board	Pico shield	1	3.50	3.50
Leg servo motors	Hiwonder HTD-45H (45 kg·cm)	12	23.80	285.60
Servo interface	BusLinker V2.5	1	5.58	5.58
Wheel motors	JGY-371 (20 RPM, 12 V)	4	8.50	34.00
Motor drivers	IBT-2 H-bridge	4	2.97	11.88
IMU	Adafruit MPU-6050 (6-DOF)	1	11.14	11.14
Battery	LiPo 4S (14.8 V, 2400 mAh)	1	33.00	33.00
DC-DC (servo rail)	300 W adjustable buck (7–40 V)	1	6.77	6.77
DC-DC (logic rail)	DFRobot DFR1015	1	3.35	3.35
Cooling fans	60×60×10 mm, 5 V	2	2.44	4.88
3D printing	PLA filament (full body)	–	–	25.00
Small parts	Screws, rollers and other hardware	–	–	30.00
Total				**577.70**

For comparison, the original Dingo, reported in May 2023, had a total cost of $1,290.69 AUD [20], equivalent to approximately €792.42 (at the average AUD/EUR exchange rate for 2023). Since this conversion is based on 2023 exchange rates and does not account for inflation between 2023 and the time of writing, it should be treated as an approximate reference rather than a precise comparison. Even so, the platform developed in this work, at €577.70, remains close to and below this converted baseline despite the addition of a full wheeled locomotion subsystem.

Appendix B

Knee Linkage Solution

This appendix gives the closure denoted $\mathscr{F}$ in Equation 5.1, which converts between the knee servo angle and the knee joint angle for the leg as built. The member lengths a to h are those measured on the robot and listed in Table 5.1, and the joint labels follow Figure 5.1.

All of the construction takes place in the leg plane, with the origin at the hip pitch axis D, the x-axis pointing forward and the y-axis upward. Two angles are fixed by the way the parts are mounted, the direction φ of the segment D–A that carries the servo axis, and the angle β between the segment G–F and the lower leg link. On the robot built here $\varphi = 54.95^\circ$ and $\beta = 95.42^\circ$, and the upper leg spans $L_2 = 130$ mm between D and G.

B.1 Intersection of Two Circles

Every joint of the mechanism is found from two lengths that are already known, which places it where two circles meet. For circles of radius r_0 and r_1 centered at $\mathbf{p}_0$ and $\mathbf{p}_1$, separated by $s = \|\mathbf{p}_1 - \mathbf{p}_0\|$, the two intersections are

$$\mathbf{p} = \mathbf{p}_0 + t\,\hat{\mathbf{u}} \pm q\,\hat{\mathbf{u}}^{\perp}, \qquad t = \frac{r_0^2 - r_1^2 + s^2}{2s}, \qquad q = \sqrt{r_0^2 - t^2}, \tag{B.1}$$

where $\hat{\mathbf{u}}$ is the unit vector from $\mathbf{p}_0$ to $\mathbf{p}_1$ and $\hat{\mathbf{u}}^{\perp}$ is perpendicular to it. The circles do not meet when $s > r_0 + r_1$ or $s < |r_0 - r_1|$, and the sign chooses between the two assemblies of that loop. This condition is the explicit test that replaces the undefined inverse cosine of the inherited routine, since a pose the linkage cannot reach is reported as such rather than returned as an invalid number.

B.2 From the Servo Angle to the Knee Angle

Given the servo crank angle θ_0 and the hip flexion angle θ_2, the joints are located in sequence. The servo axis and the knee pivot are fixed by the two input angles,

$$\mathbf{A} = -d\,(\cos\varphi,\ \sin\varphi), \tag{B.2}$$

$$\mathbf{B} = \mathbf{A} + a\,(-\cos\theta_0,\ \sin\theta_0), \tag{B.3}$$

$$\mathbf{G} = L_2\,(\cos\theta_2,\ -\sin\theta_2). \tag{B.4}$$

The first four-bar closes at C, which lies at distance b from B and c from D,

$$\mathbf{C} = \mathrm{circ}(\mathbf{B}, b) \cap \mathrm{circ}(\mathbf{D}, c). \tag{B.5}$$

The rigid triangle CDE then carries C into E, rotating by its apex angle at the fixed radius h,

$$\gamma = \arccos\left(\frac{c^2 + h^2 - e^2}{2\,c\,h}\right), \qquad \mathbf{E} = h\,\big(\cos(\alpha_C - \gamma),\ \sin(\alpha_C - \gamma)\big), \qquad \alpha_C = \mathrm{atan2}(C_y, C_x). \tag{B.6}$$

The second four-bar closes at F, at distance f from E and g from G,

$$\mathbf{F} = \mathrm{circ}(\mathbf{E}, f) \cap \mathrm{circ}(\mathbf{G}, g), \tag{B.7}$$

and the knee angle follows from the direction of the segment G to F, corrected by the fixed bend of the lower link,

$$\theta_3 = -\theta_2 - \alpha_R - \beta, \qquad \alpha_R = \mathrm{atan2}(F_y - G_y,\ F_x - G_x) + \pi, \tag{B.8}$$

wrapped to $[-\pi, \pi]$.

B.3 From the Knee Angle to the Servo Angle

The controller needs the opposite direction, since the inverse kinematics produces θ_2 and θ_3 and the servo has to be told where to go. The same construction runs backwards. The knee pivot G follows from θ_2 as above, F follows from G because the direction α_R is now known from θ_3, and the two loops are then closed in the reverse order,

$$\mathbf{E} = \mathrm{circ}(\mathbf{D}, h) \cap \mathrm{circ}(\mathbf{F}, f), \tag{B.9}$$

$$\mathbf{C} = c\,\big(\cos(\alpha_E + \gamma),\ \sin(\alpha_E + \gamma)\big), \qquad \alpha_E = \mathrm{atan2}(E_y, E_x), \tag{B.10}$$

$$\mathbf{B} = \mathrm{circ}(\mathbf{A}, a) \cap \mathrm{circ}(\mathbf{C}, b), \tag{B.11}$$

from which the crank angle is

$$\theta_0 = \operatorname{atan2}\big(B_y - A_y,\ -(B_x - A_x)\big). \tag{B.12}$$

Each sign in Equation B.1 gives a different assembly of the mechanism, so the candidates are checked by solving forward again and keeping those that reproduce the requested θ_3. The servo command finally follows from the crank angle through that leg's own zero,

$$\theta_{\text{servo}} = \frac{\theta_0 - \theta_{0,\text{zero}}}{\sigma}, \tag{B.13}$$

where $\theta_{0,\text{zero}}$ is measured for each leg and $\sigma = \pm 1$ accounts for the mirrored legs.

Appendix C

Inverse Kinematics Derivation

This appendix presents the complete inverse kinematics derivation summarized in Section 7.3, following the formulation of the original Dingo project [20], itself adapted from an earlier open-source implementation [47]. Given a desired foot position $\mathbf{p} = (x, y, z)$ expressed in the frame of one leg, the goal is to find the joint angles θ_1 (hip abduction), θ_2 (hip flexion), and θ_3 (knee flexion). The leg geometry is defined by the hip offset link $L_1 = 51.6$ mm, mounted at the fixed structural angle $\phi = 73.9°$, the upper leg $L_2 = 130$ mm, and the lower leg $L_3 = 121$ mm, measured from the knee axis to the wheel axle as explained in Section 5.1.

C.1 Leg Frame and Symmetry

The same derivation serves all four legs. Each body-frame foot target is first translated into the corresponding leg frame by subtracting the leg's mounting position, and for right-side legs the lateral coordinate is mirrored ($y \leftarrow -y$), so that a single set of equations applies to both sides.

C.2 Hip Abduction Angle

The abduction angle is found in the frontal (YZ) plane, with the nomenclature of Figure C.1. The frame is first rotated about the x-axis by $-R_1$, where

$$R_1 = \frac{\pi}{2} - \phi, \tag{C.1}$$

aligning the reference direction with the hip offset link. In the rotated frame, let (y', z') be the foot coordinates and

$$\ell_A = \sqrt{y'^2 + z'^2} \tag{C.2}$$

the distance from the abduction axis to the foot projected on the frontal plane. Three angles are then defined:

$$a_1 = \text{atan2}(z', y'), \tag{C.3}$$

$$a_2 = \arcsin\left(\frac{L_1 \sin\phi}{\ell_A}\right), \tag{C.4}$$

$$a_3 = \pi - a_2 - \phi, \tag{C.5}$$

where a_1 locates the foot relative to the positive y-axis, and a_2 follows from the sine rule applied to the triangle formed by the hip offset link L_1 (at angle ϕ) and the line ℓ_A. If the foot target lies closer to the abduction axis than $L_1 \sin\phi$, the argument of the arcsine exceeds unity and is clamped to ± 1 before a_2 is evaluated. The abduction angle is their combination,

$$\theta_1 = a_1 + a_3 \pmod{2\pi}. \tag{C.6}$$

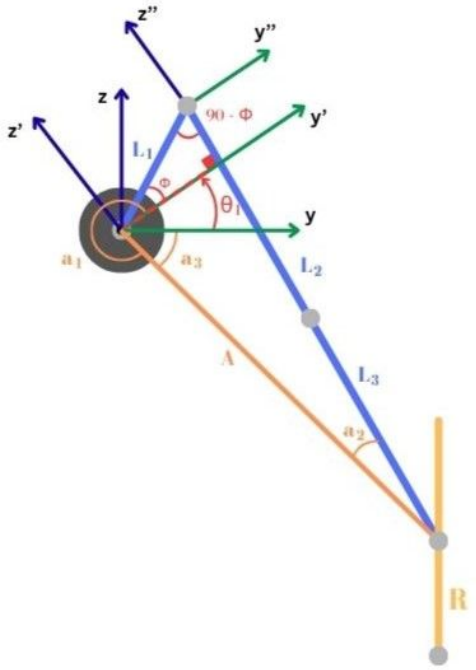


Figure C.1: Nomenclature for the hip abduction angle θ_1 in the frontal plane, with A from the abduction axis to the wheel axle. Following [20].

C.3 Reduction to the Leg Plane

With θ_1 known, the origin is translated to the end of the hip offset link,

$$\mathbf{p}' = \mathbf{p} - \begin{bmatrix} 0 \\ L_1 \cos\theta_1 \\ L_1 \sin\theta_1 \end{bmatrix}, \tag{C.7}$$

and the frame is rotated about the x-axis by $-R_2$, where

$$R_2 = \theta_1 + \phi - \frac{\pi}{2}, \tag{C.8}$$

which brings the plane containing the upper and lower leg links into the XZ plane of the rotated frame. The problem is now planar.

C.4 Planar Two-Link Solution

Let (x'', z'') be the foot coordinates in the leg plane, as shown in Figure C.2, and

$$\ell_B = \sqrt{x''^2 + z''^2} \tag{C.9}$$

the distance from the hip flexion axis to the foot. If the target is out of reach, ℓ_B is clamped:

$$\ell_B \leftarrow 0.99\,(L_2 + L_3) \quad \text{if} \quad \ell_B \geq L_2 + L_3, \tag{C.10}$$

which extends the leg as far as possible toward the unreachable target instead of producing an undefined solution. The two remaining angles follow from the law of cosines:

$$b_1 = \operatorname{atan2}(z'', x''), \tag{C.11}$$

$$b_2 = \arccos\left(\frac{L_2^2 + \ell_B^2 - L_3^2}{2 L_2 \ell_B}\right), \tag{C.12}$$

$$b_3 = \arccos\left(\frac{L_2^2 + L_3^2 - \ell_B^2}{2 L_2 L_3}\right), \tag{C.13}$$

giving

$$\theta_2 = b_1 - b_2, \tag{C.14}$$

$$\theta_3 = \pi - b_3. \tag{C.15}$$

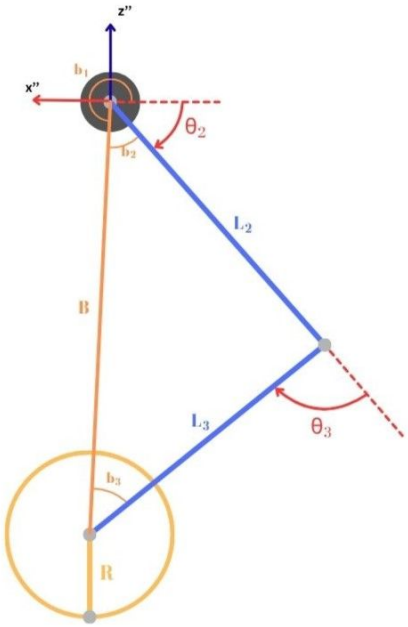


Figure C.2: Nomenclature for the hip flexion angle θ_2 and the knee angle θ_3 in the leg plane, with B the distance to the wheel axle. Following [20].

C.5 Angle Conventions

Finally, constant offsets align the results with the servo zero conventions used by the hardware interface,

$$\theta_2 \leftarrow \theta_2 - \pi, \tag{C.16}$$

$$\theta_3 \leftarrow \theta_3 - \frac{\pi}{2}, \tag{C.17}$$

and all three angles are wrapped to the interval $[-\pi, \pi]$. The conversion of θ_3 into the physical knee servo angle, which accounts for the linkage between the hip-mounted servo and the lower leg, is described in Section 5.1.