Imperial College London

DESIGN, MODELLING AND CONTROL OF SLIDER: AN ULTRA-LIGHTWEIGHT, KNEE-LESS, LOW-COST BIPEDAL WALKING ROBOT

Ke Wang

Supervised by Dr Petar Kormushev

Robot Intelligence Lab
Dyson School of Design Engineering
Imperial College London

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy at Imperial College London

March 2022

ABSTRACT

Robots can improve the world we live in by replacing humans in dangerous and tedious tasks. More specifically, bipedal robots can navigate complex and unstructured environments designed for humans which may not be possible for their wheeled counterparts. This thesis presents a comprehensive study on the design, modelling, control and motion planning for a novel ultra-lightweight, knee-less, low-cost bipedal walking robot called SLIDER.

Although many bipedal robots exist, it is still a big challenge to design and build a bipedal robot that is lightweight, able to perform agile motions while keeping the cost low. The first part of the thesis aims to tackle this challenge by proposing a novel knee-less robot design. Unlike most bipedal robots which have knees, SLIDER has no knees. Instead it has prismatic sliding joints at the hip that replace the functionality of knees. This design results in lightweight legs which in turn bring advantages for agile control and improved energy efficiency. This thesis presents the details of the design and modelling of SLIDER, a comparison between knee-less and anthropomorphic bipedal robot design as well as real robot experiments. Furthermore, lessons learned through multiple iterations of the design and modelling of SLIDER are summarized.

Besides the design and modelling, the control and motion planning algorithms are another key part for achieving bipedal locomotion. The second half of the thesis presents contributions to the walking controller, footstep planning and dynamic motion generation for bipedal robots. Three algorithms are proposed. The first algorithm enables bipedal robots to blindly walk over uneven terrains by decoupling the actuated Spring Loaded Inverted Pendulum (aSLIP) model and formulating the footstep planning as a linear Model Predictive Control (MPC). The second proposed algorithm enables bipedal robots to adapt step location and timing by solving a nonlinear MPC. The nonlinear MPC can be solved with a frequency of 200 Hz by an asynchronous framework. The third proposed algorithm generates highly dynamic jumping motions for bipedal and quadrupedal robots. A new model called Lump Leg Single Rigid Body Model (LL-SRBM) approximately models the centroidal inertia and improves the jumping performance in terms of jumping speed and distance.

The final contribution of the thesis is a novel proposed walking gait called RotoGait. Thanks to the unique design of SLIDER, the legs can continuously rotate while walking, instead of accelerating and decelerating each swing leg as in the normal gait. Trajectory optimization is used to generate the optimal energy-efficient swing leg trajectory for RotoGait. Compared with the normal walking gait, RotoGait proves to be more energy efficient at high

walking speeds. RotoGait represents a new class of walking gaits that opens the possibility of improved locomotion performance.

Looking ahead, the novel insights shown in this thesis open up future research avenues for the design of bipedal robots and performance improvement of bipedal locomotion.

There's some good in this world, Mr. Frodo... and it's worth fighting for.

— Sam Gangee, Lord of the Rings.

ACKNOWLEDGMENTS

I am very thankful for the support given to me by many people over the course of my studies, without which this work would not have been possible.

I would like to thank my supervisor, Dr Petar Kormushev, for his trust for giving me the freedom to pursue my research interests and sharing his passion for bipedal robots. I would thank my thesis committee, Dr Olivier Stasse and Dr Ravi Vaidyanathan for carefully checking my thesis.

I would also thank the lab members of the Robot Intelligence Lab at Imperial College London, especially Digby Chappell. Your big brain always produces a lot interesting ideas and it is a great pleasure to collaborate with you. I also thank Roni Permana Saputra who helped me with the experiments, discussed research ideas and reviewed my paper drafts. I would like to thank all other members: Francesco Cursi, Nemanja Rakicevic, Fabio Pardo, Arash Tavakoli, Vitaly Levdik and Ahmad Al Attar. We had a lot of discussions at the lab meetings and I have learned a lot about machine learning from you guys.

I would like to thank Prof. Sethu Vijayakumar who kindly hosted me at the University of Edinburgh so I could work on the cool robots TALOS and ANYmal. I would also like to thank Dr Vladimir Ivan for helping me set up the TALOS robot. I would like to thank Dr Songyan Xin, Dr Guiyang Xin and Dr Carlos Mastalli for supervising me and guiding me when I was lost in the jungles of different ideas. I will always miss the afternoon coffee time when a lot of interesting ideas emerged.

I would like to thank the master and bachelor students who worked on SLIDER, especially Aksat Shah, Oscar Gill, Matt Cocking, Joseph Shore, David Marsh, Akshay Raut, Stergio Chikaros, Zhonghe Jiang, Hengyi Fei, Varan Gupta, Jacopo Zhaoyang Hu, Peter Tisnikar, Tianhu Peng, Yuki Cheng, Thomas Godden, Zac Galibov and Julia Péladeau. Without your efforts SLIDER would not be able to walk.

At last, I would like to thank my family. My father Yunfu Wang and my mother Ying Liu. Without your encouragement and support, I could not have finished my PhD. I would especially like to thank my wife Xiaomeng Liao, it is you that accompanied me to go through the challenging period of my PhD and until the end. It is you that encouraged me to pursue my interests and had the patience to let me finish this long journey. I would also like to thank my grandmother who is already in heaven. Although you have not been in school for

one day in your life, you taught me optimism, hard work and perseverance. These pearls helped me to grow from a young boy born in a small village in Shandong Province, all the way to a PhD candidate at Imperial College London.

Dedicated to my parents and my beloved wife, Xiaomeng.

COPYRIGHT DECLARATION

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution Non-Commercial license (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

ORIGINALITY DECLARATION

I hereby declare that this thesis and the work herein detailed, was composed and originated by myself, except where appropriately referenced and credited.

London, March 2022

Ke Wang

CONTENTS

I INTRODUCTION

1 INTRODUCTION 2
- 1.1 Motivation . . . 2
- 1.2 Related Work . . . 3
  - 1.2.1 Robot Design . . . 4
  - 1.2.2 Whole Body Control . . . 4
  - 1.2.3 Reduced-order Model . . . 6
  - 1.2.4 Model Predictive Control . . . 6
  - 1.2.5 Nonlinear Trajectory Optimization . . . 7
  - 1.2.6 Machine Learning . . . 7
- 1.3 Thesis Contributions . . . 9
  - 1.3.1 Contributions of other people . . . 10
- 1.4 Thesis Outline . . . 10
- 1.5 Author Publications . . . 13

II DESIGN AND MODELLING OF SLIDER

2 DESIGN OF SLIDER 16
- 2.1 Overview of SLIDER Design . . . 16
- 2.2 SLIDER's Design vs Anthropomorphic Design . . . 17
  - 2.2.1 Mass Distribution . . . 17
  - 2.2.2 Workspace Comparison . . . 19
  - 2.2.3 Energy Efficiency . . . 19
- 2.3 The Design History of SLIDER . . . 22
- 2.4 Joint Mechanism . . . 24
  - 2.4.1 SLIDER v4 . . . 25
  - 2.4.2 SLIDER v5 . . . 28
  - 2.4.3 Hip Joints . . . 30
- 2.5 Actuators, Electronics and Software . . . 32
  - 2.5.1 Electronics and Software . . . 34
- 2.6 Lessons Learned . . . 35

3 MODELLING OF SLIDER 36
- 3.1 DH Parameter of SLIDER . . . 36
- 3.2 Ankle Modelling . . . 37

3.2.1 Ankle Model in SLIDER v4 . . . 38
3.2.2 Ankle Model in SLIDER v5 . . . 39
3.2.3 Algorithm Summary of Ankle Inverse Kinematics in SLIDER v5 . . . 42
3.2.4 Experimental Validation of Ankle Inverse Kinematics . . . 42
3.3 Geometric Inverse Kinematics of The Robot . . . 42
3.4 Improve Energy Efficiency with Parallel Elastic Elements . . . 45
3.4.1 Design of The Vertical Prismatic Joint . . . 47
3.4.2 Characteristics of the Bungee Cord . . . 48
3.4.3 Finding Optimal Energy-Efficient Configuration Using Bayesian Optimization . . . 49
3.4.4 Experiments . . . 52
3.4.5 Discussion . . . 55

III CONTROL AND PLANNING FOR SLIDER

4 CONTROL OF SLIDER 57
4.1 Preview Control . . . 57
4.1.1 First Iteration of Preview Control . . . 58
4.1.2 Remaining Iterations of Preview Control . . . 59
4.1.3 Implementation Details . . . 60
4.2 Inverse Dynamics Whole Body Controller . . . 60
4.2.1 Task Formulation . . . 60
4.2.2 Contact Constraint . . . 61
4.2.3 QP formulation . . . 62
4.2.4 Implementation and Discussion . . . 63
4.3 Real Robot Experiments . . . 64
4.3.1 Feedback Signals . . . 65
4.3.2 Walking Experiment Results . . . 66
5 REAL-TIME FOOTSTEP PLANNING FOR TERRAIN-BLIND WALKING 70
5.1 Introduction . . . 70
5.2 System Overview . . . 73
5.2.1 The Control Hierarchy . . . 73
5.3 Online Footstep Planning . . . 73
5.3.1 The Decoupled aSLIP Model . . . 73
5.3.2 Vertical Dynamics of decoupled aSLIP Model . . . 75
5.3.3 Horizontal Dynamics of Decoupled aSLIP Model . . . 77
5.3.4 Foot Placement and CoM Trajectory Optimization . . . 78
5.4 Trajectory Tracking . . . 81
5.4.1 Rigid Body Dynamics . . . 81
5.4.2 Contact Force Constraint . . . 82

5.4.3 Swing Foot Trajectory Generation . . . 83
5.5 Results . . . 83
5.5.1 Implementation . . . 83
5.5.2 Flat Ground Walking . . . 84
5.5.3 Walking on Smooth Uneven Terrain . . . 84
5.5.4 Walking on Discrete Uneven Terrain . . . 87
5.6 Discussion . . . 87
6 TERRAIN-AWARE REAL-TIME GAIT OPTIMIZATION 88
6.1 Introduction . . . 88
6.2 Preliminary . . . 91
6.2.1 Footstep Planning as a Discrete-time Optimal Control Problem . . . 91
6.2.2 Linear Inverted Pendulum Model . . . 92
6.2.3 Constrained Optimisation with Augmented Lagrangian . . . 93
6.3 Methods . . . 94
6.3.1 System Overview . . . 94
6.3.2 Footstep Location and Timing Optimization Formulation . . . 95
6.3.3 Augmented Lagrangian Optimizer . . . 95
6.3.4 Interior Point Optimizer . . . 96
6.3.5 Asynchronous Optimization . . . 97
6.3.6 Terrain-Aware Footstep Planning . . . 97
6.3.7 Swing Foot Trajectory Generation . . . 101
6.3.8 Whole Body Controller . . . 101
6.4 Experimental Results . . . 103
6.4.1 Experimental Platform and Scenarios . . . 103
6.4.2 Flat Ground Walking . . . 103
6.4.3 Push Recovery on Flat Ground . . . 104
6.4.4 Push Recovery on Uneven Terrain . . . 106
6.4.5 Terrain-Aware Walking . . . 107
6.5 Discussion . . . 108
6.6 CONCLUSION . . . 110
7 UNIFIED MODEL WITH INERTIA SHAPING FOR HIGHLY DYNAMIC JUMPS 111
7.1 Introduction . . . 111
7.1.1 Contributions . . . 113
7.2 System Overview . . . 114
7.3 The Lump Leg Single Rigid Body Model . . . 114
7.4 Trajectory Optimization . . . 117
7.4.1 Problem Formulation . . . 117
7.4.2 Cost Function . . . 119
7.4.3 Kinematics Constraint . . . 119

7.4.4 Contact Constraint . . . 120
7.4.5 Ground Reaction Force Constraint . . . 120
7.5 Whole Body Controller and Contact Detection . . . 121
7.5.1 QP formulation . . . 122
7.5.2 Contact detection . . . 123
7.6 Experiment Results . . . 123
7.6.1 Robot Platforms . . . 123
7.6.2 Implementation . . . 124
7.6.3 Twisting Jump . . . 124
7.6.4 Forward Jump . . . 126
7.6.5 Computation Time . . . 128
7.7 Discussion . . . 128
8 ROTOGAIT 129
8.1 Introduction . . . 129
8.2 Kinematics . . . 130
8.3 Dynamics . . . 132
8.4 Optimal Swing Leg Trajectory . . . 133
8.4.1 Cost Function . . . 133
8.4.2 Constraints . . . 134
8.4.3 Trajectory Optimization Formulation . . . 135
8.5 Footstep Planner . . . 135
8.6 Result . . . 136
8.6.1 Normal Walking Gait . . . 136
8.6.2 Achievable Gaits . . . 138
8.6.3 Energy Consumption . . . 140
8.7 Discussion . . . 141

IV CONCLUSION AND FUTURE WORK

9 CONCLUSION AND FUTURE WORK 144
9.1 Conclusion . . . 144
9.2 Limitations and Future Work . . . 145

Appendix

A.1 Reduced Order Models . . . 150
A.1.1 Notation . . . 150
A.1.2 Linear Inverted Pendulum . . . 150
A.1.3 Spring Loaded Inverted Pendulum . . . 150
A.1.4 Variable Height Inverted Pendulum . . . 151
A.1.5 Single Rigid Body Model . . . 152
A.1.6 Lumped Leg Single Rigid Body Model . . . 152

A.2 Analytical Gradients 154
A.2.1 Footstep position gradients 154
A.2.2 Footstep duration gradients 154
A.3 Quaternion to Rotation Matrix Conversion 156
A.4 Derivative of Quaternion 156
A.5 Inertia Tensor 157

BIBLIOGRAPHY 159

LIST OF FIGURES

Figure 1.1 Dynamic Legged Animals. . . . 3
Figure 1.2 Different bipedal robots. . . . 5
Figure 1.3 Thesis roadmap 8
Figure 1.4 Chapters from a view of a system 11
Figure 2.1 A model of SLIDER robot. . . . 17
Figure 2.2 Illustration of SLIDER's mass distribution in comparison to an anthropomorphic bipedal robot, COMAN. . . . 18
Figure 2.3 Height-mass relationship of selected robots. . . . 18
Figure 2.4 Comparison of SLIDER and COMAN walking with identical gait patterns. . . . 20
Figure 2.5 Comparison of the center of mass (CoM) motion between SLIDER and COMAN with the identical gait. . . . 21
Figure 2.6 SLIDER's design vs Anthropomorphic design in workspace. . . . 22
Figure 2.7 Different iterations of SLIDER design. . . . 23
Figure 2.8 A model of the SLIDER v4 of joint mechanisms. . . . 25
Figure 2.9 An exploded view of SLIDER v4's pitch and slide gearbox. . . . 25
Figure 2.10 Hip gearbox assembly. . . . 26
Figure 2.11 An exploded view of SLIDER's slide gearbox. . . . 27
Figure 2.12 An exploded view of SLIDER v4's two-stage cycloidal reduction for hip roll. . . . 28
Figure 2.13 An exploded view of SLIDER's ankle gearbox. . . . 29
Figure 2.14 An exploded view of SLIDER v5 whole assembly. . . . 30
Figure 2.15 An exploded view of SLIDER v5 hip roll assembly. . . . 31
Figure 2.16 An exploded view of SLIDER v5 hip pitch assembly. . . . 31
Figure 2.17 An exploded view of SLIDER v5 hip slide assembly. . . . 32
Figure 2.18 The SLIDER v5 ankle design. . . . 32
Figure 2.19 Torque measurement of SLIDER in Gazebo. . . . 33
Figure 2.20 T-motors: (a) R100; (b) AK10-9; (c) AK80-80. . . . 34
Figure 3.1 SLIDER link placements. . . . 37
Figure 3.2 Illustration of the inverse kinematics for the ankle. . . . 38
Figure 3.3 Geometry illustration of SLIDER's ankle. . . . 39
Figure 3.4 Illustrations of ankle kinematics in 3D. . . . 41

Figure 3.5 Comparison of desired and actual trajectories of ankle roll and pitch in the experiment. . . . 43
Figure 3.6 Illustration of SLIDER's right leg structure for solving inverse kinematics. . . . 43
Figure 3.7 Illustration that the hip roll axis doesn't intersect the leg. . . . 45
Figure 3.8 Sagittal plane view of common uses of compliant elements in walking robots. . . . 46
Figure 3.9 Coronal plane view showing the novel use of compliant elements in SLIDER. . . . 47
Figure 3.10 Experimental results of bungee cord length change with increasing applied force. . . . 49
Figure 3.11 SLIDER standing with two different torso heights. . . . 49
Figure 3.12 Trajectories of the vertical prismatic joint on the left leg in the experiment. The orange line indicates the measured position from the motor while the blue line indicates the desired position. (a) shows the slow motion with a frequency of 2.5 Hz and (b) shows the fast motion with a frequency of 5 Hz. . . . 53
Figure 3.13 Results of the Bayesian Optimization after 9 iterations. . . . 54
Figure 4.1 Workflow of preview control. . . . 59
Figure 4.2 The whole body controller can generate heel strike motion. . . . 64
Figure 4.3 The control diagram of SLIDER. . . . 64
Figure 4.4 SLIDER's contact sensors mounted on the foot. . . . 66
Figure 4.5 Contact sensor positions. . . . 66
Figure 4.6 SLIDER v4 walking in simulation and real-world. . . . 67
Figure 4.7 SLIDER v5 walking in simulation and real-world. . . . 68
Figure 4.8 The power of SLIDER v5 robot walking with forward velocity of 0.1m/s for 15 cycles. . . . 69
Figure 4.9 The energy consumption of joints in SLIDER v5. . . . 69
Figure 5.1 SLIDER robot walks on uneven terrain. . . . 72
Figure 5.2 The hierarchical controller with the execution frequency of each level. 74
Figure 5.3 The aSLIP model. . . . 75
Figure 5.4 The decoupled aSLIP model. . . . 76
Figure 5.5 SLIDER blindly walking on different kinds of terrains. . . . 82
Figure 5.6 SLIDER walks with compliant feet. . . . 83
Figure 5.7 SLIDER got pushed while walking on a wave field. . . . 85
Figure 5.8 Comparison between proposed aSLIP planner with LIP planner when walking on a 15° slope. . . . 86
Figure 6.1 SLIDER walks over slopes with ARTO-AL with perception. . . . 90
Figure 6.2 Illustration of the prediction horizon. . . . 92

Figure 6.3 Illustration of the terrain-aware footstep planner pipeline. 94
Figure 6.4 Flowchart describing the asynchronous optimization process. 98
Figure 6.5 3D illustration of constructing the plane that the CoM moves along from optimal footsteps. 98
Figure 6.6 The pipeline of generating height map from perception data. 100
Figure 6.7 The dimensions and joint configuration of SLIDER. 104
Figure 6.8 3D illustration of CoM (blue) and supporting foot position (red) when SLIDER is walking forward at 0.2 m/s with the planner ARTO-AL. . . 104
Figure 6.9 3D illustration of CoM (blue) and supporting foot position (red) when SLIDER is walking sideways at 0.3 m/s with ARTO-AL. 105
Figure 6.10 Push recovery comparison between No-Time-Adp and ARTO-AL. . . 105
Figure 6.11 Push recovery using ARTO-AL. 106
Figure 6.12 Maximum push recovery on flat ground with forces parallel to the transverse plane. 107
Figure 6.13 Maximum push recovery on inclined ground with the force vector parallel to the transverse plane. 108
Figure 6.14 Reaction step to a 70 N push when walking on a 10° slope with ARTO-AL. 108
Figure 6.15 The estimated height and slope angle from height map compared with the ground truth. 109
Figure 6.16 Plots of success rate when walking in three conditions. 109
Figure 7.1 Highly dynamic jump motions can be generated for both bipedal and quadrupedal robots using the LL-SRBM. Left: SLIDER robot. Right: ANYmal robot. 112
Figure 7.2 The planning and control framework. 114
Figure 7.3 The LL-SRBM model. 115
Figure 7.4 The shape of the centroidal inertia of the robot is affected by the leg configuration, which can be modelled by LL-SRBM. 117
Figure 7.5 Snapshots of the ANYmal robot performing a 90° twisting jump with inertia shaping and without inertia shaping. 124
Figure 7.6 The measured trajectories of ANYmal 90° twisting jump in simulation. 125
Figure 7.7 The norm of average spatial velocity when ANYmal performed a twisting jump. 126
Figure 7.8 Snapshots of ANYmal performing a twisting jump with 30°. 127
Figure 7.9 Snapshots of the ANYmal robot performing a forward jump of 30 cm with inertia shaping and without inertia shaping. 127
Figure 7.10 Snapshots of SLIDER performing a forward jump onto a 35 cm box using LL-SRBM. 128

Figure 8.1 Normal walking gait *vs* RotoGait, the difference lies in the swing leg trajectory. . . . 129
Figure 8.2 (a): the outrunner robot that applies the principle of the rimless wheel (OutRunner, 2021). (b) the miniRHex robot (Barragan et al., 2018). . . . 130
Figure 8.3 RotoGait kinematics represented in spherical coordinates. . . . 131
Figure 8.4 Illustration of swing leg dynamics of RotoGait. . . . 132
Figure 8.5 Footstep plan and CoM reference motion generation with LIP model. 136
Figure 8.6 Visualization of RotoGait with a forward step length of 0.4m. . . . 136
Figure 8.7 Visualization of swing leg trajectory with normal walking gait under different trajectory optimization constraints. . . . 137
Figure 8.8 RotoGait swing leg trajectory with maximum step length 1.38 m. . . . 138
Figure 8.9 Boundary of achievable step velocity given step length for RotoGait and normal walking gait. . . . 139
Figure 8.10 Comparison of energy consumption between the RotoGait and normal walking gait. . . . 140
Figure 8.11 Comparison of energy consumption between the RotoGait and normal walking gait on both step length and step velocity. . . . 141
Figure A.1 Linear Inverted Pendulum Model . . . 150
Figure A.2 Spring Loaded Inverted Pendulum . . . 151
Figure A.3 Variable Height Inverted Pendulum Model . . . 151
Figure A.4 Single Rigid Body Model . . . 152
Figure A.5 Lumped Leg Single Rigid Body Model . . . 153

LIST OF TABLES

Table 2.1 Overview of SLIDER and COMAN's mass distribution 19
Table 2.2 Major Versions of SLIDER 24
Table 2.3 Comparison of ServoBelt drive mechanism with other conventional prismatic joint transmission systems 27
Table 2.4 Overview of SLIDER v4's joint specifications 30
Table 2.5 Peak torque value at different walking speeds in Gazebo 33
Table 2.6 Torque Requirement for Hip Joints 33
Table 2.7 Motor Specification for SLIDER v4 34
Table 2.8 Motor Specification for SLIDER v5 34
Table 3.1 DH parameters for one side of the SLIDER v5 37
Table 3.2 Mathematical Notations for Inverse Kinematics Analysis 40
Table 3.3 Mathematical Notations for Inverse Kinematics 44
Table 3.4 Servo belt test results. 47
Table 3.5 Summary of The Experiment Results. 54
Table 4.1 Summary of Energy Consumption during SLIDER v5 Walking. 67
Table 7.1 Solve Times for Planners. 128
Table 8.1 Motor Limits for Hip Joints in SLIDER. 135

ACRONYMS

MPC	Model Predictive Control
WBC	Whole Body Controller
QP	Quadratic Programming
LIPM	Linear Inverted Pendulum Model
SLIP	Spring Loaded Inverted Pendulum
aSLIP	actuated Spring Loaded Inverted Pendulum
SRBM	Single Rigid Body Model
LL-SRBM	Lump Leg Single Rigid Body Model
CoM	Center of Mass
DoF	Degree of Freedom
AL	Augmented Lagrangian
LM	Lagrangian Multiplier
ROS	Robot Operating System
CoP	Center of Pressure
DH	Denavit–Hartenberg
URDF	Unified Robotic Description Format
IK	Inverse Kinematics
UCB	Upper Confidence Bound
GP	Gaussian Process
ZMP	Zero Moment Point
LQI	Linear Quadratic Integral
MBS	Multi-Body System
CCRBI	Centroidal Composite Rigid Body Inertia
NLP	Nonlinear Programming

Part I

INTRODUCTION

CHAPTER 1

Introduction

1.1 MOTIVATION

Robots have great potential to transform the world we live in by replacing humans in dangerous and tedious tasks. From autonomous cars to robot dogs doing inspections on sites that might be dangerous to people, more and more robots have entered human lives. Legged robotics is one of the particular areas of robotic research. One of the motivations for building legged robots is the search and rescue tasks in disaster scenarios where it is too risky to send a human. The terrain is often impossible or too hard for wheeled vehicles to traverse and drones cannot carry loads or manipulate objects as humans can. Being able to send robot systems can reduce the risk of humans and the number of fatalities.

Legs allow animals and robots to get access to areas that are inaccessible for wheeled vehicles such as stairs or extremely rough terrains. While there are some non-legged robots that are able to climb stairs, they are designed for a specific task by sacrificing other capabilities. Legged design is so powerful since it provides the solution to traverse different terrains without requiring specific designs. This is especially beneficial for robots operating in environments primarily designed for humans such as urban environments and households where discrete changes of terrain like stairs are very usual. Moreover, it is said that 24% of Earth's land mass is mountainous and one in ten people in the world live in mountainous regions (Daily, 2022). Legged robots are suitable to operate in these areas.

Some animals with legs have demonstrated remarkable mobility, as shown in Fig. 1.1. The mountain goats can climb rocky, nearly vertical cliffs without falling and dogs can catch very fast-moving objects like frisbee while coordinating the whole body motion. Thanks to the development of a novel actuation scheme (Katz, 2018, Katz et al., 2019, Dynamics, 2022a) and the recent advances in numerical optimization (Dai et al., 2014, Winkler et al., 2018, Mastalli et al., 2020a) and machine learning (Hwangbo et al., 2019, Lee et al., 2020), legged robots can perform high dynamic motions like parkour (Dynamics, 2022b) and walk on very rough terrains (Lee et al., 2020). However, unlike humans or animals, legged robots



(a) (b)

Figure 1.1: Dynamic Legged Animals. Legged animals show amazing abilities to move dynamically in unstructured environments and control their body during performing highly dynamic tasks. These abilities haven't been achieved by real robots yet. (a): Goats climb on near-straight cliffs (Geographic, 2016). (b) A dog is catching a frisbee (LOLWOT, 2021).

still can not dynamically move through unstructured environments autonomously. Moreover, legged robots especially bipedal robots that can perform highly dynamic motions are very expensive and the technology of building these robots is not fully published. The challenges that hinder legged robots from being largely deployed and rival the mobility with humans and animals include (1) little understanding of the physical system design that is essential for executing dynamic motions (2) real-time control algorithms that can handle the model complexity of legged robots such as high Degree of Freedom (DoF), underactuation and the nonlinear dynamics and (3) a motion planning pipeline that can simultaneously plan robot dynamics and contact sequences with perceptive information.

This thesis aims to make contributions that address some of the critical points of these challenges. In particular, this thesis starts with physical system design and combines real-time control and motion planning together with the design.

1.2 RELATED WORK

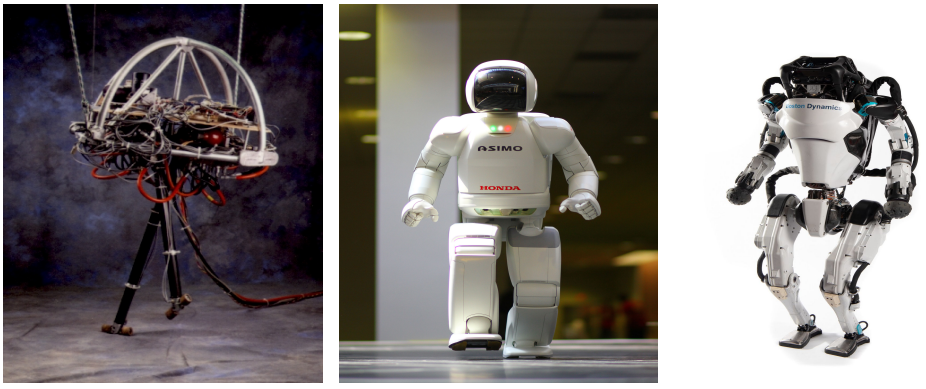
Although humans tried to build a machine that looked and moved like themselves a long time ago, it is until late 1980s a bipedal robot that could walk and balance itself started to appear. The area of dynamic locomotion has progressed rapidly thanks to the enormous amount of research on robot system design, motion control and motion planning. This section will walk through the development of legged robots, especially bipedal robots, and a rough categorization is presented.

1.2.1 *Robot Design*

One of the earliest examples are the 3D Biped from the MIT leg laboratory led by Marc Raibert, as seen in Fig. 1.2a. The robot has a very simple structure and is pneumatically actuated. It can walk, run and even somersault (Playter et al., 1992). ASIMO from Honda in 1990s is one of the first humanoids that can autonomously walk, run, climb stairs and interact with the world. ASIMO has a friendly looking and good balancing capability, as shown in Fig. 1.2b. Atlas is the most advanced humanoid robot to date, and it is made by Boston Dynamics that is a spin-off company from MIT. The robot has a height of 1.5 m and a weight of 89 kg, as shown in Fig. 1.2c. Its compact hydraulic actuation units provide high power to enable Atlas to do highly dynamic motions like parkour, and it can run at 2.5 m/s. However, due to the high complexity of hydraulic actuators, the robot is very expensive and the details of the technology are not published. Unlike Atlas which is heavy and hydraulically actuated, both the Schaft robot made by Google and the Cassie robot made by Agility Robotics are lightweight and driven by electric motors. The Schaft robot is probably the first kneeless bipedal robot to be demonstrated in the real world. It was made by a startup company from Japan which was later acquired by Google. Schaft, as seen in Fig. 1.2d, has prismatic hip joints and straight legs which is an inspiration for SLIDER's design. Although the Schaft robot demonstrated capability of uneven terrain walking and push recovery (Schaft, 2016), no design details or control algorithms have ever been published. Inspired by the animal ostrich, the Cassie robot has slim and lightweight legs, as seen in Fig. 1.2e. Cassie has springs mounted on the legs and its dynamics is much like the Spring Loaded Inverted Pendulum (SLIP), making the robot energy efficient and robust to unexpected disturbances.

1.2.2 *Whole Body Control*

The Whole Body Controller (WBC) regulates the instantaneous system dynamics around a reference trajectory computed by a higher-level motion planner. The reference trajectory contains variables of the robot in the operational space, such as Center of Mass (CoM), angular momentum and the end-effector. The advantage of WBC is that it can prioritize different tasks in a hierarchical manner that the most important ones, such as balancing, can be carried out with guarantees. The secondary ones that exploit the system redundancy, such as placing the arm, can be executed if they do not conflict with the most important ones (Feng et al., 2015, Kuindersma et al., 2016, Herzog et al., 2015, Kim et al., 2018). The WBC can be formulated as Quadratic Programming (QP) and solved very efficiently. In (Herzog et al., 2015), the robot dynamics is divided into the underactuated part and the actuated part and only controlling the underactuated part is considered. The task priority is implemented by assigning different weights for tasks. In (Kim et al., 2018) the dynamics is also separated



(a) (b) (c)

(d) (e)

Figure 1.2: Bipedal robots. (a): The 3D biped from the MIT leg laboratory (Lab, 1995). (b): ASIMO robot from Honda. (c): Atlas robot from Boston dynamics (Dynamics, 2021). (d) Schaft robot from Google (Schaft, 2016). (e): Cassie robot from Agility Robotics (Robotics, 2017b).

but the task priority is ensured with priority projections. WBC plays an important role in the control of SLIDER, because it acts as a middle layer connecting the plan generated by reduced-order models and the robot. In chapter 4 a WBC is presented to control the SLIDER robot.

1.2.3 *Reduced-order Model*

The reduced-order models for locomotion have been widely used to reduce the complexity. One of the most widely used reduced-order models is Linear Inverted Pendulum Model (LIPM). It was first introduced for bipedal locomotion in (Kajita et al., 2001a) and has been used in many humanoid robots since then. Its popularity is based on the simplicity and the coupling with MPC. (Pratt et al., 2006) reformulated the LIPM and proposed the capture point. It is defined as a point where the robot takes one step and is able to stop. Another early example is from the MIT Leg Lab where they demonstrated hopping and acrobatic biped traversing over different terrains without large optimization nor perception (Raibert, 1986). With simple but well informed heuristics, the robot was able to walk, run and even somersault. SLIP model, due to its compliance and similarity to the dynamics of human running and walking (Geyer et al., 2006), has been widely used in robot running (Arslan et al., 2011). Robots designed for energy efficiency (Hubicki et al., 2016) also use SLIP as the model to control the robot. More recently, the Single Rigid Body Model (SRBM) has gained popularity to control quadruped robots (Di Carlo et al., 2018). This model has been preferred since it well resembles the dynamics of quadruped robots and with the assumption of small roll and pitch angles, the dynamics of SRBM can be linearized. Therefore, the robot control problem can be formulated as a convex optimization problem.

1.2.4 *Model Predictive Control*

MPC attempts to solve an optimization problem in a receding planning horizon with some objectives such as tracking a reference trajectory. MPC has made a lot of success on bipedal (Powell et al., 2015, Xiong et al., 2018) and quadrupedal robots (Di Carlo et al., 2018, Ding et al., 2019, Aceituno-Cabezas et al., 2017). In (Kamioka et al., 2017, Xin et al., 2019, Kim et al., 2020), researchers showed bipedal robots resisting pushes by adapting their footsteps with MPC.

However, many of MPC implementations rely on solving strictly convex optimization problems on linear systems. Although this kind of problems can be solved fast and efficient, they are limited in the representation power. The nonlinear terms in the dynamics are not considered and therefore the controller is not informed about some important coupled

dynamics. Recently nonlinear MPC has gained more popularity. In (Naveau et al., 2016) a nonlinear MPC has been applied to generate reactive gaits that can adapt foot location and orientation and avoid convex obstacles. Results shown on the ANYmal quadruped robot demonstrate a whole-body MPC, but the limitation of the approach lies in the too many technical details of speeding up the optimization and no guarantees on algorithm convergence (Neunert et al., 2018). Chapter 5 introduces a MPC with linear dynamics to solve the problem of blind uneven terrain walking. Chapter 6 presents a MPC with nonlinear dynamics to adapt both step timing and step locations with terrain-awareness of bipedal robots.

1.2.5 *Nonlinear Trajectory Optimization*

Nonlinear optimization based controllers are more powerful than linear as they can consider complex and coupled nonlinear dynamics and result in a better solution. Trajectory optimization is a powerful framework for non-linear optimization. Different from MPC, trajectory optimization does not have the requirement of being real-time and therefore can handle more complex dynamics and nonlinear constraints. The decision variables are robot states and control inputs; the optimization constraints enforce physics law (system dynamics, kinematic constraints, actuator limits, etc) and collision avoidance. The objectives define desirable motions and sometimes penalize undesirable motions. There are many ways to formulate a trajectory optimization problem (Kelly, 2017, Li et al., n.d., Mastalli et al., 2020a, Posa et al., 2014, Manchester et al., 2019) such as single shooting, multiple shooting, direct collocation, differential dynamical programming and through contact methods. Trajectory optimization has been widely used in bipedal and quadrupedal robots to generate reference trajectories of highly dynamic motions. In (Dai et al., 2014) the author used trajectory optimization with centroidal dynamics to generate running, jumping and monkey bar motions for the Atlas robot in simulation. In (Mastalli et al., 2020b) the trajectory optimization considered whole body dynamics and generated athletic behaviors for the humanoid robot TALOS in simulation. Despite many advantages, one issue that hinders trajectory optimization from being used directly on real robots is the computation speed. Typically, trajectory optimization generates the reference trajectory and MPC tracks the trajectory. Chapter 7 presents such a framework to realize highly dynamic jumps on the robot ANYmal.

1.2.6 *Machine Learning*

Model-based optimization methods have limitations in that they have difficulties solving nonlinear optimizations in real-time. Also, some parts of dynamics are hard or impossible

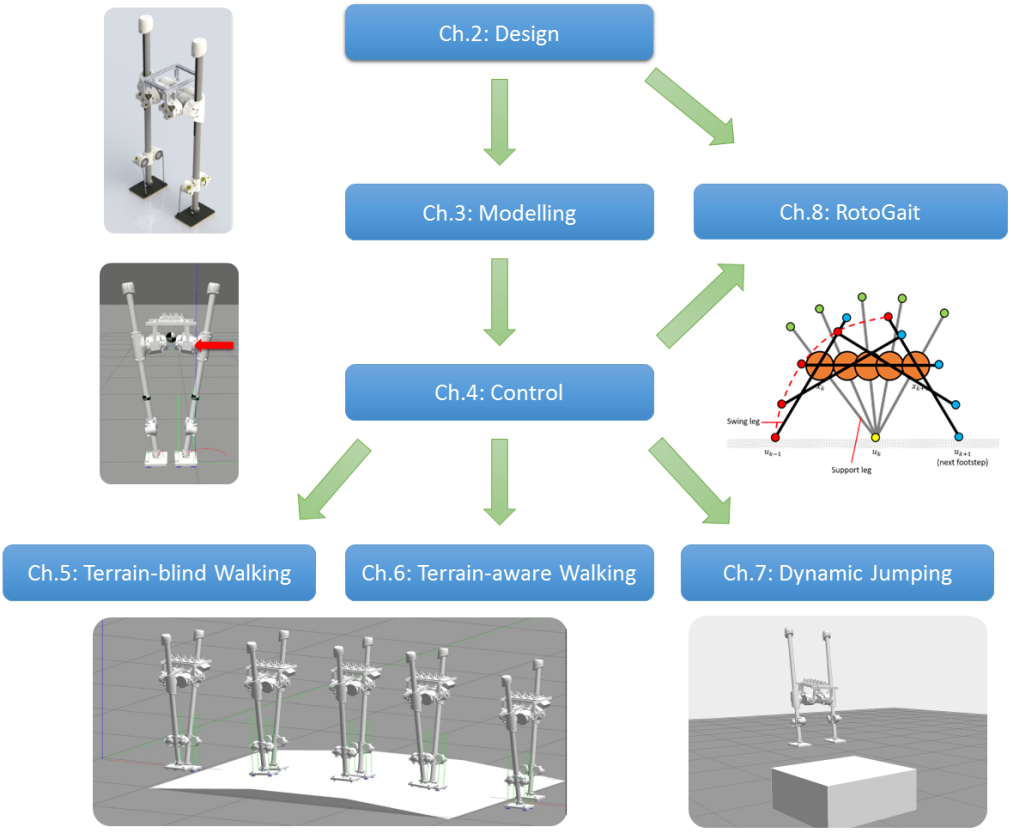


Figure 1.3: Thesis roadmap. Each block represents a chapter. The thesis overall is organized in a bottom-up structure. It starts with robot design and hardware implementation (Chapter 2), then robot modelling and WBC (Chapter 3, 4). The latter part of the thesis focuses on footstep planning (Chapter 5, 6), jump motion planning (Chapter 7) and the trajectory generation for the novel RotoGait (Chapter 8).

to model such as terrain softness. Machine learning on the other hand, can use data to build approximate models and does inference very fast. The early successful examples of applying machine learning in robotics happened in the specific area of robotics such as aerial robotics (Abbeel et al., 2004), robot manipulation (Levine et al., 2016, Levine et al., 2018) and robot locomotion (Fukuda et al., 1997). However, its applications are limited because of the lack of available data and computation power. Thanks to the fast development of deep learning, machine learning based approaches get popularity recently. There are different ways of applying machine learning to control legged robots. The work (Heess et al., 2017) from DeeMind used end-to-end reinforcement learning to make different legged robots to move in complex environments in simulation. But the motions are not realistic enough to be applied to real robots. In (Hwangbo et al., 2019) a real ANYmal robot learned to stand up

from a fallen position and walked forward with another policy. The robot could also resist pushes. The policies were learned in simulation and transferred to the real robot. The key idea is to train the neural network with realistic data collected from the real robot system. In (Xie et al., 2019), the author took a different approach by firstly imitating policies generated by model-based approaches and then improving policies using reinforcement learning. The researchers repeated this process in a iterative way until the Cassie bipedal robot could walk stably.

The machine learning approaches have shown great promise in locomotion control. However, the fundamental challenges for neural networks are designing a good reward function and tuning the parameters in an automatic way. Moreover, there are no safety guarantees in machine learning as imposing constraints in MPC. The neural network based controllers may fail at unexpected situations and cause damage to the robot. Therefore a deeper understanding of neural networks is needed. A good approach is to combine learning-based approaches with model-based controllers, as demonstrated in (Yang et al., 2021b). Either way, it is acknowledged that there is no silver bullet to solve the legged locomotion problem directly. The thesis is intended to propose approaches to improve the current state of bipedal robot abilities.

1.3 THESIS CONTRIBUTIONS

This thesis contributes to the advancements of bipedal locomotion, with specific aspects including bipedal robot design, model predictive control for footstep planning, motion planning for highly dynamic jumping motions and a novel gait – RotoGait. The detailed contributions are summarized as:

- **Design and building of the knee-less bipedal walking robot SLIDER.** A novel design of a bipedal robot that has knee-less legs. With this design, the legs are ultra-lightweight and the energy consumption can be reduced since the robot does not need to walk with bent knees.
- **Footstep planners that enable bipedal robots to walk on uneven terrains blindly and adapt step timing and locations in real-time with terrain-awareness.** Two footstep planners are proposed to improve the robustness of bipedal walking. The first footstep planner decouples the aSLIP model and formulates the footstep planning as a linear MPC, enabling robots to walk over uneven terrains blindly. The second footstep planner simultaneously adapts footstep locations and timing by solving a nonlinear MPC. This MPC can be solved in 200 Hz utilizing analytical gradients in an asynchronous framework. With terrain-awareness, the planner gives the robot the ability to walk on uneven terrains with footstep locations and timing adaptation.

- **Motion planner that generates highly dynamic jumps.** A novel framework of generating highly dynamic jumps for both bipedal and quadrupedal robots. By considering the leg dynamics, our proposed LL-SRBM can model the centroidal inertia of the robot with different leg configurations and therefore improves jumping performance in terms of speed and distance.
- **A novel walking gait: RotoGait.** The proposed novel gait rotates the leg continuously instead of moving the leg forward and backward when walking. RotoGait shows promise for walking faster and being more energy efficient under the condition of fast speed.

1.3.1 *Contributions of other people*

The CAD modelling and physical parts manufacturing for SLIDER were entirely done by other people and I would like to give them explicitly credit for this: Aksat Shah, Oscar Gill, Matt Cocking, Joseph Shore, David Marsh, Akshay Raut, Stergio Chikaros, Zhonghe Jiang, Hengyi Fei, Varan Gupta, Jacopo Zhaoyang Hu, Peter Tisnikar, Tianhu Peng, Yuki Cheng and Thomas Godden. The following students contributed to the simulation experiments of RotoGait: Zac Galibov and Julia Péladeau.

1.4 THESIS OUTLINE

The widely used framework for legged robot control is a hierarchical structure consisting of high-level footstep planning/motion planning, mid-level whole body control and low-level joint control of the robot. This thesis is composed of eight chapters covering all levels of this hierarchical structure (see Fig. 1.3 for a visual overview). The systematic overview of all chapters can be seen at Fig. 1.4:

- **Chapter 2** describes the detailed design of the knee-less bipedal robot SLIDER. The design goals and desired features of SLIDER are firstly introduced. Then the comparison between SLIDER's design and the anthropomorphic design is presented. The history of the design of SLIDER, including the description of each iteration and the advantages and disadvantages of these iterations are discussed. Following this, the mechanism used for each joint and details of actuators, electronics and the software used in SLIDER are introduced. At the last part of this chapter, the lessons I learned during the process of designing and building multiple iterations of SLIDER are summarized. This chapter includes the contents of the paper (Wang et al., 2018, Wang et al., 2020a).

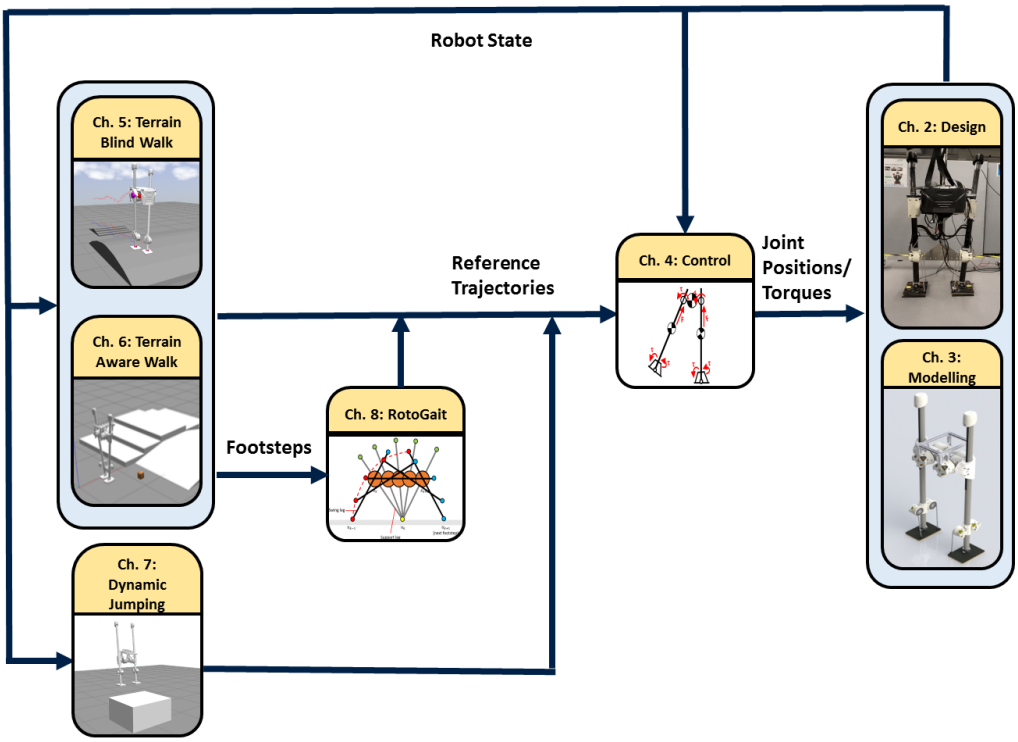


Figure 1.4: The chapters of the thesis can also be viewed as a robot system. The high-level plans such as footsteps (Chapter 5, 6) or jumping motions (Chapter 7) are given as reference trajectories to the WBC (Chapter 4). The footstep plans can also be used as inputs to RotoGait (Chapter 8) to generate the reference trajectories for the novel gait. The WBC computes instantaneous torque or joint position for each joint and the commands are sent to the robot (Chapter 2, 3).

- **Chapter 3** describes the modelling of SLIDER including the Denavit–Hartenberg (DH) parameters, parallel four-bar-linkage ankle parameters, the inverse kinematics of the ankle and the inverse kinematics of the whole robot. Then using parallel elastic actuators to improve the energy efficiency of SLIDER is discussed. This chapter includes the contents of the paper (Wang et al., 2018, Wang et al., 2020a, Wang et al., 2022).
- **Chapter 4** describes the mid-level and low-level controllers for controlling SLIDER. This includes the preview control that is used to generate walking trajectories for the real SLIDER robot, the inverse-dynamics based whole body controller used in simulation. At the final part of this chapter the experimental setup and results of SLIDER walking are presented, both in simulation and in real-world. This chapter includes the contents of the paper (Wang et al., 2020a).

- **Chapter 5** introduces a novel footstep planning algorithm that enables a bipedal robot to blindly walk over uneven terrains. In this chapter the dynamic of the decoupled aSLIP model is derived at first. Then the formulations of the footstep planing on x,y and z directions are presented. At the last part of the chapter the simulation results of SLIDER walking on different kinds of uneven terrains including a 15° slope, wave fields and stairs are shown. The result of push recovery experiments while SLIDER is walking on a slope is presented and the comparison between the proposed planner and LIP planner is shown. This chapter includes the contents of the paper (Wang et al., 2021a).
- **Chapter 6** introduces a terrain-aware footstep planner that can adapts step locations and timing in real-time based on the LIPM. In the chapter the analytical equations of optimizing locations and timing based on the LIPM are firstly derived and then z direction walking with the proposed planner is proved under the parallel plane assumption. The description of how to construct the 3D plane based on the height map generated by the perception is shown. Following up this the MPC formulation and the detailed description of solving the optimization with Augmented Lagrangian (AL) and analytical gradients are presented. The last part describes the simulation results of flat ground walking, push recovery on flat ground, uneven terrain walking, push recovery on uneven terrain walking and the performance comparison between different optimization methods. This chapter includes the contents of the paper (Chappell et al., 2020).
- **Chapter 7** describes a novel motion planning framework that improves the jumping performance by proposing a unified model for planning highly dynamic jumps. This model can approximate the centroidal inertia by using a single mass model for each leg. In this chapter the LL-SRBM and derive its dynamics are firstly introduced. Then this chapter describes the trajectory optimization formulation that generates the jumping trajectory. A contact detection method based on the average spatial velocity is introduced. At last the simulation and real robot experiments on SLIDER and the quadruped robot ANYmal are shown. These experiments include forward jump, twisting jump, jump onto a box and disturbance rejection when landing. This chapter includes the contents of the paper (Wang et al., 2021b).
- **Chapter 8** introduces a novel gait: RotoGait. In RotoGait the swing leg can rotate continuously instead of decelerating and reversing the direction as the normal gait. In this chapter the kinematics and dynamics of this novel gait are derived at first. Then the chapter describes the trajectory optimization formulation to generate this gait based on minimizing the energy consumption and visualize the generated motion. At last the comparison between RotoGait and the normal gait in terms of energy consumption with different step lengths and forward velocities is presented.

- **Chapter 9** concludes the topics discussed in this thesis and proposes several future directions related to these topics.
- **Appendix A.1** summarizes all the reduced-order model dynamics used throughout the thesis and a clear difference can be seen with a comparison.
- **Appendix A.2** provides the detailed formulation of analytical gradients of LIPM with respect to step location and timing used in Chapter 6.
- **Appendix A.3** provides the formulations of quaternion to rotation matrix transformation, derivative of quaternion and inertia tensor used in Chapter 7.

1.5 AUTHOR PUBLICATIONS

This thesis relates to work that is also publicly available in the following articles, which have been previously published:

- **Ke Wang** Aksat Shah and Petar Kormushev.
  "Slider: A Bipedal Robot with Knee-less Legs and Vertical Hip Sliding Motion."
  In International Conference Series on Climbing and Walking Robots and the Support Technologies for Mobile Machines, 2018. (Wang et al., 2018)

- **Ke Wang**, David Marsh, Roni Permana Saputra, Digby Chappell, Zhonghe Jiang, Akshay Raut, Bethany Kon and Petar Kormushev.
  "Design and Control of SLIDER: An Ultra-lightweight, Knee-less, Low-cost Bipedal Walking Robot."
  In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. (Wang et al., 2020b)
  Video: https://youtu.be/E6A5XPiC5yk

- **Ke Wang**, Roni Permana Saputra, James Paul Foster and Petar Kormushev.
  "Improved Energy Efficiency via Parallel Elastic Elements for The Straight-legged Vertically-compliant Robot SLIDER."
  In International Conference Series on Climbing and Walking Robots and the Support Technologies for Mobile Machines (**Best Paper Award for Industrial Innovation**), 2021. (Wang et al., 2022)
  Video: https://youtu.be/ZTaG9_-Dz8A

- **Ke Wang**, Hengyi Fei and Petar Kormushev.
  "Fast Online Optimization for Terrain-Blind Bipedal Robot Walking with a Decoupled Actuated SLIP Model."
  In Frontiers in Robotics and AI, 2022. (Wang et al., 2021a)
  Video: https://youtu.be/ROyV-ZP8dxA

and already submitted:

- **Ke Wang**, Guiyang Xin, Songyan Xin, Michael Mistry, Sethu Vijayakumar and Petar Kormushev.
  "A Unified Model with Inertia Shaping for Highly Dynamic Jumps of Legged Robots."
  arXiv:2109.04581, 2021. (Wang et al., 2021b)
  Video: https://youtu.be/M0HDgvkAVhM

- **Ke Wang***, Digby Chappell* and Petar Kormushev (*Equal Contribution).
  "Asynchronous Real-Time Optimization of Footstep Placement and Timing in Bipedal Walking Robots."
  arXiv:2007.00385, 2021. (Chappell et al., 2020)
  Video: https://www.youtube.com/watch?v=1h0FE9YPqbo
  Author contributions: KW and DC developed the main idea, wrote the code, did the simulation experiment and wrote the manuscript together. PK supervised the project.

This thesis also contains materials from a previous published poster:

- **Ke Wang**, Aksat Shah and Petar Kormushev.
  "SLIDER: A Novel Bipedal Walking Robot without Knees"
  In 19th International Conference Towards Autonomous Robotic Systems (TAROS), Bristol, UK. (**Best Poster Award**), 2018.

Other publications have been produced during the PhD, but are not included in this dissertation as they follow a different research direction, focused on mobile robots and reinforcement learning. The author contributed to developing control methodologies through discussions, as well as towards experiment design for validating the proposed approaches. This work is presented in the following articles which have been previously published:

- Chaochao Lu, Biwei Huang, **Ke Wang**, José Miguel Hernández-Lobato, Kun Zhang and Bernhard Schölkopf.
  "Sample-efficient Reinforcement Learning via Counterfactual-based Data Augmentation"
  In Proceedings of Neural Information Processing Systems Workshop on Offline Reinforcement Learning, 2020. (Lu et al., 2020)
  Author contribution: KW helped with developing the idea and doing experiments on control tasks, wrote the control part in the manuscript.

- Fabian Falck, Sagar Doshi, Marion Tormento, Gor Nersisyan, Nico Smuts, John Lingi, Kim Rants, Roni Permana Saputra, **Ke Wang** and Petar Kormushev.
  "Robot DE NIRO: A Human-centered, Autonomous, Mobile Research Platform for Cognitively-enhanced Manipulation"
  In Frontiers in Robotics and AI, 2020. (Falck et al., 2020)
  Author contribution: KW helped with developing the control part and doing experiments.

- Roni Permana Saputra, Nemanja Rakicevic, Digby Chappell, **Ke Wang** and Petar Kormushev.
  "Hierarchical Decomposed-objective Model Predictive Control for Autonomous Casualty Extraction"
  In IEEE Access, 2020. (Saputra et al., 2021)
  Author contribution: KW helped with developing the idea.

Part II

DESIGN AND MODELLING OF SLIDER

CHAPTER 2

Design of SLIDER

The SLIDER robot is a novel knee-less bipedal robot and there was no detailed public resource about this kind of design before. The design of SLIDER has gone through many iterations. In this chapter, I discuss about the detailed design of the SLIDER robot that includes:

- the overview of SLIDER's design, the features of SLIDER and the design goal
- the comparison of the knee-less design with the anthropomorphic design
- the different iterations of SLIDER's design and what are their advantages and disadvantages
- the mechanism used for each joint of SLIDER and their iterations
- actuators, electronics and software
- the lessons I learned from multiple iterations of SLIDER's design

2.1 OVERVIEW OF SLIDER DESIGN

Fig. 2.1 shows the design of the SLIDER robot: a knee-less bipedal walking robot design comprising of a pair of feet, legs, and main body frame. This robot has 10 joints with 5 DoF at each leg (hip roll, hip pitch, hip slide, ankle pitch and ankle roll) and has been designed to be ultra-lightweight and low-cost, but suitable as a bipedal locomotion research platform.

SLIDER's most distinctive feature, its prismatic "slide" leg actuation, imposes both inherent benefits and constraints onto its design when compared to more traditional humanoid robots. Its most notable benefit is the mass and leg complexity reduction attained because there is no knee. However, the hip is burdened with another actuator, complicating the design. To take advantage of the benefits of a knee-less design (lower leg mass) and better approximate an inverted pendulum model, SLIDER's primary design goals are: 1) lightweight and compact joints; 2) concentration of mass as close to the main body frame as

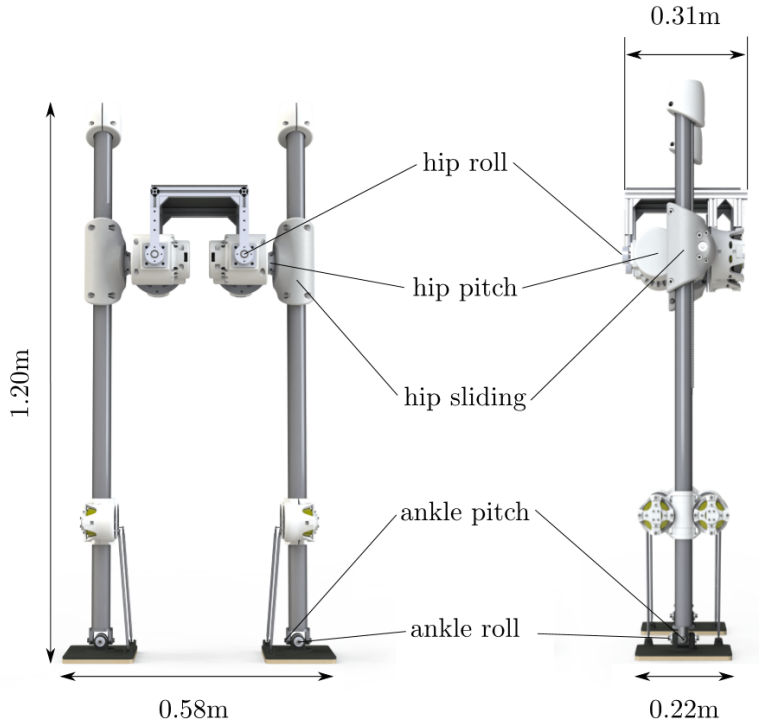


Figure 2.1: A model of SLIDER robot. It has 10 DoF and straight legs made of carbon fiber reinforced polymer. It is designed to be ultra-lightweight and suitable for agile locomotion.

possible; 3) low development cost and 4) easily modifiable design to allow the rapid addition of new sensors and mechanical improvements.

2.2 SLIDER'S DESIGN VS ANTHROPOMORPHIC DESIGN

2.2.1 *Mass Distribution*

The conventional anthropomorphic leg design has a 3-1-2 DoF distribution (3 DoF in hip, 1 DoF in knee, 2 DoF in ankle).

SLIDER's non-anthropomorphic design has a 4-0-2 DoF distribution, with 4 DoF in the hip (hip roll, hip pitch, hip yaw, and prismatic hip "slide"), and no knee.[1] This significantly changes the mass distribution of SLIDER, making the legs lighter, and shifting the CoM higher. SLIDER's mass (using off-board computer and power supply) and height are 15.0 kg and 1.2 m, respectively. Fig. 2.2 visualises the comparison of the mass distribution between SLIDER and COMAN robot (an anthropomorphic bipedal robot). It can be seen from the Fig. 2.2 that, in COMAN, a considerable amount of the weight is distributed at each leg and also

1 Please note that the hip yaw DoF is not yet implemented in the current version of SLIDER.

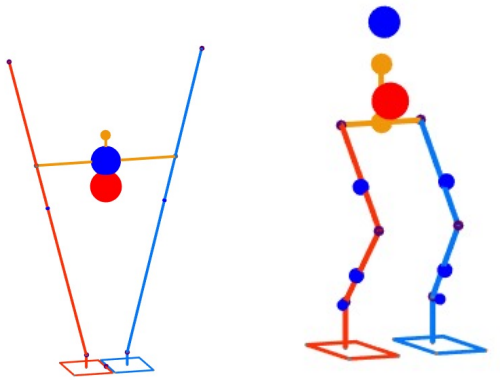


Figure 2.2: Illustration of SLIDER's mass distribution in comparison to an anthropomorphic bipedal robot, COMAN. The left robot is SLIDER and the right robot is coman. The size of spheres correspond to the mass at that specific links. The red sphere represents the CoM position.

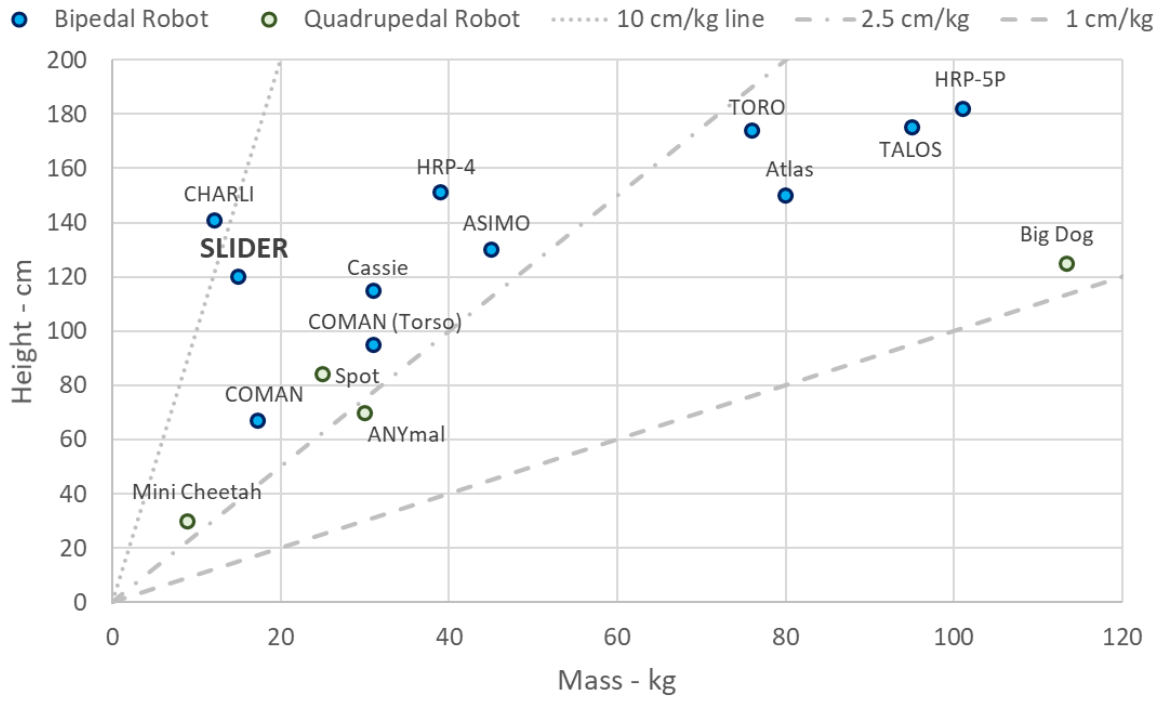


Figure 2.3: Height-mass relationship of selected robots.

Table 2.1: Overview of SLIDER and COMAN's mass distribution

	Pelvis	Hip	Upper Leg	Lower Leg	Ankle	Foot
SLIDER*	-	5.8 Kg**	0.6 Kg		1.2 Kg	0.4 Kg
COMAN*	5.5 Kg	1.2 Kg	1.7 Kg	1.4 Kg	0.7 Kg	0.9 Kg

* Each mass given is for one leg only.
** The mass of the leg sliding case is included in the mass of the hip.

at its pelvis. Meanwhile, in SLIDER's design, almost all the weight is concentrated at the main frame while a small amount of the weight is distributed at the robot legs. This mass distribution is one of the most distinctive features of the SLIDER robot design, in comparison to the conventional anthropomorphic designs. Table 2.1 presents the mass distribution at each link of SLIDER and COMAN. The table highlights SLIDER's lightweight leg design, showing that one of objectives has been achieved; ultra-lightweight bipedal robot design.

The mass distribution also affects the CoM motion. SLIDER and COMAN were both simulated walking with identical gait pattern generation and control strategies, as can be seen from Fig. 2.4. Fig. 2.5 shows that SLIDER's CoM moves less - it has both reduced sway, and remains closer to being at a constant height. This indicates that SLIDER's lightweight legs affects CoM motion less than COMAN's legs.

As seen in Fig. 2.3, further investigation of the weight comparison between SLIDER and other existing legged robots also highlights the fact that the SLIDER robot exhibits an ultra-lightweight design. Compared to other legged robots reported in the literature, SLIDER's height-mass ratio is one of the highest ratio, at approximately 6.67 [cm/kg].

2.2.2 *Workspace Comparison*

Regarding workspace, SLIDER benefits from its lack of knees in that it can navigate large obstacles or steep stairs, and larger steps are significantly easier to achieve. However, the prismatic leg means that clearance above the robot is required, meaning overhead obstacles are unsuitable for navigation (see Fig. 2.6). In relation to potential applications, this makes SLIDER well suited to agile locomotion and walking on uneven terrains.

2.2.3 *Energy Efficiency*

For the anthropomorphic design the bent knee is common, this is due to the singularity of inverse kinematics. Some efforts have been made to reduce the effect of bent knees (Griffin et al., 2018, Henze et al., 2019), but the bent knee effect cannot be fully eliminated while

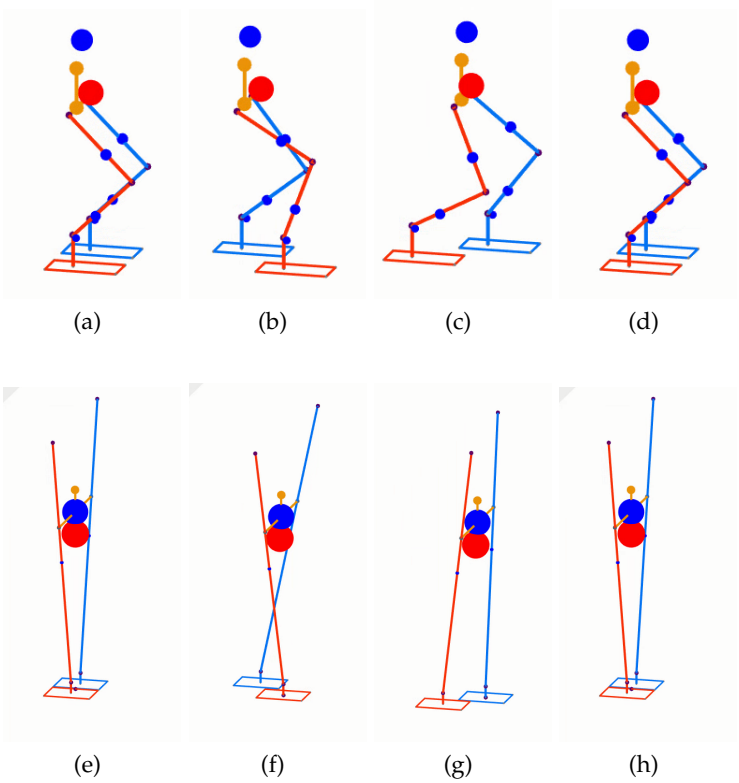
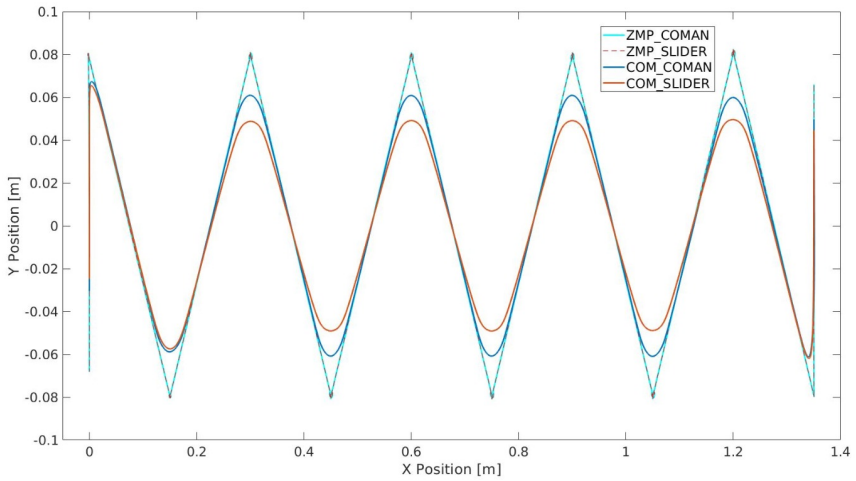
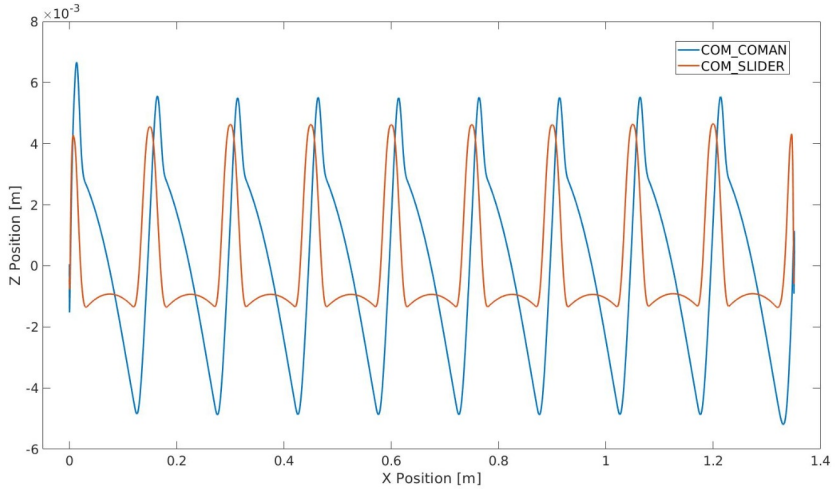


Figure 2.4: SLIDER and COMAN are walking with identical gait patterns. Blue balls represent individual link masses, the size of the ball matches the quantity of mass. The red ball represents the CoM of each robot.



(a)



(b)

Figure 2.5: Comparison of the center of mass (CoM) motion between SLIDER and COMAN with the identical gait execution. (a): x-y plane. (b) x-z plane.

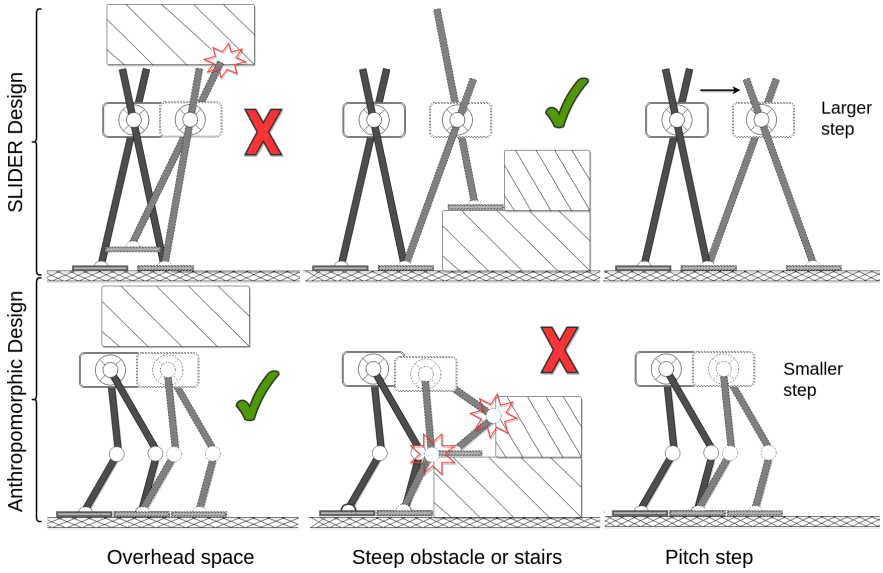


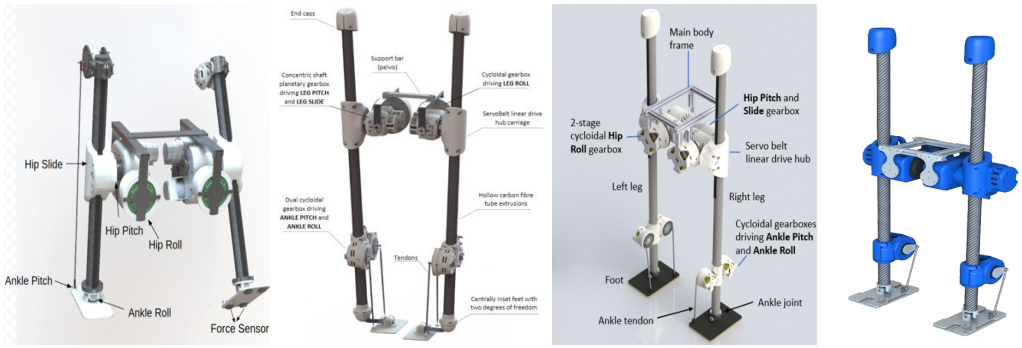
Figure 2.6: SLIDER's design vs Anthropomorphic design in workspace.

walking. The actuator have to generate more torque for the bent knee and significantly proportion of leg mass is due to the knee joint. Therefore, the anthropomorphic design consumes excess power with bent knee compared with the knee-less design.

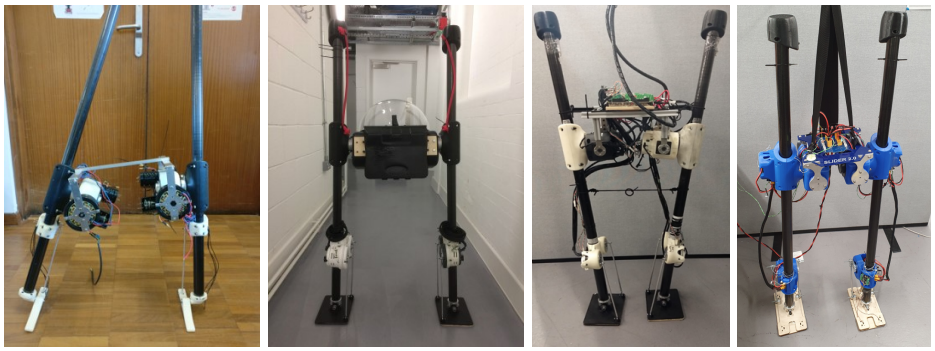
2.3 THE DESIGN HISTORY OF SLIDER

SLIDER has gone through 5 main iterations of designs, as can be seen from Fig. 2.7. Table 2.2 summarizes the main features of different versions. Fig. 2.7a to Fig. 2.7d clearly shows the evolution between different iterations. In the first iteration of design the robot was shorter than other iterations and the ankle actuation was placed on the top of the leg. The initial idea was to reduce the weight of lower part of the leg. However the rod connecting the ankle actuation and the ankle joint was too long and was easy to bent. The first iteration was not built into a prototype. In the second iteration the ankle actuation was placed much closer to the ankle joints and two ankle motors were placed symmetrically front and back. Furthermore, there was an inner offset between the leg and the foot to reduce the hip roll angles. The prototype for this iteration is Fig. 2.7e.

In the third iteration, the inner offset between the leg and the foot was removed and the base frame was strengthened to prevent the potential yaw rotation of the robot. There are two reasons why the inner offset is removed: (1) the ankle pitch and ankle roll motion was causing the undesired yaw motion because of the tilted offset link. (2) the bottom of the offset link was hitting the floor on large ankle roll rotation. The prototypes for this iteration are Fig. 2.7f and Fig. 2.7g. In the newest design iteration, the hip to hip distance is reduced



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 2.7: Different iterations of SLIDER design. The design evolves from left to right in time order. (a)-(d) are CAD designs and (e)-(h) are real robot prototypes. (e) corresponds the design of (b), which has a one DoF design of ankle and ankle offset. (f) corresponds the design of (c), where has two DoF ankle and not offset. The hip roll transmission is the improved cycloidal gearbox in which the gears inside are made of aluminum and the frame is strengthened. (g) also corresponds with (c) and the hip pitch and hip slide transmissions are improved with herringbone gears. (h) corresponds with (d).

Table 2.2: Major Versions of SLIDER

Version	Year	Features	Figure
v1	2017	It was a conceptual model with a height of 96 cm and width of 50cm, hip slide and pitch transmissions are spur gears, hip roll is 2-stage cycloidal, ankle is 1-stage cycloidal and mounted on top of the leg	Fig 2.7a
v2	2018	First iteration of the real robot, A height of 120 cm and width of 50cm, has ankle offsets to reduce lateral swing and ankle is placed on lower part of the leg with parallel bar mechanism	Fig. 2.7b and 2.7e
v3	2019	Second iteration of real robot, A height of 120 cm and width of 58cm, frame is strengthened with 4-bar structure and ankle offset is removed	Fig. 2.7c and 2.7f
v4	2019	Third iteration of real robot, A height of 120 cm and width of 51cm, the hip slide transmission is improved with herringbone gears, the hip roll backlash is reduced with more precise aluminum cycloidal gear, the first version that can walk successfully	Fig. 2.7g
v5	2021	Fourth iteration of real robot, A height of 120 cm and width of 54cm, all joints are made with integrated t-motors instead of custom-made transmissions and brushless motors to reduce backlash and friction, has much less components and easier to assemble and disassemble, the second version that can walk successfully	Fig. 2.7d and 2.7h

and both ankle actuators are placed backwards in parallel. This results in a more compact ankle design. The prototype for this iteration is Fig. 2.7h.

2.4 JOINT MECHANISM

Different from the overall design of SLIDER, the joint mechanism has gone through two major iterations, namely v4 and v5. In the first iteration both design and manufacturing were done in house. Thanks to the fast development of integrated electric motors (Katz, 2018), in the second iteration the integrated electric motors manufactured by T-Motor (LLC, 2021) were used. This results a lighter, more compact robot with less backlash.

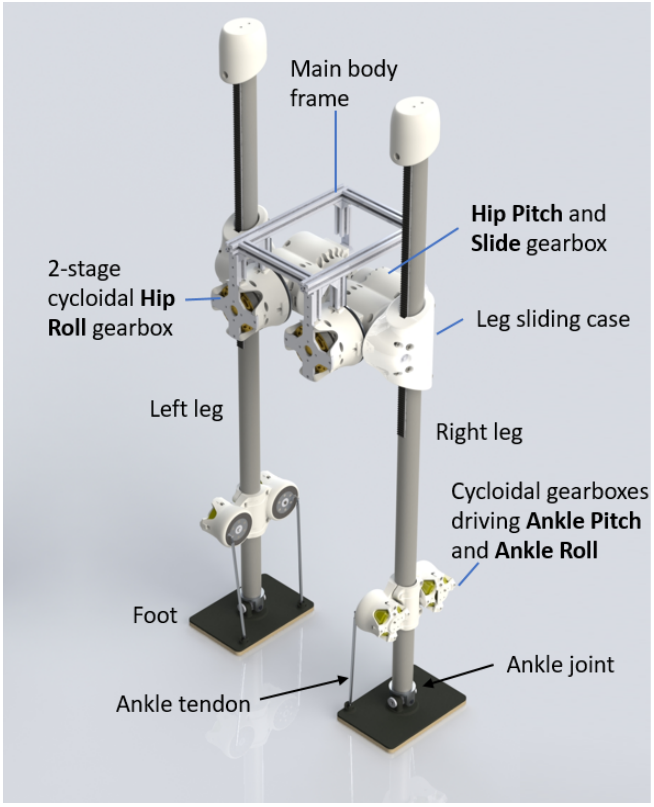


Figure 2.8: A model of the SLIDER v4 of joint mechanisms.

2.4.1 *SLIDER v4*

Hip Pitch and Slide

The hip pitch and slide motions are generated in a single gearbox (aka P&S gearbox), composed of an inner shaft (slide) and outer shaft (pitch) as shown in Fig. 2.9.

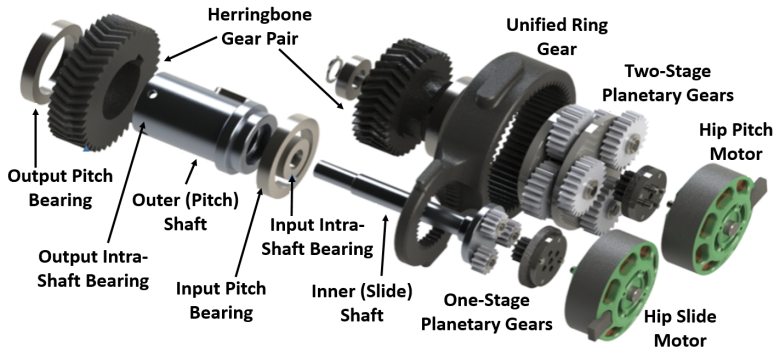


Figure 2.9: An exploded view of SLIDER v4's pitch and slide gearbox.

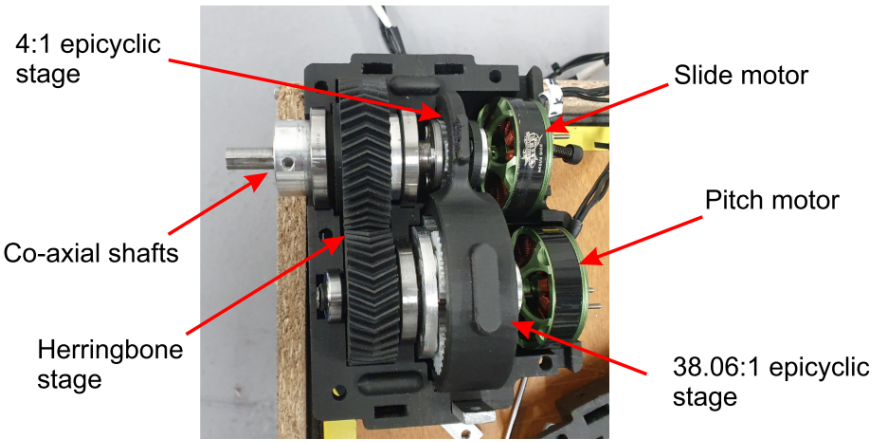


Figure 2.10: Hip gearbox assembly.

The inner shaft contains a single-stage planetary reducer, inline between the leg slide case and the slide motor. Bearings are placed between the inner and outer shafts and between the case and inner shaft. The inner shaft terminates with a key in a herringbone gear in the leg sliding case. The outer shaft is rigidly fixed to the leg slide case to generate pitch motion using four set screws for easy removal. A Multistar elite 5010-274Kv motor is connected to a herringbone gear pair with gear ratio 1.2:1 through a two-stage planetary gear reducer (31.7:1) with a single ring gear. All the planetary gear stages in this gearbox have a fixed ring gear with an input sun gear and an output carrier plate. The gearbox case is additively manufactured to decrease cost and lead time required to make design changes.

The herringbone gear pair, input sun gears (except pitch second stage sun), and unified ring gear are additively manufactured in Onyx, a tough nylon carbon fibre composite (Naranjo-Lozada et al., 2019). The planets in the two-stage pitch and one-stage slide planetary stages are nylon. They sit on roller bearings, located by circlips on shafts pressed into the aluminium carrier plates, which have a thickness of 4 mm. A unified ring gear was chosen to precisely locate each of the three planetary gear stages (two from pitch and one from slide) while allowing quick replacement for broken teeth or different gear ratio trials. Future plans to upgrade the planetary gear stages with steel can be quickly realized without the need to print new gearbox cases. One way to achieve the compliance of the leg is by adding elastic elements on the slide joint and this will be discussed in Chapter 3.

Leg Sliding Case

The leg sliding case accepts input rotary motion from the inner shaft of the P&S gearbox and outputs linear ("sliding") motion to the rack on the leg. The rack, which covers over 70% of the length of the leg, allows ± 35 cm of linear motion. It uses a two layer sandwiched belt

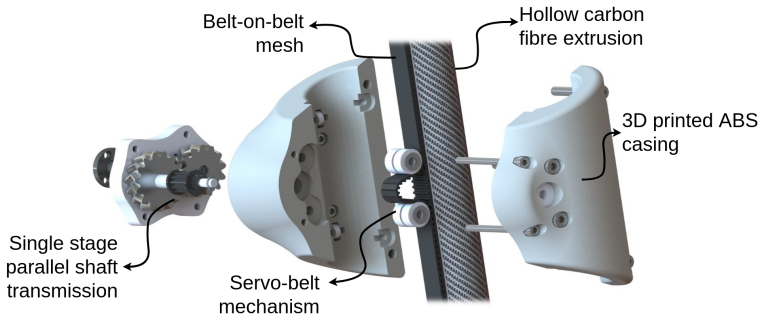


Figure 2.11: An exploded view of SLIDER's slide gearbox.

Table 2.3: Comparison of ServoBelt drive mechanism with other conventional prismatic joint transmission systems

Prismatic Joint Transmission	Light-weight	Low cost	Precision	Maximum Load
Lead screw	No	No	High	High
Ball screw	No	No	High	High
Rack and pinion drive	No	Medium	Medium (depends on gear backlash)	Medium
Conventional belt and pinion drive	Yes	Yes	Medium (depends on belt tension)	Medium
ServoBelt linear drive	Yes	Yes	High	High

system with a fixed and free belt. More detailed description can be found in (Wang et al., 2018).

Hip Roll

The hip roll gearbox is responsible for the rapid roll movement of the entire P&S gearbox, leg sliding case, leg, and ankle. Additionally, when the robot is in single support phase, it is solely responsible for resisting the roll moment generated by the mass of the in-air leg and frame. These functions, but especially resisting the moment generated during single support phase (approximately 18 Nm in the static case), require especially high torque and rigidity at low speeds. To achieve this required torque, the robot's most powerful motor is connected to a 36:1 two-stage cycloidal reducer, identical in principle to (Lin et al., 2014). Fig. 2.12 details the design of the hip roll. Its mass is 1.31 kg including support plates.

The motor drives a 3.50 mm eccentric cam, interference fit into a deep groove ball bearing. This drives a combined cycloid disc with six lobe input side and five lobe output side. The cycloid disk is aluminium, with grooves for additively manufactured protective spacer disks to prevent galling against the rolling assembly plates. The cam and bearing are located on one side by a shoulder and on the other by an Onyx spacer. The combined cycloid disk follows a path prescribed simultaneously by the input rollers and the cam. The output roller plate, attached to the output adapter by bolts, is propelled by the cycloid disc in a similar manor. A circlip locates each needle roller bearing. The pins are interference fit into the roller plates.

The hip rolls must operate with high radial loads because the output is directly responsible for supporting the leg forces and P&S gearbox mass and inertia. This was mitigated through an 80 mm bearing, which directly supports the output roller assembly. The output support plate transfers these loads directly into the frame, reducing stresses in the gearbox case and preventing high moments on the output adapter.

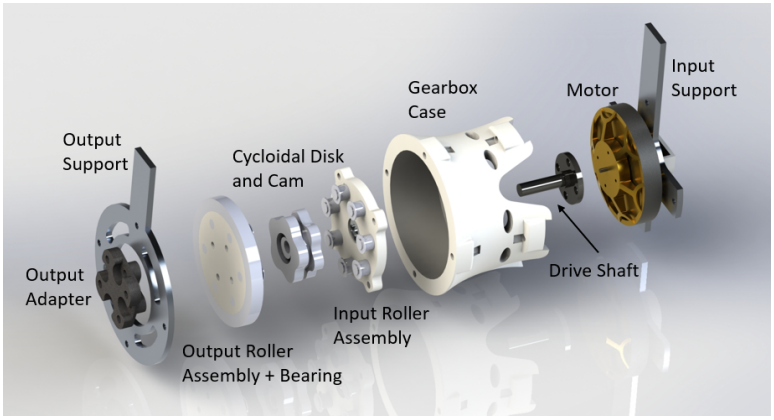


Figure 2.12: An exploded view of SLIDER v4's two-stage cycloidal reduction for hip roll.

Ankle Design

To generate both ankle pitch and roll, two actuators are attached to each foot. These are located 29 cm above the ankle joint using aluminium tendons, which decreases the inertia of the leg. To balance the power requirements from each motor, a design similar to LOLA's (Buschmann et al., 2009) was adopted for the ankle tendon placement. By locating them offset in both pitch and roll, each motor contributes to each action.

The front and back ankle motors are housed in a combined gearbox case. The front and back gearboxes are identical, utilizing 15:1 one-stage cycloidal reducers. The input rolling pins are printed into the gearbox case, introducing sliding friction but decreasing mass. The cycloid disks use 1.5 mm eccentric cam assemblies, located with circlips. The output member consists of a 6 mm plate located on the drive shaft with a bearing. The plate has six interference fit 6 mm diameter pins. Fig. 2.13 shows the inside of the gearbox.

The joint mechanisms can be summarized as 2.4:

2.4.2 *SLIDER v5*

After using the robot and carefully inspecting all the joint mechanism, I realized several issues with the previous design:

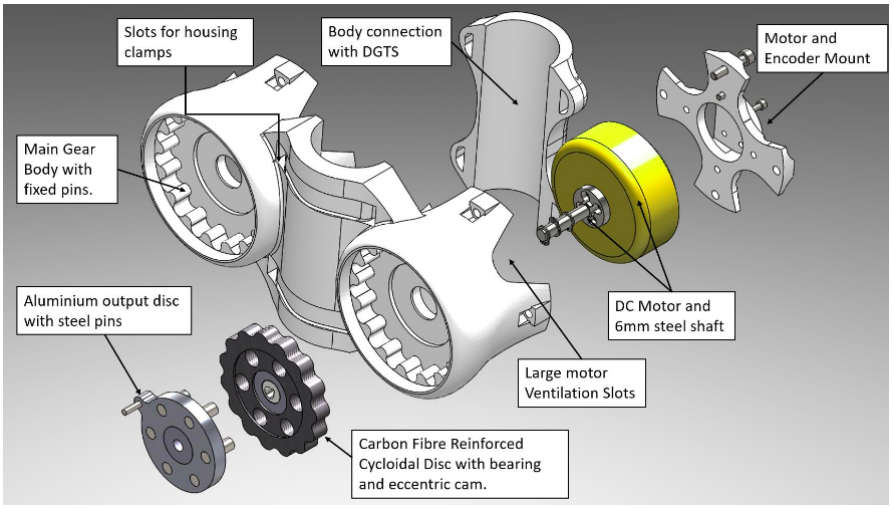


Figure 2.13: An exploded view of SLIDER's ankle gearbox (pitch and roll) using a one-stage cycloidal gear reduction.

- Backlash in hip roll and hip pitch. A backlash of approximately $\pm 3^\circ$ was found between the motor and the gearbox output of hip roll, a backlash of approximately $\pm 6^\circ$ was found between the motor and the gearbox of hip pitch. The backlash is credited to the fact that the gearbox components were manufactured in-house with loose tolerance. The hip pitch transmission is consisted of a two-stage epicyclic gear and all components are connected in series, resulting in a significant backlash.
- Hard assembly. The previous design was not design for easy assembly and the coupling between different joints are unnecessarily complicated.
- Large friction. The hip roll joint had a two-stage cycloidal gearbox with a gear ratio of 36:1 and the hip pitch had a gear ratio of 38.06:1 with the one-stage herringbone gear 3D printed. The high gear ratio and in-house manufactured gears result in high friction and the joints are not backdrivable.

Therefore the goal of new version of SLIDER is:

- Reduce the backlash of all joints.
- Easy assembly of the robot.
- Improve the backdrivability and increase compliance of joints.
- Reduce the number of components of the robot and make the robot more compact.

Fig. 2.14 shows the exploded view of the whole robot assembly. Because in the new version of design the joint transmissions are replaced with the integrated electrical motors, the robot is more compact and has less components.

Table 2.4: Overview of SLIDER v4's joint specifications

Joint	Motor	Gear Ratio	Gear Arrangement	Range
Hip Roll	Multistar 9225-160Kv	36 : 1	2 stage cycloidal reduction	$\pm 30^\circ$ to $\mp 90^\circ$
Hip Pitch	Multistar elite 5010-274Kv	38.06 : 1	2 stage epicyclic + 1 stage spur	$\pm 180^\circ$
Hip Slide	Multistar elite 5010-274Kv	1.23 cm/rev	1 stage epicyclic+ 1 stage herringbone + rack and pinion*	-30 to +40 cm
Ankle Pitch	Quanum MT 5208-335Kv	15 : 1	1 stage cycloidal reduction	$\pm 30^\circ$
Ankle Roll	Quanum MT 5208-335Kv	15 : 1	1 stage cycloidal reduction	$\pm 21^\circ$

* One-stage herringbone and rack and pinion arrangements are within the leg sliding case.

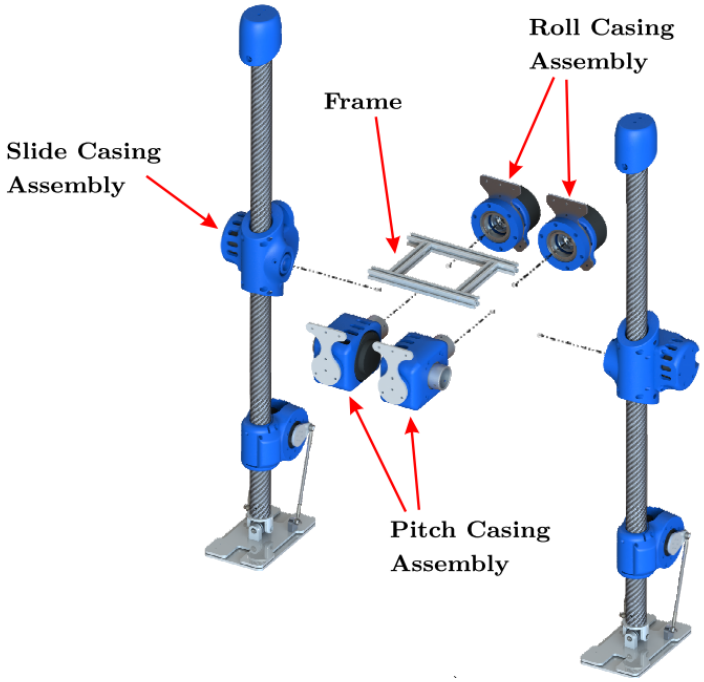


Figure 2.14: An exploded view of SLIDER v5 whole assembly.

2.4.3 *Hip Joints*

The hip joints assembly are shown in Fig. 2.15, 2.16, 2.17. Besides from the motors, the components in hip roll and hip pitch are mainly for supporting motors. The hip slide uses the same belt pulley mechanism as the previous version.

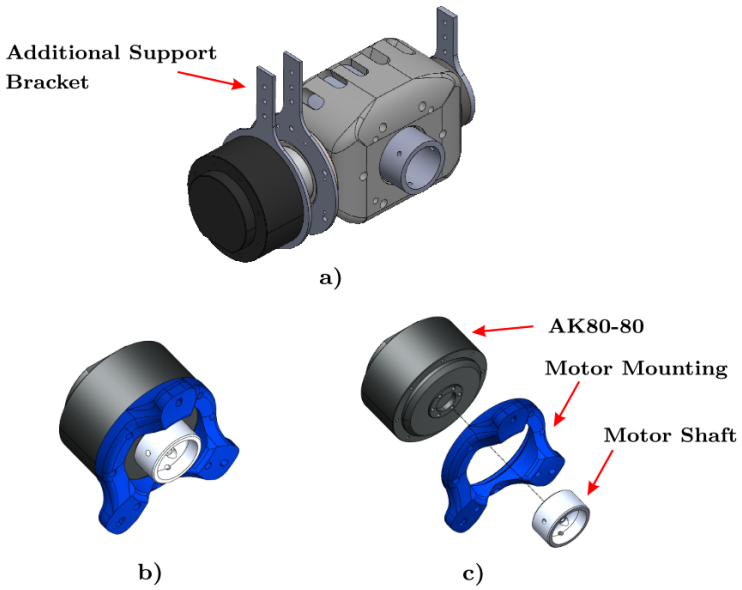


Figure 2.15: An exploded view of SLIDER v5 hip roll assembly.

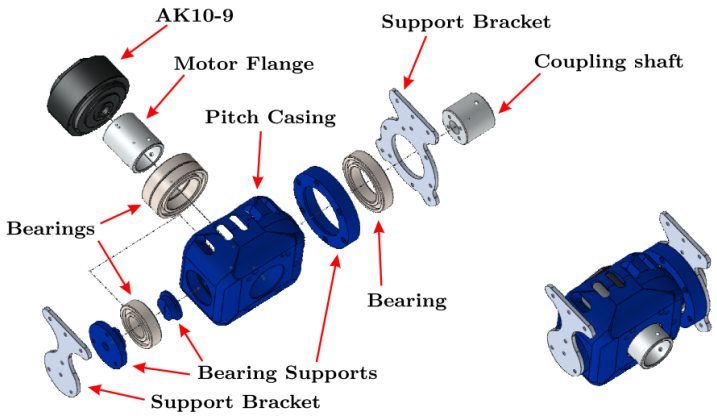


Figure 2.16: An exploded view of SLIDER v5 hip pitch assembly.

Ankle Design

The new ankle design is also a four-bar linkage structure. Compared with the previous design shown in Fig. 2.13, both actuators are chosen to be placed backwards for a more compact structure, as can be seen in Fig. 2.18.

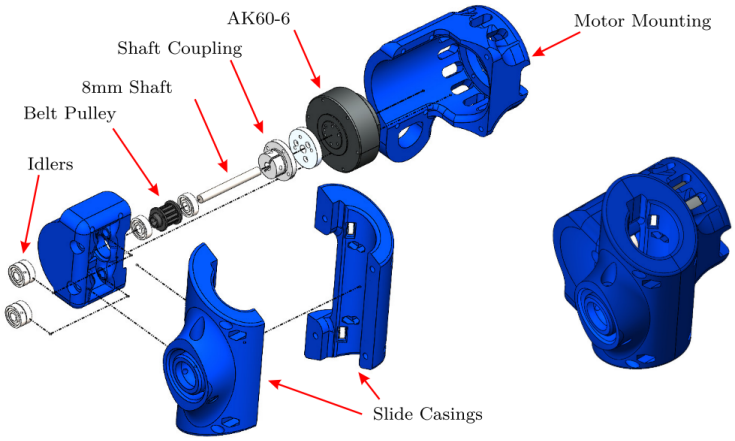


Figure 2.17: An exploded view of SLIDER v5 hip slide assembly.

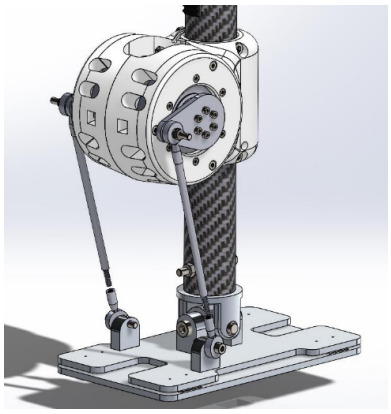


Figure 2.18: The SLIDER v5 ankle design.

2.5 ACTUATORS, ELECTRONICS AND SOFTWARE

Torque Estimation

To determine the gear ratio and motor to be used for each joint, it is important to estimate the required torques. The torques can be estimated by simple analytical approximations, however many aspects are neglected. For example the torque to compensate impacts on the ground is hard to model and often neglected. Moreover the torque to account for the link inertia during walking is complex and also neglected. A better way is to measure the joint torque values in a realistic physics simulation. Here Gazebo(Foundation, 2021) is used as the physics simulator and Robot Operating System (ROS) (Stanford Artificial Intelligence Laboratory et al., 2018) to record the data, as show in Fig. 2.19.Hip joint torques were measured and ankle joints are not measured because it is hard to simulate the closed-loop four-bar linkage in Gazebo. Furthermore, the ankle needs to be compliant so it is not very important to get the

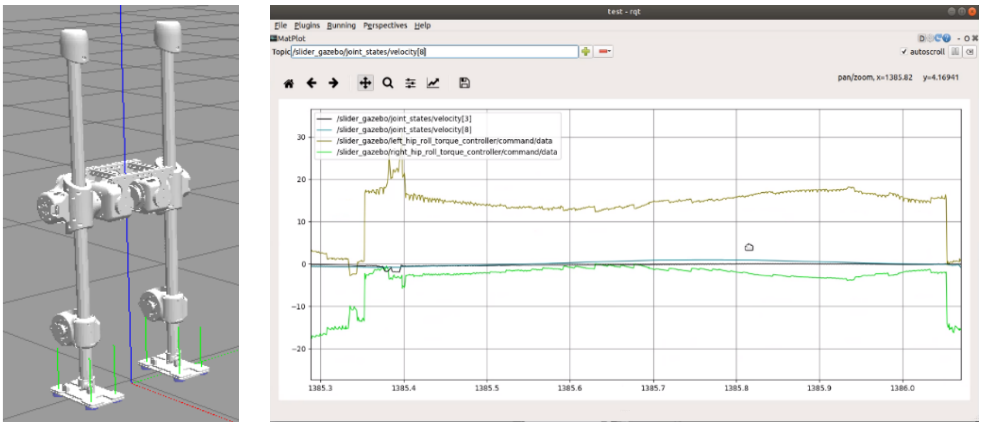
Table 2.5: Peak torque value at different walking speeds in Gazebo

Walking Speed (m/s^2)	Pitch Torque (Nm)	Roll Torque (Nm)	Slide Torque (Nm)
0.1	2.5	25	1.05
0.2	3.1	23	1.1
0.3	3.9	20	1.1
0.4	5.1	18	1.1
0.5	7	16	1.1

Table 2.6: Torque Requirement for Hip Joints

Joint Name	Torque Required (Nm)
Hip Slide	1.2
Hip Pitch	10
Hip Roll	30

peak torque needed for the ankle actuators. The result of the simulation is summarised in Table 2.5.



(a) (b)

Figure 2.19: Left: SLIDER robot in Gazebo. Right: Real-time torque measurement from Gazebo shown by rqt-plot

By analysing these results, the continuous torque requirement for each joint was decided as listed in Table 2.6.

For SLIDER v4, Multistar brushless motors are used and their spefics are listed in Table 2.7

As observed from the data, the Multistar 5010-274Kv used for hip pitch and hip slide has a low maximum torque and it is impossible to reduce the gear ratio to a great extend. In SLIDER v5, t-motor were used and the specifications can be seen in the Table 2.8. The data

Table 2.7: Motor Specification for SLIDER v4

Joint	Motor Name	Torque Constant (Nm/A)	Max Current (A)	Max Torqe (Nm)
Hip Pitch/Slide	Multistar 5010-274Kv	0.035	28	0.97
Hip Roll	Multistar 9225-160Kv	0.059	48	2.8

Table 2.8: Motor Specification for SLIDER v5

Joint	Motor Name	Torque Constant (Nm/A)	Max Torqe (Nm)	Mass (Kg)	Cost ($)
Hip Pitch	AK 10-9	18	54	0.82	698
Hip Slide	R100	3.7	11	0.75	426
Hip Roll	AK80-80	48	144	0.79	989

shows that t-motors are able to satisfy the torque requirements. The t-motors can be seen in Fig.2.20.

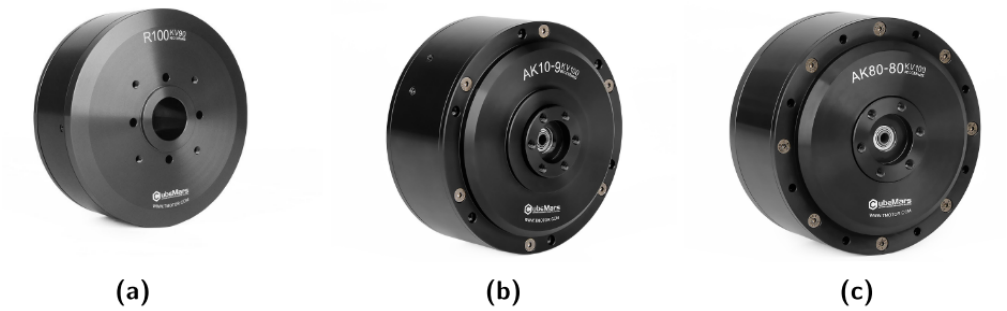


Figure 2.20: T-motors: (a) R100; (b) AK10-9; (c) AK80-80.

2.5.1 *Electronics and Software*

For SLIDER v1, Motors are controlled by ODrives (Odrive, 2019). ODrive is a open-source high performance motor controller which can handle peak currents of more than 100A per motor. Each ODrive is capable of controlling 2 motors. ODrives have a cascaded style motor controller with position, velocity and current control loop, all updated at 8kHz. A programmable interface through Python is provided to the user. Each joint has an AS5047P absolute encoder on the motor side, connected to the ODrive. To provide feedback signals on the output side, 3 Bosch BNO055 IMUs are mounted on the robot. One IMU is mounted on the base frame to measure the orientation of the base, with the other two IMUs mounted on the left and right leg to measure the pitch and roll angle of each leg with respect to the ground. Each foot has 4 load cells mounted on the bottom to provide the Center of Pressure (CoP) at 125 Hz. For SLIDER v5, the motor controllers and motor encoders are

integrated inside the motors. The computer communicates with the motors with CAN with a frequency of 500 Hz. IMUs and load cells are also mounted on the robot.

ROS (Stanford Artificial Intelligence Laboratory et al., 2018) is used to reduce the computational load of the main controller program. Local processing of ODrives, IMUs, load cells is handled by separate ROS nodes and the communication with the main controller is handled by ROS messages. This allows additional functionality to be easily added. The code is written in Python, which is fast enough for early development testing. For simulated testing, Gazebo is used as our simulation environment. Gazebo uses ODE as its default physics engine. Although ODE has worse reliability than the OpenHRP/AIST dynamics engine, as stated in (Stasse et al., 2017), the max step in ODE is set to be 0.0005 so the reliability is guaranteed for simulating walking.

2.6 LESSONS LEARNED

After several iterations of design and building two versions of the real SLIDER robot, I have learned several lessons on building a bipedal robot of human size.

- Try to avoid designing and building in-house transmissions unless necessary. Designing transmissions requires time and expertise, moreover it normally takes several iterations to design a transmission that satisfies all requirements. Manufacturing them in-house often suffers from inaccuracy of parts that cause either high friction or loose tolerance.
- Try to reduce backlash. Backlash will cause a lot of problems for control because backlash will cause a deadzone region where the motor is moving and the output does not move. It is hard to model the dynamics of backlash and a big backlash cannot be eliminated by pure feedback control. Instead, applying feedback control will cause oscillation of the joint.
- Try to increase backdrivability of the joint. A joint with good backdrivability have several advantages: (1) there is less friction and therefore more energy efficient, (2) the joint can be compliant and the robot can be more robust to disturbances; (3) it can make torque control easier.
- Try to reduce the number of parts and make the assembly easier.
- 3D printed material is a capable of most components of the bipedal robot. However metal is necessary. SLIDER has proved that 3D printed material is capable of most components for a robot with large size. However for components that resists large forces, for example the supporting components for hip roll or the rods of the ankle, the 3D printed material based components are easy to deform and break.

CHAPTER 3

Modelling of SLIDER

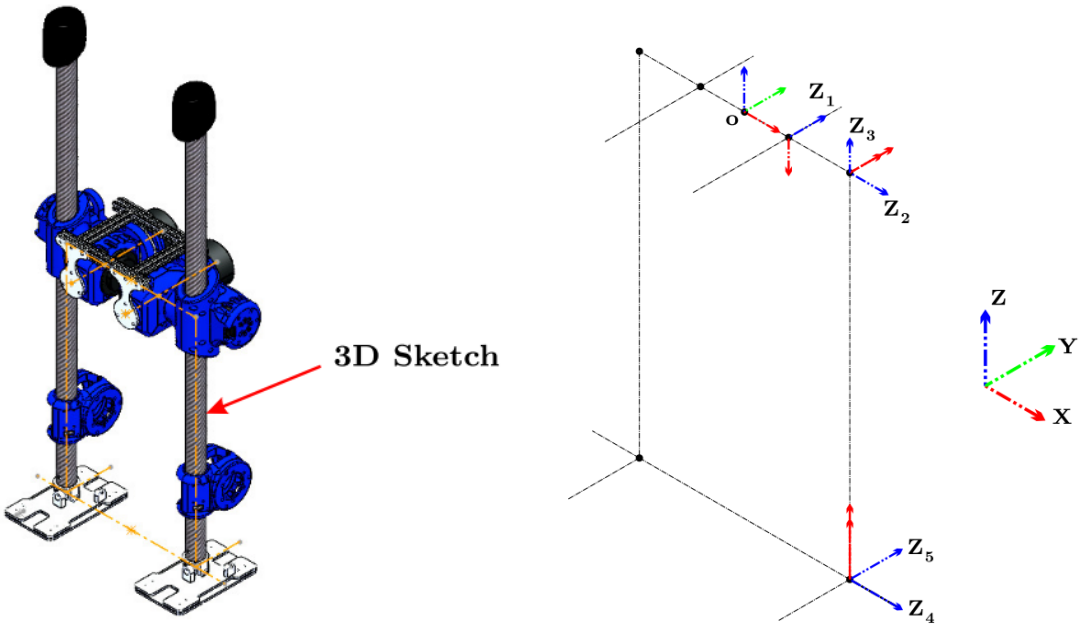
Modelling is important in the sense that it significantly affects the control performance. Modelling parameters are essential for inverse kinematics or inverse-dynamics based WBC. Furthermore, SLIDER's ankle has a parallel four-bar linkage structure that needs calculation for inverse kinematics. The modelling can also influence the energy efficiency of the robot. By adding elastic elements on the sliding joint of SLIDER, the energy efficiency can be improved. In this chapter I discuss about the modelling of SLIDER including: (1) the DH parameters, ankle parameters and the inverse kinematics of the ankle, (2) adding bungee cords and using Bayesian Optimization to improve the energy efficiency.

3.1 DH PARAMETER OF SLIDER

The DH parameters are sets of four parameters defining the reference frames associated with links and joints of a kinematic chain. Based on these parameters, transformation matrices from one link to another are then constructed. These matrices are then used to solve the kinematics of the robot. The DH parameters are needed to build the Unified Robotic Description Format (URDF) which is used in Gazebo and inverse kinematics solvers.

The placement of coordinate frames for each link of the robot is shown in Fig. 3.1. The frames are selected such that the maximum number of parameters in the DH table has zero value. Based on the axes placement, the DH parameters for the robot's leg are listed in Table 3.1.

The four DH parameters, α, a, d and θ, represents twist angle, link length, link offset and joint angle. These parameters are used to find the transformation matrix between link $i-1$ and i follow the book (Craig, 1989).



(a) (b)

Figure 3.1: Left: SLIDER robot 3D sketch. Right: Placements of coordinate frames for each link in SLIDER.

Table 3.1: DH parameters for one side of the SLIDER v5

Joint No. (i)	α_{i-1} (degree)	a_{i-1} (mm)	d_i (mm)	θ_i (degree)
1	-90	75	0	θ_1
2	90	0	104.2	θ_2
3	-90	0	d_3	0
4	90	0	0	θ_4
5	90	0	0	θ_5

3.2 ANKLE MODELLING

The ankle has a four-bar linkage structure which needs extra attention to solve the inverse kinematics. The ankle structure in SLIDER v4 and SLIDER v5 is slightly different. In v4 the ankle actuators are placed front and back while in v5 the ankle actuators are placed in parallel back-to-back.

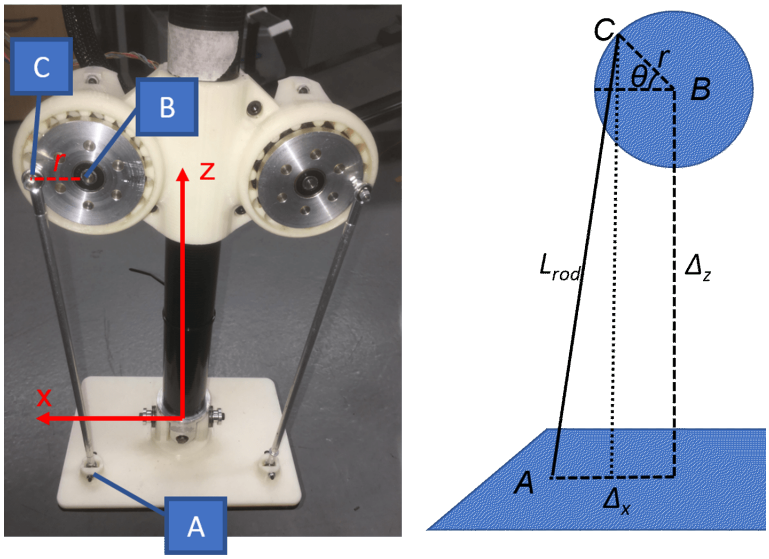


Figure 3.2: Illustration of the inverse kinematics for the ankle. left: the six bar linkage structure of ankle in real SLIDER, right: a schematic illustration of the six bar linkage structure

3.2.1 *Ankle Model in SLIDER v4*

The ankle pitch and roll motions are coupled by a six bar linkage structure, a geometric method is proposed to solve this Inverse Kinematics (Inverse Kinematics (IK)) problem. As seen in Fig. 3.2, the front point A of the right foot is driven by the motor centered at B. A base frame Σ_L is attached to the leg, with the origin located at the intersection of the two ankle axes with the z-axis pointing upward and x-axis pointing forward. A local frame Σ_F is attached to the foot with the same origin as Σ_L. It is assumed both A and B are fixed in local frame Σ_F and are expressed as ${}^F p_A$ and ${}^F p_B$ respectively. Given the desired ankle roll angle q_4 and ankle pitch angle q_5, the position of A expressed in the base frame can be calculated as follows:

$$ {}^L p_A = R_x(q_4) R_y(q_5) {}^F p_A, \tag{3.1} $$

where R_x, R_y are the rotation matrices with respect to the x and y axes. Then a relationship for the angle of rotation of the motor θ as denoted in Fig. 3.2 can be established:

$$ (r\cos\theta - \Delta_x)^2 + (r\sin\theta + \Delta_z)^2 = L_{rod}^2, \tag{3.2} $$

where $\begin{bmatrix}\Delta_x & \Delta_y & \Delta_z\end{bmatrix}$ is the absolute value of each individual element of $({}^L p_A - {}^L p_B)^T$, r is the radius of the motor output plate and L_{rod} is the length of the rod. This equation can

be solved symbolically by Symbolic Math Toolbox in MATLAB and the two solutions are shown below:

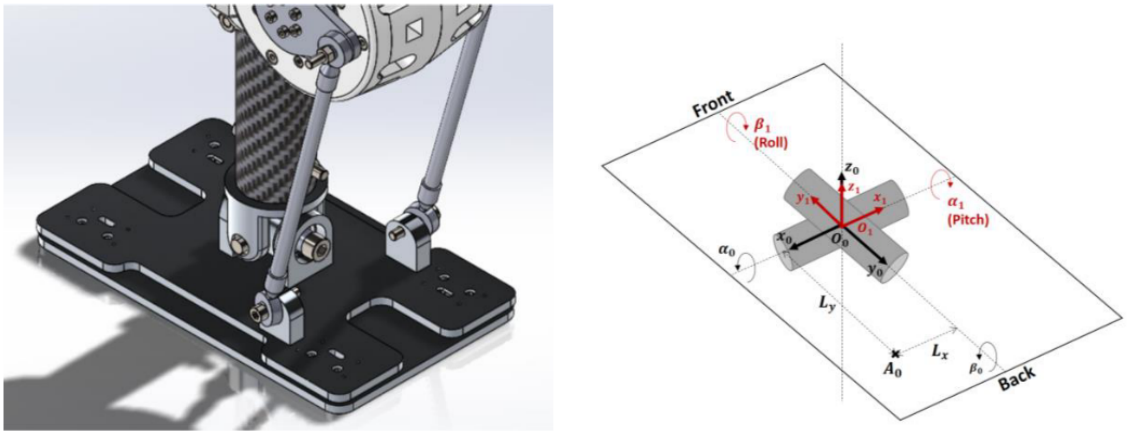
$$\begin{aligned}\theta_1 &= -\log\left(\frac{\Delta_x^2+\Delta_z^2+r^2-L_{rod}^2+s}{2(\Delta_x r+\Delta_z ri)}\right)i,\\ \theta_2 &= -\log\left(\frac{\Delta_x^2+\Delta_z^2+r^2-L_{rod}^2-s}{2(\Delta_x r+\Delta_z ri)}\right)i,\end{aligned} \tag{3.3}$$

with

$$\begin{aligned}s =&\sqrt{-(-\Delta_x^2-\Delta_z^2+r^2+2rL_{rod}+L_{rod}^2)}\\ &\times\sqrt{(\Delta_x^2+\Delta_z^2-r^2+2rL_{rod}-L_{rod}^2)}.\end{aligned} \tag{3.4}$$

The existence of two solutions is expected because the circle centered at A with radius L_{rod} has two intersections with the motor output plate. The solution is chosen in $[-\pi/2, \pi/2]$ to restrict the motion of the motor. The result can also help us determine whether a gait is executable or not, because a complex result means that the desired ankle pitch and roll motion is infeasible.

3.2.2 *Ankle Model in SLIDER v5*



(a) (b)

Figure 3.3: Left: The CAD model for the ankle with the flat foot. Right: Geometry illustration for mathematical analysis notation, including origin and axes.

Fig 3.3a shows the CAD model of the ankle model and Fig. 3.3b show the mathematical notation for the inverse kinematics analysis, as can be refereed in the Table. 3.2.

Table 3.2: Mathematical Notations for Inverse Kinematics Analysis

Symbol	Descriptions
A_0, A	Rod attachment on feed before and after the rotation
M	Center of motor
H_0, H	Motor flange position before and after rotation
A_x, A_y, A_z	Coordinates of the rod attachment
L_x, L_y	Distances between the origin and the rod attachment in x and y axes
L	Length of the rod
R	Radius of the motor flange
α, β	Pitch and roll angle of ankle

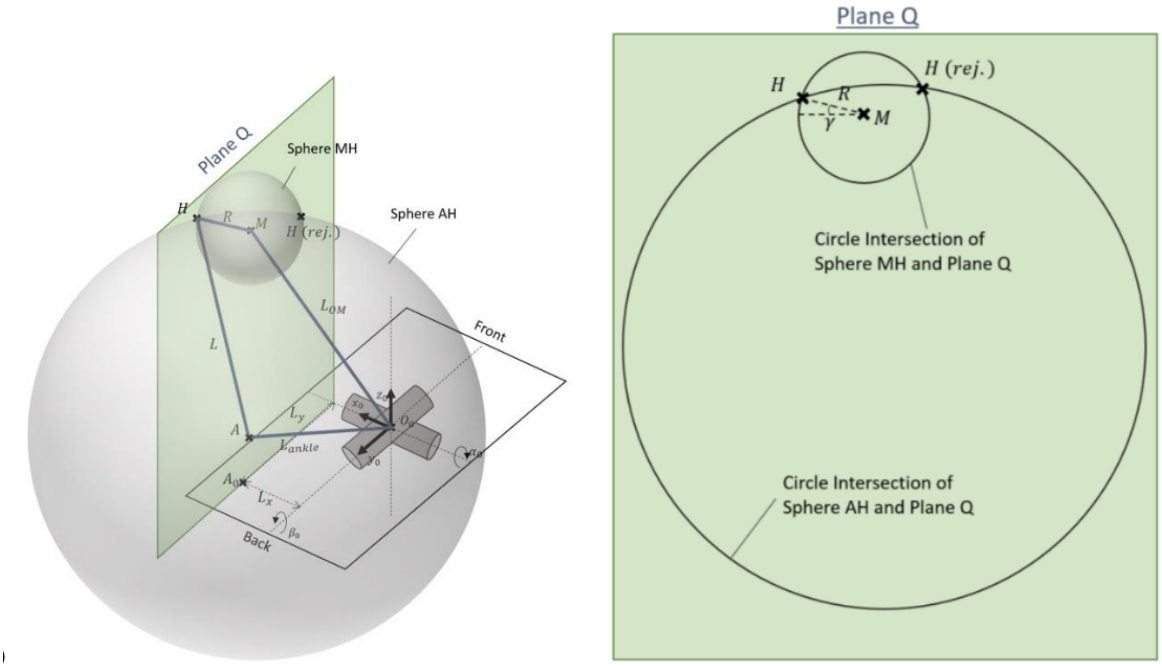
The coordinate of the ankle attachment A with roll angle α and pitch angle β can be calculated as:

$$\begin{aligned} A = R_x \times R_y \times A_0 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \times \begin{bmatrix} L_x \\ L_y \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} L_x \cos(\beta) \\ L_y \cos(\alpha) + L_x \sin(\alpha) \sin(\beta) \\ L_y \sin(\alpha) - L_x \cos(\alpha) \sin(\beta) \end{bmatrix} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \end{aligned} \tag{3.5}$$

For the previous design of ankle, the rod attachment A always located in the same plane as the center of motor and the origin, as shown in Fig. 3.2. However for the new design, it is possible that these four points are not in the same plane. Therefore a 3D spatial approach is applied for the inverse kinematics. As shown in Fig. 3.4.

Let Plane Q be the plane coincident with the motor interface; Sphere MH with radius R and Sphere AH with radius L. The intersections of both spheres with plane Q are circles as shown in Fig. 3.4b. This is due to the fact that in 3D space, the equation of circle is conveyed by a system of two equations – sphere and plane. Note that A does not lie on plane Q, therefore the radius of circle intersection AH does not equal to L. As a result, the coordinates of H is the solutions of the three equations:

$$\begin{aligned} (x_H - A_x)^2 + (y_H - A_y)^2 + (z_H - A_z)^2 &= L^2 \\ (x_H - M_x)^2 + (y_H - M_y)^2 + (z_H - M_z)^2 &= R^2 \\ x_H &= M_x \end{aligned} \tag{3.6}$$



(a) (b)

Figure 3.4: Illustrations of ankle kinematics in 3D. (a) Composition of the motor and feet in 3D space. (b) Intersections on Plane Q in 2D space.

There are two roots of these quadratic function and the solution can be simplified as:

$$z_H = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \text{ or } z_H = \frac{-B + \sqrt{B^2 - 4AC}}{2A} (rejected)$$
$$y_H = Jz_H + L \tag{3.7}$$

where A,B,C,J and L are constants. Equation 3.7 provides an accurate solution to the parallel four-bar linkage mechanism, where the ankle is not in the same plane as the rod and the origin as the motor rotates. Similar as the ankle in SLIDER v1, the solution is restricted into the range of $[-\pi/2, \pi/2]$, therefore the larger root is rejected. Moreover, it is possible that both roots are virtual numbers. In this case the solution is deprecated. The motor output angle γ can be computed as:

$$\gamma = \tan^{-1} \frac{z_H - M_z}{y_H - M_y} \tag{3.8}$$

Because two motors are place in parallel so this solution applies to both motors.

3.2.3 *Algorithm Summary of Ankle Inverse Kinematics in SLIDER v5*

The inverse kinematics for the ankle of SLIDER v5 is general and applicable for any four-bar linkage design. This highly adaptable inverse kinematics has a huge advantage of providing a framework for future ankle designs and eliminating the process of deriving their corresponding inverse kinematics. The algorithm is summarized below:

Algorithm 1: Inverse Kinematics for Ankle

Given: Ankle parameters: Rod length L, motor flange radius R, motor center position M, rod attachment position L.
Input: Desired angles of ankle roll α and pitch β.
Compute γ based on equations 3.5, 3.6, 3.7, 3.8
If $\gamma > \pi/2$ or $\gamma < -\pi/2$ or both roots from Equation 3.7 are virtual numbers: deprecate γ
Output: Desired motor angle γ.

3.2.4 *Experimental Validation of Ankle Inverse Kinematics*

The result of theoretical calculation is validated by attaching the Bosch BNO055 IMU to the bottom of the foot and measure the real pitch and roll angle. The desired ankle pitch and roll angle is commanded with the range of $[-20°, 20°]$ and the compare with the real measured angle. The results can be seen in Fig. 3.5. As can be seen, there is small error in roll and a drifting error in pitch. This is because the parameters can not be measure perfectly accurate and the IMU is drifting over time.

3.3 GEOMETRIC INVERSE KINEMATICS OF THE ROBOT

After deriving the inverse kinematics of the ankle, it is natural to derive the inverse kinematics of the whole robot. SLIDER is a knee-less robot and has a simple structure, therefore it is quite intuitive to derive the inverse kinematics through the geometric approach, as shown in Fig. 3.6. The mathematical notations for inverse kinematics are shown in Table 3.3.

Note that the position and orientation of pelvis and ankle is known, therefore $[p_1, R_1, p_3, R_3]$ are given. The hip position and orientation can be computed by Equation 3.9.

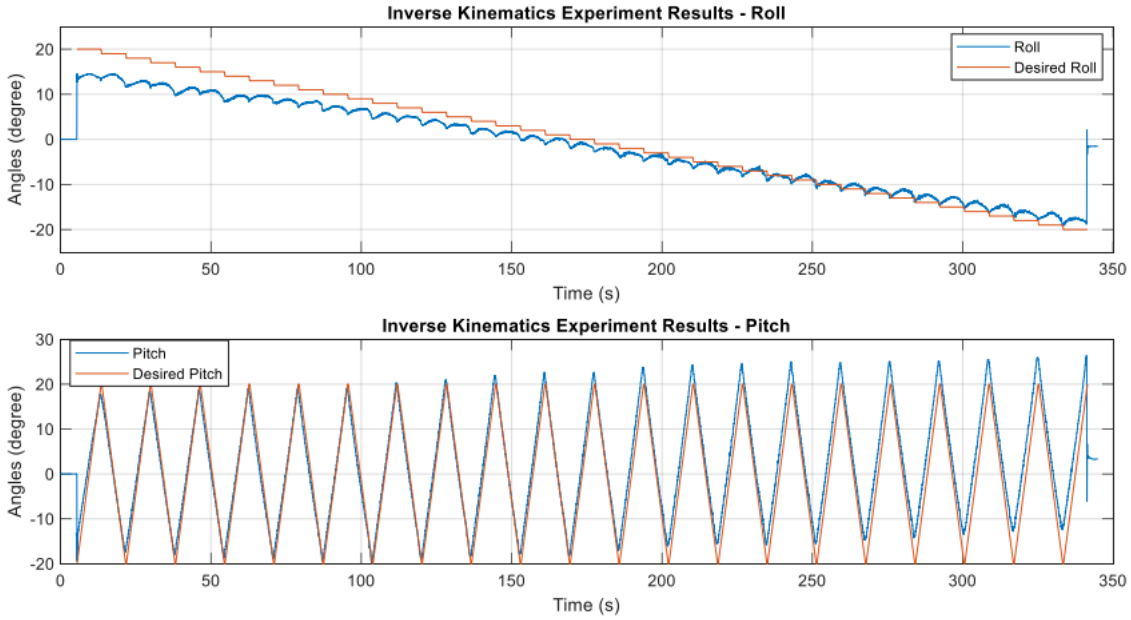


Figure 3.5: Comparison of desired and actual trajectories of ankle roll (top) and pitch (bottom) in the experiment.

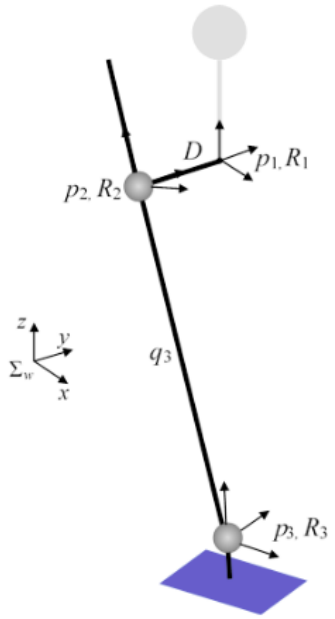


Figure 3.6: Illustration of SLIDER's right leg structure for solving inverse kinematics.

$$p_2 = p_1 + R_1 \begin{bmatrix} 0 \\ -D \\ 0 \end{bmatrix} \tag{3.9}$$

$$R_2 = R_1 R_y(q_1) R_x(q_2)$$

Table 3.3: Mathematical Notations for Inverse Kinematics

Symbol	Descriptions
p_1, R_1	Position and orientation of pelvis center
p_2, R_2	Position and orientation of right hip
p_3, R_3	Position and orientation of right ankle
D	Distance from pelvis center to right hip
q_1, q_2, q_3	Hip pitch, roll and slide joint value
q_4, q_5	Ankle pitch and roll

The vector from hip to ankle expressed in the ankle coordinate is computed by Equation 3.10 and the hip slide joint value can be computed by Equation 3.11.

$$\mathbf{r} = R_3^T(p_2 - p_3) = [r_x, r_y, r_z] \tag{3.10}$$

$$q_3^{'} = \sqrt{r_x^2 + r_y^2 + r_z^2} \tag{3.11}$$

The ankle pitch and roll angle can therefore be computed by Equation 3.12.

$$q_4 = -atan2(r_x, r_z) q_5 = atan2(r_y, \sqrt{r_x^2 + r_z^2}) \tag{3.12}$$

From the Equation 3.13 i can compute the hip joint angles with Equation 3.14, where $c_1 = \cos q_1$ and $s_1 = \sin q_1$. As the robot is symmetric, the joint angles in left leg can be computed the same way as the right leg. In real SLIDER the axis of hip roll doesn't intersect with the leg, as shown in Fig. 3.7. A compensation value for the hip slide is added by the Equation 3.15, where α is the scaling factor and L is the length from the hip to the ankle. Here the value of α is chosen to be 0.5.

$$\begin{aligned} R_3 &= R_1 R_y(q_1) R_x(q_2) R_x(q_4) R_y(q_5) \\ R_y(q_1) R_x(q_2) &= R_1^T R_3 R_1 [R_x(q_4) R_y(q_5)]^T \end{aligned} \tag{3.13}$$

$$\begin{bmatrix} c_1 + s_1 s_2 & s_1 s_2 - c_1 & c_2 s_1 \\ c_2 & c_2 & -s_2 \\ c_1 s_2 - s_1 & c_1 s_2 + s_1 & c_1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.14}$$

$$q_1 = atan2(R_{13}, R_{33})$$

$$q_2 = asin(-R_{23})$$

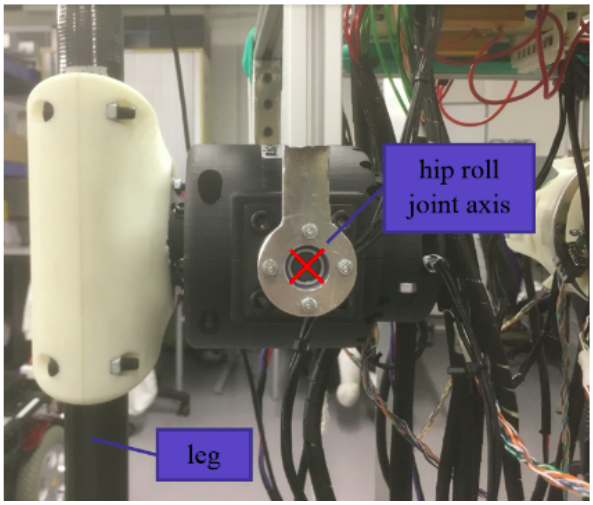


Figure 3.7: Illustration that the hip roll axis doesn't intersect the leg.

$$q_3 = q_3^{'} - \alpha L tan(q_2) \tag{3.15}$$

3.4 IMPROVE ENERGY EFFICIENCY WITH PARALLEL ELASTIC ELEMENTS

A common approach to improve energy efficiency in robots is the inclusion of compliant elements that are capable of storing and subsequently releasing energy at certain points of the robot's gait, performing work that would otherwise have to be taken up by the actuators (Alexander, 1990). In addition to energy efficiency, introducing compliant elements can also add some degree of robustness, by isolating fragile components of the mechanism from shocks resulting from foot impacts (Radkhah et al., 2014) (some exceptions to this are explained later in the introduction). Compliant elements are most commonly used in series, making use of either springs (Sreenath et al., 2011), series elastic actuators (Pratt et al., 1995) (Pratt et al., 2012), or oftentimes a combination of both (Hobbelen et al., 2008) (Kormushev et al., 2019). However, examples of compliant elements in parallel with revolute joints do exist (Yang et al., 2008) (Mazumdar et al., 2017), and research into including compliance in

the spine of a quadrupedal robot has also been performed (Folkertsma et al., 2012). But only simple spring models are used.

Unfortunately, hardware implementation issues (and their workarounds) can often nullify the benefits of energy-saving measures. When attempting to walk with straight (or near straight legs), singularity issues in the robot kinematics may occur (Lynch et al., 2017). To avoid this situation, robots can often be seen standing and walking in a squatting stance, this is also done to keep the robot's centre of mass at a near constant height for trajectory planning purposes (Kajita et al., 2003). Furthermore, as an exception to the earlier argument of compliant elements adding robustness to robots, with the leg fully extended, any compliance in the design is severely reduced upon foot touchdown, because impacts travel straight through the links of the leg directly to the hip, bypassing the joint compliance and essentially transferring the impact shock to the torso (Dallali et al., 2014).

By leveraging SLIDER's unique design, here lightweight bungee cords are introduced which is anchored in parallel with the prismatic joints that extend the robot's legs, and demonstrate vertical compliance and energy efficiency without the implementation issues seen on robots with articulated legs. To illustrate how our approach differs from the current literature, Fig. 3.8 shows common uses of compliant elements in legged robots, while Fig. 3.9 shows our usage of compliant elements.

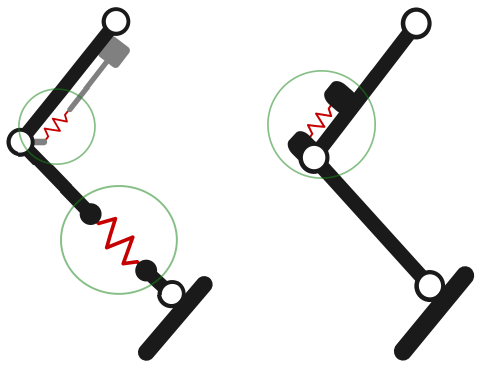


Figure 3.8: Sagittal plane view of common uses of compliant elements in walking robots, circled in green. In (a), compliant elements are either in series with robot links to store energy or dampen impacts, or used as part of series elastic actuators to directly drive joints. In (b), the more rare case of a spring being mounted in parallel with a revolute joint is shown.

Different optimization methods have previously been used in the literature to characterize walking motion that results in energy efficiency. (Kormushev et al., 2011) use reinforcement learning with evolving policy parameterization, however the uncertainty is not modelled. (Arakawa et al., 1996) use genetic algorithms but this requires a lot of samples. Bayesian Optimization is a sample-efficient black-box optimization method that can be directly applied onto robot problems (Calandra et al., 2014a). Bayesian Optimization is used

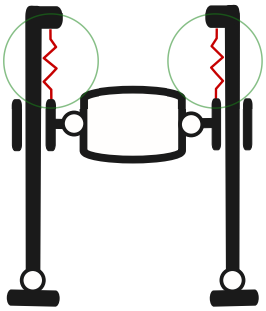


Figure 3.9: Coronal plane view showing the novel use of compliant elements in SLIDER, circled in green. As the legs extend above the torso, compliant elements (in this case, bungee cords) can be mounted in parallel with the prismatic joints.

for our problem because it works well with dynamics that are hard to model, and it is robust to noisy measurements.

3.4.1 *Design of The Vertical Prismatic Joint*

SLIDER's vertical prismatic joints are designed to be compact, lightweight and backdrivable, this latter property ensures that vertical compliance is always maintained. A single-stage planetary gear provides a 1:4 slide transmission reduction. Inspired by the rack and pinion mechanism, the servo belt mechanism is used, the design detail can be found in (Wang et al., 2020a). The linear actuation is powered by a tightly meshed pair of 3D printed carbon fiber reinforced plastic double helical gears. The chosen gear type ensures maximum efficiency and low noise production. To increase the contact surface with the pulley, two timing belts are used together, creating a servo belt mechanism. The testing results of the servo belt mechanism are summarized in Table 3.4.

Table 3.4: Servo belt test results.

Mass	Max Load	Max Speed	Max Travel Length	Backlash
0.996 Kg	32 Kg	0.45 m/s	0.6 m	negligible

3.4.2 *Characteristics of the Bungee Cord*

According to (Kockelman et al., 2004), the relationship between the applied force on the bungee cord and the length of the cord can be represented using generic stress-strain equations:

$$\sigma = \frac{F}{A} \tag{3.16}$$

where σ is the stress on the cord, A is the cross-sectional area of the cord when relaxed and F is applied force on the bungee. The strain ε is defined as maximum cord length x divided by the relaxed cord length L:

$$\varepsilon = \frac{x}{L} \tag{3.17}$$

The modulus of elasticity E is the slope of the stress-strain curve, written as:

$$E = \frac{\sigma}{\varepsilon} \tag{3.18}$$

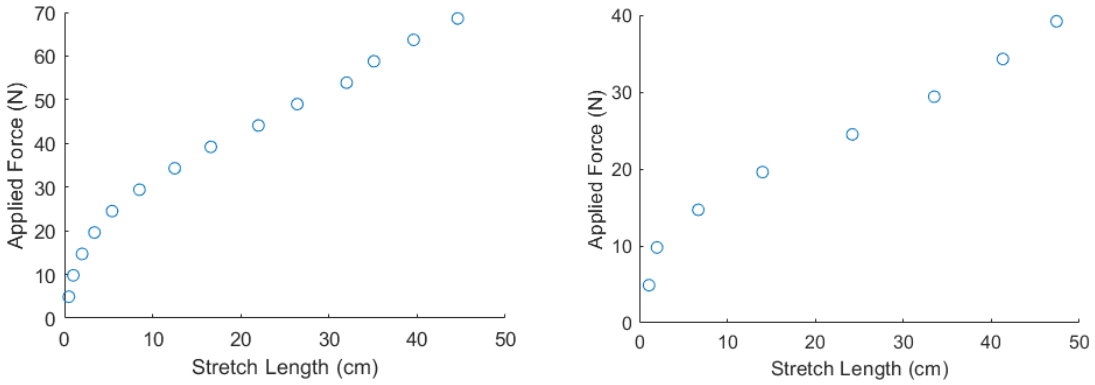
The relationship between stiffness k and modulus E is:

$$k = \frac{AE}{L} \tag{3.19}$$

Compared with the elastic modulus E that is a material property independent of cross-sectional area A and cord length L, according to Eq. 3.19 the value of stiffness k is dependent on both A and L, making it difficult to calculate the stiffness when the length changes. This is reinforced in Fig. 3.10, which shows that the bungee cord exhibits nonlinear behaviour.

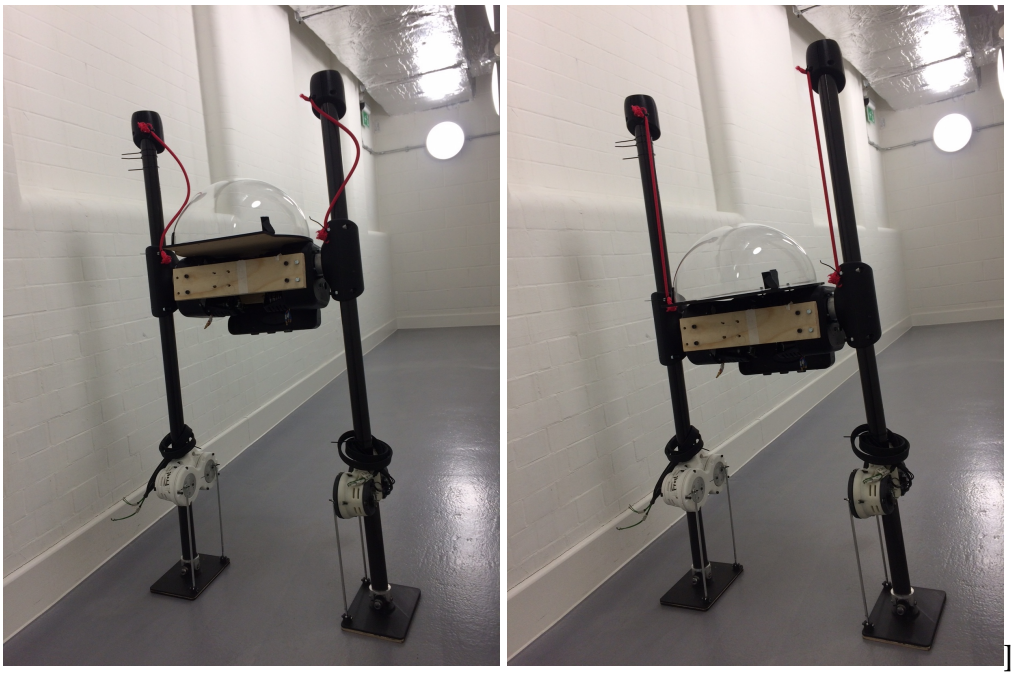
Attaching Bungee Cords in Parallel

The design of SLIDER's leg make it easy to attach elastic elements parallel to the leg. The end cap located on the top of each leg and the long bolt of the linear drive hub carriage act as anchor points for attaching the bungee cords, as is shown in Fig. 3.11. Compared with springs which are widely used as elastic elements on the robots, bungee cords are made of natural rubber and are incredibly light. A unique property of bungee cords is their ability to stretch to roughly six times their relaxed length without breaking, which is an incredibly useful property to have considering the large leg extensions that SLIDER is capable of.



(a) (b)

Figure 3.10: Experimental results of bungee cord length change with increasing applied force. In (a), the test results of a bungee cord with a diameter of 10 mm is shown. In (b) the test results of a bungee cord with a diameter of 8 mm was is shown. Both cords have a relaxed length of 40 cm. The results show that the stiffness of the bungee cords is nonlinear with respect to the changing length as force is applied, especially when the stretch length is small.



(a) (b)

Figure 3.11: SLIDER standing with two different torso heights. When the torso is high in (a), the red bungee cords are relaxed. In (b), the torso is low and the bungee cords are in tension.

3.4.3 *Finding Optimal Energy-Efficient Configuration Using Bayesian Optimization*

Evaluation of Energy Consumption

The bungee cord can be utilized to improve the energy efficiency of SLIDER's motion, as the cord stores potential energy when it is stretched by downward motion of the pelvis. In order

to verify that the robot is saving energy compared to when it does not have bungee cords installed, a way to evaluate energy consumption must be found. One approach is to estimate the mechanical energy from motor torque measurements and angular velocities. However, this approach incorrectly includes the work done by the bungee cord and gravity and can only infer the real electrical energy used for walking indirectly. The approach proposed by us is to directly measure the electrical energy used by the motors of the robot. SLIDER is equipped with open-source ODrive (Odrive, 2019) motor controllers, which provide direct measurement of motor current and voltage. The electrical energy can be calculated using the power Equation $P = IU$, where I is the electric current and U is the voltage. The consumed energy is calculated by integrating the power over time using

$$E_i = \int_{t_1}^{t_2} I_i(t) U_i(t)\, dt \tag{3.20}$$

where $[t_1, t_2]$ is the time interval and i is the selected joint for which the energy consumption is calculated. One rollout is defined as one walking cycle motion and experiments are performed with multiple rollouts in order to reduce the effect of noise in the measurement. The average energy consumption of one cycle is calculated and estimated using the formula:

$$E(\tau) = \frac{1}{c} \sum_{i \in I} E_i(t_1, t_2) \tag{3.21}$$

where τ denotes a given rollout of the experiment, I is the set of joints tested (in our experiment, the vertical slide joint), and c is the number of rollouts included in the experiment.

Bayesian Optimization for Finding Optimal Energy Efficient Motion

The stiffness of the bungee cord is nonlinear with respect to the stretched length. Moreover, mechanical friction from the gearbox and the servo belt is not easy to measure. Both factors make it hard to build a precise dynamic model of SLIDER and optimize the motion analytically. Instead, Bayesian Optimization is used to characterize the stretch length of bungee cords that results in SLIDER being as energy-efficient as possible.

Bayesian Optimization does sequential optimization over a surface response (i.e. surrogate model) instead of optimizing a known objective function directly (Agnihotri et al., 2020). The method iteratively creates a data set $D = \{\theta, f(\theta)\}$ in which θ is the parameter and $f(\theta)$ are the corresponding function evaluations. This data set is used to build a surrogate model $\hat{f}(\cdot) : \theta \rightarrow f(\theta)$, that maps θ to the corresponding function evaluations $f(\theta)$. The response

surface is subsequently used to replace the optimization of a real objective function which is unknown, with an optimization process which optimizes the surrogate objective function:

$$\theta^* \in \underset{\theta \in \mathbb{R}}{\operatorname{argmax}}\, \hat{f}(\theta) \tag{3.22}$$

In this problem the objective is the energy consumption defined in 3.21 and Gaussian Process (GP) (Rasmussen, 2003) is used to build the surrogate function. GP is a widely used probabilistic model that allows us to model noisy observations and to explicitly take the uncertainty of the model into account, making the probabilistic model more robust to the effect of model errors. GP also allows us to use priors to encode expert knowledge in a principled way. GP is a distribution defined by the mean and kernel function:

$$f(\mathbf{x}) \sim GP(\mu(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j)), \tag{3.23}$$

where μ is the mean function and k is the kernel function. The prior mean function $\mu(x)$ is set to be 0 in the experiment. There are a variety of kernel functions to choose from. The most widely used one is the the Squared Exponential (SE) kernel, here the Matérn kernel is used which is a generalization of the SE kernel.

$$k(x_i, x_j) = \sigma^2 \frac{1}{\Gamma(\nu)2^{\nu-1}} (\gamma\sqrt{2\nu}\rho)^{\nu} K_{\nu}(\gamma\sqrt{2\nu}\rho) \tag{3.24}$$

where ρ is the Eulicdian distance and $\Gamma(\nu)$ is the gamma function. This kernel has an additional parameter ν which controls the smoothness of the resulting function, γ is the length scale factor which also influences the smoothness of the function.

Given the data set $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$ with n training inputs where $\mathbf{X} = [\theta_1, ..., \theta_n]$ and corresponding function evaluations $\mathbf{y} = [f_1(\theta), ..., f_n(\theta)]$, the predicted distribution of GP is

$$p(f(\theta) \mid \mathbb{D}, \theta) = \boldsymbol{N}(\mu(\theta), \sigma^2(\theta)), \tag{3.25}$$

where the mean function $\mu(\theta)$ and the variance σ are:

$$\mu(\theta) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}, \sigma^2 = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*, \tag{3.26}$$

$\mathbf{K}$ is the matrix with $k_{ij} = k(\theta_i, \theta_j)$, $k_{**} = k(\theta, \theta)$ and $\mathbf{k}_* = k(\mathbf{X}, \theta)$.

In order to find next best point to sample, Bayesian Optimization uses an acquisition function $\alpha(\cdot)$. The acquisition function performs a trade-off between exploration and exploitation, which is extremely important for the optimization process when the data set has

only a few samples. The Upper Confidence Bound (UCB) is used as the acquisition function because it is reported to be the best performing acquisition function for the task of learning optimal walking gaits (Calandra et al., 2014b). UCB is defined as

$$\alpha(\theta) = \mu(\theta) + \kappa\sigma(\theta) \tag{3.27}$$

where κ is the factor to trade-off between exploration and exploitation, the choice of κ will be discussed in section 3.4.4. The whole Bayesian Optimization algorithm is implemented using an open source package (`https://github.com/fmfn/BayesianOptimization`) and is summarized as Algorithm 2.

Algorithm 2: Bayesian Optimization

Given: A black-box function $f(\theta)$ (defined in Eq. 3.21) for evaluation, an acquisition function $\alpha(\theta)$ (defined in Eq. 3.27), maximum iteration.
Output: Current optimal θ^* which minimizes $f(\theta)$.
Initialization: A dataset with pre-collected samples $D = \{\theta_i, f_i\}_{i=1,2,...}$ and find current optimal θ^* and $f(\theta^*)$.
while *i < max_iteration* **do**
  Find next query parameter θ_i by maximizing the acquisition function $\alpha(\theta)$:

$$\theta_i = \underset{\theta}{\operatorname{argmax}}\, \alpha(\theta|D)$$

  Evaluate the objective function $f(\theta)$ at θ_i, if $f(\theta_i) < f(\theta^*)$, then $\theta^* = \theta_i, f(\theta^*) = f(\theta_i)$;
  Add the new data point $\{\theta_i, f(\theta_i)\}$ to the dataset:

$$D \leftarrow D \cup \{\theta_i, f(\theta_i)\}$$

  Update Gaussian Process $GP(\mu(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j))$ and acquisition function $\alpha(\theta)$
  $i = i + 1$
end

3.4.4 *Experiments*

Experiments were performed directly on SLIDER because in simulation it is very difficult to model the nonlinear dynamics of the bungee cords and the mechanical friction and other losses present in the robot. To measure the improved energy efficiency via the bungee cord, the experiment is conducted using the vertical prismatic joints at the hips[1]. In the experiment setup, all other joints of the robot (hip roll, hip pitch, ankle pitch, ankle roll) were frozen by keeping the motor position fixed to zero positions. For the vertical prismatic joint motion, a

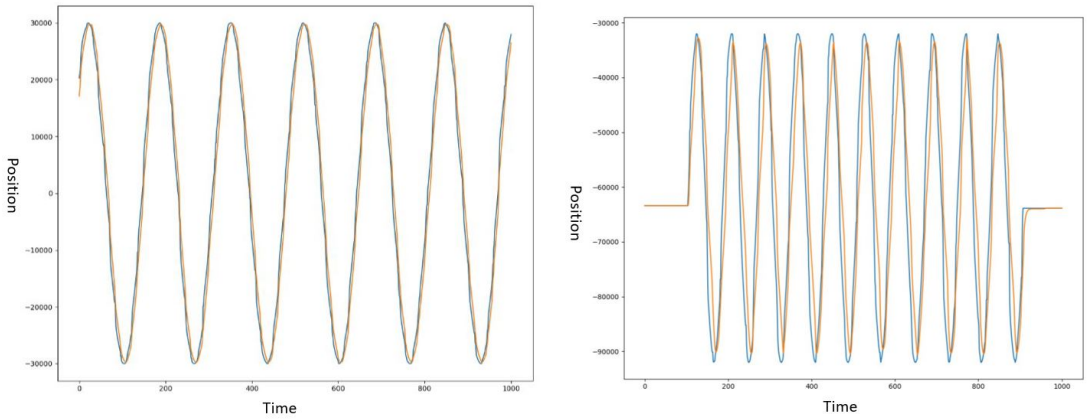
1 Due to the COVID-19 pandemic, access to the physical robot has been limited. The focus is on validating the concept and not on whole-body walking experiments.

bobbing motion was triggered as this motion is similar to the vertical component of walking. To achieve this bobbing motion, the reference trajectory was defined as a sine wave:

$$x(t) = A \sin \omega t + B, \tag{3.28}$$

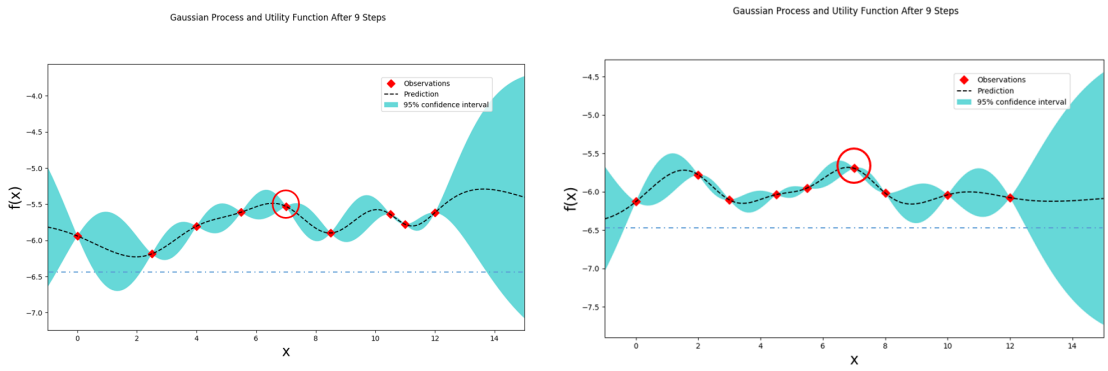
where A is the amplitude, ω is the frequency of the sine wave, and B is the vertical offset of the bobbing motion. The value of B equals 0 when the stretch force of the bungee cord equals the sum of the gravitational force and the frictional forces from transmissions. This makes the value of B the same as the default stretched length of the bungee cord.

The bungee cord has a diameter of 8 mm at rest and its stiffness can be seen in Fig. 3.10. The amplitude A and the frequency ω were fixed in order to find the optimal default stretched length of the bungee cord which minimizes the energy consumption via Bayesian optimization. For each trial of Bayesian Optimization 10 rollouts were ran, (recall that one rollout is one period of the sinusoidal motion). The response surface of the Bayesian optimization is the average evaluated energy consumption defined by Eq. 3.20. The range of B is $[0, 12]$ cm and the relaxed length of the bungee cord is 40 cm. The data is collected sequentially: one trial is performed and the energy consumption is evaluated, then the sample is attached to the data set and fed to the optimizer. The optimizer computes the posterior and gives the value of the B parameter for the next sample using the acquisition function defined by Eq. 3.27. In our experiments I use $\kappa = 5$ for the collection of the first 5 trials, which means that the optimization process encourages exploration. κ is then changed to 1.5 for later sample collections so that the optimization would prefer exploitation of gathered data.



(a) (b)

Figure 3.12: Trajectories of the vertical prismatic joint on the left leg in the experiment. The orange line indicates the measured position from the motor while the blue line indicates the desired position. (a) shows the slow motion with a frequency of 2.5 Hz and (b) shows the fast motion with a frequency of 5 Hz.



(a) (b)

Figure 3.13: Results of the Bayesian Optimization after 9 iterations. (a) shows results of the 2.5 Hz motion and (b) shows results of the 5 Hz motion. The x axis is the vertical offset B and the y axis is the negative value of the evaluation of energy consumption (because the Bayesian Optimization maximizes the objective function). The blue dashed line is the borderline which is the energy consumption without the bungee cord and the red circle indicates the optimal solution.

Experiments were performed with two different frequencies representing fast and slow motion. The value of ω in the two experiments is 2.5 $rads^{-1}$ and 5 $rads^{-1}$ respectively, the resultant input and motor trajectories can be seen in Fig. 3.12. Experimental results show that with bungee cords attached in parallel, the energy efficiency is improved significantly for both fast and slow motion cases, as shown in Fig. 3.13. In all trials conducted, the energy consumption of the robot with bungee cords attached is smaller than the borderline (energy consumption without attaching bungee cords). In the case of slow motion which has the frequency of 2.5 HZ, the optimal solution computed from only 9 iterations of Bayesian Optimization, saves 15.5% of SLIDER's energy consumption compared to the borderline. In the case of fast motion which has the frequency of 5 HZ, the optimal solution saves 11.65% of energy consumption compared to the borderline. In both cases, the optimal default stretched length of bungee cords is around 7 cm, indicating that the same bungee cords configuration can be applied onto various speeds. The experimental results are summarized in Table 3.5.

Table 3.5: Summary of The Experiment Results.

Frequency (Hz)	2.5	5
Energy Consumption without Bungee Cord (J)	65.43	64.45
Optimal Energy Consumption with Bungee Cord (J)	55.29	56.94
Optimal Default Stretched Length (cm)	6.8	7
Improvement of Energy Consumption (%)	15.5	11.65

3.4.5 *Discussion*

The novel design of SLIDER is leveraged to mount compliant elastic elements (bungee cords) in parallel with prismatic actuation mechanisms, as opposed to current approaches that only consider elastic elements in series, or in parallel with revolute joints. Bayesian Optimization is used to find the optimal configuration of the bungee cords because of the nonlinear dynamics of the elastic elements and various sources of friction. The optimal solution found by only 9 trials is up to 15% more energy-efficient compared to the robot configuration without parallel elastic elements.

In the future, the focus of experiments will be optimizing bungee cords configurations for SLIDER walking and jumping. Another future direction will be co-optimizing the walking motion together with the bungee cords.

Part III

CONTROL AND PLANNING FOR SLIDER

CHAPTER 4

Control of SLIDER

As is shown in the roadmap of my thesis in Chapter 1, to make the robot walk and jump, a hierarchical structure is used which is composed of high-level planning, mid and low-level control to track the generated motion plan. In this chapter the mid-level and low-level control are discussed. Firstly, the preview control which is widely used because of its practical and easy implementation in the real robot is introduced. Then, the inverse-dynamics based whole body controller is discussed. Finally, the results of real robot walking experiments are presented.

4.1 PREVIEW CONTROL

Preview control (Kajita et al., 2003) is used to generate the CoM trajectory given the Zero Moment Point (ZMP) reference given the LIP dynamics, which is planned without taking into account the dynamic properties of the robot. The dynamics of LIP is 4.1, where x, y are CoM x and y coordinates and z_c is the CoM height which is constant, p_x, p_y are ZMP x and y coordinates.

$$\begin{aligned} \ddot{x} &= \frac{g}{z_c}(x - p_x) \\ \ddot{y} &= \frac{g}{z_c}(y - p_y) \end{aligned} \tag{4.1}$$

The basic idea of preview control is to discretize the continuous LIP dynamics and rewrite into the discrete-time system using the jerk of CoM as input and the ZMP as output:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + Bu(k)p(k) = C\mathbf{x}(k) \tag{4.2}$$

where

$$\mathbf{x} = \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \end{bmatrix}, A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \tag{4.3}$$

The problem then is reduced to an output tracking problem given reference ZMP trajectory. This is solved by formulating an unconstrained QP with the following cost function and optimal solution:

$$\begin{aligned} \mathbf{J} &= \sum_{n=1}^{\infty} Q_e^2 + \Delta x(i)^T Q(x) \Delta x(i) + R \Delta u(i)^2 \\ u(k) &= -G_i \sum_{i=0}^{k} -G_x x(k) - \sum_{i=1}^{N_L} G_p(j) p^{ref}(k+j) \end{aligned} \tag{4.4}$$

where $e(k) = p(k) - p^{ref}(k)$ is the output tracking error, $\Delta x(k) = x(k) - x(k-1)$ is the incremental state vector, $\Delta u(k) = \Delta u(k) - \Delta u(k-1)$ is the incremental input,N_L denotes the future target references up to N_L steps, the weighting matrices Q_e, Q_x, R are symmetric and positive definite. In our implementation, Qe and Qx are identity matrix.Since the acceleration can be changed instantaneously, the weighting R is only 10^{-6}.

Note that the incremental control input in the performance index J introduces an integral action which eliminates the tracking error $e(k)$ and is known as Linear Quadratic Integral (LQI) technique. The detailed derivation of the optimal gain matrices G_i, G_x, G_p can be found in (KATAYAMA et al., 1985) and the values can be easily calculated using MATLAB with less than 20 lines of code. The workflow of a preview controller can be shown in Fig. 4.1.

4.1.1 *First Iteration of Preview Control*

At each time step, the optimal jerk can be calculated based on Equation 4.4, which will then be used to compute the desired CoM position at next time step using 4.1. Since the CoM position cannot be used to solve the inverse kinematics, it is necessary to obtain the desired pelvis position as well. Assuming the difference between the pelvis and CoM position is known at the current time step, the desired pelvis position can then be obtained by adding this difference to the desired CoM location. This is followed by performing the inverse kinematics to obtain the joint positions, which will be further used to compute the link positions and the actual CoM location. Finally, the difference between the pelvis and CoM

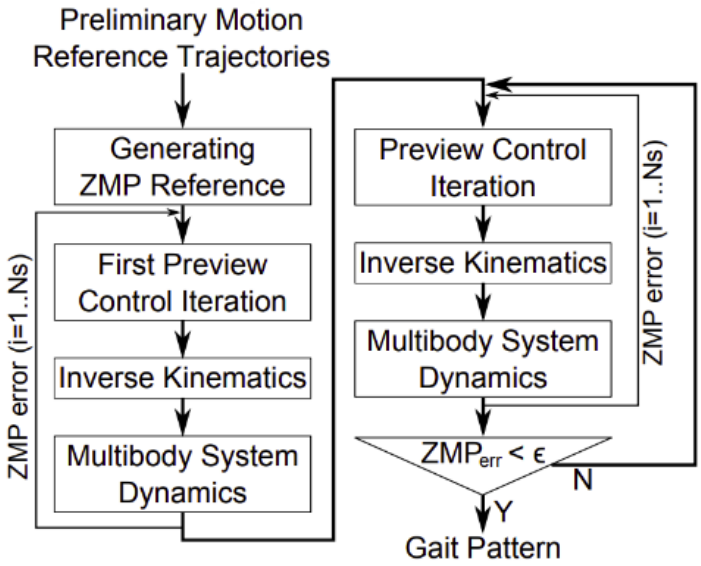


Figure 4.1: Workflow of preview control.

position is updated for the next time step using the desired pelvis position and the actual CoM location. After this whole process, the joint positions at each time step will be obtained.

However, the first iteration of preview control does not utilizes the dynamics of the robot, which implies that the generated joint trajectory may not be directly applied to the real robot.

4.1.2 *Remaining Iterations of Preview Control*

In the remaining iterations, the preview control can correct the trajectory generated with the first iteration by considering the Multi-Body System (MBS) dynamics. The MBS dynamics neglects the effect of the inertia tensor of each link about the robot's center of mass and assumes the robot is composed of point masses. With this simplified dynamics, the ZMP of the robot can be computed with Equation 4.5. Therefore, by using forward kinematics to update the position of each link, a new ZMP can be computed. The update of ZMP is performed at each iteration. The ZMP difference between two adjacent iterations is denoted as δp. The preview control is terminated until δp is below the given threshold. In practice the difference is negligible after only three iterations.

$$p_x = \frac{\sum_{i=1}^{N} m_i[(\ddot{z}_i + g)x_i - (z_i - p_z)\ddot{x}_i]}{\sum_{i=1}^{N} m_i(\ddot{z}_i + g)}, p_y = \frac{\sum_{i=1}^{N} m_i[(\ddot{z}_i + g)y_i - (z_i - p_z)\ddot{y}_i]}{\sum_{i=1}^{N} m_i(\ddot{z}_i + g)}, \quad (4.5)$$

4.1.3 *Implementation Details*

Because of the inherent unstable dynamics of bipedal robots, the frequency of the controllers (both high-level and low-level) should be as fast as possible to generate a smooth trajectory and react quickly to disturbances. For preview control, the sampling time is a major factor affecting the computational speed. This is directly related with the size of the optimal gain matrix G_d and the number of computations performed at each preview iteration. The time step for the high-level LIP model is usually 1ms. In practice 100 Hz is not able to generate smooth joint trajectory for real robot in the low-level. In preview control the frequency is 1K Hz.

4.2 INVERSE DYNAMICS WHOLE BODY CONTROLLER

The use of preview control as a dynamics filter discussed in the previous section is a traditional way to generate a dynamically consistent joint trajectory for a position controlled robot. However this approach does not utilize the full potential of robot dynamics and has several issues. For example, the foot may hit the ground ahead of the planned time and the robot can be easily unstable. When the robot walks fast, the fast swing motion of the leg generates extra angular momentum and makes the robot unstable. Therefore a controller that can fully use the potential of the robot dynamics is needed.

The inverse dynamics based whole body controller uses the dynamics of the robot to track the desired tasks from the high-level planner while respecting a set of constraints. This controller is applicable to a torque controlled robot.

4.2.1 *Task Formulation*

The commonly used tasks of interest are the CoM position, the pelvis orientation, the angular momentum of the robot, the foot positions and orientations. Each task is comprised of a desired acceleration as a feed-forward term and a state feedback term to stabilize the trajectory. Generally, the task for the linear motion can be expressed as:

$$\mathbf{J}_T \ddot{\mathbf{q}} = \ddot{\mathbf{x}}^{cmd} - \dot{\mathbf{J}}_T \dot{\mathbf{q}},$$

$$\ddot{\mathbf{x}}^{cmd} = \ddot{\mathbf{x}}^{des} + \mathbf{K}_P^{pos}(\mathbf{x}^{des} - \mathbf{x}) + \mathbf{K}_D^{pos}(\dot{\mathbf{x}}^{des} - \dot{\mathbf{x}}),$$

where $\mathbf{J}_T$ is the translational Jacobian for the task, $\mathbf{x}$ is the actual position of the link, and the superscript des indicates the desired motion.

For the task of angular motion, the command can be formulated as:

$$\mathbf{J}_R \ddot{\mathbf{q}} = \dot{\boldsymbol{\omega}}^{cmd} - \dot{\mathbf{J}}_R \dot{\mathbf{q}},$$

$$\dot{\boldsymbol{\omega}}^{cmd} = \dot{\boldsymbol{\omega}}^{des} + \mathbf{K}_P^{ang}(\text{AngleAxis}(\mathbf{R}^{des}\mathbf{R}^T)) + \mathbf{K}_D^{ang}(\boldsymbol{\omega}^{des} - \boldsymbol{\omega}),$$

where $\mathbf{J}_R$ is the rotational Jacobian for the task, $\mathbf{R}$ and $\mathbf{R}^{des}$ denote the actual and desired orientation of the pelvis link respectively, AngleAxis() maps a rotation matrix to the corresponding axis-angle representation, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the link.

To track the momentum of the robot, the angular momentum formulation is discussed at first. The angular momentum can be computed from the velocity with Equation 4.6:

$$\mathbf{h} = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \tag{4.6}$$

where $\mathbf{A}_G$ is the centroidal momentum matrix (Orin et al., 2013) and $\mathbf{h}$ contains the linear and angular momentum expressed at CoM. Differentiating Equation 4.6 yields Equation 4.7

$$\dot{\mathbf{h}} = \mathbf{A}_G\ddot{\mathbf{q}} + \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \tag{4.7}$$

The momentum task can therefore be written as Equation 4.8:

$$\mathbf{A}_G\ddot{\mathbf{q}} + \dot{\mathbf{A}}_G(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{F}_{des} \\ \dot{\mathbf{h}}_{ref} + K_p(\mathbf{h} - \mathbf{h}_{ref}) \end{bmatrix} \tag{4.8}$$

For balancing or walking, angular momentum task are often defined as a damping task that damps out excess angular momentum. However for highly dynamic motion such as twisting jump, angular momentum varies a lot during the process and it plays a vital role to achieve the jump motion. In this case, the reference angular momentum is needed and it comes from our motion planner.

4.2.2 *Contact Constraint*

Friction is a important contributing factor to the stability of walking to prevent the foot from slipping. The criterion is that the contact force should satisfy the inequality below,

$$f_z \geqslant 0, \quad \sqrt{f_x^2 + f_y^2} \leqslant \mu f_z \tag{17}$$

where μ is the friction coefficient. However, this formulation is a quadratic inequality constraint and rises the problem of constraining fast optimization. Therefore, the pyramidal friction model is used which is a linear inequality constraint for fast optimization:

$$f_z \geqslant 0, \quad |f_x| \leqslant \frac{\mu f_z}{\sqrt{2}}, \quad |f_y| \leqslant \frac{\mu f_z}{\sqrt{2}} \tag{18}$$

4.2.3 *QP formulation*

Inspired by (Herzog et al., 2015), the full dynamics can be decomposed into the underactuated part and actuated part:

$$\begin{bmatrix} \mathbf{M}_f \\ \mathbf{M}_a \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{H}_f \\ \mathbf{H}_a \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{S}_a \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{J}_f^T \\ \mathbf{J}_a^T \end{bmatrix} \mathbf{f},$$

where $\mathbf{M}$, $\mathbf{H}$, $\mathbf{S}_a$, $\boldsymbol{\tau}$, $\mathbf{J}$ and $\mathbf{f}$ are the mass matrix, Coriolis force matrix and gravitation force vector, the actuator selection matrix, joint torques vector, the stacked contact Jocabian and reaction force vector. The subscript, f and a, indicates the floating part and actuated part respectively. The weighted sum formulation is applied, in which one QP problem is solved at each control loop. The formulation of the QP problem can be written as

$$\begin{aligned} \min_{\ddot{\mathbf{q}}, \mathbf{f}} \quad & \frac{1}{2}\|\mathbf{A}\ddot{\mathbf{q}} + \dot{\mathbf{A}}\dot{\mathbf{q}} - \mathbf{B}^{cmd}\|_{W_1}^2 + \frac{1}{2}\|\mathbf{f} - \mathbf{f}_{des}\|_{W_2}^2 && (4.9) \\ \text{s.t.} \quad & \mathbf{M}_f\ddot{\mathbf{q}} - \mathbf{J}_f^T\mathbf{f} = -\mathbf{H}_f && \text{(floating base dynamics)} \\ & \mathbf{P}\mathbf{f} \leqslant 0 && \text{(friction cone)} \\ & \mathbf{S}_a^{-1}(\mathbf{M}_a\ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T\mathbf{f}) \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}] && \text{(input limits)} \end{aligned}$$

where $\mathbf{A}$ is a stack of the Jacobian matrices for the tasks of interest, $\mathbf{B}^{cmd}$ is a stack of the commanded accelerations and $\mathbf{W}_i$ $(i = 1, 2)$ are the weighting matrices, $\mathbf{P}$ denotes the linearized friction cone matrix. Similar to (Apgar et al., 2018)(Kuindersma et al., 2016) , the unilateral contact constraint is treated as a soft constraint by simply assigning a large weight on the desired zero acceleration. It is reported in (Feng et al., 2014) that this gives a better stability.

The output torque commands $\boldsymbol{\tau}$ at each control iteration is computed by

$$\boldsymbol{\tau} = \mathbf{S}_a^{-1}(\mathbf{M}_a\ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T\mathbf{f}) \tag{4.10}$$

4.2.4 *Implementation and Discussion*

Although the original quadratic problem can now be passed into the QP solver, it is still not enough to make the whole body controller work. The first problem is that even when the weight matrix in Equation 4.9 is chosen to be positive definite, the Hessian matrix is only positive semidefinite because the objective function does not contain the other two optimization variables τ and f. Hence, the optimization problem is not strictly convex and is difficult to converge fast to the expected solution. The trick is to add a diagonal matrix with small positive values to make the Hessian matrix positive definite. This is essentially minimizing the 2 norm of each optimization variable. Since the additional positive values on the diagonal are very small, they will not affect the final result and make the solver successfully find the expected solution.

Another issue is that SLIDER robot does not have hip yaw joint and this means that the angular acceleration of base hip yaw has to be constrained. Constraining the other entries does not yield a satisfying improvement on the performance and it is quite hard to quantify the limits on the accelerations.

The third issue is tuning the gains. There are different objectives in the whole body controller and their PD gains vary in different tasks. However in all walking tasks, it is very important to track the swing foot because in practice a foothold position with around 5 cm error in z axis will make the robot fall down. Therefore high PD gains are assigned to track the swing foot and relative low PD gains to track the stand foot.

I wrote the whole body controller with C++ using the Pinocchio library (Carpentier et al., 2015–2021) to compute full rigid body dynamics and qpOASES (Ferreau et al., 2014) to solve the QP problem. For SLIDER robot the whole body controller runs at 1k Hz to track the desired trajectory. Based on our measurement it takes only less than 0.05 ms for Pinocchio to compute the dynamics and most of time is spent on solving QP problems. The controller is tested with the SLIDER robot in Gazebo simulation.

Compared with the preview control, the inverse dynamics based whole body controller has the advantage of being able to keep dynamically stable instead of statically stable. This can result in many interesting behaviours and one of the most significant is the ankle compliance. As seen in Fig. 4.2, with small PD gains to track the ankle orientation, the heel strike motion can be generated naturally by the whole body controller. Another case is the ankle can adapt to height variations when walking blindly on uneven terrains, which is demonstrated in Fig. 5.6.

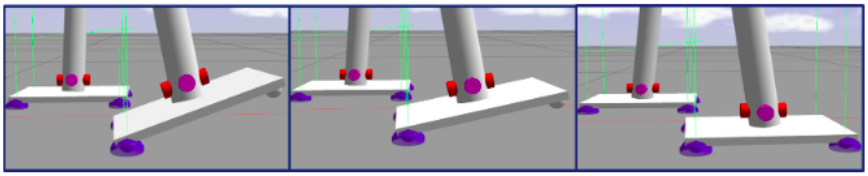


Figure 4.2: The whole body controller can generate heel strike motion.

4.3 REAL ROBOT EXPERIMENTS

Because SLIDER has light-weight straight legs, the LIP model can be used for generating flat terrain walking motion. The inverse kinematics is adjusted for SLIDER's special design. To make physical SLIDER walk, trajectory compensation and feedback control are necessary. The control structure of real SLIDER is shown in Fig. 4.3. The walking gait is generated offline by preview control which is composed of two stages. The first stage accepts the reference CoM velocity as the input and uses the simplified LIP mode to generate the CoM and ZMP reference trajectories. In the second stage a preview controller calculates optimal joint trajectories with a dynamic filter. The dynamic filter uses MBS dynamics to make the generated gait more dynamically feasible for the real robot. It uses the reference trajectories of the CoM and ZMP from the first stage to generate the final walking pattern.

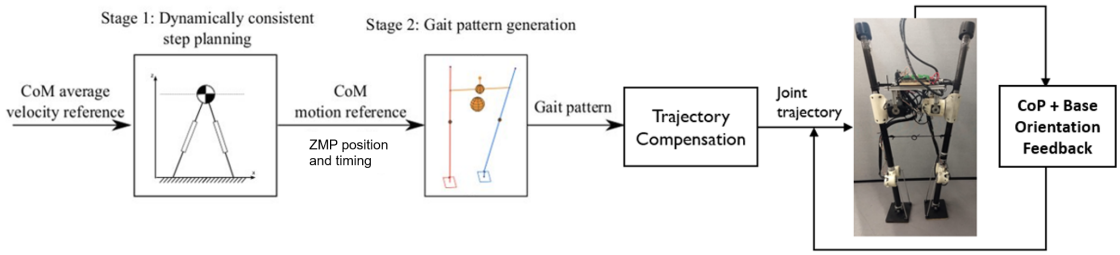


Figure 4.3: The control diagram of SLIDER. The original joint reference trajectory is generated offline by preview control, then the trajectory is compensated for backlash. The trajectory is tracked with CoP and base orientation feedback on the real robot.

The physical SLIDER uses a trajectory tracked at a frequency of 500 Hz. Due to the backlash and deformation of hip pitch and hip roll joints, tracking offline generated trajectory doesn't result in stable walking. To achieve stable walking on the physical SLIDER, both feedforward compensation and feedback controller for stablization are used. SLIDER v4 has a relatively large backlash, three compensations were added to the offline generated trajectory to help counteract various mechanical imperfections:

1. Linear Offset - Constant offsets were added to precisely transit from the calibrated home position (orthogonal axes at each joint with slide at a reference location) to the neutral position (static, balanced standing, which serves as the starting position for

walking gaits). The offsets can be adjusted to account for imprecise calibration of the home position.

2. Non-Scheduled Scaling - Because of the lightweight construction, SLIDER has large loaded deformations in its joints, which introduces error between the requested and actual positions during movements. To adjust for these errors, the commanded changes in position can be linearly scaled. The hip pitch joints make most use of this compensation to counteract the particularly high deformation present.
3. Scheduled Scaling - To allow scaling of commanded position changes during only certain portions of the walking gait, scheduled scaling is used. Due to loaded deformation which almost exclusively occurs during single support phase, the commanded roll positions of the support leg are scaled during single support phase only.

For SLIDER v5, the backlash is not as big as SLIDER v4 and there is no need to add feedforward compensation.

During early walking tests, non-flat foot impacts with the floor often caused loss of balance. It was determined that a control strategy would need to be applied, utilizing the real data about the CoP to adjust the ankle angles. Load cells are used to measure the CoP and calculate the actual ZMP in single and double support phase, based on the algorithm proposed by (Kajita et al., 2001a). A PD feedback control law is used to track the desired ZMP in both phases. The base of the robot tilts due to the backlash and the deformation of 3D printed parts, therefore the IMU is added to measure the orientation of the base and use a PD control to regulate the base orientation.

4.3.1 *Feedback Signals*

There are two feedback signals in the controller for real robot experiments, the CoP and base orientation. The CoP is estimated with the load cells and the base orientation is estimated with Bosch BNO055 IMU mounted on the base. As shown in Fig. 4.8 four load cells are mounted on the four corners of the bottom of the foot and connected with the 1046 Phidget Bridge. An illustration can be seen at Fig. 4.5.

The load cells output the value of contact force in z axis and the CoP can be calculated as:

$$p_x = \alpha \frac{F_{12} - F_{03}}{F}, p_y = \beta \frac{F_{01} - F_{23}}{F} \tag{4.11}$$

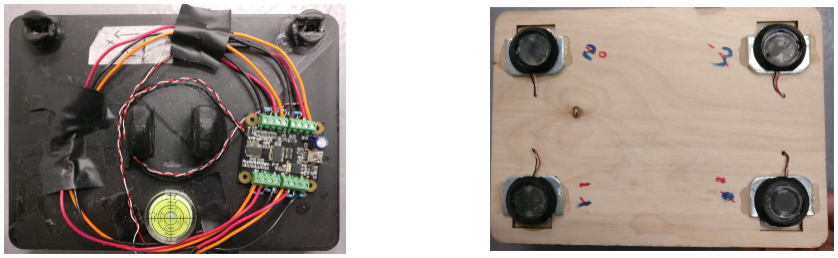


Figure 4.4: Top (left) and bottom (right) view of the SLIDER's contact sensors mounted on the foot.

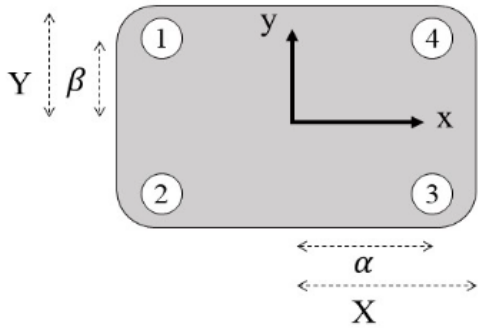


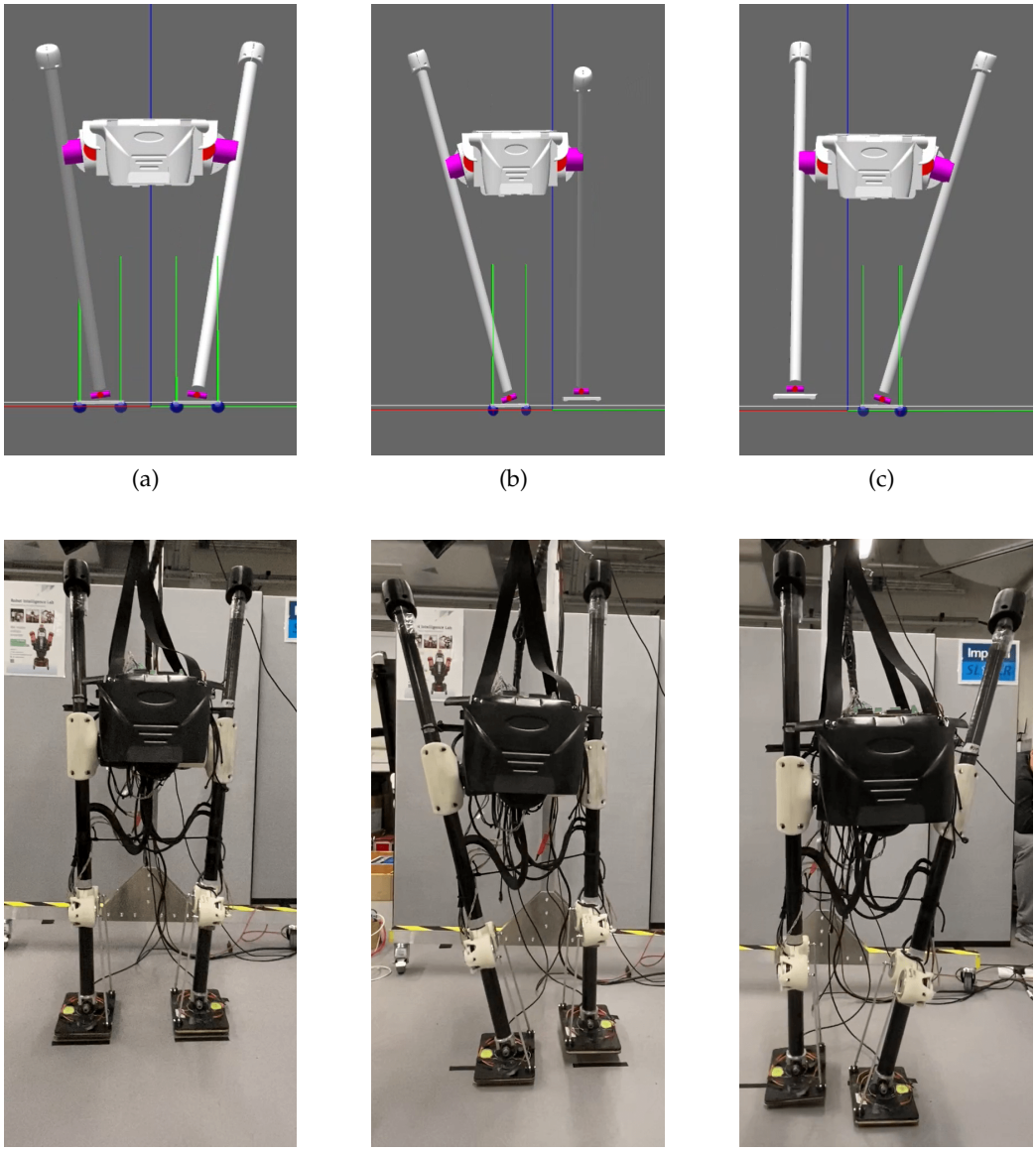
Figure 4.5: Contact sensor positions. α and β are distances between contact sensor and the center of foot in x and y axis.

where α and β are distances between contact sensor and the center of foot in x and y axis. F is the sum of all four sensor values, F_{12} is the sum of values of sensor 1 and sensor 2 and so on. The preview control generates the reference ZMP trajectory. The ZMP can be transformed into local coordinate of the foot and it is the same as CoP when the robot is in single support phase. Our controller only does CoP feedback in single support phase.

4.3.2 *Walking Experiment Results*

With a combination of offline trajectory generation and online CoP feedback control, SLIDER v4 walks with a speed of 0.10 m/s, see Fig 4.7.

Fig. 4.8 shows the energy consumption with time of SLIDER v5 walking for 15 cycles and Table 4.1 summarizes the key data of energy consumption. As can be seen from the table, the average power is only 12.36 watt and the peak power is 97.13 watt.



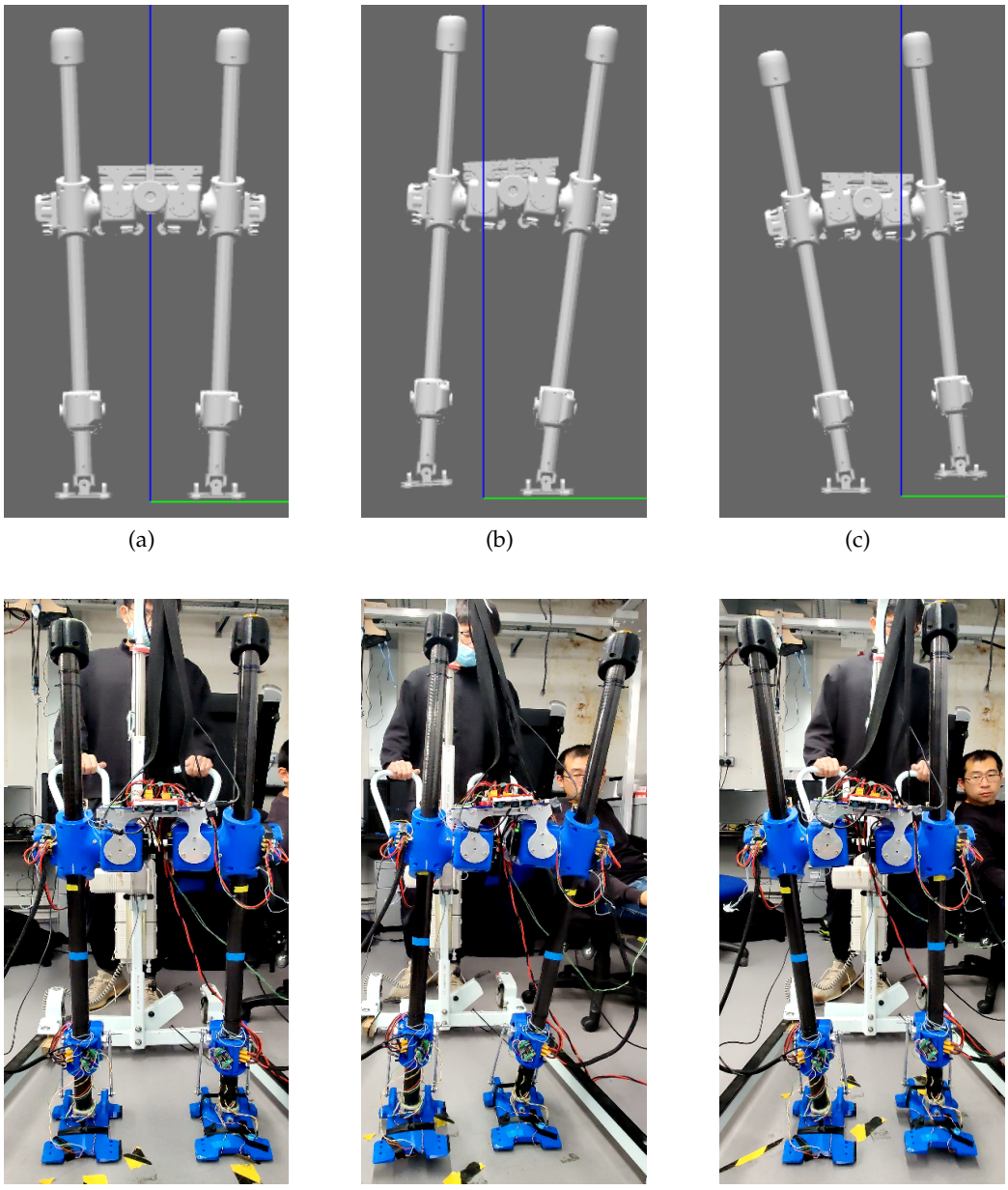
(a) (b) (c)

(d) (e) (f)

Figure 4.6: The top 3 snapshots show SLIDER v4 walking in simulation, with green vectors indicating the ground reaction forces. *From left to right:* double support phase, single support phase with right leg as support leg, single support phase with left leg as support leg. The bottom 3 photos show the physical SLIDER robot walking on flat floor. *From left to right:* same order as the top images.

Table 4.1: Summary of Energy Consumption during SLIDER v5 Walking.

Peak Power (W)	97.13
Average Power (W)	12.36
Overall Energy Consumption (J)	246.746



(a) (b) (c)

(d) (e) (f)

Figure 4.7: The top 3 snapshots show SLIDER v5 walking in simulation, with green vectors indicating the ground reaction forces. *From left to right:* double support phase, single support phase with right leg as support leg, single support phase with left leg as support leg. The bottom 3 photos show the physical SLIDER robot walking on flat floor. *From left to right:* same order as the top images.

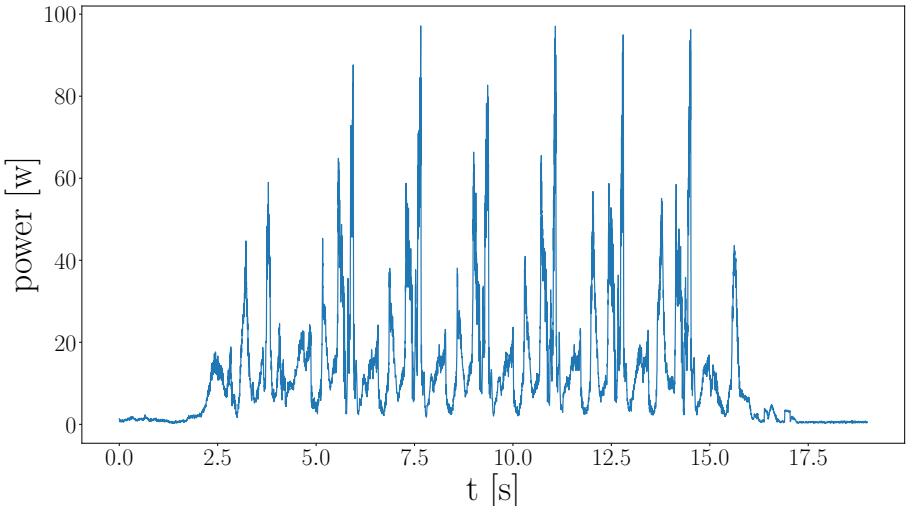
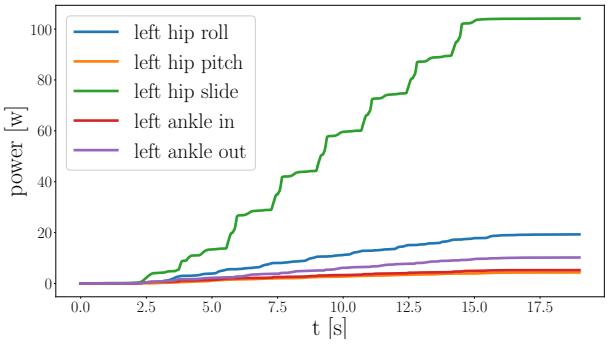
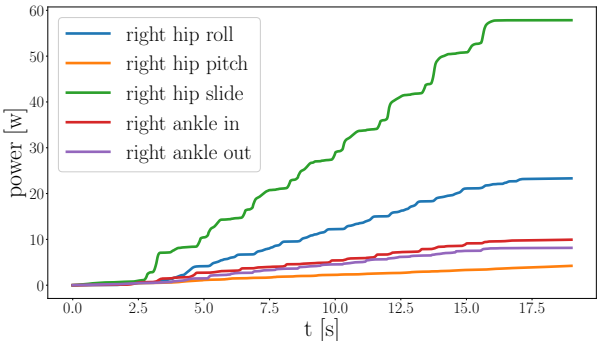


Figure 4.8: The power of SLIDER v5 robot walking with forward velocity of 0.1m/s for 15 cycles.



(a)



(b)

Figure 4.9: The energy consumption of joints in SLIDER v5 during a 15 walk cycles with a forward velocity of 0.1m/s. Hip slide joint consumes the highest energy due to the fact that it supports the base weight all the time. Hip roll joint consumes second highest energy because this joint supports whole leg and base weight during the single support phase.

CHAPTER 5

Real-Time Footstep Planning for Terrain-Blind Walking

As is shown in the roadmap in Chapter 1.4, footstep planning gives high-level plans (CoM, foot) that are tracked by controllers. This chapter presents an online footstep planning algorithm that enables bipedal robots to blindly walk over various kinds of uneven terrains while resisting pushes. The proposed optimization algorithm performs high level motion planning of footstep locations and center-of-mass height variations using the decoupled aSLIP model. The decoupled aSLIP model simplifies the original aSLIP with LIPM dynamics in horizontal states and spring dynamics in the vertical state. The motion planning can be formulated as a discrete-time MPC problem and solved at a frequency of 1 kHz. The output of the motion planner is fed into an inverse-dynamics based whole body controller for execution on the robot. A key result of this controller is that the feet of the robot are compliant, which further extends the robot's ability to be robust to unobserved terrain variations. The method is evaluated in simulation with the bipedal robot SLIDER. Results show the robot can blindly walk over various uneven terrains including slopes, wave fields and stairs. It can also resist pushes of up to 40 N for a duration of 0.1 s while walking on uneven terrain.

5.1 INTRODUCTION

To make bipedal robots really suitable for many applications, it is important that they can go out of the lab and walk in the complex real world environment. Real world environments contain various kinds of uneven terrains: slopes, stairs, hills, etc. Most existing controllers that allow a bipedal robot to walk over uneven terrains require predefined footstep locations or exact information about the terrain height variations (Mordatch et al., 2010, Englsberger et al., 2015, Liu et al., 2015). However, even with most advanced sensors there are some uncertainties on the perception of the terrain. In contrast, humans can easily walk on uneven

terrains, such as outdoor environments. without extra thought or careful planning. Therefore, it is important to have a reactive controller that is robust to unobserved uneven terrain variations.

The SLIP model has become a popular model for walking and running in the legged robotic research (Raibert, 1986). Despite its simplicity, it has been proven to capture essential dynamics properties of walking and running (Geyer et al., 2006). The standard setting of SLIP model is energy-conservative: it assumes there is no energy loss at impact. Though this assumption simplifies the control analysis, it does not resemble the reality. There is energy loss on physical systems, and robots (Ahmadi et al., 2006, Robotics, 2017a) designed to approximate SLIP dynamics have added actuation to compensate for the energy dissipation. As a result, the aSLIP model (Ernst et al., 2010) is proposed for a better approximation of the real robot dynamics. The aSLIP model has been successfully used to design controllers not only for SLIP-like robots (Apgar et al., 2018, Green et al., 2020, Xiong et al., 2018) but also as a template model for humanoid robots on uneven terrain walking (Liu et al., 2016).

An important step in making a controller reactive is to achieve real-time constrained optimization. (Wieber, 2006, Xin et al., 2019) used the simpler LIP model to design a reactive controller for flat ground walking, but this model cannot be applied to walking on uneven terrains. However, due to the nonlinear dynamics that arise from the 3D aSLIP model, fast optimization is difficult. (Liu et al., 2016) used a gait synthesized from a library of gaits acquired from off-line optimization, but this requires a large computation load offline and cannot cover all possible situations. (Apgar et al., 2018) decoupled 3D aSLIP model to facilitate fast computation, but the continuous dynamics of decoupled aSLIP model is still nonlinear and there is no theoretical guarantee of fast convergence.

This chapter proposes a reactive controller that enables robots to blindly walk over uneven terrains by optimizing horizontal footstep locations and CoM height online. "Terrain-blind" means the robot is not provided with the local map that specifies terrain height, but the robot can get access to its own states through state estimation, for example CoM height. Under the assumption that the angle the modelled inverted pendulum makes with the vertical is relatively small, the 3D aSLIP model can be decoupled into a 1D actuated spring model responsible for z direction and 2D LIP model responsible for x and y directions. The dynamics of all three dimensions can be written as linear equations in a discrete-time state space manner. The online step planner is formulated as a discrete-time MPC problem and solve it by QP. To facilitate fast computation, the spring length is constrained to change linearly. As a result, a step planner runs at a frequency of 1000 HZ. The step planner using a simple model is embedded into the inverse-dynamics based whole body controller (Herzog et al., 2015, Kim et al., 2019) which tackles the inconsistency between the simple model used in high level planning, and full robot dynamics. With the whole body controller, the feet of the robot show great compliance and this helps the robot to transition between different

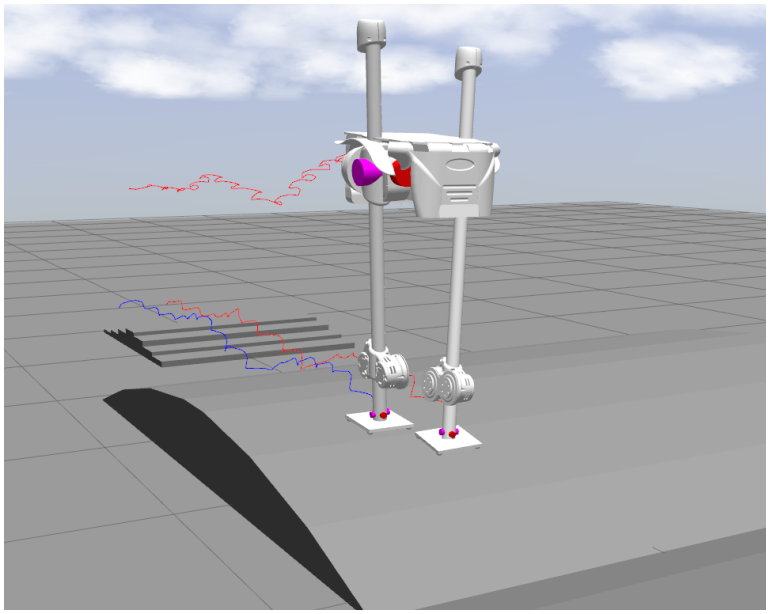


Figure 5.1: SLIDER robot walks on uneven terrain. The three traces are trajectories of CoM, left foot and right foot respectively.

terrains without any information about the terrain. Due to the fast execution frequency and compliance of the foot, our proposed controller enables the robot to blindly walk over various kinds of moderately uneven terrains including slopes, wave fields and stairs, as shown Figure 5.1. Our controller can also handle disturbances from all direction while walking on uneven terrain. Our controller is validated on the straight-legged bipedal robot SLIDER (Wang et al., 2020a) in Gazebo simulation.

The main contribution of the chapter is the fast online optimization algorithm which enables bipedal robots to blindly walk over various uneven terrains while resisting disturbances. With reasonable assumptions, the nonlinear 3D aSLIP dynamics can be decoupled and the footstep location and CoM height can be optimized separately. By proper reformulation, the control of the vertical dynamics can be formulated as QP and therefore a solution is guarenteed to be found. Our controller is simple to implement and computationally efficient; the optimization of both vertical and horizontal motions can be solved by QP and executed at 1 kHz[1].

1 Video summary: https://youtu.be/R0yV-ZP8dxA

5.2 SYSTEM OVERVIEW

5.2.1 *The Control Hierarchy*

The controller presented has a hierarchical structure as shown in Fig 5.2. Due to the computational complexity of using full body motion planning, a high level planner optimizes online only the foot placement in x, y directions and the CoM height using the decoupled aSLIP model. The low level whole body controller (Herzog et al., 2015, Feng et al., 2015) tracks the trajectory generated by high level planner. The whole body controller considers the full dynamics of the robot and generates the consistent torque command for each joint.

5.3 ONLINE FOOTSTEP PLANNING

Motion planning with the full dynamics of the robot is too computationally expensive to be executed at a fast frequency. Instead reduced order models can be used for online footstep planning: the LIP model is used to plan footsteps in the horizontal plane and the 1D actuated spring model is used generate in the vertical direction.

5.3.1 *The Decoupled aSLIP Model*

The aSLIP model is different from the classical SLIP model in that the aSLIP model is a combination of the SLIP model and a virtual linear actuator, as shown in Figure 5.3. Introducing such a virtual actuator gives the aSLIP model the capability to actively modify the reference spring length during each walking phase and makes aSLIP model better at handling vertical changes than the SLIP model. However, similar to the SLIP model, the dynamics of aSLIP model presented above is nonlinear because of the coupling in vertical and horizontal dynamics, and the exact dynamic equation needs to be numerically integrated. An approximate model which decouples the CoM states into horizontal and vertical directions will be used in this chapter. This decoupled approximate model simplifies the nonlinear dynamics of the original aSLIP model and makes the fast online optimization of footstep locations and CoM height possible. To make the approximation valid, two assumptions are made. The first assumption is that the vertical deviation of CoM over each walking period is small relative to the height of the CoM, such that individual steps can be modelled using the LIP model. The second assumption is that the angle the inverted pendulum makes with the vertical is small, such that the spring contributes to the CoM's vertical behavior only. This is reasonable, considering a typical small step length is small compared to the height of the CoM. These assumptions are summarised in Figure 5.3.

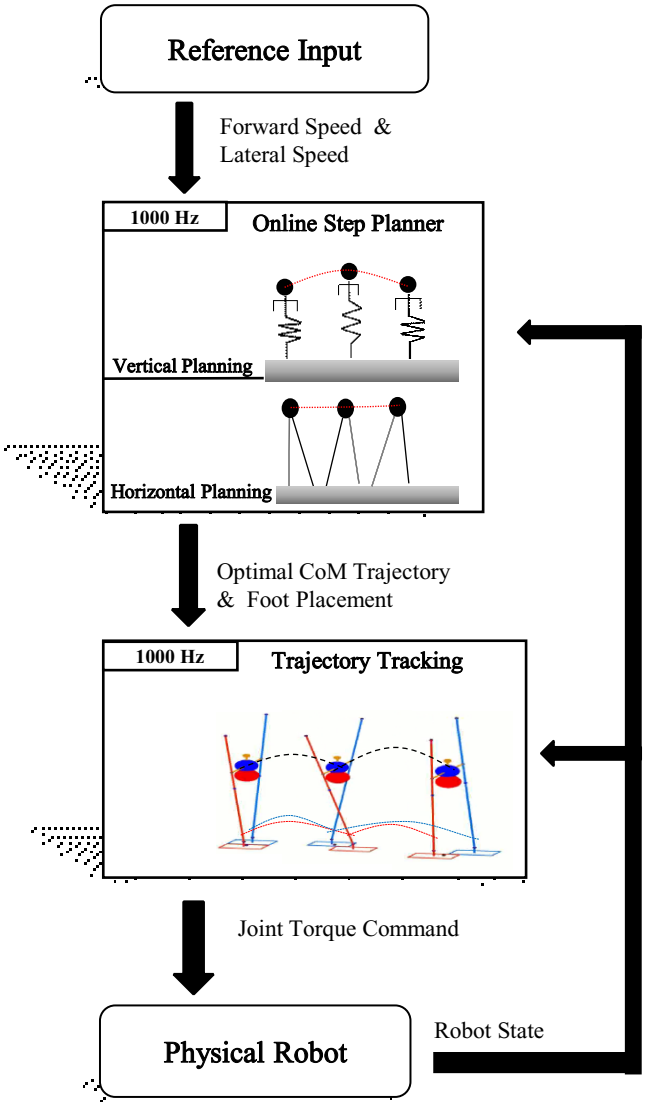


Figure 5.2: The hierarchical controller with the execution frequency of each level. The high level trajectory planner takes the desired velocity as input and generates the optimal CoM trajectory along with the foot placement. The low level whole body controller (Herzog et al., 2015, Feng et al., 2015) considers full dynamics of the robot and tracks the trajectory at a frequency of 1 kHz.

5.3.2 *Vertical Dynamics of decoupled aSLIP Model*

The original dynamics equation of the SLIP model is

$$\begin{bmatrix} m\ddot{l} \\ ml^2\ddot{\theta} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2 - k(l - l_0) - mg\cos\theta \\ -2m\dot{l}\dot{\theta} + mgl\sin\theta \end{bmatrix} \tag{5.1}$$

where l is the pendulum length and θ is the angle between the pendulum and the vertical axis, as demonstrated in Figure 5.3. Under the assumption that θ is negligible and l can be well approximated by the CoM height z, the dynamics can be simplified into the decoupled aSLIP model as shown in Figure 5.4. The dynamics in vertical direction Equation A.1, where r denotes the reference spring length, and k is the stiffness of the spring.

$$m\ddot{z} = -mg + k(z - r) \tag{1}$$

Compared with the classical SLIP model, the reference spring length $r(t)$ is treated as a time-varying optimization variable, which is contributed by the virtual linear actuator. $r(t)$ can be constrained to change linearly between initial and final values over the phase to facilitate fast computation. Therefore, the reference spring length should satisfy the condition below,

$$r(t) = r_0 + \frac{t}{T}(r_T - r_0) \tag{2}$$

where, r_0 and r_T represents the reference spring length at the start and at the end of one step respectively, T is the step duration. The dynamic Equation (A.1) can be formulated into a state space equation as,

$$\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} (r - \frac{g}{\omega_z^2}) \tag{3a}$$

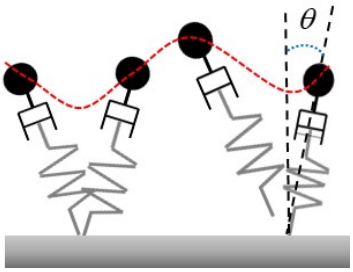


Figure 5.3: The aSLIP model. This model combines the SLIP model with a virtual linear actuator. θ is the angle the inverted pendulum makes with the vertical.

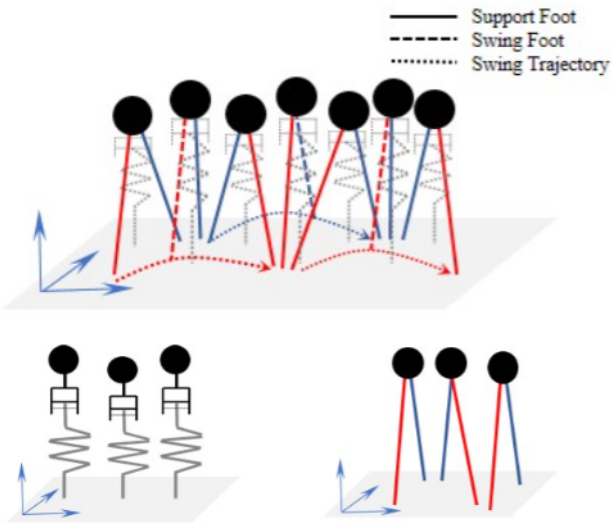


Figure 5.4: The decoupled aSLIP model during walking (top), with vertical dynamics (bottom left) and horizontal dynamics (bottom right).

where the states of CoM in vertical direction are defined as $Z = [z, \dot{z}]^T$ and $\omega_z = \sqrt{\frac{k}{m}}$. Further, $r - \frac{g}{\omega_z^2}$ can be denoted as u_z, and Equation (7.5) can be written into a linear state space equation as,

$$\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} u_z \tag{3b}$$

Equation (7.5) can be discretized with sampling time T_s and can obtain,

$$Z_{k+1} = \Phi(T_s) Z_k + \int_0^{T_s} \Phi(T_s) dt \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} u_{z,k} \tag{4a}$$

where,

$$\Phi(T_s) = \begin{bmatrix} \cos(\omega_z T_s) & \sin(\omega_z T_s)/\omega \\ \omega_z \cdot \sin(\omega_z T_s) & \cos(\omega_z T_s) \end{bmatrix} \tag{4b}$$

$$\int_0^{T_s} \Phi(T_s) dt \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} = \begin{bmatrix} 1 - \cos(\omega_z T_s) \\ \omega_z \cdot \sin(\omega_z T_s) \end{bmatrix} \tag{4c}$$

This equation can also be written as:

$$Z_{k+1} = A_z(T_s)Z_k + B_z(T_s)u_{z,k} \tag{4d}$$

where Z_k and Z_{k+1} are CoM states in the vertical direction at time k and k+1 respectively, $A_z(T_s)$ and $B_z(T_s)$ are time-varying matrices, $u_{z,k}$ is the control input for the spring at time k. Therefore, N future steps can be computed given fixed sampling time T_s:

$$\begin{bmatrix} Z_1 = A_z(T_s)Z_0 + B_z(T_s)u_{z,0} \\ Z_2 = A_z(T_s)Z_1 + B_z(T_s)u_{z,1} \\ \dots \\ Z_N = A_z(T_s)Z_{N-1} + B_z(T_s)u_{z,N-1} \end{bmatrix} \tag{5a}$$

where N is the number of steps to be optimized which should be more than 1. In this formulation, the control input vector for the spring $\bar{U} = [u_{z,0}, u_{z,1}, \dots u_{z,N-1}]$ is the optimization variable. Equation (5a) can also be written in a matrix form for MPC:

$$\bar{Z} = \begin{bmatrix} A_z \\ A_z^2 \\ \dots \\ A_z^N \end{bmatrix} Z_0 + \begin{bmatrix} B_z & 0 & 0 & 0 \\ A_z B_z & B_z & 0 & 0 \\ \dots & \dots & \dots & \dots \\ A_z^{N-1} B_z & A_z^{N-2} B_z & \dots & B_z \end{bmatrix} \bar{U}_z \tag{5b}$$

where $\bar{Z}$ and $\bar{U}_z$ are the vectors of states and control inputs in z direction.

5.3.3 *Horizontal Dynamics of Decoupled aSLIP Model*

Under the assumption that the leg angle θ is relatively small and therefore the horizontal and the vertical dynamics can be decoupled, the horizontal dynamics then becomes a classical LIPM. Similar to (Xin et al., 2019), given the continuous dynamics of LIPM in the x direction,

$$\ddot{x} = \frac{g}{z_0}(x - p_x) \tag{6}$$

the discrete state space equation can be reformulated into a linear state space as below, with state defined as $X = [x, \dot{x}]^T$, and input u_x defined as the footstep positions in x direction.

$$X_{k+1} = A_x(T_s)X_k + B_x(T_s)u_{x,k} \tag{7a}$$

Where

$$A_x(T_s) = \begin{bmatrix} \cosh(\omega_x T_s) & \sinh(\omega_x T_s)/\omega_x \\ \omega_x \cdot \sinh(\omega_x T_s) & \cosh(\omega_x T_s) \end{bmatrix} \tag{7b}$$

$$B_x(T_s) = \begin{bmatrix} 1 - \cosh(\omega_x T_s) \\ \omega_x \cdot \sinh(\omega_x T_s) \end{bmatrix} \tag{7c}$$

Similarly, states in the predicted time horizon can also be obtained by,

$$\bar{X} = \begin{bmatrix} A_x \\ A_x^2 \\ \dots \\ A_x^N \end{bmatrix} X_0 + \begin{bmatrix} B_x & 0 & 0 & 0 \\ A_x B_x & B_x & 0 & 0 \\ \dots & \dots & \dots & \dots \\ A_x^{N-1} B_x & A_x^{N-2} B_x & \dots & B_x \end{bmatrix} \bar{U}_x \tag{8}$$

where $\bar{X}$ and $\bar{U}_x$ are the vectors of states and control inputs in x direction. The dynamics in the y direction has identical formulation to the dynamics in the x direction.

5.3.4 *Foot Placement and CoM Trajectory Optimization*

The footstep planning can be formulated as a QP problem due to the fact that both horizontal and vertical dynamics are linear and the constraints are linear. The formulation is:

$$\min_{u_x, u_y, u_z} \Gamma \qquad \text{(cost function)} \tag{9a}$$

$$\text{s.t.} \quad X_{k+1} = A(T_s)X_k + B(T_s)u_k \qquad \text{(dynamics)} \tag{9b}$$

$$h(\mathbf{p}_j) < 0 \qquad \text{(reachability)} \tag{9c}$$

where $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$ is the cost function term, X and u are general representations of states and control inputs respectively and $\mathbf{p}_j$ is the left or right foot position.

The cost formulation of the vertical and horizontal control problems are both composed of the three parts: Γ_1, Γ_2 and Γ_3. The first part is minimizing the difference between the predicted state and the referenced state, which drives the CoM state to reach the desired one from the given current state. The formulation is:

$$\Gamma_1 = \|X_N - X_N^{ref}\|_P^2 + \sum_{k=0}^{N-1} \|X_k - X_k^{ref}\|_Q^2 \tag{10}$$

where P and Q are the weight matrix respectively. For the horizontal trajectory planning, only the velocity of CoM in both directions are cared about, because a reactive footstep planner which is not constrained by a absolute reference trajectory is needed. From another point of view, tracking the reference velocity is equivalent to tracking the relative CoM reference position. X^{ref} is easy to define with the desired sagittal and frontal velocity. However, the reference velocity in vertical direction is not constant during one step. To get the reference velocity, Equation (A.1) can be integrated and the continuous dynamics equation can be derived as:

$$z(t) = d_1 \cos \omega_z t + d_2 \sin \omega_z t + r(t) - g/{\omega_z}^2 \tag{11a}$$

where

$$d_1 = z_0 - r_0 + g/{\omega_z}^2 \tag{11b}$$

$$d_2 = \dot{z}_0/{\omega_z}^2 - (r(t) - r_0)/(T\omega) \tag{11c}$$

In this equation, z_0 and $\dot{z}_0$ are the CoM vertical position and velocity at the start of the current step respectively. Then the time-varying reference velocity in z direction can be derived from the continuous dynamics.

The second part is minimizing the difference between the reference control input u_k^{ref} and the optimized control input u_k and R is the weight matrix.

$$\Gamma_2 = \sum_{k=0}^{N-1} \|u_k - u_k^{ref}\|_R^2 \tag{12}$$

For the reference control input in z direction, recall that the spring length is constrained to change linearly, as indicated by Equation (7.2). If Δr_i is denoted as the change of spring

length during one sampling time T_s at step i, then it is clear that Δr_i is constant throughout step i, and the reference control input in z direction $\bar{U}_z^{ref}$ is therefore obtained as below,

$$\bar{U}_z^{ref} = \begin{bmatrix} r_0 I_{N_r} \\ 0_{N_s} \\ \dots \\ 0_{N_s} \end{bmatrix} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} 0_{N_r} \\ \Delta r_1 I_{N_s} \\ \dots \\ \Delta r_{N_{steps}} I_{N_s} \end{bmatrix} \tag{13}$$

where N_r is the remaining number of sampling points for the current step and N_s is the number of sampling points for one step. N_{steps} is the number of predicted steps.

For the horizontal trajectory planning, the input of the system is the footstep position. The reference control input in x direction $\bar{U}_x^{ref}$ is then obtained according to the current support foot position P_0 and the difference between two consecutive steps $\Delta P_{x,i}$.

$$\bar{U}_x^{ref} = \begin{bmatrix} I_{N_r} \\ 0_{N_s} \\ \dots \\ 0_{N_s} \end{bmatrix} P_{x,0} + \begin{bmatrix} 0_{N_r} & \dots & 0_{N_r} \\ I_{N_s} & \dots & 0_{N_s} \\ \dots & \dots & \dots \\ 0_{N_s} & \dots & I_{N_s} \end{bmatrix} \begin{bmatrix} P_{x,1} \\ P_{x,2} \\ \dots \\ P_{x,N_{steps}} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} P_{x,1} \\ P_{x,2} \\ \dots \\ P_{x,N_{step}} \end{bmatrix} = \begin{bmatrix} I_{N_r} \\ I_{N_s} \\ \dots \\ I_{N_s} \end{bmatrix} P_{x,0} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta P_{x,1} \\ \Delta P_{x,2} \\ \dots \\ \Delta P_{x,N_{step}} \end{bmatrix}$$

The third part is tracking the desired change of the control input between two consecutive steps, with d_i denoting the desired difference and W denoting weights. For the vertical trajectory planning the desired difference is set to equal zero, this means the footstep planner tries to keep a constant CoM height during one step to make the decoupled aSLIP assumption valid.

$$\Gamma_3^z = \sum_{i=1}^{N_{steps}} \|\Delta r_i - d_i^z\|_W^2 = \sum_{i=1}^{N_{steps}} \|\Delta r_i\|_W^2 \tag{15a}$$

For horizontal trajectory planning,

$$\Gamma_3^{x,y} = \sum_{i=1}^{N_{steps}} \|\Delta P_i^{x,y} - d_i^{x,y}\|_W^2 \tag{15b}$$

In x direction, d_i^x is the step length which is calculated by the desired speed multiplied by the step time. In y direction, d_i^y is defined as $d_i^y = d_{step} * 2 * (-1)^j$, where d_{step} is the desired inter-feet clearance distance, and j is the flag for the supporting foot, 0 stands for left support and 1 stands for right support. This relative distance regularization term is introduced to keep the feet away from each other to avoid self-collision. Morever, as this part of the cost function only includes relative distances, it helps to produce a reactive footstep planner which keeps the robot walking even when unexpected disturbance is applied.

The reachability constraint is responsible for making sure the footstep location is physically possible,

$$\mathbf{L} - \mathbf{r}^{x,y} < \mathbf{p}_j - \mathbf{c} < \mathbf{L} + \mathbf{r}^{x,y} \tag{16}$$

where $\mathbf{L}$ is the nominal offset of the foot position from the CoM of the robot, $\mathbf{r}^{x,y}$ is the reachability constraint in the x and y directions and $\mathbf{c}$ is the CoM position.

5.4 TRAJECTORY TRACKING

The low level trajectory tracking generates corresponding torques for each joint to minimize the difference between the actual body trajectory and the desired trajectory given by the high level trajectory planner. Since the trajectory planning does not include the full body dynamics, the dynamic inconsistency of the torque command generated by the high level planner is significant. The whole body controller solves the inverse dynamics based on the full robot dynamics.

5.4.1 *Rigid Body Dynamics*

The walking robot is modeled as a floating based rigid body system with coordinates $q = [q_b, q_r]$. Here, $q_b \in \mathbb{R}^7$ represents the position and orientation of the floating base using quaternions, and $q_r \in \mathbb{R}^{10}$ represents the joint configuration. Inspired by (Herzog et al., 2015), the full dynamics can be decomposed into an underactuated part and an actuated part:

$$\begin{bmatrix} M_f \\ M_a \end{bmatrix} \ddot{q} + \begin{bmatrix} H_f \\ H_a \end{bmatrix} = \begin{bmatrix} 0 \\ S_a \end{bmatrix} \tau + \begin{bmatrix} J_f^T \\ J_a^T \end{bmatrix} f, \tag{16}$$

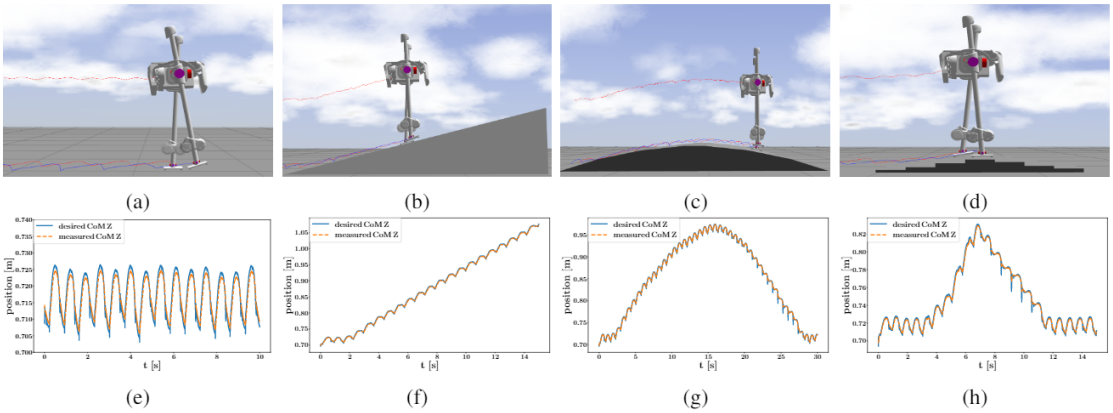


Figure 5.5: Snapshots and plots of SLIDER blindly walking on different kinds of terrains, with a constant forward velocity of 0.3 m/s (a ~ c) and 0.6 m/s (d). On top row, from left to right: (a) ~ (d): Walk on flat terrain. Walk on a slope with an angle of 15 degrees. Walk on a wave field approximated by slopes with 3 different angles: 15, 10 and 5 degree. Walk over stairs, the successive elevation changes are +2 cm, +2 cm, +3 cm, +3 cm, -2 cm, -3 cm, -2 cm, -3 cm. On bottom row, from left to right, (e) ~ (f): The plots of desired and measured CoM z position in corresponding scenarios.

where $\mathbf{M}$, $\mathbf{H}$, $\mathbf{S}_a$, $\boldsymbol{\tau}$, $\mathbf{J}$ and $\mathbf{f}$ are the mass matrix, the Coriolis force vector and the gravitation force vector, the actuator selection matrix, the joint torques vector, the stacked contact Jacobian and the reaction force vector respectively. The subscript f and a, indicates the floating part and the actuated part respectively.

5.4.2 *Contact Force Constraint*

Friction is a important contributing factor to the stability of walking to prevent the foot from slipping. The criterion is that the contact force should satisfy the inequality below,

$$f_z \geqslant 0, \quad \sqrt{f_x^2 + f_y^2} \leqslant \mu f_z \tag{17}$$

where μ is the friction coefficient. However, this formulation is a quadratic inequality constraint and rises the problem of constraining fast optimization. Therefore, the pyramidal friction model is used which is a linear inequality constraint for fast optimization:

$$f_z \geqslant 0, \quad |f_x| \leqslant \frac{\mu f_z}{\sqrt{2}}, \quad |f_y| \leqslant \frac{\mu f_z}{\sqrt{2}} \tag{18}$$

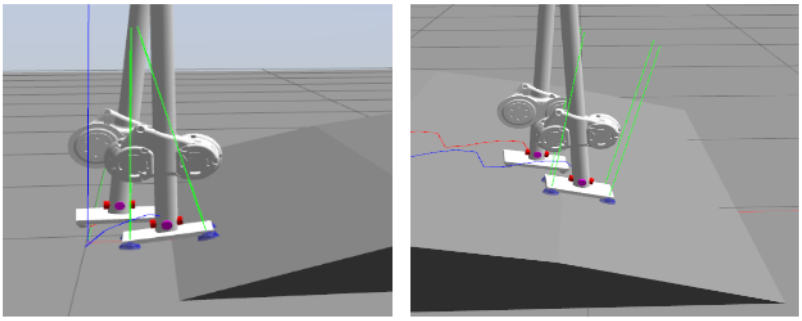


Figure 5.6: With the whole body controller the foot is compliant and can do transitions to different terrains even the robot doesn't know how the terrain looks like. This makes the robot robust to unobserved terrain variations.

5.4.3 *Swing Foot Trajectory Generation*

The swing foot trajectory is generated using a fifth order polynomial. The start and final positions, velocities and accelerations are specified and a parametric quintic curve is generated in x, y and z directions. The polynomial in z direction has two halves, with the predefined foot height and zero velocity at the midpoint. The start and final positions in z direction are calculated using the estimated CoM z position at the start of current step subtracted by a constant offset. Because the robot is walking blindly a trajectory for the foot orientation is not specified, and this allows the foot to be compliant to a range of surfaces.

5.5 RESULTS

This section discusses implementation details and simulation results of SLIDER robot walking on different kinds of uneven terrains including slopes, wave fields and stairs. All the experiments are shown in detail in the accompanying video.

5.5.1 *Implementation*

Both the high level footstep planner and low level trajectory tracking controller are implemented in C++ for real-time performance. In the whole body controller Pinocchio (Carpentier et al., 2015–2021) is used to compute the full rigid body dynamics and qpOASES (Ferreau et al., 2014) to solve the related QP problem in both levels. All experiments were carried out in robot simulation environment Gazebo (Koenig et al., 2004) with the physics engine ODE (Drumwright et al., 2010), using the full dynamics of the real SLIDER robot. The communication through different levels of the control hierarchy is achieved through ROS.

The decoupled aSLIP parameters are set to match with the physical SLIDER robot, here r_0 = 0.715 m, m = 14.5 kg, k = 1470 N/m. The step duration is chosen to be 0.7 s and the footheight in the swing foot trajectory generation is 5 cm. The sampling time in the discrete-time MPC is 0.1 s in x and y directions and 0.05 s in z direction. The motion planner predicts 4 steps in horizontal states and 1 step in the vertical state. The same parameters are used among all walking experiments except that the forward velocity is different. In the stair walking experiment the robot has to walk faster otherwise the foot might hit the edge of the stair.

5.5.2 *Flat Ground Walking*

Our approach is firstly validated on flat ground. Because there are no changes in the z direction on flat ground, the z position of CoM oscillates around r_0, as shown in Figure 6a, 6e.

5.5.3 *Walking on Smooth Uneven Terrain*

Our approach is then validated on slopes and wave fields where the change of terrain height is smooth. With a forward velocity of 0.3 m/s, the SLIDER robot can walk on a slope with 15 degree and a wave field approximated by slopes with 3 different angles: 15, 10 and 5 degrees, as shown in Figure 6b, 6c, 6f, 6g. In the experiment the whole body controller plays an important role in making the robot remain robust to unobserved terrain variations. By properly tuning the PD gain of the foot orientation task in the whole body controller, the foot is compliant to adapt to large terrain variations, as shown in Figure 5.6.

Further experiments like pushing the robot while it was walking on uneven terrains were performed. As shown in Figure 5.7, the robot was pushed in x or y direction twice while it walking on a wave field. When the robot was pushed in x direction, the robot quickly took a large step in x direction to regulate the CoM velocity back to the desired velocity. There is also a big change in z direction because the wave field is ascending in x direction when the robot got pushed, but the motion quickly got stabilized. In the case of y direction push, the robot also took a large step in y direction and stabilized in one step. As the terrain height is only changing along x direction, there is no big change in z direction in this case.

our proposed planner is also compared with the planner with LIP, as shown in Figure 5.8. The robot with LIP planner is unable to walk up a 15° slope as LIP planner assumes a constant CoM height, as shown in the bottom plot in Figure 5.8. The aSLIP planner enables SLIDER to walk up because it is adaptive to height changes.

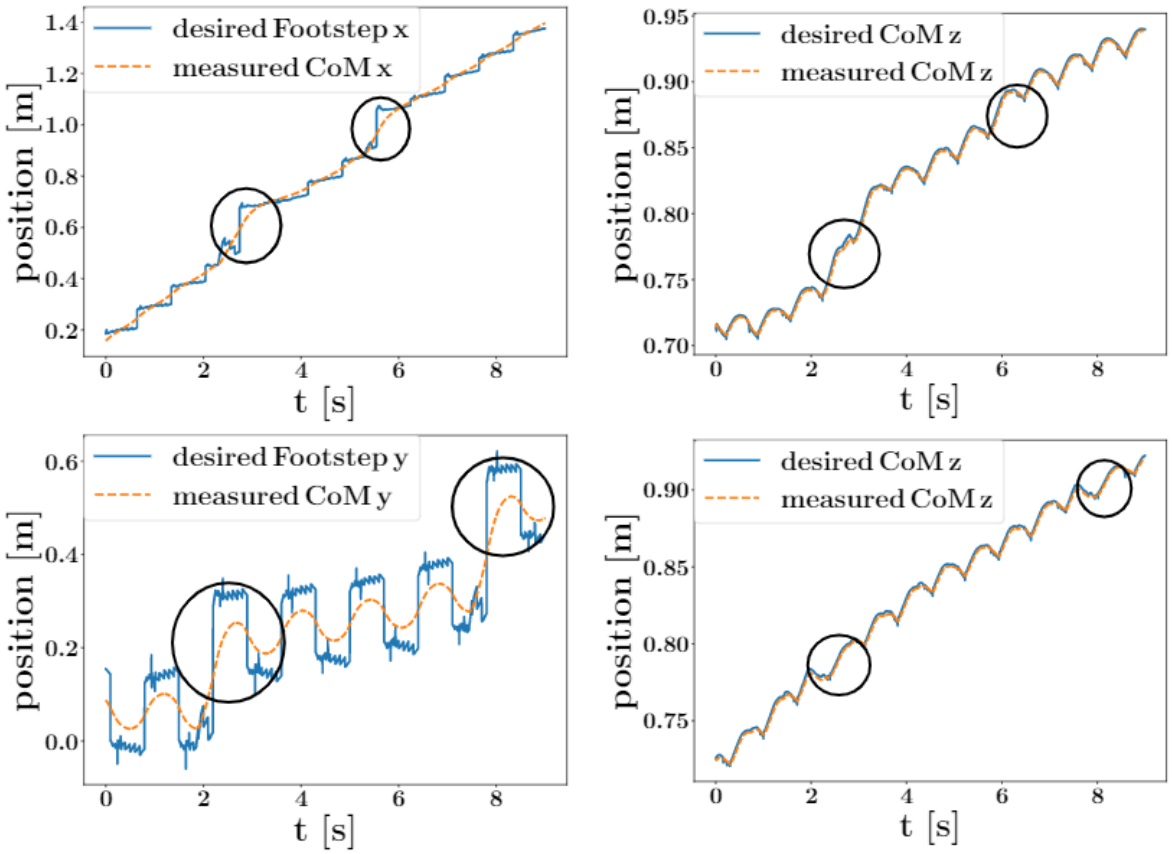


Figure 5.7: Plots showing the states of CoM when the robot got pushed in x or y direction while walking on a wave field at a speed of 0.3 m/s. The pushes are indicated with circles. All impulses were applied with a value of 40 N for a duration of 0.1 s. Top row: states of CoM when got pushed in x direction. Bottom row: states of CoM when got pushed in y direction.

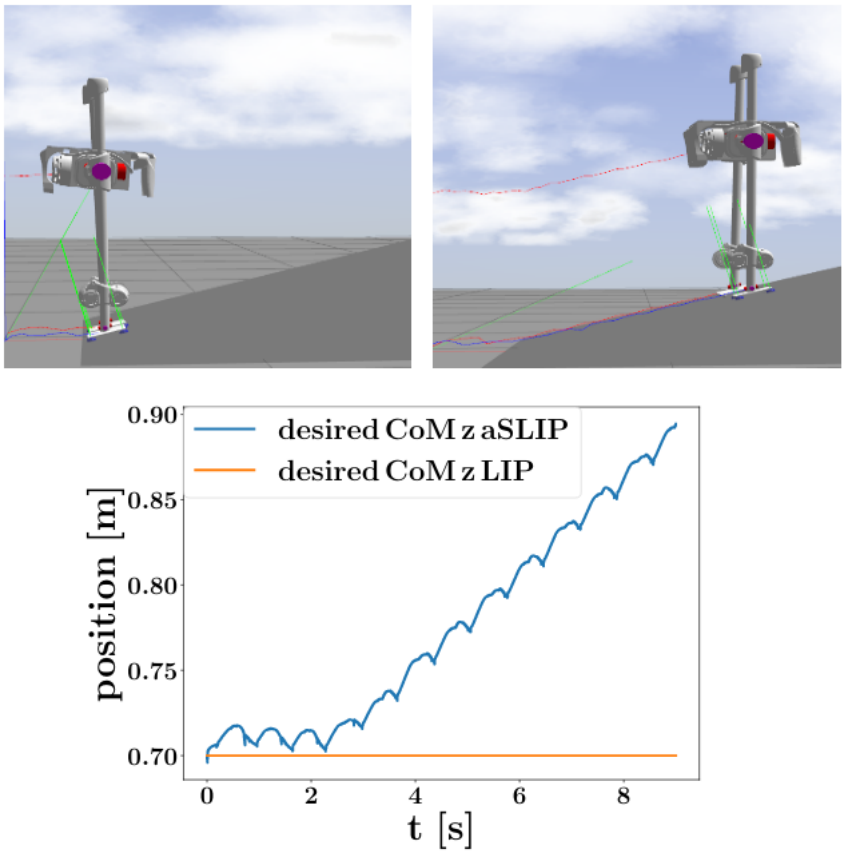


Figure 5.8: Comparison between proposed aSLIP planner with LIP planner when walking on a 15° slope. Top left is the picture showing SLIDER walking with LIP planner, the robot is unable to walk forward after a short distance because LIP does not take height change into account. Top right is the picture showing SLIDER walking with aSLIP planner and the robot is able to walk successfully. Bottom is the plot showing the comparison between desired CoM height of LIP and aSLIP planner.

5.5.4 *Walking on Discrete Uneven Terrain*

Discrete uneven terrain provides bigger instantaneous variations to terrain height compared with smooth uneven terrain. As shown in Figure 6d, the robot walked blindly over a set of stairs with a biggest height change of 3 cm. The robot walked with a forward velocity of 0.6 m/s and a foot height of 5 cm in swing foot phase. As shown in Figure 6h the CoM Z position exhibits large variations when walking over stairs but was stabilized quickly, which demonstrates the robustness of our approach. In our experiment the highest steps the robot can walk over is 3 cm, for higher steps the foot would hit the edge of the stair and get stuck.

5.6 DISCUSSION

The proposed methods is successfully applied to the SLIDER robot and demonstrates its performance of blind walking on various uneven terrains and robustness to disturbances. Because of SLIDER's lightweight legs, the robot can perform fast leg movements which helps to stabilize the robot. Furthermore, the proposed method is general and can easily be applied to other legged robots.

There are other techniques dealing with variable height CoM trajectories, for example DCM based approaches (Englsberger et al., 2013, Caron et al., 2017). But these techniques require the controller to be terrain-aware to plan future CoM motion. Our proposed controller adjusts the CoM height online within each step by following a spring dynamics. This has two advantages: firstly, the controller doesn't require terrain information; secondly, the spring dynamics enables the vertical compliance of the robot so that the robot is robust to unexpected height variations.

The author observed that the tracking error is larger when the robot walks down the wave field or stairs than walking up, as shown in Figure 6g, 6h. This happens because the feet are still in the air at the end of one step when the robot walks down. The unexpected sudden drop of the feet at the start of next step gives the robot a large impact. For robot walking upwards, the effect of early touchdown can be alleviated by the compliant feet and vertical compliance of the CoM. To improve, a controller with contact detection or terrain information can achieve a smaller tracking error.

CHAPTER 6

Terrain-Aware Real-Time Gait Optimization

Online footstep planning is essential for bipedal walking robots, allowing them to walk in the presence of disturbances and sensory noise. Most of the literature on the topic has focused on optimizing the footstep placement while keeping the step timing constant. Last chapter talks about terrain-blind footstep planning. In this chapter, a footstep planner is introduced capable of optimizing both footstep placements and step timing online with terrain-awareness. The proposed planner, consisting of IPOPT and an optimizer based on the AL method with analytical gradient descent, solves the full dynamics of the LIP model in real time to optimize for footstep location as well as step time simultaneously at a rate of 200 Hz. It is shown that such asynchronous real-time optimization with the AL method (ARTO-AL) provides the required robustness and speed for successful online footstep planning. Compared to an algorithm with no footstep time adaptation, ARTO-AL demonstrates increased stability in simulated walking experiments as it can resist pushes on flat ground and on a 10° ramp up to 120 N and 100 N respectively. Furthermore, ARTO-AL can be extended to plan footsteps in 3 dimensions, which allows terrain-aware planning on uneven terrain.

6.1 INTRODUCTION

In order for legged robots to dynamically balance themselves in the presence of disturbance, it is essential for them to plan footstep positioning and duration in real time. Much of existing work focuses on optimizing footstep locations while keeping step timing fixed, according to the dynamics of LIP model (Motoi et al., 2009, Kajita et al., 2010, Wieber, 2006). The LIP model is a widely used reduced-order model of the dynamics of a bipedal walking robot; the legs are assumed to be massless, and the body is a concentrated point mass located at the robot's CoM and the height of the CoM with respect to the ground is assumed to be constant

(Kajita et al., 2001). Using this model, online footstep planning can be characterised by a real-time nonlinear MPC problem. Due to the second order differential equation governing the LIP model, optimizing footstep duration with respect to time produces a non-linear optimization problem which is difficult to solve in real time. Thus, real-time footstep timing adaptation work has largely been focused on simplifying the dynamics used in optimization, or carefully tailoring heuristic solutions.

One popular method to simplify the control problem is to consider only the DCM of the body and predict only 'one step' ahead. By only considering the DCM of the next step, dynamics are reduced to a first order problem, where the optimal step time to control the DCM about a nominal point can be computed with QP (Khadiv et al., 2016, Khadiv et al., 2020). DCM can also be combined with the virtual repellent point (Englsberger et al., 2015) to encode all forces on a robot's body in one single point. (Englsberger et al., 2017) controls the DCM in this way by interpolating the virtual repellent point between desired locations. However, the DCM only presents a simplified version of the LIP dynamics, potentially limiting the control capabilities and versatility of footstep planning. Further to this, a reference trajectory for the DCM must be specified, which is generated heuristically (Khadiv et al., 2016).

Another approach is to optimize the full system dynamics, only considering the CoM state at each footstep. (Kryczka et al., 2015) achieves this, interpolating between regenerated gait patterns when a disturbance is detected. (Ding et al., 2019) also only considers the CoM state at each step, but reformulates the nonlinear optimization problem to a sequential quadratic programming problem by writing the nonlinear terms directly as decision variables. However, although (Ding et al., 2019) is fast, the decision variables, including footstep positions require references to track - this again requires heuristics provided by the algorithm designer and may not represent the optimal behaviour.

Rather than simplifying the dynamics of the system, an alternative method is to simplify the optimization itself. Such approaches focus on generating approximate solutions that are iteratively improved with each control step. Real Time Iteration (RTI), introduced in (Diehl et al., 2002) is an online method for approximately solving MPC problems, where the solution is found by first linearising the system at its current state, then performing one QP step (Diehl et al., 2005). RTI has been shown to significantly reduce computation time when solving nonlinear MPC problems (Gros et al., 2020). (Neunert et al., 2017) followed this approach and applied nonlinear MPC to control a quadruped robot's dynamic motion. However, their optimization is unconstrained and requires careful engineering. This has been improved recently by (Mastalli et al., 2020a) with adding 'soft' constraints - penalty terms included in the cost function to shape it such that its' optima satisfy constraints. Constraints can more concretely be handled by using the AL method (POWELL, 1969), which 'augments' the standard Lagrange formulation with an added quadratic component. This method has

gained popularity in robotics due to its fast convergence and numerical robustness (Howell et al., 2019).

Terrain-aware (also known as terrain-adaptive) locomotion has recently started attracting growing attention. (Wu et al., 2010) has achieved terrain-aware locomotion for a 3D simulated humanoid. A gait library is built offline that includes 5 periodic gaits and a set of transition gaits. The gait library is parameterized so that the desired pose planned in real-time based on the height map can choose the right gait to reach the goal. (Jenelten et al., 2020-10) has demonstrated terrain-aware walking on the quadruped robot ANYmal by tightly integrating terrain perception for foothold planning. By generating the elevation map in real time and integrating the map's information into the optimization for the footholds, ANYmal robot can climb stairs and obstacles of heights up to 33 % of the robot's leg length. (Xie et al., 2020, Da et al., 2019) have shown impressive results of uneven terrain walking with Cassie robot. However, their controller doesn't use terrain information, as it relies on proprioception (such as IMU and joint encoder signals) to complete a locomotion task.

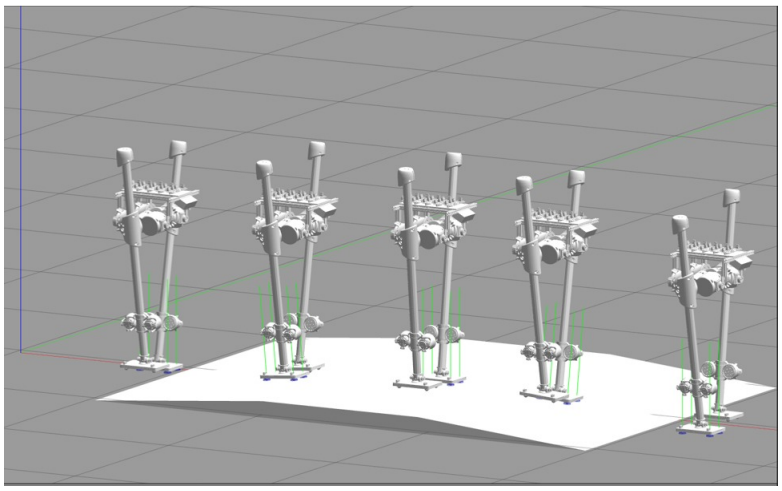


Figure 6.1: SLIDER walks over slopes with ARTO-AL with perception.

In this chapter the use of the augmented Lagrangian method is investigated to solve the terrain-aware footstep planning of bipedal walking as a nonlinear MPC in real time, as shown in Fig. 6.1. Our contributions are as follows:

- a footstep planner is proposed that adapts both footstep location and timing in real time. Instead of using simplified dynamics or doing local approximation, our optimizer uses analytical gradients of the full LIP model, allowing each gradient descent step to be computed in sub-millisecond time, resulting in a fast and robust footstep planner.
- a structure that combines the best of two optimizers asynchronously is proposed: the secondary interior point optimizer produces a near-optimal solution at a slower rate,

which is used to initialise the main optimizer such that the solution can be quickly updated. This improves robustness of the planner.

- It is shown how the dynamics used in the planner can be extended to three dimensions under some conditions, allowing footstep planning to be performed in real time on uneven terrains.

6.2 PRELIMINARY

6.2.1 *Footstep Planning as a Discrete-time Optimal Control Problem*

Footstep planning problem can be formulated as a discrete-time optimal control problem. A cost function is specified to allow the CoM state $\mathbf{x}_k$ to follow a reference $\mathbf{x}_{ref}$. At each control step k, this cost is minimised by two control inputs: footstep location $\mathbf{u}_k$, and the duration of the step Δt_k. This optimisation is constrained by the CoM dynamics which are not only dependent on the robot's physical characteristics, but also on the model used to represent them. Furthermore, there is a set of physical inequality and equality constraints that this cost function is subject to:

$$\min_{\mathbf{u}_{k:N},\, \Delta t_{k:N}} \sum_{k=0}^{N} \|\mathbf{x}_k - \mathbf{x}_{ref}\|_Q^2 \tag{6.1}$$

$$\text{s.t.} \tag{6.2}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \Delta t_k, \mathbf{u}_k) \tag{6.3}$$

$$\mathbf{g}(\mathbf{x}_k, \Delta t_k, \mathbf{u}_k) \leqslant \mathbf{0} \tag{6.4}$$

$$\mathbf{h}(\mathbf{x}_k, \Delta t_k, \mathbf{u}_k) = \mathbf{0}. \tag{6.5}$$

The prediction horizon in this formulation is finite.

Importantly, in this work each footstep is considered as a single control step, as opposed to considering the centre of pressure of the foot during each step. This greatly simplifies the optimisation problem while allowing us to plan further into the future. In this work, a prediction horizon of two footsteps ahead is considered: optimizing two footstep positions and three footstep durations (including the current footstep duration). The footstep positions and durations optimized can be illustrated graphically in Fig. 6.2.

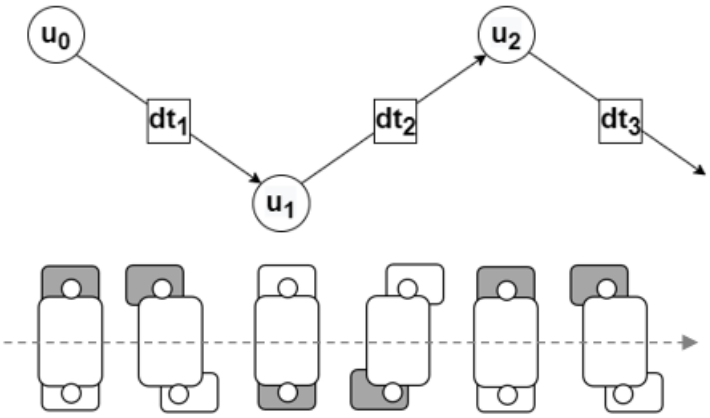


Figure 6.2: Illustration of the prediction horizon. The current footstep position of the active supporting foot (grey area) is represented by u_0, which will switch to the next footstep locations u_1 and u_2 in the duration dt_1, dt_2, and dt_3.

6.2.2 *Linear Inverted Pendulum Model*

The dynamics of bipedal walking are commonly modelled using the LIP model. This model consists of an inverted pendulum that has its entire mass located at the top end of the pendulum with a rotation occurring at the lower end (Kajita et al., 2001a). The LIP model requires that the position of the lower end is constantly updated so that the CoM of the pendulum is stable. The additional constraint of the LIP model is that the CoM remains in the same plane at all times, maintaining a constant distance to the ground:

$$\ddot{x} = \frac{g}{h}(x - u_x) \tag{6.6}$$

$$\ddot{y} = \frac{g}{h}(y - u_y)\,. \tag{6.7}$$

The full analytical solution to this second-order differential Equation (6.6) can be expressed as an exponential Equation (6.8). Here Δt_k is defined as the step duration, x_k and $\dot{x}_k$ as the CoM position and velocity at the beginning of footstep k, and $u_{x,k}$ as the position of the foot.

$$x_{k+1} = a_k \exp(\omega \Delta t_k) + b_k \exp(-\omega \Delta t_k) + u_{x,k}. \tag{6.8}$$

Where $a_k = 0.5(x_k - u_{x,k} + \frac{\dot{x}_k}{\omega})$, and $b_k = 0.5(x_k - u_{x,k} - \frac{\dot{x}_k}{\omega})$ and $\omega = \sqrt{\frac{g}{h}}$. Note that the positive exponential term is linked to the divergent component of motion, as it diverges as Δt_k increases.

As can be seen, Equation (6.8) is non-linear with footstep duration, meaning real-time optimization is difficult. This is further exacerbated as the optimization horizon increases, as subsequent optimisations are recursively linked to their predecessors. For more detailed formulation of these relationship, please see the Appendix.

6.2.3 *Constrained Optimisation with Augmented Lagrangian*

Constrained optimization problems are commonly solved by converting the problems into unconstrained ones by using Lagrangian Multiplier (LM). These problems can be expressed in a general form as:

$$\begin{aligned} &\min_x f(x) \\ \text{subject to} \quad &c(x) \leqslant 0 \end{aligned} \tag{6.9}$$

with

$$c(x) = \begin{cases} c_j(x) & j \in E \\ \max\left[0, c_j(x)\right] & j \in I \end{cases} \tag{6.10}$$

where $c(x)$ is the is the equality (E) or inequality (I) constraint, as shown in Equation (7.5). A popular way to solve the constrained optimization problem is perform dual gradient descent on the Lagrangian of the problem:

$$\mathcal{L} = f(x) + \sum_j \lambda_j c(x)_j \tag{6.11}$$

where λ_j is the jth Lagrange multiplier. As λ becomes larger, the violation of the constraints is penalised more severely, forcing the optimizer to find a solution within the feasible set. This behaviour leads to a λ that updates at each iteration to achieve a value of infinity in order to obtain the optimal solution, however an infinite coefficient would cause an infinite penalty when the constraint is violated (Bertsekas, 2014). As shown in (Nocedal et al., 2006), an improved method of optimization is to use the Augmented Lagrangian method, which adds a quadratic penalty term to the Lagrangian:

$$\mathcal{L} = f(x) + \sum_j \lambda_j c(x)_j + \sum_j \mu c^2(x)_j. \tag{6.12}$$

where μ is the positive coefficient of the AL penalty term. The solutions and coefficients of Augmented Lagrangian method are iteratively updated using gradient descent as follows:

$$dx_{k+1} = dx_k - \alpha \frac{dL}{dx} \tag{6.13}$$

$$\lambda_{k+1} = \lambda_k + \mu_k c(x) \tag{6.14}$$

$$\mu_{k+1} = \varphi \mu_k \quad with \;\; \varphi > 1. \tag{6.15}$$

where x is the solution to the optimization problem, α is the update step size, and ψ is a scaling parameter that regulates the update of μ by monotonically increasing it.

6.3 METHODS

6.3.1 *System Overview*

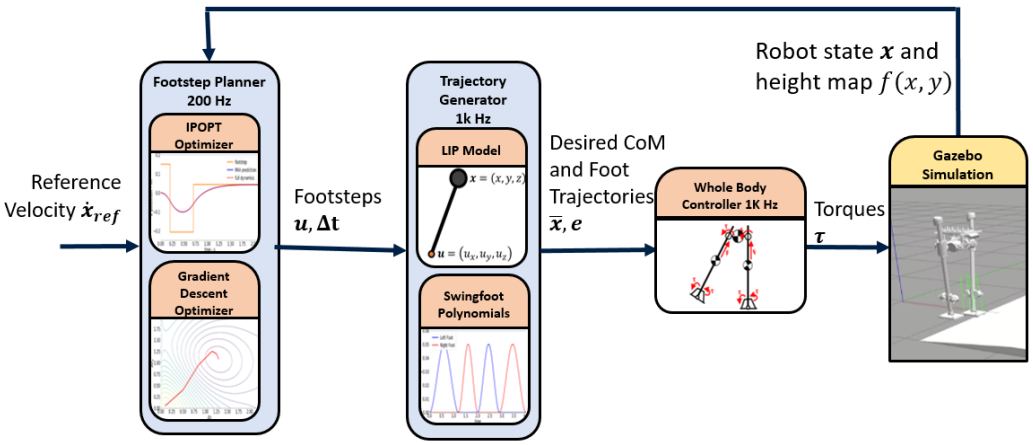


Figure 6.3: Illustration of the terrain-aware footstep planner pipeline. Given the reference velocity from the operator, robot states and the height map representing terrain information, the planner outputs reference footstep location in real time.

The proposed footstep planner solves the optimal control problem introduced in Equation (6.1), producing footstep locations and durations at a rate of 200 Hz. As summarized in Fig. 6.3, given a reference CoM velocity, the footstep planner makes use of two optimizers to generate optimal solutions. The main optimizer makes use of the AL method to rapidly update the footstep plan at 200 Hz, while a secondary interior point optimizer produces more optimal solutions at 20 Hz. From the generated footstep plan, a CoM trajectory is generated following the LIP dynamics and a swing foot trajectory is generated using fifth

order polynomials. Finally, an inverse dynamics-based whole body controller tracks the generated trajectories and outputs joint torques at a frequency of 1 kHz.

6.3.2 *Footstep Location and Timing Optimization Formulation*

As introduced in Section 6.2.1, at each control step, each optimizer must produce optimal footstep locations and durations. In this work a reference CoM trajectory is defined by a reference velocity, such that weighting matrix $Q = [w_x\ 0;\ 0\ w_y]$. The dynamic equality constraint defined by the linear inverted pendulum model defined in Equations (6.6) and (6.7) is added:

$$x_{k+1} = \frac{1}{2}(x_k - u_{x,k} + \frac{\dot{x}_k}{\omega})\exp(\omega\Delta t_k) + \frac{1}{2}(x_k - u_{x,k} - \frac{\dot{x}_k}{\omega})\exp(-\omega\Delta t_k) + u_{x,k}. \quad (6.16)$$

and physical inequality constraints are added:

$$\begin{aligned} \left\|\mathbf{x}_{pos,k} - \mathbf{u}_k\right\|_2 &\leqslant l_{max}, && \text{(next step length)} \\ \left\|\mathbf{x}_{pos,k} - \mathbf{u}_{k-1}\right\|_2 &\leqslant l_{max}, && \text{(current step length)} \\ \left|\mathbf{u}_{k,y} - \mathbf{u}_{k-1,y}\right| &\geqslant r_{foot}\,, && \text{(no crossing feet)} \\ c_{t,k,\,lower} \leqslant \Delta t_k &\leqslant c_{t,k,\,upper}\,. && \text{(step time)} \end{aligned}$$

Step length constraints define a maximum leg extension such that the robot does not travel too far on its support foot, and doesn't place the next foot too far away. No crossing feet constraint prevents the current and next support foot from crossing each other in the coronal plane. Time limit constraint defines safety limits to the step duration.

6.3.3 *Augmented Lagrangian Optimizer*

The AL optimizer makes use of analytically calculated gradients in order to find approximate optima as quickly as possible. However, gradient methods often have difficulty handling constraints, so the AL method introduced in Section 6.2.3 is used. The analytical gradient descent is computed by taking the derivative of Equation (6.1) with respect to the x-y footstep location $\mathbf{u}_i$, shown in Equation (6.17), and footstep duration Δt_i, shown in Equation (6.18):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_i} = 2\sum_{k=1}^{N} \frac{\partial \mathbf{x}_k}{\partial \mathbf{u}_i}^T \mathbf{Q}(\mathbf{x}_k - \mathbf{x}_{ref}) + \frac{\partial}{\partial \mathbf{u}_i}\sum_{i=1}^{P} \lambda_i c(\mathbf{x})_i + \frac{\partial}{\partial \mathbf{u}_i}\sum_{i=1}^{P} c(\mathbf{x})_i^T \mu_i c(\mathbf{x})_i \quad (6.17)$$

$$\frac{\partial \mathcal{L}}{\partial \Delta t_i} = 2 \sum_{k=1}^{N} \frac{\partial \mathbf{x}_k}{\partial \Delta t_i} \mathbf{Q}(\mathbf{x}_k - \mathbf{x}_{ref}) + \frac{\partial}{\partial \Delta t_i} \sum_{i=1}^{P} \lambda_i c(\mathbf{x})_i + \frac{\partial}{\partial u_i} \sum_{i=1}^{P} c(\mathbf{x})_i^T \mu_i c(\mathbf{x})_i. \tag{6.18}$$

Individual expressions for the derivatives of CoM position x_k and velocity $\dot{x}_k$ can be found in the Appendix.

The criteria of convergence is that the difference for norm of all gradients is smaller than 0.05. The footstep planner algorithm uses the projection method to "project" the solution on the lower and upper bounds of constraints when the search direction converges outside the admissible solutions, this can also prevent oscillations observed during the convergence. One special case is the remaining time of current step, as shown in Equation (6.19):

$$dt_{0,k+1} = dt_{0,k} - 1/\text{LoopRate}, \quad \text{if } dt_0 < 0 \tag{6.19}$$

6.3.4 *Interior Point Optimizer*

The secondary optimizer uses Interior Point methods to perform the optimization. In this work the IPOPT optimizer (Wächter et al., 2006) is used via the software framework CasADi (Andersson et al., 2012). This optimizer is considerably slower, but more accurate than the AL optimizer. To warm start the solver, the initial value of the solution, $\mathbf{u}^{(i)}_{k,j}$, is set as the previous iteration's final solution, $\mathbf{u}^{(f)}_{k,j-1}$, for each step in the prediction horizon, $k \in [1, ..., N]$. The current step duration, $\Delta t^{(i)}_{1,j}$ is adjusted for the time elapsed between iterations, τ_{j-1}:

$$\mathbf{u}^{(i)}_{k,j} = \mathbf{u}^{(f)}_{k,j-1} \tag{6.20}$$

$$\Delta t^{(i)}_{1,j} = \Delta t^{(f)}_{1,j-1} - \tau_{j-1}. \tag{6.21}$$

When a step has been taken between iterations, the initial value of the solution is provided as follows for the first $k \in [1, ..., N-1]$ predicted steps:

$$\mathbf{u}^{(i)}_{k,j} = \mathbf{u}^{(f)}_{k+1,j-1} \tag{6.22}$$

$$\boldsymbol{\Delta t}^{(i)}_{k,j} = \boldsymbol{\Delta t}^{(f)}_{k+1,j-1}. \tag{6.23}$$

With the final predicted step initialized to repeat the motion of the penultimate step, mirrored in the y direction.

$$\mathbf{u}^{(i)}_{N,j} = \mathbf{u}^{(i)}_{N-1,j-1} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (\mathbf{u}^{(i)}_{N,j-1} - \mathbf{u}^{(i)}_{N-1,j-1}) \tag{6.24}$$

$$\Delta t^{(i)}_{N,j} = \Delta t^{(i)}_{N,j-1}. \tag{6.25}$$

6.3.5 *Asynchronous Optimization*

The AL and Interior Point optimizers run in parallel threads communicating via ROS topics. Each time a feasible solution is computed from IPOPT, the solutions, as well as the values of Lagrange multipliers, are fed into the gradient descent optimizer to reinitialize it, as shown in Fig. 6.4. Importantly, it is observed that the AL optimizer alone can get stuck in local solutions, or in some situations the analytical gradient is extremely large (for example at the switching point between feet), meaning the AL optimizer's solution is not feasible. For periods of time where an AL optimizer solution is infeasible, the IPOPT solution is used as a warm-start for the AL optimizer to generate feasible solutions. The interaction between the solvers is shown in Fig. 6.4. This method allows the footstep planner to run at a rate of 200 Hz: an order of magnitude increase of speed over the interior point optimizer alone. This structure combines the best of both optimizers: IPOPT has the advantage of finding accurate optima, and the AL optimizer is very fast.

6.3.6 *Terrain-Aware Footstep Planning*

Inspired by the parallel height constraint of the LIP model proposed by (Kajita et al., 2001b), the LIP model can be actually extended as a viable model for dealing with uneven terrain by constraining the CoM to move along a plane parallel to the slope of the terrain, as shown in Fig. 6.5. Because the plane has a linear relationship with the x and y position of the robot, no new decision variables are added to the optimization problem.

If the dynamics of an inverted pendulum is considered, its motion in the sagittal $(x - z)$ plane is described by the following equation:

$$\frac{\ddot{x}}{x - u_x} = \frac{\ddot{z} + g}{z - u_z}, \tag{6.26}$$

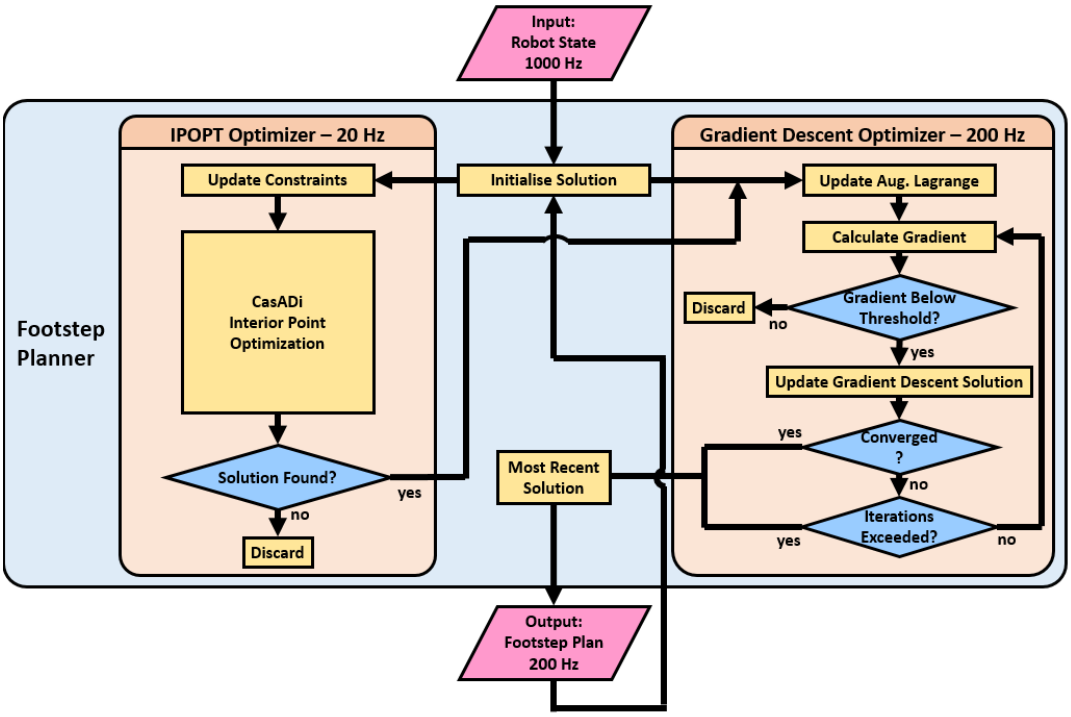


Figure 6.4: Flowchart describing the asynchronous optimization process.

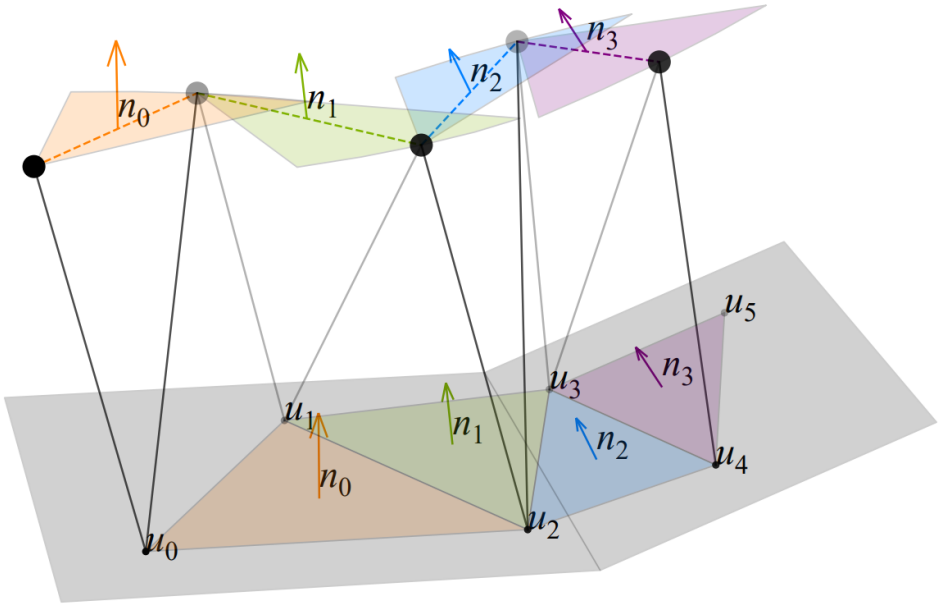


Figure 6.5: 3D illustration of constructing the plane that the CoM moves along from optimal footsteps.

and in the coronal $(y - z)$ plane:

$$\frac{\ddot{y}}{y - u_y} = \frac{\ddot{z} + g}{z - u_z}. \tag{6.27}$$

It is assumed that the local terrain around the robot can be described as a plane defined by gradients in the x and y directions of α and β, respectively. The motion of the centre of mass is then constrained to move parallel to this plane at a vertical offset (height) of h_0:

$$z - u_z = \alpha(x - u_x) + \beta(y - u_y) + h_0, \tag{6.28}$$

The acceleration in the z direction can then be defined as:

$$\ddot{z} = \alpha\ddot{x} + \beta\ddot{y}, \tag{6.29}$$

This constraint leads to Equation (6.26) and Equation (6.27) becoming non-linear, and difficult to optimise. However, in two cases, they can be reduced to the popular linear inverted pendulum equations.

By substituting Equation (6.29) into Equation (6.26) and Equation (6.27), giving

$$\frac{\ddot{x}}{x - u_x} = \frac{\alpha\ddot{x} + \beta\ddot{y} + g}{\alpha(x - u_x) + \beta(y - u_y) + h_0}, \tag{6.30}$$

$$\frac{\ddot{y}}{y - u_y} = \frac{\alpha\ddot{x} + \beta\ddot{y} + g}{\alpha(x - u_x) + \beta(y - u_y) + h_0}. \tag{6.31}$$

The first case in which these equations reduce to the linear inverted pendulum model, as noted in (Kajita et al., 2001), is when α and β are very small (i.e. flat ground). This is of little interest to a robot walking on uneven terrain.

The second case occurs when the forward (x) direction of the robot aligns with the direction of steepest gradient of the plane. This reduces the gradient in the y direction to zero, namely $\beta = 0$. Then Equation (6.30) reduces to the LIP equation. Next, it is assumed that the distance between the centre of mass and the foot in the y direction is significantly larger than in the x direction, due to the inter-foot spacing of a bipedal robot:

$$y - u_y \gg x - u_x. \tag{6.32}$$

Finally, it is assumed that the acceleration of the sway of the robot side-to-side is much greater in magnitude than the robot's forward acceleration:

$$\ddot{y} \gg \ddot{x}. \tag{6.33}$$

Under these assumptions, the x terms from Equation (6.31) are negligible, meaning that it, too, reduces to the equation of a linear inverted pendulum.

The linear inverted pendulum dynamic model can therefore be applied to a walking robot on uneven terrain, provided that the robot is able to perceive the terrain and the robot is aligned with the direction of the local gradient of the terrain. This is the case for many structured environments, such as stairs and ramps.

The mapping of the terrain is achieved by adapting the state-of-the-art Robot-Centric Elevation Mapping ROS package(Stanford Artificial Intelligence Laboratory et al., 2018) provided by ANYbotics (Fankhauser et al., 2018),(Fankhauser et al., 2014) to our robot. This provides multiple 2.5d probabilistic mappings of the environment with information on the height, normals, variance and corresponding uncertainties. This mapping is updated based on new observations with the uncertainty being determined by the time of the most recent observation. The choice of a robot centric representation allows for planning on a local level when only local knowledge is necessary while not compromising on the possibility of global planning with reliable odometry estimates. The pipeline of mapping can be seen in Fig. 6.6

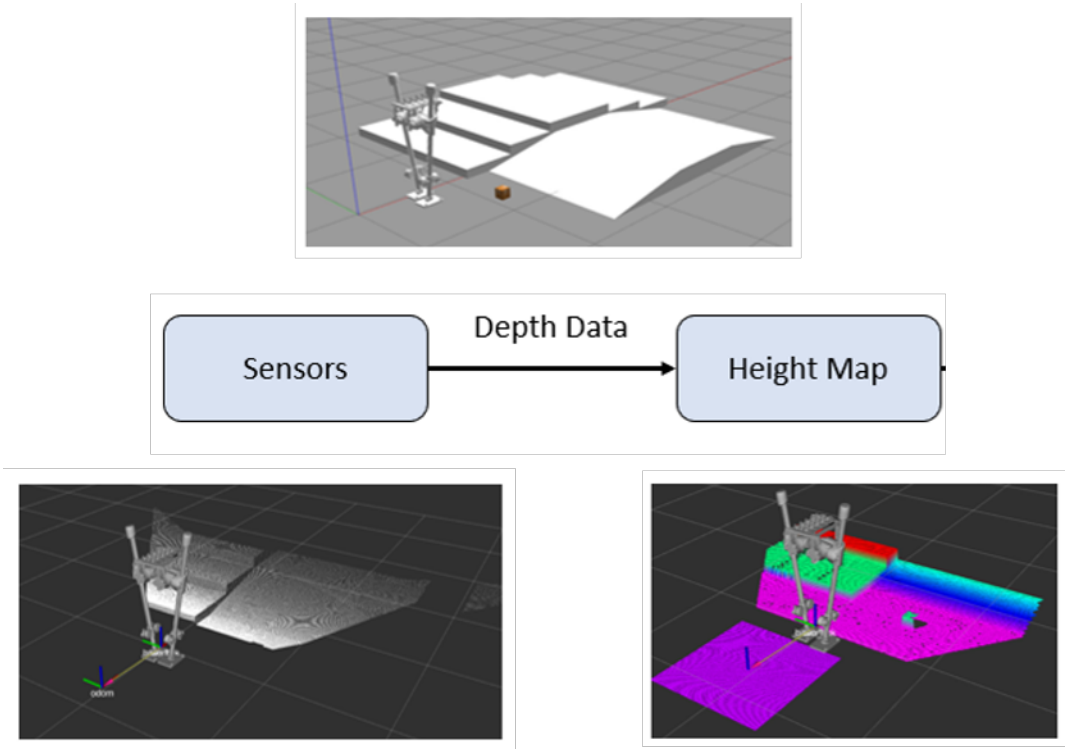


Figure 6.6: The pipeline of generating height map from perception data.

With these assumptions, a controller can be constructed that computes the control inputs based on the LIP dynamics, while it is able to adapt to uneven terrain using perceptive feedback. The movement planes are constructed using a height map, which provides heights for queried sets of points. The queried points are centered around the locations of the next 2 footsteps and contain the averaged value of all points within the size of a footstep, as computed by the model predictive controller shown in Fig. 6.5. These queried points and

their heights are then used to fit a plane passing through them, and are passed to the controller which moves the CoM onto those planes. The planes can get updated mid-step without affecting performance or walking stability. The orientation of the swing foot is also commanded to align with the normals queried around the footstep position. This allows the robot to walk around increasingly complex terrains where the orientation of the CoM plane is different than the orientation of the individual footholds. In our method, the plane is updated at a rate of 5 Hz.

6.3.7 *Swing Foot Trajectory Generation*

The swing foot trajectory is generated using fifth order polynomials: current and final positions, velocities and accelerations are specified, and a parametric quintic curve in the x, y, and z directions is generated to produce smooth trajectories. The z polynomial is generated in two halves, with a midpoint foot height with zero velocity and acceleration specified to ensure no collision with the ground. The LIP model is used to generate the CoM trajectory during footsteps. Each trajectory point is calculated at the whole body controller rate of 1000 Hz.

6.3.8 *Whole Body Controller*

An inverse dynamics-based whole body controller is used to track the foot motion while respecting a set of constraints. In our method, the tasks of interest are the CoM position and velocity, the pelvis orientation, the foot positions and orientations. Each task is comprised of a desired acceleration as a feed-forward term and a state feedback term to stabilize the trajectory. Generally, the task for the linear motion can be expressed as:

$$\mathbf{J}_T \ddot{\mathbf{q}} = \ddot{\mathbf{x}}^{cmd} - \dot{\mathbf{J}}_T \dot{\mathbf{q}},$$

$$\ddot{\mathbf{x}}^{cmd} = \ddot{\mathbf{x}}^{des} + \mathbf{K}_P^{pos}(\mathbf{x}^{des} - \mathbf{x}) + \mathbf{K}_D^{pos}(\dot{\mathbf{x}}^{des} - \dot{\mathbf{x}}),$$

where $\mathbf{J}_T$ is the translational Jacobian for the task, $\mathbf{x}$ is the actual position of the link, and the superscript des indicates the desired motion.

For the task of angular motion, the command can be formulated as:

$$\mathbf{J}_R \ddot{\mathbf{q}} = \dot{\boldsymbol{\omega}}^{cmd} - \dot{\mathbf{J}}_R \dot{\mathbf{q}},$$

$$\dot{\boldsymbol{\omega}}^{cmd} = \dot{\boldsymbol{\omega}}^{des} + \mathbf{K}_P^{ang}(\text{AngleAxis}(\mathbf{R}^{des}\mathbf{R}^T)) + \mathbf{K}_D^{ang}(\boldsymbol{\omega}^{des} - \boldsymbol{\omega}),$$

where $\mathbf{J}_R$ is the rotational Jacobian for the task, $\mathbf{R}$ and $\mathbf{R}^{des}$ denote the actual and desired orientation of the pelvis link respectively, AngleAxis() maps a rotation matrix to the corresponding axis-angle representation to avoid the gimbal lock of using euler angles, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the link.

QP formulation

Inspired by (Herzog et al., 2015), the full dynamics of the walking robot can be decomposed into the underactuated part and actuated part:

$$\begin{bmatrix} \mathbf{M}_f \\ \mathbf{M}_a \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{H}_f \\ \mathbf{H}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{S}_a \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{J}_f^T \\ \mathbf{J}_a^T \end{bmatrix} \mathbf{f},$$

where $\mathbf{M}$, $\mathbf{H}$, $\mathbf{S}_a$, $\boldsymbol{\tau}$, $\mathbf{J}$ and $\mathbf{f}$ are the mass matrix, Coriolis force matrix and gravitation force vector, the actuator selection matrix, joint torques vector, the stacked contact Jocabian and reaction force vector. The subscript, f and a, indicates the floating part and actuated part respectively. The weighted sum formulation is applied, in which one QP problem is solved at each control loop. The formulation of the QP problem can be written as

$$\begin{aligned} \min_{\ddot{\mathbf{q}}, \mathbf{f}} \quad & \frac{1}{2} \|\mathbf{A}\ddot{\mathbf{q}} + \dot{\mathbf{A}}\dot{\mathbf{q}} - \mathbf{B}^{cmd}\|_{W_1}^2 && (6.34) \\ \text{s.t.} \quad & \mathbf{M}_f \ddot{\mathbf{q}} - \mathbf{J}_f^T \mathbf{f} = -\mathbf{H}_f && \text{(floating base dynamics)} \\ & \mathbf{P}\mathbf{f} \leqslant \mathbf{0} && \text{(friction cone)} \\ & \mathbf{S}_a^{-1}(\mathbf{M}_a \ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T \mathbf{f}) \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}] && \text{(input limits)} \end{aligned}$$

where $\mathbf{A}$ is a stack of the Jacobian matrices for the tasks of interest, $\mathbf{B}^{cmd}$ is a stack of the commanded accelerations and $\mathbf{W}_i$ $(i = 1, 2)$ are the weighting matrices, $\mathbf{P}$ denotes the linearized friction cone matrix. The unilateral contact constraint is treated as a soft constraint by simply assigning a large weight on the desired zero acceleration. It is reported in (Feng et al., 2014) that this gives a better stability.

The output torque commands $\boldsymbol{\tau}$ at each control iteration is computed by

$$\boldsymbol{\tau} = \mathbf{S}_a^{-1}(\mathbf{M}_a \ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T \mathbf{f}) \tag{6.35}$$

Dynamic Gains

To aid walking on uneven terrains, a different set of gains is applied for each state between the swing and the stance foot. The stance foot is generally stable (low velocity) and low gains are preferred. Adversely on the swing foot the trajectory is aimed to be accurately tracked

and thus high gains are applied. The foot step planner generates the step time for the given step. This timing however can be inaccurate due to mid step instabilities and is especially problematic when dealing with uneven surfaces. The complication lies in the timing of the switch from high gains to low gains when switching from swing to stance foot after a step is taken. If the timing is too late and the foot makes impact with the ground before switching gains then the high gains causes a strong reaction force with the surface.

To address this problem the concept of dynamic gains was utilized. The idea stems from using the step time as an estimate of when a step would end. Around this time the gains can be adjusted to decrease from the swing foot configuration to that of the stance foot. Similarly the gains from the stance foot can be increased linearly to that of the swing foot as the foot leaves the ground.

6.4 EXPERIMENTAL RESULTS

6.4.1 *Experimental Platform and Scenarios*

The SLIDER robot is used as our experimental platform. SLIDER is a knee-less bipedal robot designed by the Robot Intelligence Lab at Imperial College London (Wang et al., 2020a). As shown in Fig. 6.7, SLIDER is 1.2 m tall and has 10 DoF: hip pitch, hip roll, hip slide, ankle roll and ankle pitch on each leg. The robot is lightweight (14 kg in total) and most of its weight is concentrated in the pelvis as its legs are made of carbon fiber reinforced polymer. The prismatic knee joint design is SLIDER's unique feature that differentiates it from other robots with anthropomorphic design. The sliding joint has a large range of motion, which combined with the light weight makes SLIDER suitable for agile locomotion.

6.4.2 *Flat Ground Walking*

SLIDER successfully walked 50 m of simulation ground at forward speed of 0.2 m/s in Gazebo. Fig. 6.8 and Fig. 6.9 illustrate the CoM position and active support foot against time in 3D during the two locomotion tasks: walking forward and sideways, respectively. The results show that in both scenarios the CoM remains approximately at constant height and follows a serpentine pattern on the transverse plane.

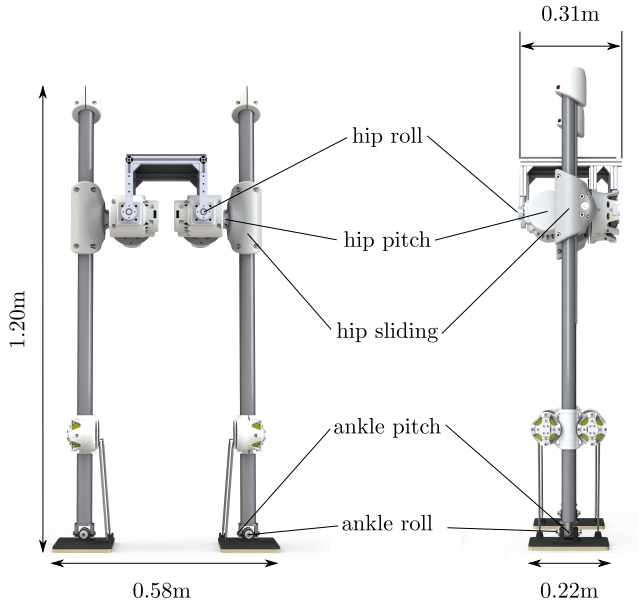


Figure 6.7: The dimensions and joint configuration of SLIDER.

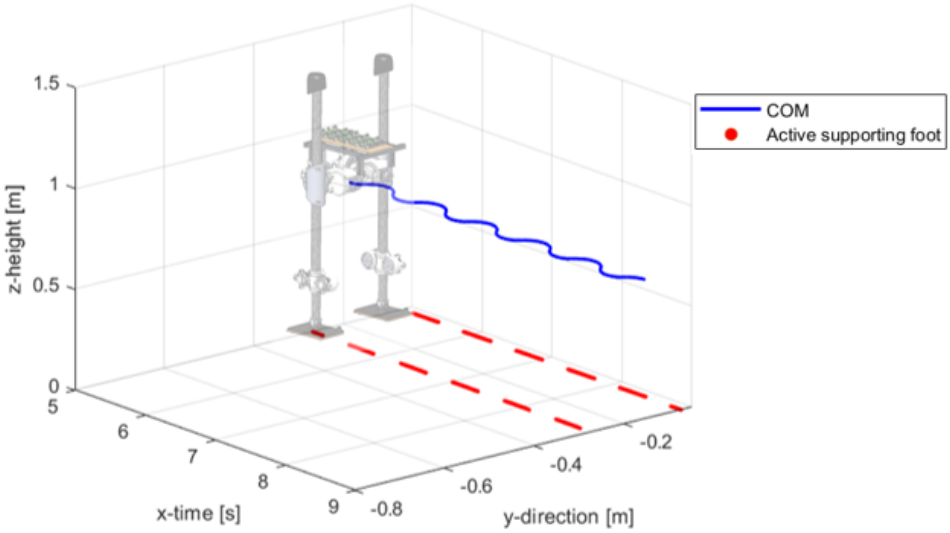


Figure 6.8: 3D illustration of CoM (blue) and supporting foot position (red) when SLIDER is walking forward at 0.2 m/s with the planner ARTO-AL.

6.4.3 *Push Recovery on Flat Ground*

The push recovery experiment consisted of the analysis of SLIDER's performances to external forces to compare two footstep planners: Footstep planner with no time adaptation (No-Time-Adp), and ARTO-AL. The aim of the experiment was to observe the push recovery of one footstep planner that adapts time and one that does not, demonstrating that optimizing

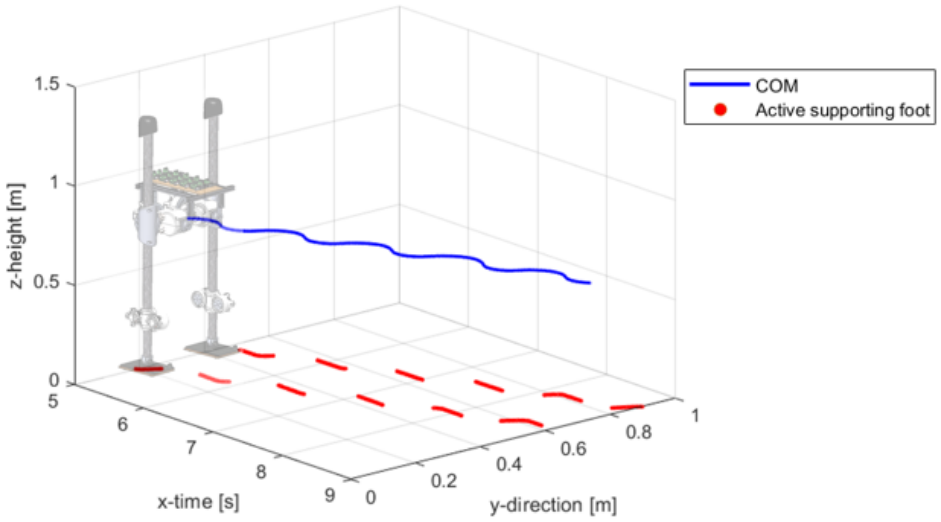


Figure 6.9: 3D illustration of CoM (blue) and supporting foot position (red) when SLIDER is walking sideways at 0.3 m/s with ARTO-AL.

both position and time offers better performances to external disturbances than when time is not optimized. For the first part of the experiment,SLIDER was stepping in place and then pushed laterally with a force of 30 N. Fig. 6.10 shows the comparison between a method that does not optimize time and one that does, i.e. No-Time-Adp and ARTO-AL, respectively. Fig.6.11 illustrates the five recovery stages performed by SLIDER using ARTO-AL with a

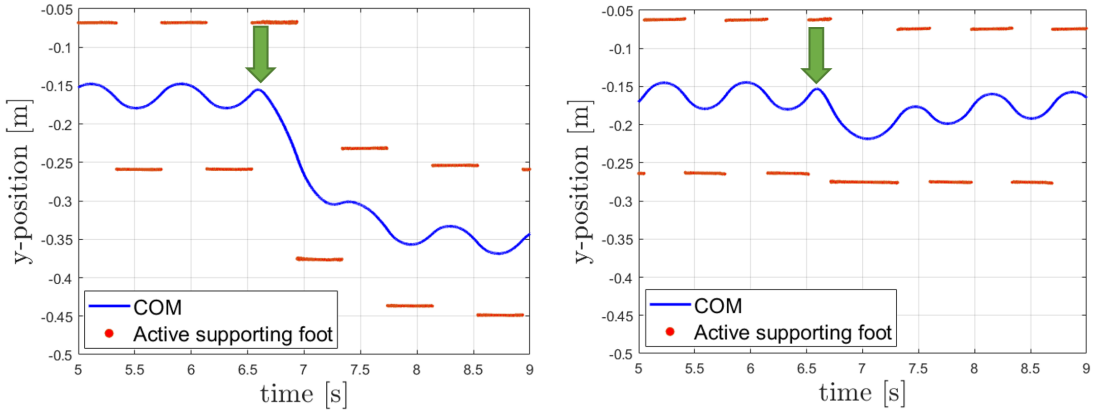


Figure 6.10: Push recovery comparison between No-Time-Adp (a) and ARTO-AL (b). The figure highlights the behaviour between a planner that does not optimize time (a) and one that optimizes both footstep position and duration (b). The red arrow represents a 30 N side push of duration 0.1 s.

30 N push. Compared to No-Time-Adp, ARTO-AL is able to stabilise the robot after the

push in 2 steps, returning to a normal stepping in place behaviour. However, without time adaptation, the robot requires at least 5 steps to return to normal stepping in place.

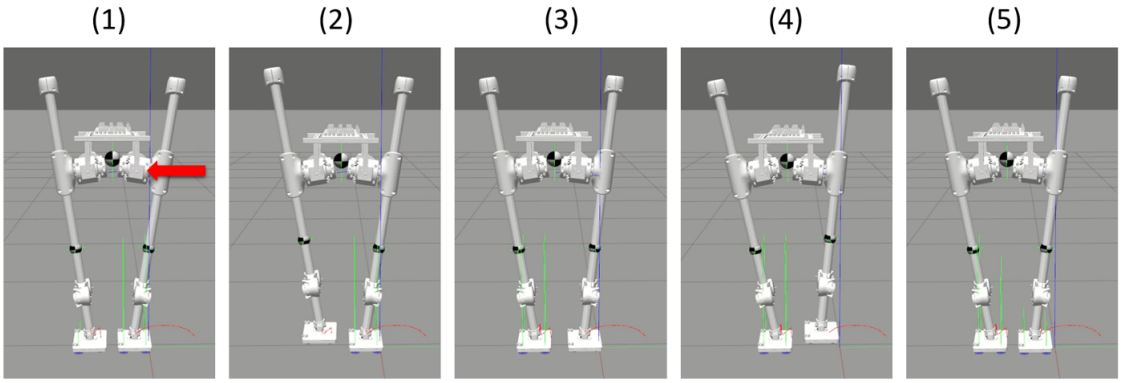


Figure 6.11: Push recovery using ARTO-AL. The model of SLIDER is stepping in place and is pushed sideways with a force of 30 N on the base and with a duration of 0.1 s. Red arrow indicates the direction of push. (1-3-5) show the moments corresponding to the switch of active supporting foot. (2-4) correspond to the sideways steps of the robot's CoM to absorb the external force.

In the second experiment, the push recovery analysis was performed with the five footstep planners by measuring the maximum push force before failing. The five planners are: No-Time-Adp, IPOPT only, AL-only, DCM and ARTO-AL. The DCM step planner is adapted from the open source code [1] which is the implementation of the paper (Khadiv et al., 2016). The robot was pushed in 8 directions spaced 45 degrees from each other. As a force is applied, the robot needs to react by taking a step in the same direction of the push, in order to prevent its body from falling. A successful push recovery consists of the robot returning to a stable walk after the push. The forces were applied at the touchdown moment of the left foot in a direction from left to right, using an impulse with a duration of 0.1 s, as shown in Fig. 6.11. Overall, the maximum resist force is bigger in angle range $[0°, 180°]$ than angle range of $[180°, 360°]$ because of the direction of push force. ARTO-AL has the best performance over all planners.

6.4.4 *Push Recovery on Uneven Terrain*

Fig. 6.13 shows the push recovery performance of 5 planners while SLIDER walks up a 10° ramp given the slope angle. The maximum force of forward push is bigger than backward push. The reason is that the foot cannot touch the ground as a result of leg length limit when pushed backward. ARTO-AL performs best among 5 planners. Fig. 6.14 shows the reaction of

1 https://github.com/machines-in-motion/reactive_planners

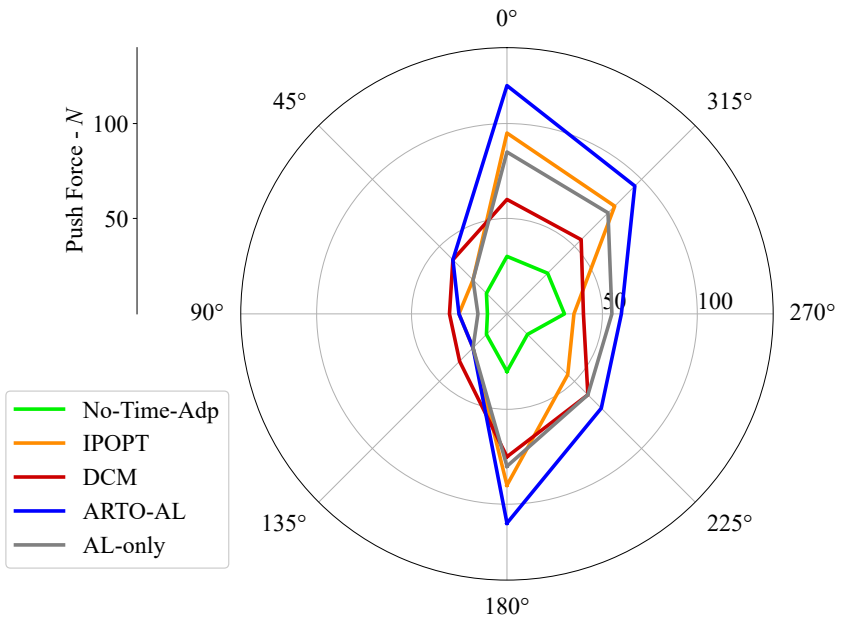


Figure 6.12: Maximum push recovery on flat ground with forces parallel to the transverse plane. The impulse was applied with a duration of 0.1 s on the base while SLIDER was stepping in place. 0° corresponds to the robot being pushed forward.

SLIDER when pushed forward with ARTO-AL. The DCM planner is not compared because constructing a plane needs two steps prediction ahead.

6.4.5 *Terrain-Aware Walking*

Fig. 6.15 shows the comparison between the estimated height and slope angle from the height map compared with the ground truth. The average error is 0.86 cm for height and 1.09° for the slope angle. Fig. 6.15 shows the success rate of walking with three different conditions. Benchmark: no perception and no dynamic gains. ARTO-AL+No DG: ARTO-AL without dynamic gains. ARTO-AL + DG: ARTO-AL with dynamic gains. As can be seen the inclusion of terrain-aware ARTO-AL provides large improvements in performance over the benchmarks bringing the critical failure rate for steps from 3 to 4 cm. The largest improvements can be seen on slopes where the critical failure rate is above 10° where it is 2.5° for benchmark. Dynamic gains further improves the performance, for step height of 3 cm the success rate reaches around 90%. The success rate also increases to 90% with dynamic gains when the robot walks on the 10° slope.

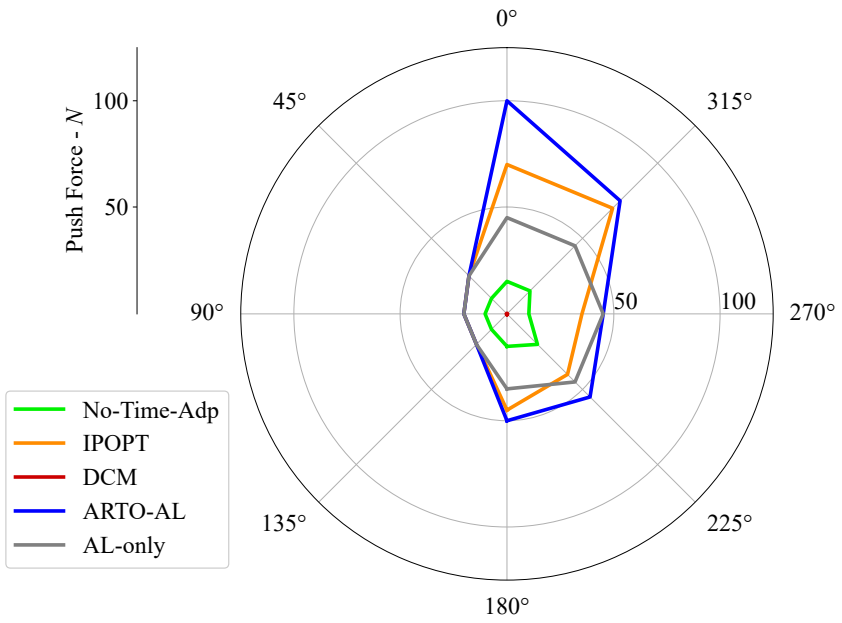


Figure 6.13: Maximum push recovery on inclined ground with the force vector parallel to the transverse plane. The impulse of the disturbance lasted 0.1 s and was applied while walking on a 10° ramp at 0.1 m/s. 0° corresponds to the robot being pushed forward.

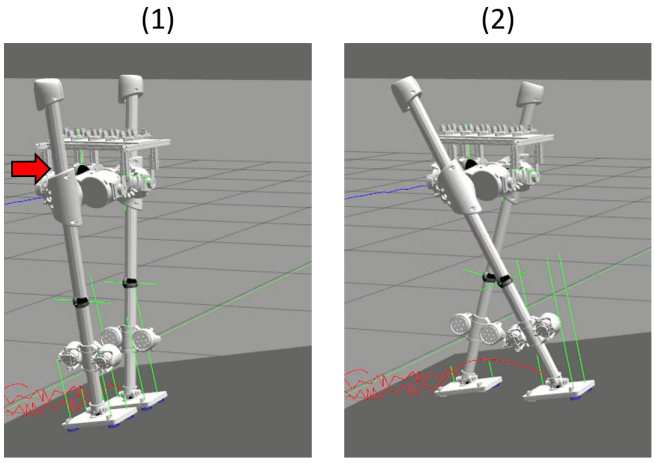
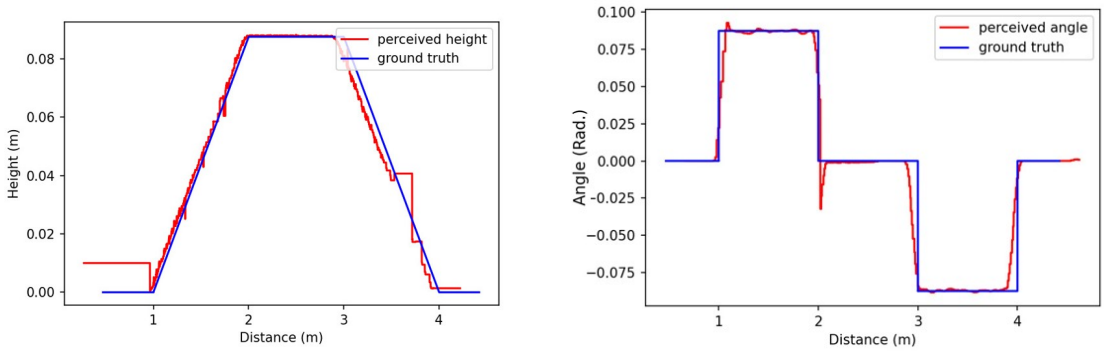


Figure 6.14: Reaction step to a 70 N push when walking on a 10° slope with ARTO-AL. Red arrow indicates the push direction.

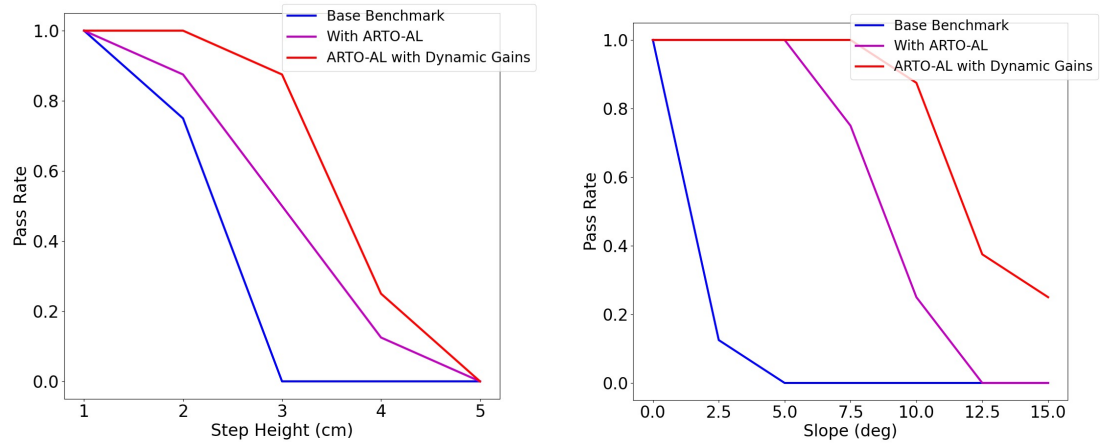
6.5 DISCUSSION

- *Number of steps in prediction*: It is still unclear what is the best number of prediction steps. As (Khadiv et al., 2020) found and compared to human studies (Patla et al., 2003), it is sufficient to consider just the next step when walking normally, and two steps ahead when walking on difficult terrains or with considerable constraints. In our



(a) (b)

Figure 6.15: The estimated height and slope angle from height map compared with the ground truth, when the robot walks up and down on a 5° slope. (a) The estimated height vs ground truth. (b) The estimated slope angle vs ground truth.



(a) (b)

Figure 6.16: Plots of success rate when walking in three conditions. Base Benchmark: no perception and no dynamic gains. ARTO-AL+No DG: ARTO-AL without dynamic gains. ARTO-AL + DG: ARTO-AL with dynamic gains. (a), Success rate with different step height. (b), Success rate with different slope angles.

method two steps is predicted so the robot can walk over uneven terrains. It is possible to predict more steps (Gibson et al., 2021), however predicting more steps requires more computation power and might affect the real-time performance.

- *DCM vs full LIP dynamics*: In (Khadiv et al., 2020) the footstep location and timing adaptation problem is formulated using DCM and it can be solved with QP. While solving with QP gives the real-time guarantee, it is only viable for *one* step prediction. Because of this, the approach using DCM cannot easily extend to uneven terrain walking. In the meantime, as shown in the push recovery result, approach using

DCM performs worse than ARTO-AL. Furthermore, using DCM needs extra steps and heuristics to compute DCM offset. While our approach is straightforward and requires no extra steps.

- *Robustness and Sensitivity Analysis*: Robustness under sensory noise is important for implementing this footstep planner on real robots. Although it is difficult to do analytical analysis, quantitative analysis can be done. It is also interesting to investigate which parameters are more sensitive to sensory noise.

6.6 CONCLUSION

This section proposes a terrain-aware footstep planning algorithm that can adapts step position and timing based on the LIP dynamics. With an asynchronous structure which combines an Augmented Lagrangian solver with analytical gradient descent and IPOPT optimizer, the planner can achieves real-time performance with 200 Hz update frequency. Furthermore, the planner is extended from flat ground walking to uneven terrain with perceptive information. We have shown experimental results of the robot walking on flat ground, uneven terrains and push recovery on both flat ground and a slope with 10°, demonstrating its performance and robustness to disturbances.

CHAPTER 7

Unified Model with Inertia Shaping for Highly Dynamic Jumps

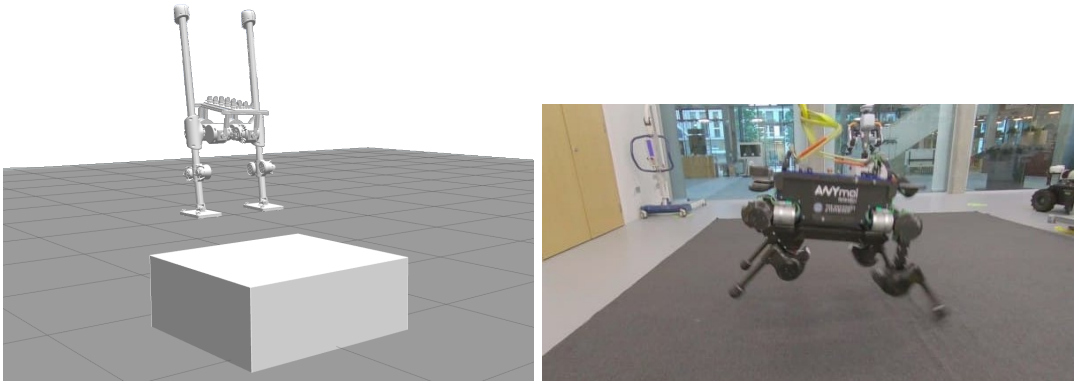
To achieve highly dynamic jumps with legged robots, it is essential to control the rotational dynamics of the robot. This chapter introduces a planning framework that aims to improve the jumping performance by proposing a unified model for planning highly dynamic jumps that can approximately model the centroidal inertia. This model abstracts the robot as a single rigid body for the base and point masses for the legs. The model is called the LL-SRBM and can be used to plan motions for both bipedal and quadrupedal robots. By taking the effects of leg dynamics into account, LL-SRBM provides a computationally efficient way for the motion planner to change the centroidal inertia of the robot with various leg configurations. Concurrently, a novel contact detection method by using the norm of the average spatial velocity is proposed. After the contact is detected, the controller is switched to force control to achieve a soft landing. Twisting jump and forward jump experiments on the bipedal robot SLIDER and quadrupedal robot ANYmal demonstrate the improved jump performance by actively changing the centroidal inertia. These experiments also show the generalization and the robustness of the integrated planning and control framework.

7.1 INTRODUCTION

Early studies on highly dynamic motion such as jumping and running in legged robots are largely influenced by the heuristic control implemented on Raibert's hoppers (Raibert, 1986). The hopper is able to achieve the dynamic behaviors through simple composition of a set of simple controllers that control hopping height, speed, and posture separately. This is possible due to their prismatic leg design which differs from most humanoid robots with human-like morphology.

Despite its success, heuristic control is quite limited since it needs large amount of work on parameter tuning. More recently, model based approaches are becoming more

and more popular. The SLIP model is a well recognized template model for running and jump. An approximated SLIP model has been used to represent the translational motion, robust running and jump are planned with model predictive control (Mordatch et al., 2010). 3D-SLIP model has been used to generate high speed running motion (Wensing et al., 2013) and long jump (Wensing et al., 2014). However, the SLIP model only considers the point mass dynamics and the angular momentum is ignored. It is therefore difficult to consider motions involving body rotation, such as a twisting jump.



(a) (b)

Figure 7.1: Highly dynamic jump motions can be generated for both bipedal and quadrupedal robots using the LL-SRBM. Left: SLIDER robot. Right: ANYmal robot.

Compared to humanoid robots, quadrupedal robots are often built with lower DoF legs and point feet. This equips them with lightweight legs which are often ignored during the motion planning phase or the control phase without introducing significant modeling error. With the assumption of massless legs, dynamic gaits including trotting, pronking, bounding and pacing have been demonstrated on Mini Cheetah through online convex model predictive control (Di Carlo et al., 2018, Kim et al., 2019, Katz et al., 2019). The convex formulation is made possible due to a small angle assumption for roll and pitch angle of the SRBM and predefined footstep locations. Unfortunately, these assumptions do not hold for generating versatile and highly dynamic motions. Recently, a more versatile and acrobatic motion generation framework, the kino-dynamic planner, is proposed in (Chignoli et al., 2021). However, the kino-dynamic planner is complex and computationally inefficient. Therefore, a good trade-off between model complexity and computation efficiency is needed.

At the same time, the robustness of the controller is important for guaranteeing the successful execution of the dynamic motion. The robot experiences a large impact when landing; these large contact forces and fast changing velocities can easily make the robot unstable. Roboticists have made various efforts to reduce the effect of the impact. One direction is changing the reference trajectory in real time. (Corbères et al., 2021) shows a quadrupedal robot walking with 50 Hz re-planning with a slightly simplified SRBM. But no

results have been shown with more dynamic motions such as jumping. In (Halm et al., 2019, Stouraitis et al., 2020) the impacts are explicitly modelled, however the simplified model and large uncertainties of state estimation make the control more difficult. (Yang et al., 2021a) presents a feedback controller in an impact invariant space. Although this approach is not affected by the rapid change of velocities, it can not reduce the effects caused by the impact. To reduce the effect of the impact, good contact detection with a soft landing is a promising direction. External force sensors are commonly used to monitor contacts for robot arms (Haddadin et al., 2017). But in practice force signals are too noisy for legged robots. The average spatial velocity, first introduced by (Lee et al., 2007), is a synthetic representation of all link velocities and an indicator for the kinetic energy of the robot. The average spatial velocity is relied for contact detection.

7.1.1 *Contributions*

In this chapter, the aim is to improve the performance of highly dynamic jumps with the LL-SRBM and the robustness of the tracking controller with contact detection. Additionally, it is show that the proposed planning and control methods can be applied to both bipedal and quadrupedal robots. The twisting jump and forward jump are demonstrated in simulation and real robot experiments. Here, the contributions of the chapter is highlighted [1]:

- A general framework that can generate highly dynamic jumping motions for both bipedal and quadrupedal robots. Each layer of the framework is generalized for both types of robots, including the lump leg concept.
- The LL-SRBM provides a computationally efficient way to optimize the shape of the robot's inertia while planning the jump motion. By taking the motion of legs into consideration, the jump motion generated by the LL-SRBM achieves better performance (e.g. faster twisting jump, more stable forward jump etc.) than the motion generated by using the SRBM.
- Safe landing is ensured by using a novel contact detection method and switching to force control in landing phase. this chapter proposes to use the norm of the average spatial velocity to detect contacts. By taking account of all link states, the average spatial velocity is more robust to noises and general enough for detecting impulses from all directions. After the contact is detected, contact force tracking control is switched on for a period to achieve soft landing. It is difficult to use force control for the whole jump process since the LL-SRBM is still an approximated model for planning.

1 Video summary: https://youtu.be/M0HDgvkAVhM

7.2 SYSTEM OVERVIEW

The highly dynamic jumping behaviour is achieved via a hierarchical structure which includes the motion planner and the whole body controller as shown in Fig. 7.2. The high-level user selects the motion (e.g. twisting jump, forward jump, etc.) and inputs to the motion planner with initial and final states and contact positions for non-flight phases. The trajectory optimization with LL-SRBM outputs the reference centroidal and foot trajectories to the whole body controller. The whole body controller computes optimal joint torques for the robot to track these task space targets while considering all physical constraints. The contact detection runs after the robot takes off and monitors the average spatial velocity. Once the contact is detected, the whole body controller switches to force control for a short time to achieve a soft landing. After landing, trajectory tracking is reinstated to achieve the desired steady finishing pose.

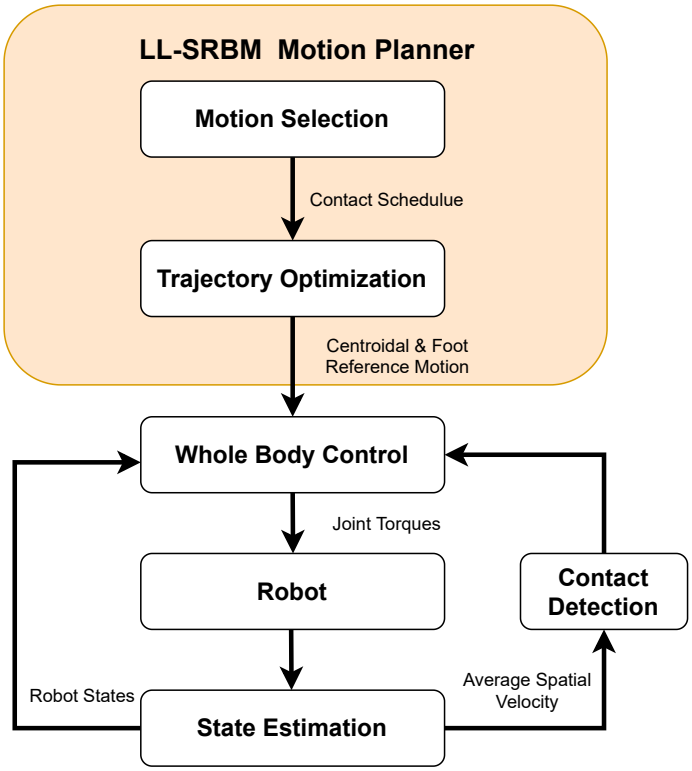


Figure 7.2: The planning and control framework.

7.3 THE LUMP LEG SINGLE RIGID BODY MODEL

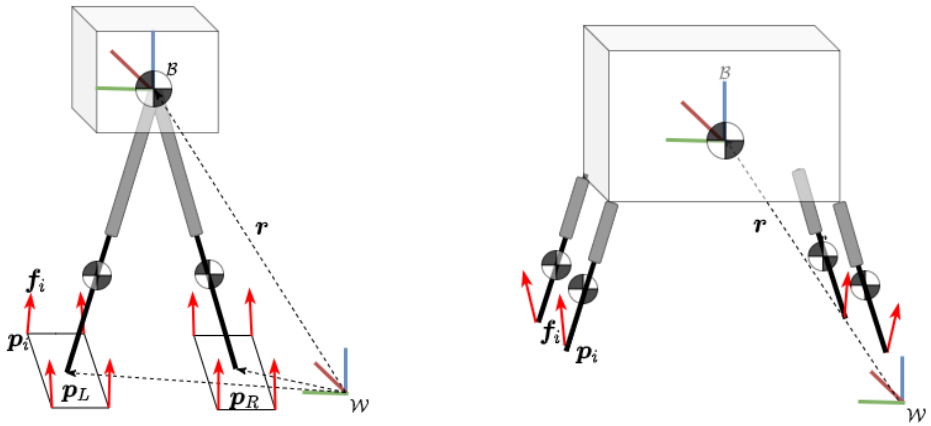
I am interested in generating high fidelity dynamic jump motions while keeping the model simple enough. The novel aspect of LL-SRBM model is that it abstracts the robot as a

single rigid body for the base and point masses for the legs, as illustrated in Fig. 7.3. The point mass of the leg is located at the CoM of the leg at the default configuration and the distance between the point mass and the leg contact location is proportional to the leg length. Compared to SRBM, the LL-SRBM models the leg dynamics as well, which is important for computing the centroidal inertia, as show in Fig. 7.4. The full Newton-Euler dynamics of the LL-SRBM with multiple contacts with the environment can be written as:

$$\begin{bmatrix} \dot{\mathbf{H}} = \sum_{i=1}^{n} \mathbf{f}_i + m\mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix} \tag{7.1}$$

where $\mathbf{H}$ and $\mathbf{L}$ are the linear and angular momentum of the model, $\mathbf{f}_i$ is the ith ground reaction force acting on $\mathbf{p}_i$, n forces are assumed. $\mathbf{r}$ is the position of the CoM, m and $\mathbf{g}$ are the mass of the model and the gravity vector. The linear momentum is directly related to the translational velocity of the CoM:

$$\dot{\mathbf{r}} = \mathbf{H}/m \tag{7.2}$$



(a) (b)

Figure 7.3: The LL-SRBM is a general model for bipedal and quadrupedal robots, it takes the leg dynamics into consideration. It also unifies planar, line and point feet with the distributed Ground Reaction Force representation. (a) shows LL-SRBM applied to a bipedal robot with planar feet. The ground reaction forces are located at four corners of the foot with arbitrary directions. (b) shows LL-SRBM applied to a quadrupedal robot with point feet. For both figures $\mathcal{W}$ and $\mathcal{B}$ stand for the world frame and the body frame.

For LL-SRBM, the CoM is computed as

$$\mathbf{r} = \frac{\mathbf{r}_B \cdot m_B + \sum_i \mathbf{r}_l^i \cdot m_l^i}{m_B + \sum_i m_l^i} \tag{7.3}$$

where $\mathbf{r}_B$ and m_B are CoM and mass of the body, $\mathbf{r}_l^i$ and m_l^i are the position and mass of the ith leg.

Angular momentum of the LL-SRBM, $\mathbf{L}$, can be expressed as:

$$\mathbf{L} = \mathbf{I}_W \boldsymbol{\omega} \tag{7.4}$$

where $\mathbf{I}_W$ represents the centroidal inertia of the model in the world frame and $\boldsymbol{\omega}$ is the angular velocity of the body. The Centroidal Composite Rigid Body Inertia (CCRBI) matrix in (Lee et al., 2007) can be computed as:

$$\mathbf{I}_W = \mathbf{R}_B \mathbf{I}_G \mathbf{R}_B^T = \mathbf{R}(\mathbf{q}_B)\mathbf{I}_G \mathbf{R}^T(\mathbf{q}_B) \tag{7.5}$$

where $\mathbf{R}(\mathbf{q}_B)$ is a transformation from local to world frame as given in the Appendix A.3. $\mathbf{I}_G \in \mathbb{R}^{3\times3}$ is the centroidal inertia matrix at CoM frame and can be computed as:

$$\mathbf{I}_G = \mathbf{I}_B + \sum_i \mathbf{I}_l^i = \mathbf{I}_B + \sum_i m_i(\mathbf{r} - \mathbf{p}_i + \Delta\mathbf{p}_i)^2 \tag{7.6}$$

where $\mathbf{I}_B$ is the inertia of the single rigid body projected to the CoM, and $\Delta\mathbf{p}_i$ is the offset between the leg's contact location and the position of leg mass, the second term represents the inertia of legs projected to CoM.

In (Di Carlo et al., 2018) the Euler angles are used to represent the orientation, however this approach suffers from the problem of gimbal lock, so quaternion representation is preferred for the orientation. The dynamics of the quaternion $\mathbf{q}_B$ can be expressed as:

$$\dot{\mathbf{q}}_B = \frac{1}{2}\mathbf{q}_B \circ \boldsymbol{\omega} = \mathbf{Q}(\mathbf{q}_B)\boldsymbol{\omega} \tag{7.7}$$

where $\circ$ represents the quaternion product and $\mathbf{Q}(\mathbf{q}_B) \in \mathbb{R}^{4\times3}$ is the corresponding matrix representation. By associating Equations (7.4), (7.5), (7.7), the dynamics of orientation can be represented by quaternion:

$$\dot{\mathbf{q}}_B = \frac{1}{2}\mathbf{Q}(\mathbf{q}_B)\mathbf{I}_W^{-1}(\mathbf{q}_B)\mathbf{L} \tag{7.8}$$

So the complete dynamics of the LL-SRBM can be summarized as:

$$\begin{bmatrix} \dot{\mathbf{r}} = \mathbf{H}/m \\ \dot{\mathbf{q}}_B = \frac{1}{2}\mathbf{Q}(\mathbf{q}_B)\mathbf{I}_W^{-1}(\mathbf{q}_B)\mathbf{L} \\ \dot{\mathbf{H}} = \sum_{i=1}^n \mathbf{f}_i + m\mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^n (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix} \tag{7.9}$$

with

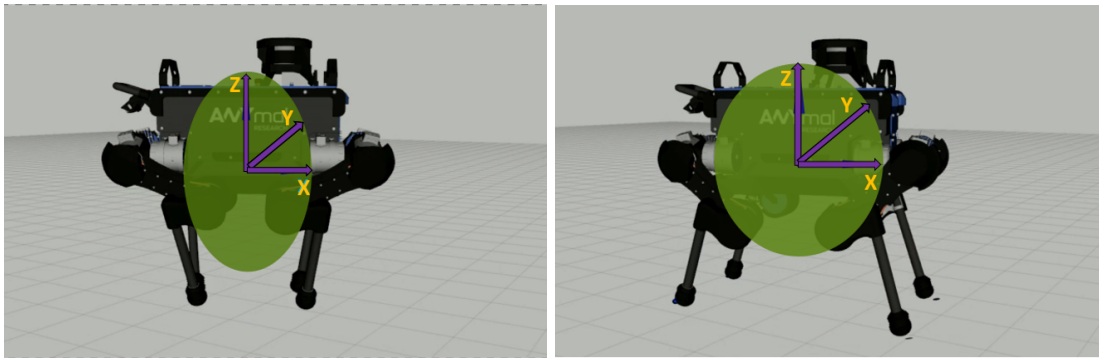
$$\mathbf{r} = \frac{\mathbf{r}_B \cdot m_B + \sum_i \mathbf{r}_l^i \cdot m_l^i}{m_B + \sum_i m_l^i} \tag{7.3}$$

$$\mathbf{I}_W = \mathbf{R}(\mathbf{q}_B)(\mathbf{I}_B + \sum_i m_i(\mathbf{r} - \mathbf{p}_i + \Delta\mathbf{p}_i)^2)\mathbf{R}^T(\mathbf{q}_B) \tag{7.10}$$

7.4 TRAJECTORY OPTIMIZATION

This section describes the mathematical formulation of trajectory optimization and how the trajectory optimization is formulated into a Nonlinear Programming (NLP) problem.

7.4.1 *Problem Formulation*



(a) (b)

Figure 7.4: The shape of the centroidal inertia of the robot is affected by the leg configuration, which can be modelled by LL-SRBM. (a) and (b) show two different leg configurations when ANYmal is performing a twisting jump. Compared with the centroidal inertia in (b), the centroidal inertia in (a) has smaller values on y and z axes.

The trajectory optimization (Kelly, 2017) is a powerful tool to find optimal trajectories for complex tasks that contain nonlinear dynamics and involves multiple phases. In this method, the *multiple shooting method* is employed to transcribe the trajectory optimization problem into a NLP formulation. In the formulation, multiple phases are considered since versatile dynamic jumps are focused that contains multiple pre-defined contact phases: *takeoff*, *flight phase* and *post-landing phase*.

The system has been discretised into N segments and therefore $N+1$ knots exists. Since m phases are assumed, each phase gets N/m segments. Because timing is important for

generating highly dynamic motions, time is not fixed for each phase ΔT. However the same time interval dt is kept between knots inside each phase to reduce the computation cost. For knot k, the state $\mathbf{x}[k]$ and control $\mathbf{u}[k]$ are defined as:

$$\begin{aligned}\mathbf{x}[k] =&[\mathbf{r}[k], \mathbf{q}_B[k], \mathbf{H}[k], \mathbf{L}[k]]\\ \mathbf{u}[k] =&[\mathbf{f}_1[k], ..., \mathbf{f}_{N_l}[k], \mathbf{p}_1[k], ..., \mathbf{p}_{N_l}[k]]\\ \boldsymbol{\xi}[k] =&[\mathbf{x}[k], \mathbf{u}[k], dt[k]]\end{aligned}$$

where $dt[k]$ is the time interval from knot k to $k+1$, N_l is the leg number, $\mathbf{f}_1[k], ..., \mathbf{f}_{N_l}[k]$ are the ground reaction forces acting on the feet. For bipedal robots with planar feet, the left foot and right foot collect four corresponding corner forces: $\mathbf{f}_L[k] = [\mathbf{f}_{L_1}[k], \mathbf{f}_{L_2}[k], \mathbf{f}_{L_3}[k], \mathbf{f}_{L_4}[k]]$, $\mathbf{f}_R[k] = [\mathbf{f}_{R_1}[k], \mathbf{f}_{R_2}[k], \mathbf{f}_{R_3}[k], \mathbf{f}_{R_4}[k]]$ as shown in Fig. 7.3a. $\mathbf{p}_L[k]$ and $\mathbf{p}_R[k]$ are the center position of the left foot and right foot respectively. For quadrupedal robots, $\mathbf{u}[k] = [\mathbf{f}_1[k], \mathbf{f}_2[k], \mathbf{f}_3[k], \mathbf{f}_4[k], \mathbf{p}_1[k], \mathbf{p}_2[k], \mathbf{p}_3[k], \mathbf{p}_4[k]]$, as represented by Fig. 7.3b.

Given the open parameters $\boldsymbol{\xi}[k]$, the complete trajectory optimization problem can be formulated as follows:

$$\begin{aligned}
&\min_{\boldsymbol{\xi}} \sum_{k=0}^{N} l(\boldsymbol{\xi}[k]) + \phi(\boldsymbol{\xi}[N]) && \text{(cost function)}\\
&\text{s.t.} \quad \mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k]) && \text{(LL-SRBM dynamics)}\\
&\qquad \mathbf{x}[0] = \mathbf{x}_{ini} && \text{(initial states)}\\
&\qquad \mathbf{x}[N] = \mathbf{x}_{fin} && \text{(final states)}\\
&\qquad t_{min} \leqslant dt[k] \leqslant t_{max} && \text{(time limits)}\\
&\qquad \|\mathbf{q}_B[k]\| = 1 && \text{(quaternion norm)}
\end{aligned}$$

$$l_{min} \leqslant \|\mathbf{r}[k] - \mathbf{p}_i[k]\| \leqslant l_{max} \quad (i \in \{1, ..., N_l\}) \qquad \text{(kinematics limits)}$$

if foot in contact:

$$\begin{aligned}
&0 \leqslant f_i^z \leqslant f_{max}^z \quad (i \in \{1, ..., N_l\}) && \text{(ground force limit)}\\
&\mathbf{P}\mathbf{f}_i \leqslant 0 \quad (i \in \{1, ..., N_l\}) && \text{(friction cone)}\\
&\mathbf{p}_i[k] = \mathbf{p}_i^* \quad (i \in \{1, ..., N_l\}) && \text{(given contacts)}
\end{aligned}$$

if foot in the air:

$$\mathbf{f}_i = 0 \quad (i \in \{1, ..., N_l\}) \qquad \text{(no contact force)}$$

7.4.2 *Cost Function*

The items inside the cost function can be categorized into 4 groups: *control inputs smoothness*, *energy consumption*, *time penalization* and *final state target*.

Control Inputs Smoothness: To generate a smooth motion, the difference of ground reaction forces and foot movements between adjacent nodes are minimized:

$$\sum_{k=0}^{N}\sum_{i=0}^{N_l}\left(\dot{\mathbf{p}}_i^2[k]+\dot{\mathbf{f}}_i^2[k]\right) \tag{7.11}$$

Energy Consumption: the squared momentum is minimized to achieve minimal energy consumption:

$$\sum_{k=0}^{N}(\mathbf{H}^2[k]+\mathbf{L}^2[k]) \tag{7.12}$$

Time Penalization: it is desirable to generate a minimum-time trajectory to finish a task while respecting the constraints. Also, this cost term encourages the robot to change the shape of the centroidal inertia in the *flight phase* to achieve a fast motion:

$$\sum_{i=0}^{N}(dt^2[i]) \tag{7.13}$$

Final State Target: Though final state targets are formulated as constraints, it is found that putting final orientation target into cost function can speed up the solving process. This is especially the case in tasks requiring large change of orientation, like twisting jump. So the final orientation targets is put into the cost function:

$$(\mathbf{q}_B[N]-\mathbf{q}_B^{fin})^2+(\dot{\mathbf{q}}_B[N]-\dot{\mathbf{q}}_B^{fin})^2 \tag{7.14}$$

7.4.3 *Kinematics Constraint*

The kinematics constraint bounds the distance between CoM and the feet to be within $[l_{min}, l_{max}]$,

$$l_{min}\leqslant\|\mathbf{r}[k]-\mathbf{p}_i[k]\|\leqslant l_{max}\quad(i\in\{1,...,N_l\}) \tag{7.15}$$

For the bipedal robot SLIDER, $l_{min} = 0.35$ m and $l_{max} = 0.75$ m. For the quadrupedal robot ANYmal, $l_{min} = 0.31$ m and $l_{max} = 0.6$ m.

7.4.4 *Contact Constraint*

The contact sequence is pre-defined as *takeoff phase*, *flight phase* and *landing phase*. The start and target position of the foot are pre-defined as well, namely:

$$\mathbf{p}_i[k] = \mathbf{p}_i^{ini} \quad (i \in \{1, ..., N_l\}) \qquad \text{(takeoff phase)}$$
$$\mathbf{p}_i[k] = \mathbf{p}_i^{fin} \quad (i \in \{1, ..., N_l\}) \qquad \text{(post-landing phase)}$$

7.4.5 *Ground Reaction Force Constraint*

In (Chignoli et al., 2021) the actuator limits are introduced in the planner by approximating the configuration-dependent reaction forces. Although joint angles of the robot are not included in LL-SRBM, the ground reaction force limit can still be approximated using the default configuration by:

$$\mathbf{f}_{max} = \mathbf{J}^{-T}(\mathbf{q}_0)\boldsymbol{\tau}_{max} \tag{7.16}$$

where $\mathbf{J}$ is the jacobian matrix, $\mathbf{q}_0$ represents the joint coordinates at the default configuration, $\boldsymbol{\tau}_{max}$ is the maximum joint torque vector. In *takeoff phase* and *landing phase* both feet need to be in contact with the ground, so all the ground reaction forces would be 0 or pushing against the ground, therefore the ground force limit is:

$$0 \leqslant f_i^z \leqslant f_{max}^z \quad (i \in \{1, ..., N_l\}) \tag{7.17}$$

No slippage of foot contact points is also ensured. The tangential forces are constrained to remain inside the Coulomb friction cone defined by the friction coefficient μ. The friction cone is approximated by the friction pyramid, which is a common approach to make the constraints linear and speed up the computation. The friction cone constraint is given by:

$$-\mu f_i^z \leqslant f_i^x \leqslant \mu f_i^z \tag{7.18}$$
$$-\mu f_i^z \leqslant f_i^y \leqslant \mu f_i^z \tag{7.19}$$

where f_i^x, f_i^y, f_i^z are components of the ground reaction force $\mathbf{f}_i$ $(i \in \{1, ..., N_l\})$.

If all feet is in the air, which is the case of *flight phase*, all ground reaction forces are enforced to be exactly 0:

$$\mathbf{f}_i = \mathbf{0} \quad (i \in \{1, ..., N_l\}) \tag{7.20}$$

7.5 WHOLE BODY CONTROLLER AND CONTACT DETECTION

The whole-body controller takes responsibility of computing the joint torques to achieve the desired motions while respecting a set of constraints. In our method, the tasks of interest are the CoM position, the pelvis orientation, the angular momentum of the robot, the foot positions and orientations. Each task is comprised of a desired acceleration as a feed-forward term and a state feedback term to stabilize the trajectory. Generally, the task for the linear motion can be expressed as:

$$\mathbf{J}_T \ddot{\mathbf{q}} = \ddot{\mathbf{x}}^{cmd} - \dot{\mathbf{J}}_T \dot{\mathbf{q}}, \tag{7.21}$$

$$\ddot{\mathbf{x}}^{cmd} = \ddot{\mathbf{x}}^{des} + \mathbf{K}_P^{pos}(\mathbf{x}^{des} - \mathbf{x}) + \mathbf{K}_D^{pos}(\dot{\mathbf{x}}^{des} - \dot{\mathbf{x}}), \tag{7.22}$$

where $\mathbf{J}_T$ is the translational Jacobian for the task, $\mathbf{x}$ is the actual position of the link, and the superscript des indicates the desired motion.

For the task of angular motion, the command can be formulated as:

$$\mathbf{J}_R \ddot{\mathbf{q}} = \dot{\boldsymbol{\omega}}^{cmd} - \dot{\mathbf{J}}_R \dot{\mathbf{q}}, \tag{7.23}$$

$$\dot{\boldsymbol{\omega}}^{cmd} = \dot{\boldsymbol{\omega}}^{des} + \mathbf{K}_P^{ang}(\text{AngleAxis}(\mathbf{R}^{des}\mathbf{R}^T)) + \mathbf{K}_D^{ang}(\boldsymbol{\omega}^{des} - \boldsymbol{\omega}), \tag{7.24}$$

where $\mathbf{J}_R$ is the rotational Jacobian for the task, $\mathbf{R}$ and $\mathbf{R}^{des}$ denote the actual and desired orientation of the pelvis link respectively, AngleAxis() maps a rotation matrix to the corresponding axis-angle representation, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the link.

For the CoM task and angular momentum task, the centroidal momentum matrix (Orin et al., 2013) is used as the task jacobian. For balancing or walking, angular momentum task are often defined as a damping task that damps out excess angular momentum. However for highly dynamic motion such as twisting jump, angular momentum varies a lot during the process and it plays a vital role to achieve the jump motion. In this case, the reference angular momentum is needed and it comes from our motion planner. Since the single rigid

body model is used in the motion planner, its orientation and associated angular momentum are both well defined.

7.5.1 *QP formulation*

Inspired by (Herzog et al., 2015), the full dynamics can be decomposed into the underactuated part and actuated part:

$$\begin{bmatrix} \mathbf{M}_f \\ \mathbf{M}_a \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{H}_f \\ \mathbf{H}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{S}_a \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{J}_f^T \\ \mathbf{J}_a^T \end{bmatrix} \mathbf{f}, \tag{7.25}$$

where $\mathbf{M}$, $\mathbf{H}$, $\mathbf{S}_a$, $\boldsymbol{\tau}$, $\mathbf{J}$ and $\mathbf{f}$ are the mass matrix, Coriolis force matrix and gravitation force vector, the actuator selection matrix, joint torques vector, the stacked contact Jocabian and reaction force vector. The subscript, f and a, indicates the floating part and actuated part respectively. The weighted sum formulation is applied, in which one QP problem is solved at each control loop. The formulation of the QP problem can be written as

$$\begin{aligned} \min_{\ddot{\mathbf{q}}, \mathbf{f}} \quad & \frac{1}{2}\|\mathbf{A}\ddot{\mathbf{q}} + \dot{\mathbf{A}}\dot{\mathbf{q}} - \mathbf{B}^{cmd}\|_{W_1}^2 + \frac{1}{2}\|\mathbf{f} - \mathbf{f}_{des}\|_{W_2}^2 && (7.26) \\ \text{s.t.} \quad & \mathbf{M}_f\ddot{\mathbf{q}} - \mathbf{J}_f^T\mathbf{f} = -\mathbf{H}_f && \text{(floating base dynamics)} \\ & \mathbf{P}\mathbf{f} \leqslant \mathbf{0} && \text{(friction cone)} \\ & \mathbf{S}_a^{-1}(\mathbf{M}_a\ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T\mathbf{f}) \in [\boldsymbol{\tau}_{min}, \boldsymbol{\tau}_{max}] && \text{(input limits)} \end{aligned}$$

where $\mathbf{A}$ is a stack of the Jacobian matrices for the tasks of interest, $\mathbf{B}^{cmd}$ is a stack of the commanded accelerations and $\mathbf{W}_i$ $(i = 1, 2)$ are the weighting matrices, $\mathbf{P}$ denotes the linearized friction cone matrix. Similar to (Apgar et al., 2018)(Kuindersma et al., 2016) , the unilateral contact constraint is treated as a soft constraint by simply assigning a large weight on the desired zero acceleration. It is reported in (Feng et al., 2014) that this gives a better stability.

The output torque commands $\boldsymbol{\tau}$ at each control iteration is computed by

$$\boldsymbol{\tau} = \mathbf{S}_a^{-1}(\mathbf{M}_a\ddot{\mathbf{q}} + \mathbf{H}_a - \mathbf{J}_a^T\mathbf{f}) \tag{7.27}$$

Since the SRBM is an approximate model, $\mathbf{W}_2 = 0$ is set to track the desired trajectory in the jump motion, except the landing phase. In the landing phase, in order to achieve soft landing, tracking desired contact forces is switched by increasing $\mathbf{W}_2$ while decreasing the weights for centroidal momentum control to be 0. After a short safe landing phase (around 0.2 second), Trajectory tracking control is switched back in order to achieve the desired

steady pose. This strategy of switching to force control during landing is similar to (Chignoli et al., 2021), which is particular critical for real robot experiments in the existing of trajectory tracking and state estimation errors.

7.5.2 *Contact detection*

According to (Orin et al., 2013), the spatial average velocity of a robot is defined as

$$\mathbf{v}_G = \begin{bmatrix} \boldsymbol{\nu}_{com} \\ \boldsymbol{\omega}_G \end{bmatrix} = \mathbf{I}_f^{-1} \mathbf{h}_G \tag{7.28}$$

where $\mathbf{h}_G \in \mathbb{R}^6$ is the centroidal momentum of the robot, $\mathbf{I}_f \in \mathbb{R}^{6\times6}$ is the centroidal inertia matrix of the robot, $\boldsymbol{\nu}_{com}$ is the center of mass velocity, $\boldsymbol{\omega}_G$ denotes the average angular velocity of the robot. Please refer to (Orin et al., 2013) for the details of $\mathbf{h}_G$ and $\mathbf{I}_f$.

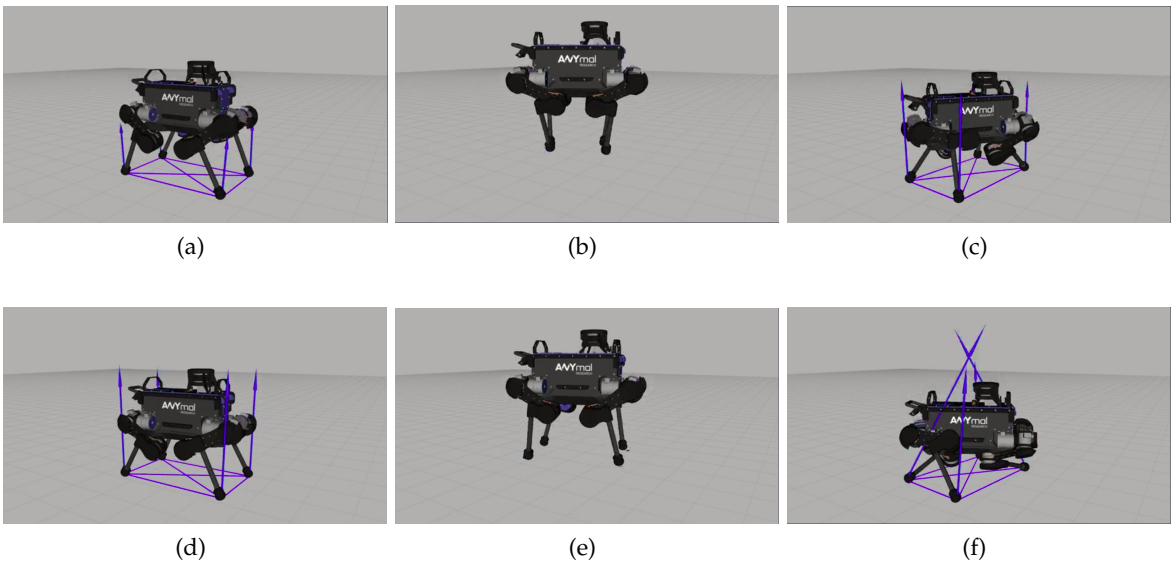
The change rate of $\|\mathbf{v}_G\|_2$ is used to judge the contact. During the flight phase, the gravity is the only external force that changes the state of the robot. The norm of $\boldsymbol{\nu}_{com}$ will increase under the gravity while the robot is dropping down from the peak. When contacts happen, the velocities will decrease. In order to respond to all kinds of contact, $\|\mathbf{v}_G\|_2$ is used to detect contacts which is a comprehensive metric including all the changes of velocities of each link of a robot. When the numerical derivatives of $\|\mathbf{v}_G\|_2$ in a sampling window are all less than a threshold, the controller is told that the robot gets contact. In practice, the proposed method is more reliable than using force signals that are very noisy.

7.6 EXPERIMENT RESULTS

7.6.1 *Robot Platforms*

SLIDER is a knee-less bipedal robot designed by the Robot Intelligence Lab at Imperial College London (Wang et al., 2020a, Wang et al., 2018). It is 1.2 m tall and has 10 actuated joints with a total weight of 16 Kg. Most of its weight is concentrated in the pelvis. The prismatic knee joint design is an unique feature of this robot that differentiates it from many other robots with anthropomorphic design. Also the sliding joint has a relatively large range of motion. The overall light weight and large range of leg motion make the robot suitable for agile locomotion.

The quadrupedal robot ANYmal is made by ANYbotics. It has 12 SEAs (Series Elastic Actuators) and weighs approximately 35 Kg. When ANYmal standing still in the default configuration, it is about 0.5 m tall.



(a) (b) (c)

(d) (e) (f)

Figure 7.5: Snapshots of the ANYmal robot performing a 90° twisting jump with inertia shaping and without inertia shaping. First row: twisting jump with inertia shaping. Second row: twisting jump without inertia shaping.

7.6.2 *Implementation*

The trajectory optimization framework is implemented in CasADi (Andersson et al., 2019) with Python using the interior-point solver IPOPT (Wächter et al., 2006). The whole body controller is written with C++ using the Pinocchio library (Carpentier et al., 2015–2021) to compute full rigid body dynamics and qpOASES (Ferreau et al., 2014) to solve the QP problem. For SLIDER robot the whole body controller runs at 1k Hz to track the desired trajectory. For ANYmal the whole-body controller is running at 400 Hz in simulation and real experiments as well. For both robots the simulation is done in Gazebo with ODE as the physics engine.

7.6.3 *Twisting Jump*

The twisting jump is experimented on both ANYmal and SLIDER robots and compared the performance between the trajectories generated by LL-SRBM and SRBM. In the trajectory optimization a simple linear interpolation of the orientation is provided as the initial guess.

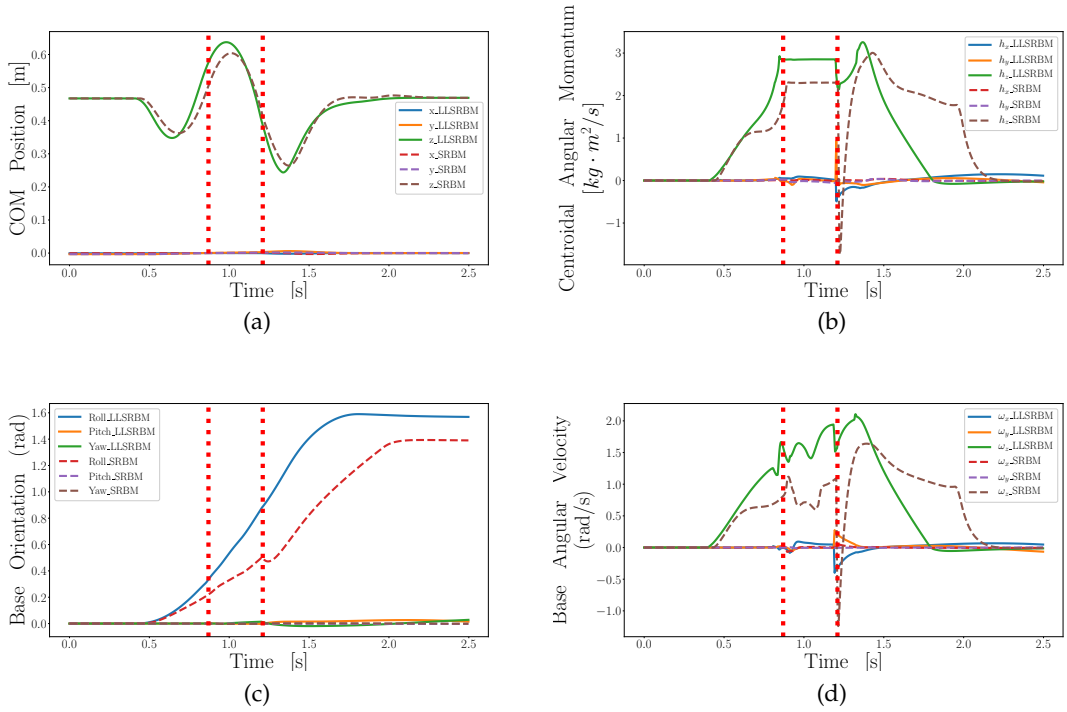


Figure 7.6: The measured trajectories of ANYmal 90° twisting jump in simulation, with measured trajectories generated by LL-SRBM and SRBM repectively. In each plot the vertical dashed lines split the entire trajectory into three phases: *takeoff*, *flight* and *landing*. The full lines show the measured trajectories with LL-SRBM and the dashed lines show measured trajectories with SRBM.

Also a large weight to the *final state target* and *time penalization* in the cost function are assigned to encourage the solver to find a trajectory that reaches the target pose in a minimum time.

As shown in Fig. 7.5, the foot motion generated with LL-SRBM has a big difference compared to the one generated with SRBM. In the flight phase, the feet try to keep close to the central rotating axis with the LL-SRBM generated trajectories. The SRBM generated foot trajectories just do an interpolation from the start to the goal position. This interesting behaviour emerges as LL-SRBM takes leg dynamics into consideration in the planning and tries to maximize the angular velocity in flight phase by modifying the shape of inertia with the leg motion. Fig 7.6 shows the measured trajectories of the twisting jump experiments. It can be seen clearly that in flight phase the base angular velocity in experiments with LL-SRBM does not drop as much as the base angular velocity in experiments with SRBM. The robot reaches the goal orientation earlier with LL-SRBM. To accomplish the twisting jump for SLIDER, two joints for the robot have been added, one yaw joint per leg. SLIDER can perform a 90° twisting jump and the motion can be seen in the accompany video.

In real experiments, the effectiveness of the proposed approaches is demonstrated by twisting jumps on unknown objects, as shown in Fig 7.8. A wooden board with a height of

25 mm was put under ANYmal when it was in the flight phase. Fig 7.7 shows the norm of the average spatial velocity. Even though collected from real robot experiments, the norm of the average spatial velocity is smooth and can be used as a reliable judgement for contact detection. The robot successfully detected the contact event and switched to a soft landing.

7.6.4 *Forward Jump*

In the trajectory optimization, a large weight to the *final state target* and *energy consumption* is assigned. For ANYmal, a forward jump motion of 30 cm is generated with LL-SRBM and SRBM, as show in Fig. 7.9. It can be seen that for trajectories generated with LL-SRBM model, the feet stretch out more in the air than trajectories generated with SRBM. This is because the optimizer tries to increase the inertia around y axis to reduce the change of base pitch angle. The stretched feet in the flight phase also help to prevent early touchdown of the robot.

Jumping onto a box for SLIDER with LL-SRBM is also tried. The maximum height of the box SLIDER can jump on is 35 cm, with a forward jump length of 20 cm, as shown in Fig. 7.10. Considering that SLIDER is 1.2 m high, the robot can jump onto a box equal to 30% of its total height. Compared with the anthropomorphic robot design which has knees, SLIDER's straight legs have a larger range of motion, allowing SLIDER to reach the same jump height with a relatively smaller CoM height. Because there are no knee joints on legs of SLIDER, the feet can go all the way up until the ankles touch the pelvis. That is the reason why SLIDER can jump onto a high box.

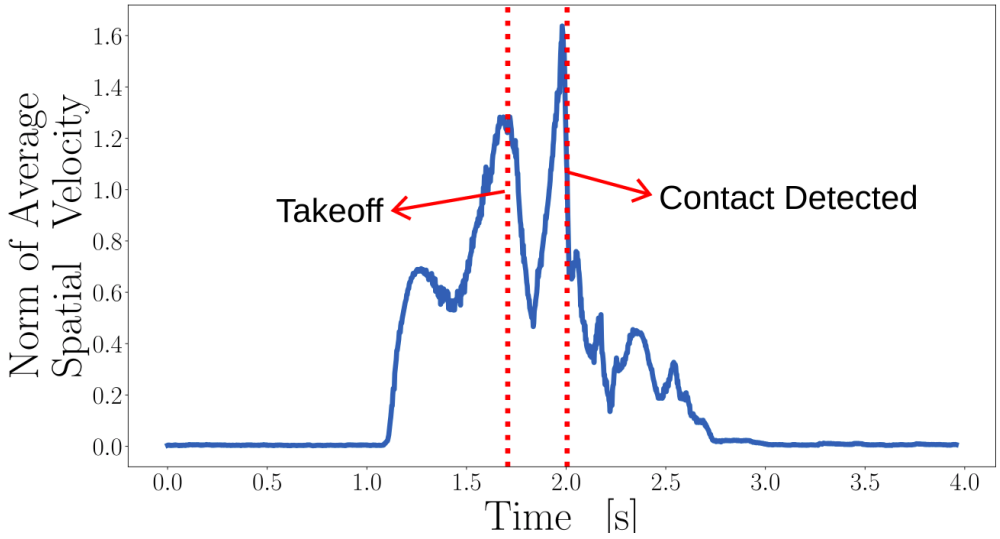
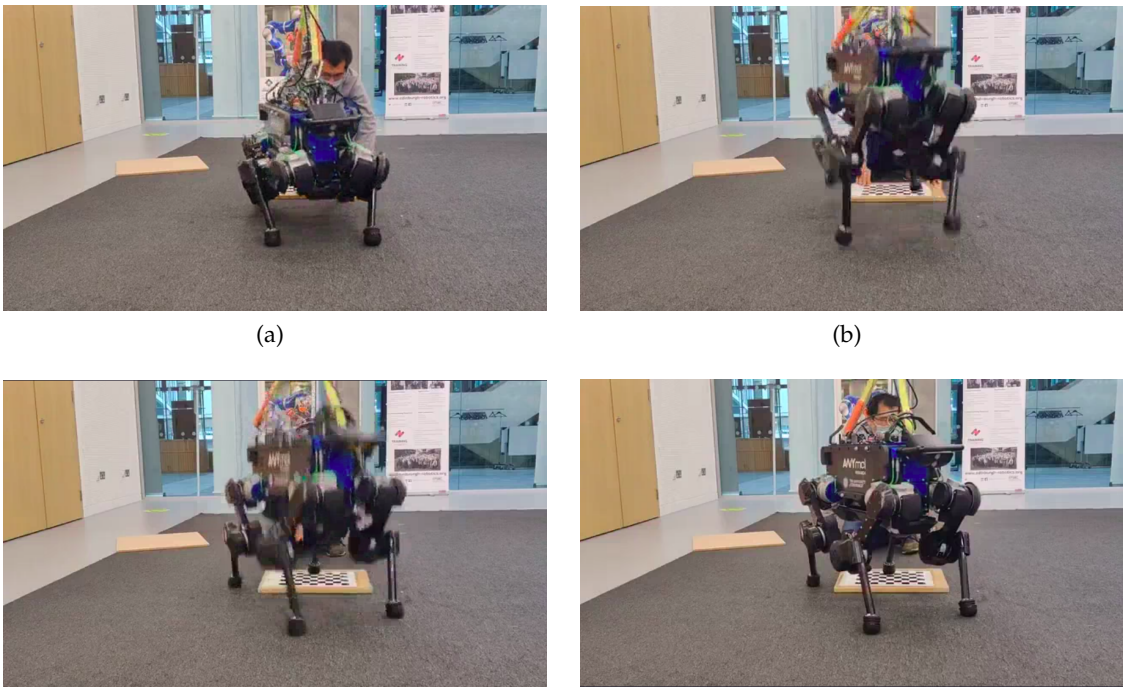


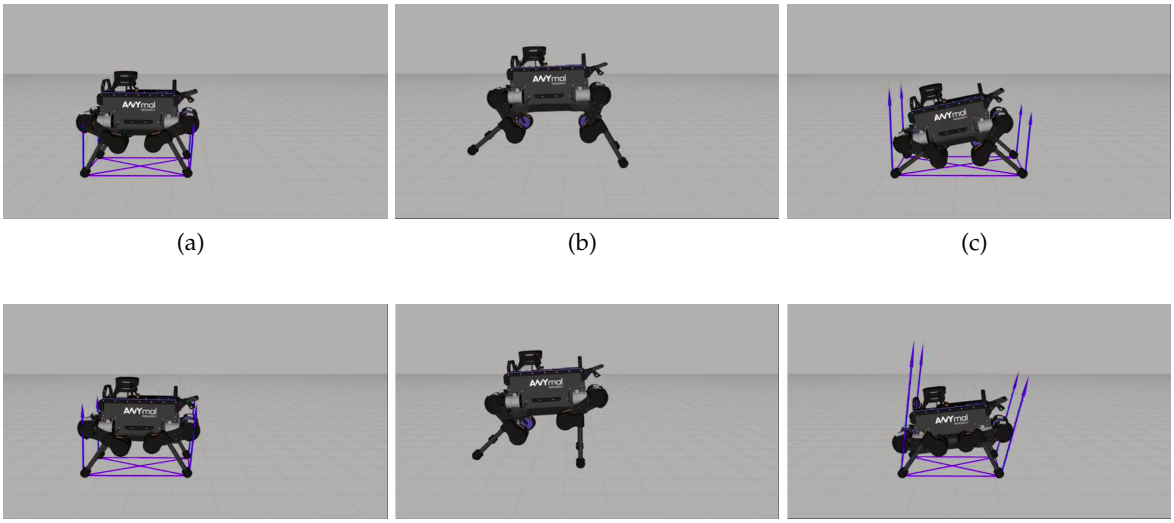
Figure 7.7: The norm of average spatial velocity when ANYmal performed a twisting jump. A wooden board was put under the robot while it was flying. A big drop of the value after getting contact when the robot was dropping down.



(a) (b)

(c) (d)

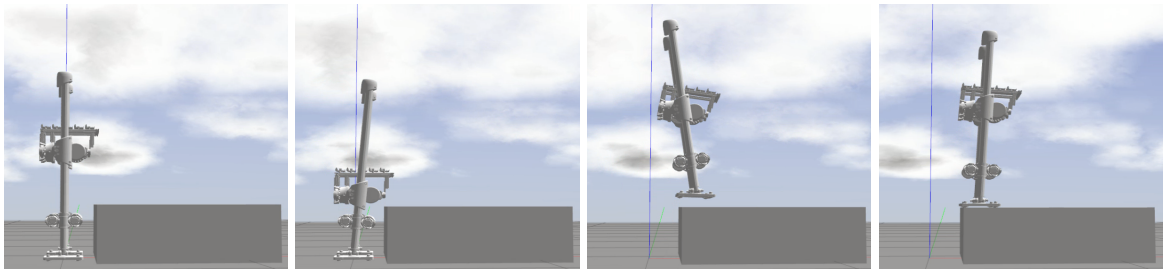
Figure 7.8: Snapshots of ANYmal performing a twisting jump with 30°. A wooden board with a height of 25 mm was put under the robot when it was in the *flight* phase to test the contact detection and the robustness of our controller. From left to right: (a): takeoff phase, (b): flight phase, (c): contact is detected, (d): the robot is stable after a soft landing.



(a) (b) (c)

(d) (e) (f)

Figure 7.9: Snapshots of the ANYmal robot performing a forward jump of 30 cm with inertia shaping and without inertia shaping. First row: forward jump with inertia shaping. Second row: forward jump without inertia shaping.



(a) (b) (c) (d)

Figure 7.10: Snapshots of SLIDER performing a forward jump onto a 35 cm box using LL-SRBM. (a) ~ (b): takeoff phase, (c): flight phase, (d): landing phase.

Table 7.1: Solve Times for Planners.

	quadruped forward jump	quadruped twist jump	biped forward jump	biped twist jump
SRBM	1.50s	2.45s	6.33s	5.44s
LL-SRBM	1.53s	2.56s	9.54s	7.37s

7.6.5 *Computation Time*

The computation times of trajectory optimizations is recorded for various dynamic jump motions. For the bipedal robot 60 knots are used and each knot has 59 variables (26 for states and 33 for control inputs). For the quadrupedal robot 50 knots are used and each knots has 53 variables (26 for states and 27 for control inputs). Although the complexity of LL-SRBM has increased compared with SRBM, the computation time with LL-SRBM only increase by an average of 33.6% with respect to that with SRBM, as shown in Table 7.1.

7.7 DISCUSSION

This chapter proposes an unified model with inertia shaping for planning highly dynamic jumps of legged robots. This model allows the motion planner to improve the jumping performance by actively changing the centroidal inertia. In the meanwhile, this chapter also proposes a novel contact detection method using the norm of average spatial velocity. The twisting jump and forward jump experiments on bipedal robot SLIDER and quadrupedal robot ANYmal show the improved jump performance after using the proposed model and the robustness of the controller to unforeseen impacts. In the future, it is interesting to speed up the computation to re-plan the jump motion with the proposed LL-SRBM.

CHAPTER 8

RotoGait

8.1 INTRODUCTION

In this chapter a novel gait — RotoGait is introduced. This gait is a step beyond evolution. For anthropomorphic robots, the gait is usually to emulate the nature that inspired the design. For example, a humanoid robot walks like a human and a quadruped robot has trotting gait to mimic a dog. SLIDER, on the other hand, does not try to mimic nature and it is designed to improve the walking ability. A unique feature of SLIDER is that it has knee-less legs and its pitch joint can rotate continuously. By rotating the leg and placing a foot on the other side of the leg, a novel gait can be performed and it is termed "RotoGait" for short. This novel gait is illustrated in Fig. 8.1. Interestingly, the motion of the stance leg and CoM are the same as the normal walking gait, meaning the footstep plan can be the same and only the swing leg trajectory is different. This gait shows promise for walking faster and being more energy efficient at fast speed, as the leg can rotate continuously when touching the floor instead of decelerating and reversing the direction as in the normal gait.

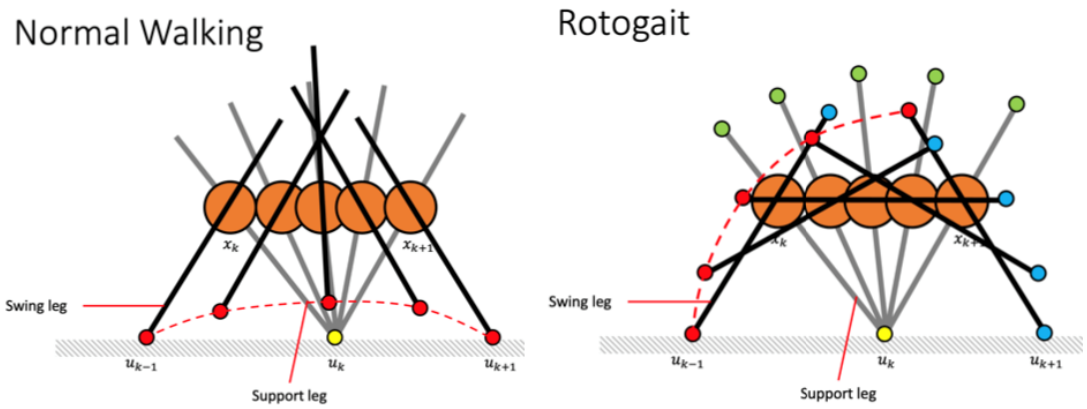
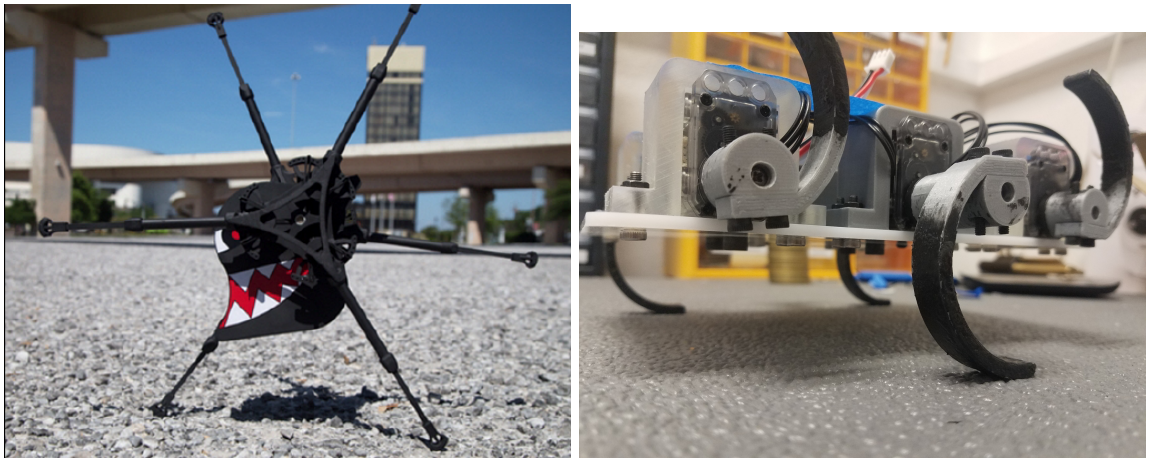


Figure 8.1: Normal walking gait *vs* RotoGait, the difference lies in the swing leg trajectory.

There has no published material on RotoGait as it is inspired by the unique design of SLIDER. The closest research is the rimless wheel. These are multi-spoke robots that use rolling motion to move around. Normally they have more than 2 spokes since the increased number of spokes resembles more like a wheel and the motion is therefore smoother and the robot is easier to control. One outstanding example of the rimless wheel robot is outrunner, as seen in Fig and an earlier version of it sets the world speed record (OutRunner, 2021). However the angles between spokes are fixed and the length of spoke cannot be adjusted. Moreover, it can only walk in 2D instead of 3D. Another robot which is relevant is the miniRHex robot (Barragan et al., 2018). MiniRHex is a hexapod built to mimic the way of cockroaches move through the environment. Each leg has only one motor and the leg has a passive compliant shape. The robot moves by rotating its leg. However, similar as the constraints of rimless wheels, one DoF on each leg restricts the robot to move only in 2D and the robot cannot adjust its CoM height. One of the closest examples to RotoGait exists only in animation and has not been physically built. This is the robot TARS from the movie Interstellar (Interstellar, 2015). It exhibited a rolling gait similar as RotoGait, but TARS has 4 legs instead of 2 legs.



(a) (b)

Figure 8.2: (a): the outrunner robot that applies the principle of the rimless wheel (OutRunner, 2021). (b) the miniRHex robot (Barragan et al., 2018).

8.2 KINEMATICS

The kinematics is derived to convert from joint space to task space and vice versa. A similar kinematics has been derived in chapter 4 but RotoGait is slightlt different. Spherical coordinate is used as it matches the configuration of RotoGait perfectly. However, different from the most common seen spherical coordinate which involves yaw rotation, in RotoGait

roll and pitch rotations are involved. As shown in Fig. 8.3, given the hip roll angle θ and hip pitch angle ϕ, the forward kinematics can be calculated as:

$$x = X_{CoM} + L \sin \phi \tag{8.1}$$

$$y = Y_{CoM} + L \cos \phi \cos \theta \tag{8.2}$$

$$z = Z_{CoM} + L \cos \phi \sin \theta \tag{8.3}$$

where $X_{CoM}, Y_{CoM}, Z_{CoM}$ are coordinates of CoM of SLIDER and L is the length from slide joint to the top of the rod, which is used as the hip slide value. The inverse kinematics can

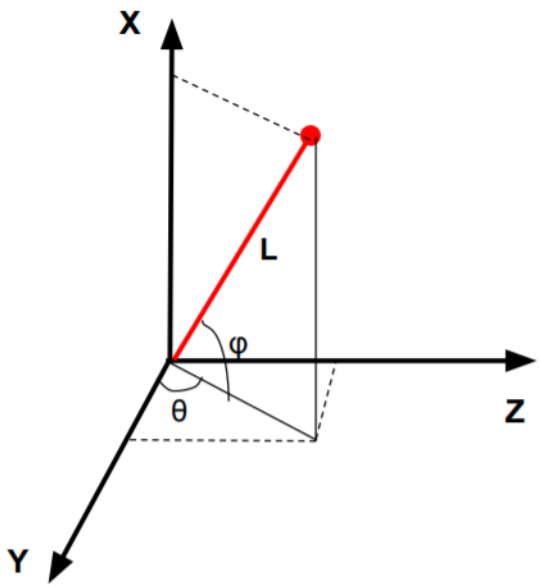


Figure 8.3: RotoGait kinematics represented in spherical coordinates.

be calculated as:

$$L = \sqrt{(x - X_{CoM})^2 + (y - Y_{CoM})^2 + (z - Z_{CoM})^2} \tag{8.4}$$

$$\theta = \arctan(\frac{z - Z_{CoM}}{y - Y_{CoM}}) \tag{8.5}$$

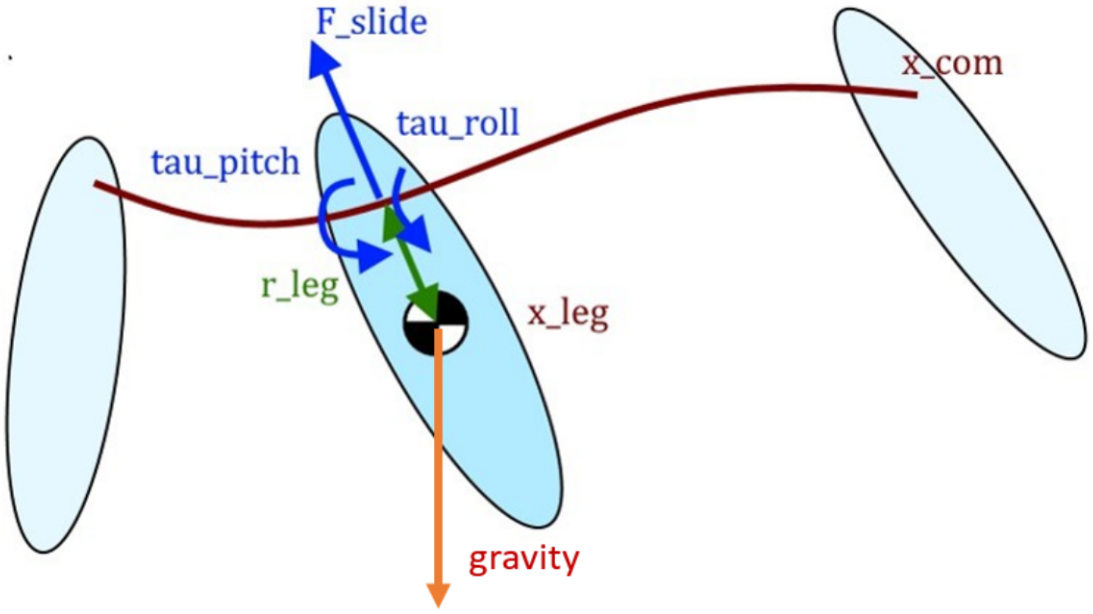


Figure 8.4: Illustration of swing leg dynamics of RotoGait. The external forces and torques act on the swing leg are torques of hip pitch and hip roll, forces from hip slide and gravity.

$$\phi = \arctan\left(\frac{(x - X_{CoM})\cos\theta}{y - Y_{CoM}}\right) \tag{8.6}$$

8.3 DYNAMICS

The essential difference between RotoGait and normal walking gait is the swing leg trajectory. To generate the optimal swing leg trajectory, it is necessary to derive the dynamics in joint space. The dynamics analysis is illustrated in Fig. 8.4 and the dynamics is very similar with Single Rigid Body Dynamics. The only difference is that because the leg is in the air, there is no ground reaction forces. However, to facilitate the trajectory optimization to be mentioned later, the dynamics is derived in joint space (spherical coordinate) instead of the commonly used task space (cartesian coordinate). The dynamics of the swing leg, analysed by Newton's method, is composed of the effect of gravity, acceleration and centripetal force. The forward dynamics in joint space is below:

$$\ddot{\phi} = \frac{-mg\sin\phi\cos\theta - m\ddot{X}_{CoM}\cos\phi - m\ddot{Y}_{CoM}\sin\phi\cos\theta + m\ddot{Z}_{CoM}\sin\phi\cos\theta + \tau_{pitch}}{I_{xx}} \tag{8.7}$$

$$\ddot{\theta} = \frac{-mg\cos\phi\sin\theta - m\ddot{Y}_{CoM}\cos\phi\sin\theta + m\ddot{Z}_{CoM}\cos\phi\cos\theta + \tau_{roll}}{I_{yy}} \tag{8.8}$$

$$\ddot{L} = g\cos\phi\sin\theta - \ddot{X}_{CoM}\sin\phi - \ddot{Y}_{CoM}\cos\phi\cos\theta + \ddot{Z}_{CoM}\cos\phi\sin\theta + \dot{\phi}^2 L + \dot{\theta}^2 L - \frac{F_L}{m} \tag{8.9}$$

where $\tau_{pitch}, \tau_{roll}, F_L$ are joint torques of hip pitch, hip roll and joint force of hip slide. I_{xx}, I_{yy} are inertia about pitch and roll axes, respectively. m is the mass of swing leg. For SLIDER m is 0.6 kg and $I_{xx} = I_{yy} = 0.0737 kgm^2$. Here because the distance between the leg CoM and the rotate axis is relatively small compared with the whole leg length, I_{xx} and I_{yy} are constant.

8.4 OPTIMAL SWING LEG TRAJECTORY

As mentioned at introduction, the advantage of RotoGait is the energy efficiency when walking fast. Therefore it is important to generate optimal swing leg trajectory in terms of energy consumption.

8.4.1 *Cost Function*

For the real robot, the energy consumption can be calculated by integrating power which is represented by the multiplication of current and voltage. For theoretical calculation and simulation, the energy can be represented as mechanical energy and electrical losses. The electrical losses, I^2R, where R is the motor resistance) can be measured as a function of the motor torque (τ) and the motor torque constant (K), by $I = \frac{\tau_i}{K_i}$. Table 8.1 lists the relevant parameters of motors used on SLIDER. Therefore the energy can be calculated as Equation 8.10:

$$E = \sum_{i\in J}\int_0^T (I_i(t)^2 R_i + \tau_i(t)\omega_i(t))dt = \sum_{i\in J}\int_0^T (\frac{\tau_i^2 R_i}{K_i^2} + \tau_i(t)\omega_i(t))dt \tag{8.10}$$

where T is the total time of walking, i is the index of the motor, J is the set of motors, ω is the angular velocity. For hip slide joint which has a linear motion, a gear of radius 9.55 mm is used to convert angular motion to linear motion. However, optimizing mechanical energy will cause a behaviour known as "bang-bang" solution where the solver applies maximum

torque at zero velocity or zero torque at nonzero velocity to bring the mechanical energy into zero. Therefore only electrical losses are minimized and can the cost be summarized as:

$$E = \sum_{i \in J} \int_0^T (\frac{\tau_i^2 R_i}{K_i^2}) dt \tag{8.11}$$

8.4.2 *Constraints*

State Constraints

State constraints include position and velocity constraints. The start and end position of the foot is determined by the footstep planner and should be constrained. The corresponding joint angles can be calculated by inverse kinematics.

The foot velocity is also constrained to be zero at the start and end of the walking cycle, this allows the holding of non-slip condition. The zero velocity constrain also help to minimize the impact the foot acts on the floor. The pitch and roll joint velocities at this condition can be represented as below:

$$\dot{\phi} = \frac{\dot{X}_{CoM} \cos\phi + \dot{Y}_{CoM} \sin\phi \cos\theta + \dot{Z}_{CoM} \sin\phi \sin\theta}{L + 0.5 \cdot L_{total}} \tag{8.12}$$

$$\dot{\theta} = \frac{\dot{Y}_{CoM} \cos\phi \sin\theta + \dot{Z}_{CoM} \cos\phi \cos\theta}{L + 0.5 \cdot L_{total}} \tag{8.13}$$

Interestingly, the leg is rotated during the RotoGait, therefore the foot is flipped at the end of the walking cycle compared with the start. The slide velocity at the start is:

$$\dot{L} = -\dot{X}_{CoM} \sin\phi - \dot{Y}_{CoM} \cos\phi \sin\theta + \dot{Z}_{CoM} \cos\phi \sin\theta \tag{8.14}$$

the slide velocity at the end is:

$$\dot{L} = \dot{X}_{CoM} \sin\phi - \dot{Y}_{CoM} \cos\phi \sin\theta + \dot{Z}_{CoM} \cos\phi \sin\theta \tag{8.15}$$

For the normal walking, in which the leg does not rotate, the constraints for roll and pitch are the same, the slide velocity constraints are same as Equation 8.14 for both start and end.

Physical Constraints

The physical constraints, namely joint position, velocity and torque limits are listed in table 8.1.

Table 8.1: Motor Limits for Hip Joints in SLIDER.

Joint	Joint Limit	Torque Limit (N · m)	Velocity Limit (rad/s)	K_t (Nm/A)	$R_m(\Omega)$
hip pitch	N/A	18	55.1	0.095	0.270
hip roll	+20°/-90°	48	6.9	0.095	0.483
hip slide	+/-0.4m	3	57.8	0.068	0.786

8.4.3 *Trajectory Optimization Formulation*

Multiple shooting is used to formulate our trajectory optimization problem, where the trajectories are sampled into a certain number of grid points. The control input is joint torques and states are joint position and velocities, the formulation is below:

$$\min_{\mathbf{x},\dot{\mathbf{x}},\mathbf{u}} \sum_{k=0}^{N} f(\mathbf{u}[k])) \qquad \text{(cost function)}$$

$$\text{s.t.} \quad \mathbf{x}[k+1] = f(\mathbf{x}[k], \mathbf{u}[k]) \qquad \text{(swing foot dynamics)}$$

$$\mathbf{x}[0] = \mathbf{x}_{ini} \qquad \text{(initial states)}$$

$$\mathbf{x}[N] = \mathbf{x}_{fin} \qquad \text{(final states)}$$

$$l_{min} \leqslant \mathbf{x}[k] \leqslant l_{max} \qquad \text{(kinematic limits)}$$

$$v_{min} \leqslant \dot{\mathbf{x}}[k] \leqslant v_{max} \qquad \text{(velocity limits)}$$

$$\tau_{min} \leqslant \mathbf{u}[k] \leqslant \tau_{max} \qquad \text{(torque limits)}$$

8.5 FOOTSTEP PLANNER

The same footstep planner can be employed which is used by normal walking gait for the RotoGait. Here the commonly used LIP model is used to generate footsteps and reference CoM motion and give it as inputs to the trajectory optimization. The CoM height is set to be 0.58 m. As seen in Fig. 8.5, the height of CoM is constant and the step length can be generated by LIP planner and the optimization details can be seen in chapter 5. The reference CoM motion can be calculated based on LIP dynamics given the footsteps.

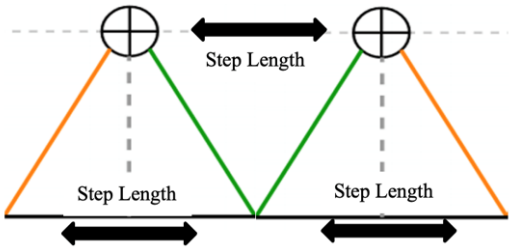


Figure 8.5: Footstep plan and CoM reference motion generation with LIP model.

8.6 RESULT

The trajectory optimization is implemented in Python with Casadi and IPOPT is used as the nonlinear optimizer. The number of nodes N in the trajectory is 220 and step time T_s is 0.4s. Fig. 8.6 visualizes the result of trajectory optimization with a forward step length of 0.4m. It is can be seen that the trajectory is smooth and the energy consumption for this trajectory is 115.62 J.

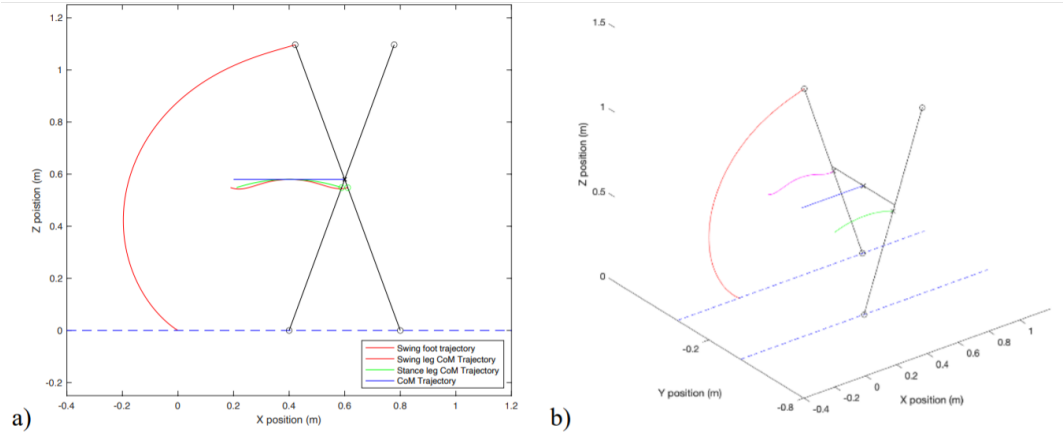


Figure 8.6: Visualization of RotoGait with a forward step length of 0.4m. (a) shows the xz plane and (b) shows the 3D trajectory.

8.6.1 *Normal Walking Gait*

The swing foot trajectory for normal walking gait is commonly generated by polynomials (Khadiv et al., 2020), however it is not optimal in terms of energy consumption. It is realized that the same trajectory optimization formulation can be applied to generating swing leg trajectory of normal walking. Besides the original formulation that is used for RotoGait, two

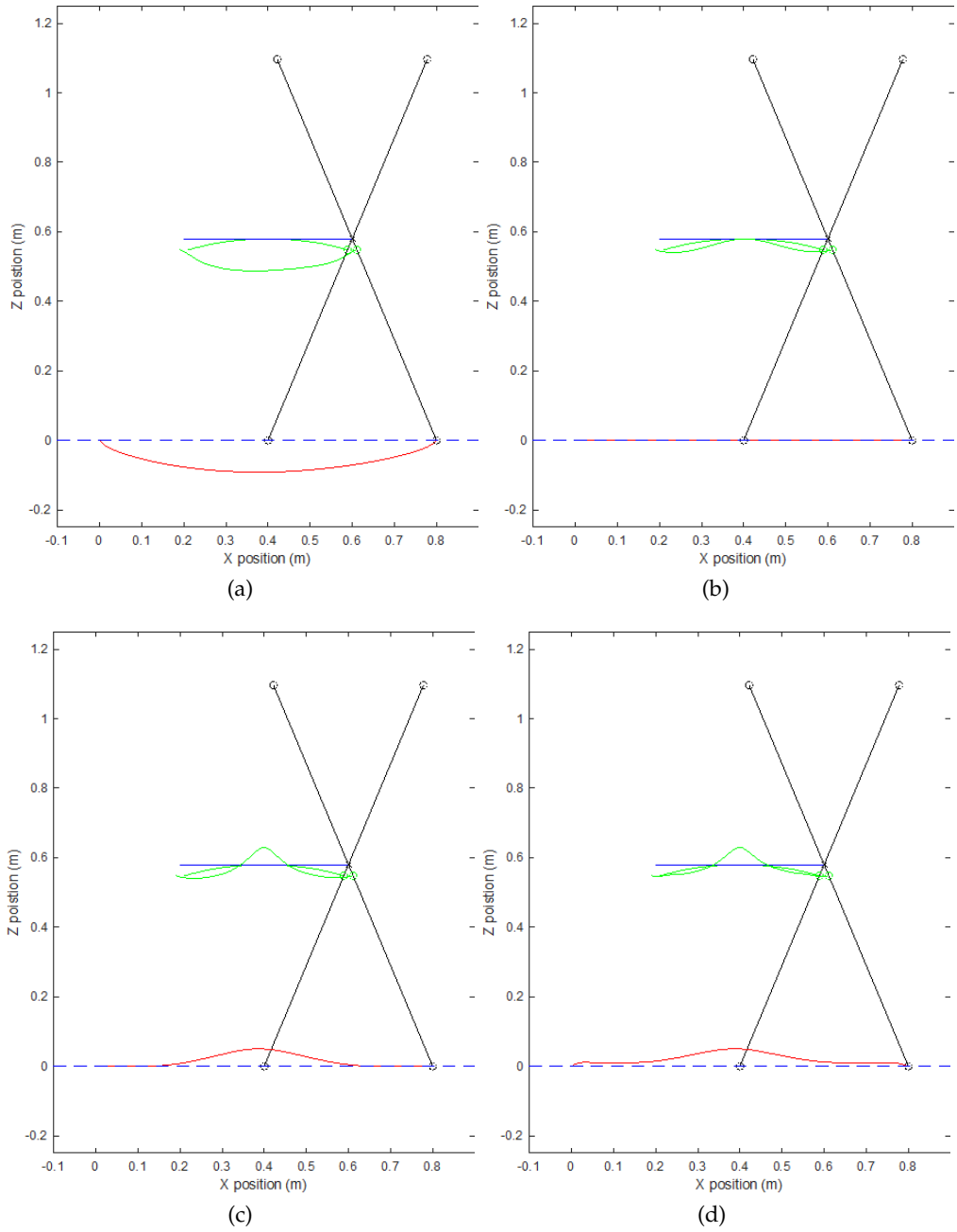


Figure 8.7: Visualization of swing leg trajectory with normal walking gait under different trajectory optimization constraints. (a) same as RotoGait constraint, leg penetrates the floor. (b) add floor constraint to prevent penetrating, leg is close to the surface of floor. (c) add foot height constraint + floor constraint, start and end part is still close to the floor. (d) add foot height and foot clearance constraint, feasible trajectory for the SLIDER robot.

additional constraint has to be added. Firstly, one constraint has to prevent the leg from penetrating the ground and this constraint can be written as:

$$(L[i] + \frac{L_{total}}{2}) \cos\phi[i] \cos\theta[i] > \epsilon, i \in [0 : N-1] \tag{8.16}$$

where ϵ is the foot clearance. This value is set as 10 mm at the first $1/10$ and last $1/10$ of the trajectory and 0 mm at other parts of the trajectory. As the energy is minimized, the foot will be close to the surface if a step height is not set. Therefore the second additional constraint is the step height as shown below:

$$(L[\frac{N}{2}] + \frac{L_{total}}{2}) \cos\phi[\frac{N}{2}] \cos\theta[\frac{N}{2}] == h \tag{8.17}$$

where h is the step height and is set as 5 cm. Fig. 8.7 shows the results and it can be seen that only by adding both constraints can the trajectory optimization generate feasible swing leg trajectory for normal walking gait. The energy consumption of the feasible trajectory is 66.14 J, which is smaller than RotoGait with same step length and step time, this is reasonable because the forward velocity is not fast and has a relatively short step.

8.6.2 *Achievable Gaits*

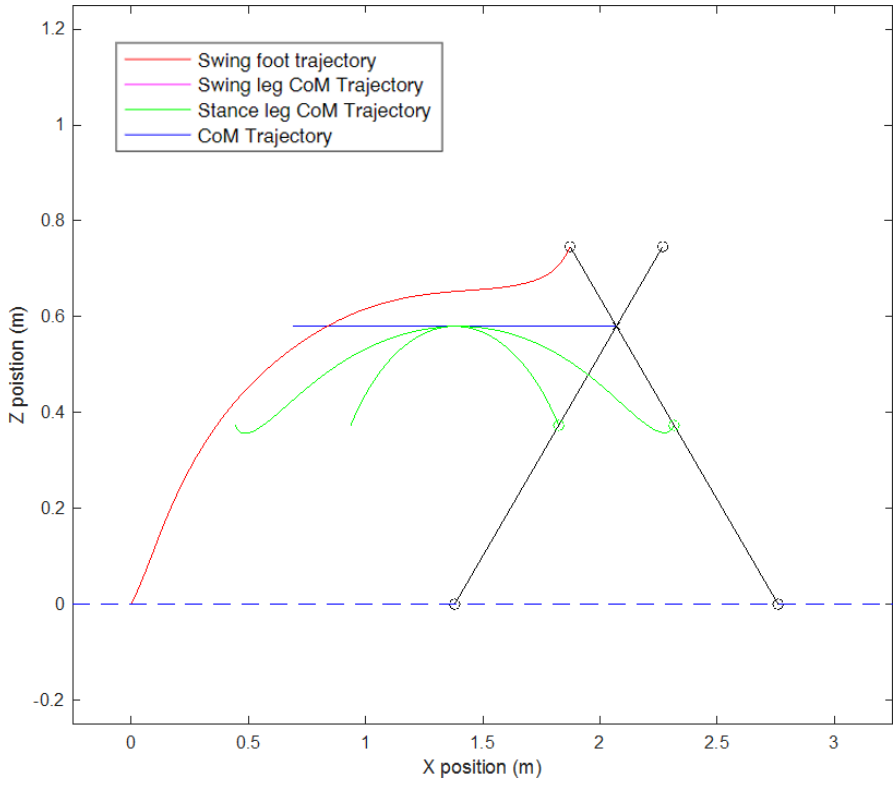


Figure 8.8: RotoGait swing leg trajectory with maximum step length 1.38 m.

Experiments are conducted to find out the limit of step length and forward velocity of RotoGait. The maximum step length is 1.38 m when the optimization respecting the states and physical limits. The trajectory can be seen in Fig. 8.8. If geometric limits are considered regardless of motor limits, with the CoM height at 0.58 m, the maximum is 1.58 m. Then the

motor limits is removed and the maximum step length from the optimizer is set to be 1.55 m which is very close to the theoretical limit. There is no minimum step length as the state and physical constraints are not violated at this case. I also tried to find the maximum forward velocity given a fixed step length and compared with the normal walking gait, as seen in the plot 8.9. It can be seen at smaller step length, RotoGait has smaller maximum step velocity

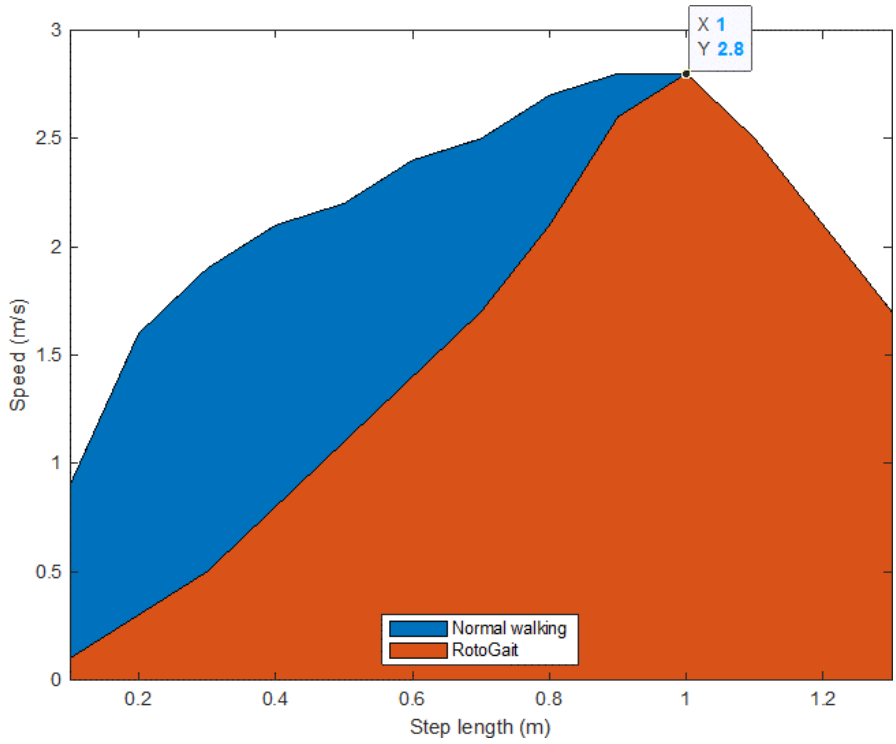
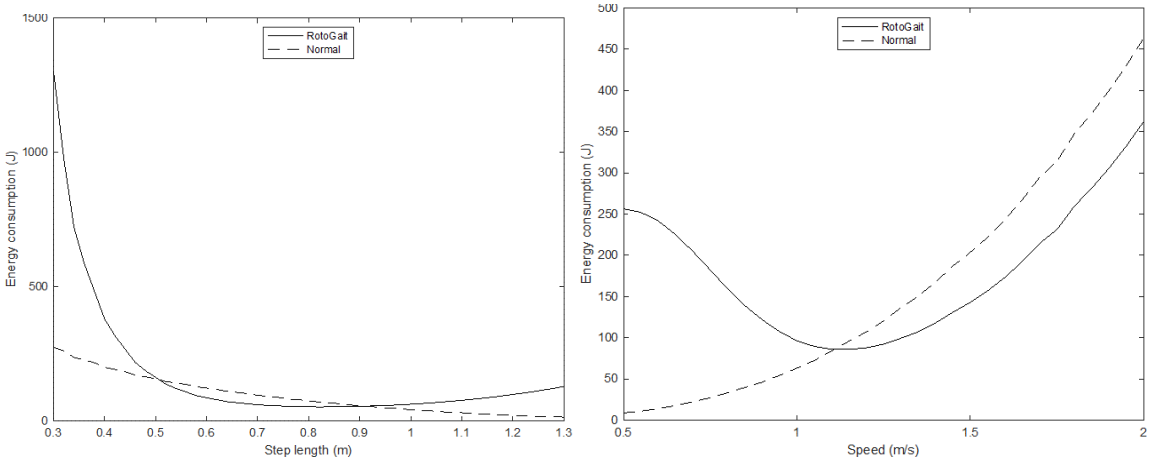


Figure 8.9: Boundary of achievable step velocity given step length for RotoGait and normal walking gait.

than normal walking gait. It is reasonable since RotoGait needs to rotate the legs and the hip pitch travels more than that of the normal walking gait. Given the same step time and motor velocity limit, RotoGait has smaller maximum step velocity. As the step length increases, the pitch rotation required for RotoGait decreases while for normal walking gait increases. The crossover point, calculated from geometry, is a step length of 1.18 m where the pitch rotate angle is 90°. But interestingly, both gaits reaches peak step velocity at step length 1 m with 2.8 m/s, this is because of the motor torque limits. Beyond that the maximum step velocity for both gaits decreases. This is because for larger step length, the slide joint requires large value. Therefore in this region the slide joint limit is the restricting factor.



(a) (b)

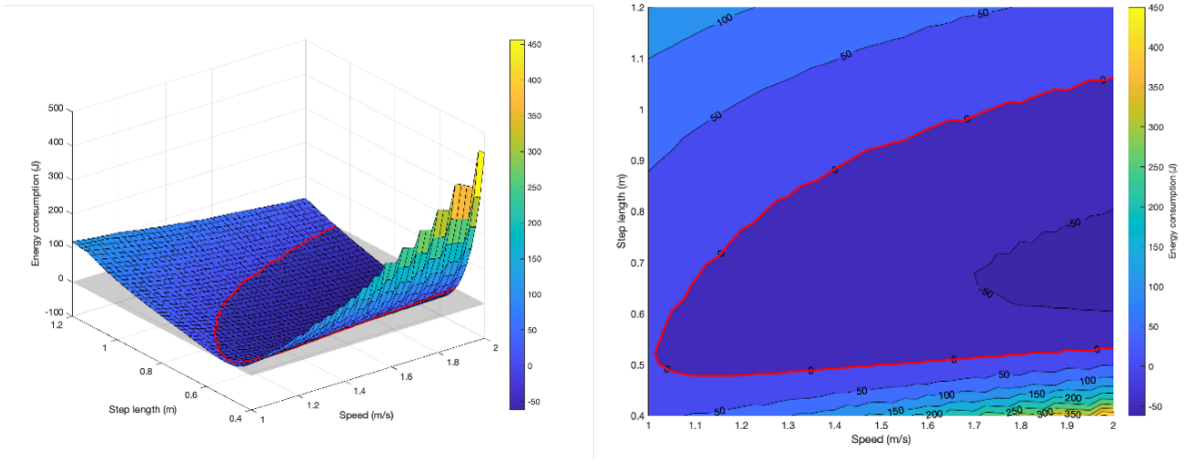
Figure 8.10: Comparison of energy consumption between the RotoGait and normal walking gait. (a) fixed velocity of 1.5 m/s with varying step length. (b) fixed step length 0.6 m with varying velocity.

8.6.3 *Energy Consumption*

The energy consumption of both gaits are compared. Fixed velocity and step length are investigated separately, as seen in Fig. 8.11. The velocity is fixed at 1.5 m/s and the step length is sampled between 0.3-1.2 m. As seen from Fig. 8.10a, at small step lengths, where the pitch rotation for RotoGait is much larger than that of normal walking gait, RotoGait consumes more energy. As the step length increases, the energy consumption gap between the two gaits decreases. The crossing point is at approximate 0.5 m where RotoGait becomes more efficient. It is noticed that 0.5 m is well below the geometric crossover point of 1.18 m. It can be said that the larger pitch rotation required by RotoGait is outweighed by the energy efficiency advantage. Interestingly, at step length larger than 0.9 m, RotoGait becomes less efficient than normal walking gait. For large step lengths, the leg CoM is far away from the CoM of SLIDER and this large moment caused by the gravity increased the torque requirement on the pitch joint. This increased gravitational term has the opposite effect on normal walking gait and it explains the decrease in energy consumption as the step length increases.

Fig. 8.10b compares the energy consumption with fixed step length of 0.6 m while varying step velocity sampled between 0.5-2 m/s. RotoGait consumes more energy at lower speed while becomes more efficient at higher speed. This matches the prediction that RotoGait is more energy efficient at high speed. This is because RotoGait can continue this large initial angular velocity when touching the floor while for normal walking gait the leg has to decelerate and reverse its direction. This reduces the pitch torque required for RotoGait and at higher speed out-weights the effect of energy consumption needed rotate

the leg. For fixed step length of 0.6 m the crossover point is 1.1 m/s. Noticeably, at very low velocity when the leg rotates very slowly, the optimal trajectory commands the leg to balance at a horizontal position for a short moment, this behaviour slightly negates the high energy consumption in this case.



(a) (b)

Figure 8.11: Comparison of energy consumption between the RotoGait and normal walking gait on both step length and step velocity. (a) 3D surface plot of energy difference. (b) Contour plot of the 3D surface, red line marks the crossover where both gaits have the same energy consumption. Negative value means RotoGait is more energy efficient than normal walking gait.

The energy consumption is compared by varying both step length and step velocity. It can be seen from Fig. 8.11a clearly that RotoGait consumes more energy at small and very large step length. In the contour plot in Fig. 8.11b, it is possible to see the effect of increasing step velocity out-weights the inefficiency caused by large step length.

8.7 DISCUSSION

This chapter presents a novel walking gait for SLIDER: RotoGait. Different from the rimless wheel, RotoGait can actively control its CoM height as well as the touching down angle by controlling the slide and pitch joint. It can also walk horizontally by controlling its hip roll joint. By integrating the footstep planners presented in previous chapters, SLIDER with RotoGait has the ability to resist disturbances and transverse uneven terrains. The analytical kinematics and dynamics are derived and the optimal energy-efficient trajectory is generated with the trajectory optimization. In this chapter the energy consumption with normal walking gait is compared and the advantage of RotoGait to save energy in high step

velocity case is highlighted. This research opens a new opportunity to apply this novel gait to SLIDER and to demonstrate its unique features.

Part IV

CONCLUSION AND FUTURE WORK

CHAPTER 9

Conclusion and Future Work

9.1 CONCLUSION

The major advantage of legged robots is their ability to navigate through complex and unstructured environments. This thesis provides a comprehensive study on the design, manufacturing, control and motion planning of the ultra-lightweight, knee-less bipedal robot SLIDER. The contributions of this thesis encompass design, footstep planning, motion planning and a novel gait.

More specifically, this thesis proposed the design of the knee-less bipedal robot SLIDER. Instead of having knees as most bipedal robots do, the knee joint was removed and a prismatic joint (slide joint) was added on the hip for SLIDER. This design could greatly reduce the weight of the leg and did not have the issue of the bent knee walking. Therefore the energy consumption could be reduced. Furthermore, the lightweight leg made the robot dynamics more like the reduced order models such as LIP and SLIP. The detailed mechanical design, lessons learnt from several iterations of design, modelling of the robot were presented.

The second part of the thesis focused on the control part of the SLIDER robot. To control the robot, both inverse kinematics and inverse dynamics based WBC were presented. Experimental results of the real robot and in simulation were presented.

The third part of the thesis described the contributions to the footstep planning algorithms which enabled bipedal robots to walk over uneven terrains blindly and to adapt step location and timing in real-time. The planning problem could be formulated as MPC and solved within milliseconds. These algorithms enhanced the bipedal robot's ability to navigate complex terrains and to resist disturbances. This part also introduced a motion planning framework of highly dynamic jumps. With the proposed LL-SRBM which can approximately model the centroidal inertia of the robot, the jumping performance could be improved in terms of velocity and distance.

The last part of the thesis introduced the novel gait — RotoGait. RotoGait showed promise to increase the walking speed and improved the energy efficiency under the high-speed walking situation. This opened up new possibilities towards applying RotoGait to legged robots and improving the walking performance.

9.2 LIMITATIONS AND FUTURE WORK

The progress presented in this thesis can serve as a starting point for future legged locomotion research. Here I list the limitations of the methods presented in this thesis and possible research directions to improve:

- **Robot Design**: It is clear that lightweight legs bring a lot of advantages to the control of bipedal robot. Not only SLIDER, the bipedal robot Cassie (Robotics, 2017a) also adopts the similar philosophy. In the future I expect to see more bipedal robots have lightweight legs.

  Recently the actuator technology has progressed rapidly and quasi-direct drive motors have gained popularity in quadruped robots. SLIDER v5 uses such quasi-direct drive motors and this greatly simplifies the design and manufacturing process. Although this kind of motors have not been able to produce enough power for heavier bipedal robots (SLIDER v5 only weights 15 kg), there is some attempt to improve the motor ability (Chignoli et al., 2021). Hopefully in the near future quasi-direct drive motors can be widely used to drive bipedal and humanoid robots.

  Another promising future research direction is co-optimizing the design together with the dynamics by trajectory optimization or machine learning. Bipedal and humanoid robots are complex systems with many components and the design is mostly based on heuristics. It is important to find a systematic way to generate an optimized design to meet different requirements (Fadini et al., 2021).

- **SLIDER**: The SLIDER robot has shown great potential for performing agile locomotion and several footstep planning and motion planning algorithms have been developed. In the future it is important to implement the proposed motion planning algorithms and RotoGait on the real robot. It is also important to improve the design of the robot in terms of weight and functionality. For example, yaw joints can be added so that the robot can turn as it walks. Moreover, the upper body with arms can be added for manipulating objects.

- **Footstep Planning**: The proposed footstep planning algorithms have demonstrated their performance on agile locomotion and can be directly applied to quadruped robots. However, the ability of the footstep planners proposed is constrained by the express

power of reduced-order models. Planning motions with whole dynamics model in a MPC fashion is still challenging in terms of computational efficiency. Therefore faster and more efficient numerical solvers are needed (Mastalli et al., 2020a, Howell et al., 2019). Another approach to solve this problem is using machine learning techniques to generate dynamically feasible footsteps or as oracles for nonlinear optimizers.

Furthermore, quantitative analysis of the stability and parameter sensitivity needs to be done for the proposed footstep planner. It is important to know the robustness of the proposed footstep planner under sensory noise and which parameter is more sensitive to the noise. An interesting future direction is combining stochastic MPC with the proposed footstep planner (Gazar et al., 2021).

- **RotoGait**: The trajectory optimizer can generate the swing leg trajectory in 3D for RotoGait. Next step will be integrating the trajectory optimizer with high-level footstep planner (Wang et al., 2021a, Chappell et al., 2020) so that SLIDER can resist pushes and walking over uneven terrains (such as climb stairs, walk/run on slopes) with the RotoGait. Better numerical solvers that can generate swing leg trajectory in real-time could be investigated (Mastalli et al., 2020a, Howell et al., 2019). Moreover, a gait scheduler that can automatically choose between normal gait and RotoGait based on the environment and robot states is another future direction.

Looking ahead, I feel very excited about three research directions on legged locomotion:

- The advancement of the electric actuator technology makes the design and building of bipedal and humanoid robots easier. Furthermore, with the better modelling of the actuator dynamics, either through model-based approach or through machine learning, the potential of electric actuators can be further released. Future research can be building agile and dynamic bipedal or humanoid robots with electric motors and optimizing the design together with dynamics.
- The development of faster numerical optimizers. To perform more agile and dynamic motions, it is important to plan dynamic motions and contact sequences with more complex models. To develop faster optimizers, better understanding of complex dynamics especially the hybrid dynamics, parallelization of the optimization algorithms and machine learning methods can be used.
- Coupling motion planning with perception. Perception can provide useful information that help robots become reactive. With the fast development of deep learning, it is promising to incorporate perception into the motion planning framework so the robot can walk reactively on unstructured environments.

In conclusion, better electric motors, more efficient and faster numerical solvers, machine learning techniques together with the framework proposed in this thesis have the potential

to enable bipedal robots to autonomously move in a complex unstructured environment in the real world.

APPENDIX

A.1 REDUCED ORDER MODELS

A.1.1 *Notation*

f: Force
H: Linear Momentum
L: Augular Momentum
r: CoM position
k: Stiffness
p: Contact location
θ: Angle between the pendulum and the z axis

A.1.2 *Linear Inverted Pendulum*

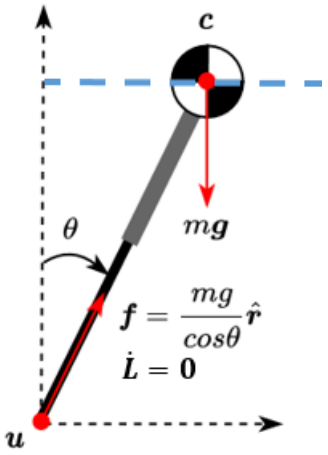


Figure A.1: Linear Inverted Pendulum Model

Linear Inverted Pendulum assumes a point mass and the pendulum height is constant. Moreover, it assumes the change of angular momentum is 0.

A.1.3 *Spring Loaded Inverted Pendulum*

Spring Loaded Inverted Pendulum assumes a point mass and the pendulum length changes based on the physics of a spring. Moreover, it assumes the change of angular momentum is 0.

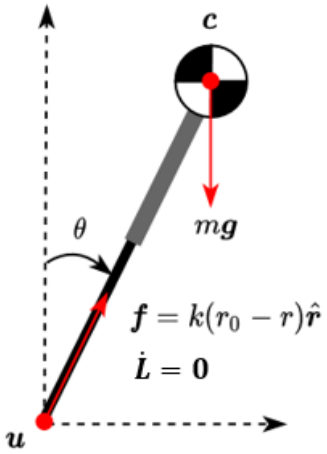


Figure A.2: Spring Loaded Inverted Pendulum

A.1.4 *Variable Height Inverted Pendulum*

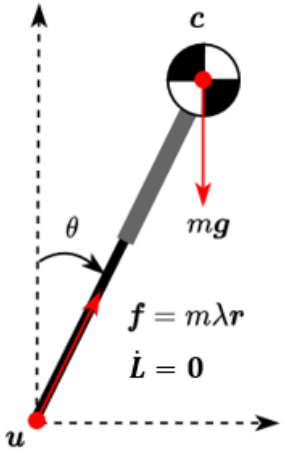


Figure A.3: Variable Height Inverted Pendulum Model

Variable Height Inverted Pendulum assumes a point mass and the pendulum length can freely change. Moreover, it assumes the change of angular momentum is 0.

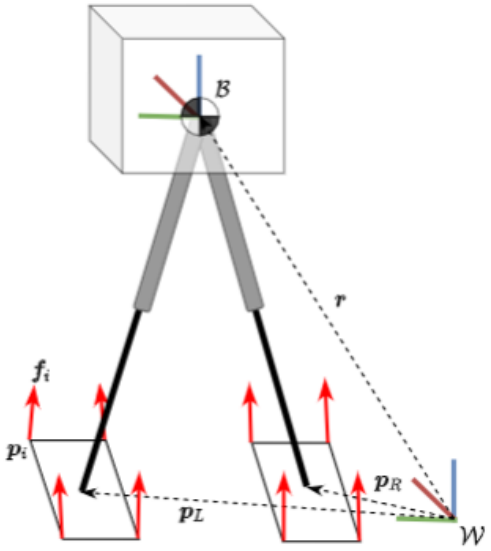


Figure A.4: Single Rigid Body Model

A.1.5 *Single Rigid Body Model*

Single Rigid Body Model simplifies the whole robot as a single rigid body and no extra masses. The dynamics is:

$$\begin{bmatrix} \dot{\mathbf{H}} = \sum_{i=1}^{n} \mathbf{f}_i + m\mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix} \tag{A.1}$$

A.1.6 *Lumped Leg Single Rigid Body Model*

Lumped Leg Single Rigid Body Model simplifies the whole robot as a single rigid body plus point mass for each leg. The dynamics is:

$$\begin{bmatrix} \dot{\mathbf{r}} = \mathbf{H}/m \\ \dot{\mathbf{q}}_B = \frac{1}{2}\mathbf{Q}(\mathbf{q}_B)\mathbf{I}_W^{-1}(\mathbf{q}_B)\mathbf{L} \\ \dot{\mathbf{H}} = \sum_{i=1}^{n} \mathbf{f}_i + m\mathbf{g} \\ \dot{\mathbf{L}} = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix} \tag{A.2}$$

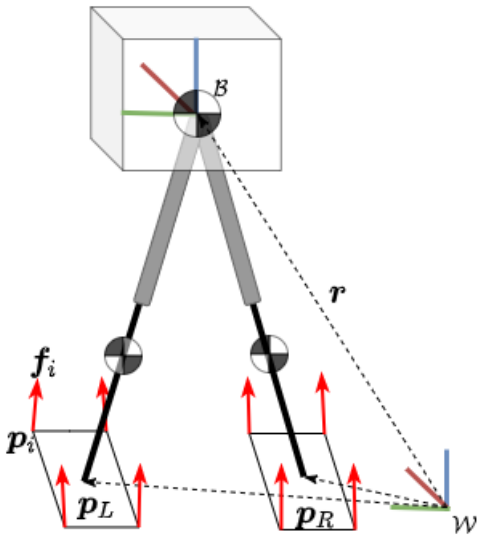


Figure A.5: Lumped Leg Single Rigid Body Model

with

$$\mathbf{r} = \frac{\mathbf{r}_B \cdot m_B + \sum_i \mathbf{r}_l^i \cdot m_l^i}{m_B + \sum_i m_l^i} \tag{7.3}$$

$$\mathbf{I}_W = \mathbf{R}(\mathbf{q}_B)(\mathbf{I}_B + \sum_i m_i(\mathbf{r} - \mathbf{p}_i + \Delta\mathbf{p}_i)^2)\mathbf{R}^T(\mathbf{q}_B) \tag{A.3}$$

where $\mathbf{q}_B$ is the body orientation expressed with quaternion, I_W is the centroidal inertia expressed in world frame and m is the overall mass of the robot.

A.2 ANALYTICAL GRADIENTS

A.2.1 *Footstep position gradients*

$$\frac{\partial x_j}{\partial u_{x,k}} = \begin{cases} 0 & j \leqslant k \\ -\frac{1}{2}(e^{\omega \Delta t_{j-1}} + e^{-\omega \Delta t_{j-1}}) + 1 & j = k+1 \\ \\ \frac{1}{2}\frac{\partial x_{j-1}}{\partial u_{x,k}}(e^{\omega \Delta t_{j-1}} + e^{-\omega \Delta t_{j-1}}) + & j \geqslant k+2 \\ \frac{1}{2\omega}\frac{\partial \dot{x}_{j-1}}{\partial u_{x,k}}(e^{\omega \Delta t_{j-1}} - e^{-\omega \Delta t_{j-1}}) & \end{cases}$$

$$\frac{\partial \dot{x}_j}{\partial u_{x,k}} = \begin{cases} 0 & j \leqslant k \\ -\frac{1}{2}\omega(e^{\omega \Delta t_{j-1}} - e^{-\omega \Delta t_{j-1}}) & j = k+1 \\ \\ \frac{\omega}{2}\frac{\partial x_{j-1}}{\partial u_{x,k}}(e^{\omega \Delta t_{j-1}} - e^{-\omega \Delta t_{j-1}}) + & j \geqslant k+2 \\ \frac{1}{2}\frac{\partial \dot{x}_{j-1}}{\partial u_{x,k}}(e^{\omega \Delta t_{j-1}} + e^{-\omega \Delta t_{j-1}}) & \end{cases}$$

With identical formulation in the y direction.

A.2.2 *Footstep duration gradients*

$$\frac{\partial x_j}{\partial \Delta t_k} = \begin{cases} 0 & j < k \\ \dot{x}_j & j = k \\ \frac{1}{2}\frac{\partial x_{j-1}}{\partial \Delta t_k}(e^{\omega \Delta t_j} + e^{-\omega \Delta t_j}) + & j \geqslant k+1 \\ \frac{1}{2\omega}\frac{\partial \dot{x}_{j-1}}{\partial \Delta t_k}(e^{\omega \Delta t_j} - e^{-\omega \Delta t_j}) & \end{cases}$$

$$\frac{\partial \dot{x}_j}{\partial \Delta t_k} = \begin{cases} 0 & j < k \\ \ddot{x}_j & j = k \\ \frac{\omega}{2}\frac{\partial x_{j-1}}{\partial \Delta t_k}(e^{\omega \Delta t_j} - e^{-\omega \Delta t_j}) + & j \geqslant k+1 \\ \frac{1}{2}\frac{\partial \dot{x}_{j-1}}{\partial \Delta t_k}(e^{\omega \Delta t_j} + e^{-\omega \Delta t_j}) & \end{cases}$$

Again, with identical formulation in the y direction.

A.3 QUATERNION TO ROTATION MATRIX CONVERSION

Given a quaternion $\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$ which represents the orientation of the single rigid body in world frame, the equivalent rotation matrix representation can be derived as:

$$\mathbf{R} = \begin{bmatrix} 1-2\left(q_y^2+q_z^2\right) & 2q_xq_y-2q_wq_z & 2q_wq_y+2q_xq_z \\ 2q_xq_y+2q_wq_z & 1-2\left(q_x^2+q_z^2\right) & 2q_yq_z-2q_wq_x \\ 2q_xq_z-2q_wq_y & 2q_wq_x+2q_yq_z & 1-2\left(q_x^2+q_y^2\right) \end{bmatrix}$$

A.4 DERIVATIVE OF QUATERNION

Followed by (Graf, 2008), given a rigid body with quaternion $\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$ and with angular velocity $\boldsymbol{\omega}$, the derivative of the quaternion $\dot{\mathbf{q}}$ can be calculated as:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \boldsymbol{\omega}$$

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \\ \dot{q}_w \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix} \circ \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \\ \dot{q}_w \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_w & -q_z & q_y & q_x \\ q_z & q_w & -q_x & q_y \\ -q_y & q_x & q_w & q_z \\ -q_x & -q_y & -q_z & q_w \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \\ 0 \end{bmatrix}$$

or

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \\ \dot{q}_w \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{bmatrix}$$

A.5 INERTIA TENSOR

Inertia tensor in the local body frame:

$$\mathbf{I}_B = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

$$\mathbf{I}_B^{-1} = \begin{bmatrix} 1/I_{xx} & 0 & 0 \\ 0 & 1/I_{yy} & 0 \\ 0 & 0 & 1/I_{zz} \end{bmatrix}$$

Inertia tensor in the world inertia frame:

$$\mathbf{I}_W = \mathbf{R}_B \mathbf{I}_B \mathbf{R}_B^T$$

$$\mathbf{I}_W^{-1} = \mathbf{R}_B \mathbf{I}_B^{-1} \mathbf{R}_B^T$$

where $\mathbf{R}_B$ is the body rotation matrix expressed in the world inertia frame.

BIBLIOGRAPHY

Abbeel, P. and A. Y. Ng (2004). 'Apprenticeship learning via inverse reinforcement learning'. In: *Proceedings of the twenty-first international conference on Machine learning*, p. 1 (cit. on p. 8).

Aceituno-Cabezas, B., C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López and C. Semini (2017). 'Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization'. In: *IEEE Robotics and Automation Letters* 3.3, pp. 2531–2538 (cit. on p. 6).

Agnihotri, A. and N. Batra (2020). 'Exploring Bayesian Optimization'. In: *Distill* (cit. on p. 50).

Ahmadi, M. and M. Buehler (2006). 'Controlled passive dynamic running experiments with the ARL-monopod II'. In: *IEEE Transactions on Robotics* 22.5, pp. 974–986 (cit. on p. 71).

Alexander, R. (1990). 'Three uses for springs in legged locomotion'. In: *International Journal of Robotics Research* 9.2, pp. 53–61 (cit. on p. 45).

Andersson, J. A., J. Gillis, G. Horn, J. B. Rawlings and M. Diehl (2019). 'CasADi: a software framework for nonlinear optimization and optimal control'. In: *Mathematical Programming Computation* 11.1, pp. 1–36 (cit. on p. 124).

Andersson, J., J. Åkesson and M. Diehl (2012). 'CasADi: A Symbolic Package for Automatic Differentiation andOptimal Control'. In: *Recent Advances in Algorithmic Differentiation*. Ed. by S. Forth, P. Hovland, E. Phipps, J. Utke and A. Walther. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 297–307 (cit. on p. 96).

Apgar, T., P. Clary, K. Green, A. Fern and J. W. Hurst (2018). 'Fast Online Trajectory Optimization for the Bipedal Robot Cassie.' In: *Robotics: Science and Systems*. Vol. 101, p. 14 (cit. on pp. 62, 71, 122).

Arakawa, T. and T. Fukuda (1996). 'Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization'. In: *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929)*. Vol. 2. IEEE, pp. 1495–1500 (cit. on p. 46).

Arslan, Ö. and U. Saranli (2011). 'Reactive planning and control of planar spring–mass running on rough terrain'. In: *IEEE Transactions on Robotics* 28.3, pp. 567–579 (cit. on p. 6).

Barragan, M., N. Flowers and A. M. Johnson (2018). 'MiniRHex: A Small, Open-source, Fully Programmable Walking Hexapod'. In: *Robotics: Science and Systems Workshop on "Design and Control of Small Legged Robots"*. Pittsburgh, PA (cit. on p. 130).

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press (cit. on p. 93).

Buschmann, T., S. Lohmeier and H. Ulbrich (2009). 'Humanoid robot lola: Design and walking control'. In: *Journal of physiology-Paris* 103.3-5, pp. 141–148 (cit. on p. 28).

Calandra, R., N. Gopalan, A. Seyfarth, J. Peters and M. P. Deisenroth (2014a). 'Bayesian gait optimization for bipedal locomotion'. In: *International Conference on Learning and Intelligent Optimization*. Springer, pp. 274–290 (cit. on p. 46).

Calandra, R., A. Seyfarth, J. Peters and M. P. Deisenroth (2014b). 'An experimental comparison of Bayesian optimization for bipedal locomotion'. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1951–1958 (cit. on p. 52).

Caron, S. and A. Kheddar (2017). 'Dynamic walking over rough terrains by nonlinear predictive control of the floating-base inverted pendulum'. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5017–5024. DOI: 10.1109/IROS.2017.8206385 (cit. on p. 87).

Carpentier, J., F. Valenza, N. Mansard et al. (2015–2021). *Pinocchio: fast forward and inverse dynamics for poly-articulated systems*. https://stack-of-tasks.github.io/pinocchio (cit. on pp. 63, 83, 124).

Chappell, D., K. Wang and P. Kormushev (2020). 'Asynchronous Real-Time Optimization of Footstep Placement and Timing in Bipedal Walking Robots'. In: *CoRR* abs/2007.00385. arXiv: 2007.00385. URL: https://arxiv.org/abs/2007.00385 (cit. on pp. 12, 14, 146).

Chignoli, M., D. Kim, E. Stanger-Jones and S. Kim (2021). *The MIT Humanoid Robot: Design, Motion Planning, and Control For Acrobatic Behaviors*. arXiv: 2104.09025 [cs.RO] (cit. on pp. 112, 120, 123, 145).

Corbères, T., T. Flayols, P.-A. Léziart, R. Budhiraja and N. Mansard (May 2021). 'Comparison of predictive controllers for locomotion and balance recovery of quadruped robots'. In: *2021 IEEE International Conference on Robotics and Automation - ICRA*. Xi'an, China. URL: https://hal.laas.fr/hal-03034022 (cit. on p. 112).

Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*. 2nd. USA: Addison-Wesley Longman Publishing Co., Inc. ISBN: 0201095289 (cit. on p. 36).

Da, X. and J. Grizzle (2019). 'Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots'. In: *The International Journal of Robotics Research* 38.9, pp. 1063–1097. DOI: 10.1177/0278364919859425. eprint: https://doi.org/10.1177/0278364919859425. URL: https://doi.org/10.1177/0278364919859425 (cit. on p. 90).

Dai, H., A. Valenzuela and R. Tedrake (2014). 'Whole-body motion planning with centroidal dynamics and full kinematics'. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 295–302 (cit. on pp. 2, 7).

Daily, S. (2022). *Mountain*. URL: https://www.sciencedaily.com/terms/mountain.htm (visited on 07/01/2022) (cit. on p. 2).

Dallali, H., P. Kormushev, N. Tsagarakis and D. G. Caldwell (2014). 'Can Active Impedance Protect Robots from Landing Impact?' In: *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids 2014)*. Madrid, Spain (cit. on p. 46).

Di Carlo, J., P. M. Wensing, B. Katz, G. Bledt and S. Kim (2018). 'Dynamic locomotion in the mit cheetah 3 through convex model-predictive control'. In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 1–9 (cit. on pp. 6, 112, 116).

Diehl, M., H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy and F. Allgöwer (2002). 'Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations'. In: *Journal of Process Control* 12.4, pp. 577 –585. ISSN: 0959-1524. DOI: https://doi.org/10.1016/S0959-1524(01)00023-3. URL: http://www.sciencedirect.com/science/article/pii/S0959152401000233 (cit. on p. 89).

Diehl, M., H. G. Bock and J. P. Schlöder (2005). 'A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control'. In: *SIAM Journal on Control and Optimization* 43.5, pp. 1714–1736. DOI: 10.1137/S0363012902400713. eprint: https://doi.org/10.1137/S0363012902400713. URL: https://doi.org/10.1137/S0363012902400713 (cit. on p. 89).

Ding, J., X. Xiao and N. Tsagarakis (2019). 'Nonlinear optimization of Step Duration and Step Location'. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2259–2265 (cit. on p. 89).

Ding, Y., A. Pandala and H.-W. Park (2019). 'Real-time model predictive control for versatile dynamic motions in quadrupedal robots'. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 8484–8490 (cit. on p. 6).

Drumwright, E., J. Hsu, N. Koenig and D. Shell (2010). 'Extending open dynamics engine for robotics simulation'. In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, pp. 38–50 (cit. on p. 83).

Dynamics, B. (2021). *Atlas Robot*. URL: https://www.geeky-gadgets.com/boston-dynamics-atlas-robot-demonstrates-its-parkour-skills-18-08-2021/ (visited on 19/02/2022) (cit. on p. 5).

Dynamics, B. (2022a). *Atlas, the Next Generation*. URL: https://www.youtube.com/watch?v=rVlhMGQgDkY&ab_channel=BostonDynamics (visited on 07/01/2022) (cit. on p. 2).

Dynamics, B. (2022b). *What's new, Atlas?* URL: https://www.youtube.com/watch?v=fRj34o4hN4I&ab_channel=noobqoou (visited on 07/01/2022) (cit. on p. 2).

Englsberger, J., G. Mesesan and C. Ott (2017). 'Smooth trajectory generation and push-recovery based on Divergent Component of Motion'. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4560–4567 (cit. on p. 89).

Englsberger, J., C. Ott and A. Albu-Schäffer (2013). 'Three-dimensional bipedal walking control using Divergent Component of Motion'. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2600–2607 (cit. on p. 87).

Englsberger, J., C. Ott and A. Albu-Schäffer (2015). 'Three-dimensional bipedal walking control based on divergent component of motion'. In: *Ieee transactions on robotics* 31.2, pp. 355–368 (cit. on pp. 70, 89).

Ernst, M., H. Geyer and R. Blickhan (2010). 'Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant'. In: *Mobile Robotics: Solutions and Challenges*. World Scientific, pp. 639–644 (cit. on p. 71).

Fadini, G., T. Flayols, A. Del Prete, N. Mansard and P. Souères (2021). 'Computational design of energy-efficient legged robots: Optimizing for size and actuators'. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9898–9904. DOI: 10.1109/ICRA48506.2021.9560988 (cit. on p. 145).

Falck, F., S. Doshi, M. Tormento, G. Nersisyan, N. Smuts, J. Lingi, K. Rants, R. P. Saputra, K. Wang and P. Kormushev (2020). 'Robot DE NIRO: A Human-Centered, Autonomous, Mobile Research Platform for Cognitively-Enhanced Manipulation'. In: *Frontiers in Robotics and AI* 7, p. 66. ISSN: 2296-9144. DOI: 10.3389/frobt.2020.00066. URL: https://www.frontiersin.org/article/10.3389/frobt.2020.00066 (cit. on p. 14).

Fankhauser, P., M. Bloesch, C. Gehring, M. Hutter and R. Siegwart (2014). 'Robot-Centric Elevation Mapping with Uncertainty Estimates'. In: *International Conference on Climbing and Walking Robots (CLAWAR)* (cit. on p. 100).

Fankhauser, P., M. Bloesch and M. Hutter (2018). 'Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization'. In: *IEEE Robotics and Automation Letters (RA-L)* 3.4, pp. 3019–3026. DOI: 10.1109/LRA.2018.2849506 (cit. on p. 100).

Feng, S., E. Whitman, X Xinjilefu and C. G. Atkeson (2014). 'Optimization based full body control for the atlas robot'. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 120–127 (cit. on pp. 62, 102, 122).

Feng, S., E. Whitman, X Xinjilefu and C. G. Atkeson (2015). 'Optimization-based full body control for the darpa robotics challenge'. In: *Journal of Field Robotics* 32.2, pp. 293–312 (cit. on pp. 4, 73, 74).

Ferreau, H. J., C. Kirches, A. Potschka, H. G. Bock and M. Diehl (2014). 'qpOASES: A parametric active-set algorithm for quadratic programming'. In: *Mathematical Programming Computation* 6.4, pp. 327–363 (cit. on pp. 63, 83, 124).

Folkertsma, G. A., S. Kim and S. Stramigioli (2012). 'Parallel stiffness in a bounding quadruped with flexible spine'. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 2210–2215 (cit. on p. 46).

Foundation, O. S. R. (2021). *Gazebo simulator*. URL: http://gazebosim.org/ (visited on 16/11/2021) (cit. on p. 32).

Fukuda, T., Y. Komata and T. Arakawa (1997). 'Stabilization control of biped locomotion robot based learning with GAs having self-adaptive mutation and recurrent neural networks'. In: *Proceedings of International Conference on Robotics and Automation*. Vol. 1, 217–222 vol.1. DOI: 10.1109/ROBOT.1997.620041 (cit. on p. 8).

Gazar, A., M. Khadiv, A. D. Prete and L. Righetti (2021). 'Stochastic and Robust MPC for Bipedal Locomotion: A Comparative Study on Robustness and Performance'. In: *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 61–68. DOI: 10.1109/HUMANOIDS47582.2021.9555783 (cit. on p. 146).

Geographic, N. (2016). *Amazing Footage: Goats Climbing on a Near-Vertical Dam*. URL: https://www.youtube.com/watch?v=f0jMzdbEui0&ab_channel=NationalGeographic (visited on 19/02/2022) (cit. on p. 3).

Geyer, H., A. Seyfarth and R. Blickhan (2006). 'Compliant leg behaviour explains basic dynamics of walking and running'. In: *Proceedings of the Royal Society B: Biological Sciences* 273.1603, pp. 2861–2867 (cit. on pp. 6, 71).

Gibson, G., O. Dosunmu-Ogunbi, Y. Gong and J. Grizzle (2021). *Terrain-Aware Foot Placement for Bipedal Locomotion Combining Model Predictive Control, Virtual Constraints, and the ALIP*. arXiv: 2109.14862 [cs.RO] (cit. on p. 109).

Graf, B. (2008). 'Quaternions and dynamics'. In: *arXiv preprint arXiv:0811.2889* (cit. on p. 156).

Green, K., R. L. Hatton and J. Hurst (2020). 'Planning for the unexpected: Explicitly optimizing motions for ground uncertainty in running'. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1445–1451 (cit. on p. 71).

Griffin, R. J., G. Wiedebach, S. Bertrand, A. Leonessa and J. Pratt (2018). 'Straight-Leg Walking Through Underconstrained Whole-Body Control'. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5747–5754. DOI: 10.1109/ICRA.2018.8460751 (cit. on p. 19).

Gros, S., M. Zanon, R. Quirynen, A. Bemporad and M. Diehl (2020). 'From linear to nonlinear MPC: bridging the gap via the real-time iteration'. In: *International Journal of Control* 93.1, pp. 62–80. DOI: 10.1080/00207179.2016.1222553. eprint: https://doi.org/10.1080/00207179.2016.1222553. URL: https://doi.org/10.1080/00207179.2016.1222553 (cit. on p. 89).

Haddadin, S., A. De Luca and A. Albu-Schäffer (2017). 'Robot collisions: A survey on detection, isolation, and identification'. In: *IEEE Transactions on Robotics* 33.6, pp. 1292–1312 (cit. on p. 113).

Halm, M. and M. Posa (2019). 'Modeling and analysis of non-unique behaviors in multiple frictional impacts'. In: *arXiv preprint arXiv:1902.01462* (cit. on p. 113).

Heess, N., D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami et al. (2017). 'Emergence of locomotion behaviours in rich environments'. In: *arXiv preprint arXiv:1707.02286* (cit. on p. 8).

Henze, B., M. A. Roa, A. Werner, A. Dietrich, C. Ott and A. Albu-Schäffer (2019). 'Experiments with Human-inspired Behaviors in a Humanoid Robot: Quasi-static Balancing using Toe-off Motion and Stretched Knees'. In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2510–2516. DOI: 10.1109/ICRA.2019.8794096 (cit. on p. 19).

Herzog, A., N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti (2015). 'Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid'. In: *Autonomous Robots* 40.3, 473–491. ISSN: 1573-7527. DOI: 10.1007/s10514-015-9476-6. URL: http://dx.doi.org/10.1007/s10514-015-9476-6 (cit. on pp. 4, 62, 71, 73, 74, 81, 102, 122).

Hobbelen, D., T. de Boer and M. Wisse (2008). 'System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking'. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 2486–2491 (cit. on p. 45).

Howell, T. A., B. E. Jackson and Z. Manchester (2019). 'Altro: A fast solver for constrained trajectory optimization'. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 7674–7679 (cit. on pp. 90, 146).

Hubicki, C., J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate and J. Hurst (2016). 'Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot'. In: *The International Journal of Robotics Research* 35.12, pp. 1497–1521 (cit. on p. 6).

Hwangbo, J., J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun and M. Hutter (2019). 'Learning agile and dynamic motor skills for legged robots'. In: *Science Robotics* 4.26, eaau5872. DOI: 10.1126/scirobotics.aau5872. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aau5872 (cit. on pp. 2, 8).

Interstellar (2015). *Interstellar - Waves Scene 1080p HD*. URL: https://www.youtube.com/watch?v=4Hf_XkgE1d0&ab_channel=JayM (visited on 29/12/2021) (cit. on p. 130).

Jenelten, F., T. Miki, A. Elanjimattathil Vijayan, M. Bjelonic and M. Hutter (2020-10). 'Perceptive Locomotion in Rough Terrain – Online Foothold Optimization'. en. In: *IEEE Robotics and Automation Letters* 5.4, pp. 5370 –5376. ISSN: 2377-3766. DOI: 10.3929/ethz-b-000425596 (cit. on p. 90).

KATAYAMA, T., T. OHKI, T. INOUE and T. KATO (1985). 'Design of an optimal controller for a discrete-time system subject to previewable demand'. In: *International Journal of Control* 41.3, pp. 677–699. DOI: 10.1080/00207185089611 56. eprint: https://doi.org/10.1080/00207185089611 56. URL: https://doi.org/10.1080/00207185089611 56 (cit. on p. 58).

Kajita, S., F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa (2001). 'The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation'. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. Vol. 1, 239–246 vol.1 (cit. on pp. 89, 99).

Kajita, S., M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi (2010). 'Biped walking stabilization based on linear inverted pendulum tracking'. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496 (cit. on p. 88).

Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa (2003). 'Biped walking pattern generation by using preview control of zero-moment point'. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 2. IEEE, pp. 1620–1626 (cit. on pp. 46, 57).

Kajita, S., F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa (2001a). 'The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation'. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE, pp. 239–246 (cit. on pp. 6, 65, 92).

Kajita, S., F. Kanehiro, K. Kaneko, K. Yokoi and H. Hirukawa (2001b). 'The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation'. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE, pp. 239–246 (cit. on p. 97).

Kamioka, T., H. Kaneko, M. Kuroda, C. Tanaka, S. Shirokura, M. Takeda and T. Yoshiike (2017). 'Dynamic gait transition between walking, running and hopping for push recovery'. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 1–8. DOI: 10.1109/HUMANOIDS.2017.8239530 (cit. on p. 6).

Katz, B. G. (2018). 'A low cost modular actuator for dynamic robots'. PhD thesis. Massachusetts Institute of Technology (cit. on pp. 2, 24).

Katz, B., J. Di Carlo and S. Kim (2019). 'Mini cheetah: A platform for pushing the limits of dynamic quadruped control'. In: *2019 international conference on robotics and automation (ICRA)*. IEEE, pp. 6295–6301 (cit. on pp. 2, 112).

Kelly, M. (2017). 'An introduction to trajectory optimization: How to do your own direct collocation'. In: *SIAM Review* 59.4, pp. 849–904 (cit. on pp. 7, 117).

Khadiv, M., A. Herzog, S. A. A. Moosavian and L. Righetti (2016). 'Step timing adjustment: A step toward generating robust gaits'. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 35–42 (cit. on pp. 89, 106).

Khadiv, M., A. Herzog, S. A. A. Moosavian and L. Righetti (2020). 'Walking Control Based on Step Timing Adaptation'. In: *IEEE Transactions on Robotics*, pp. 1–15 (cit. on pp. 89, 108, 109, 136).

Kim, D., J. Di Carlo, B. Katz, G. Bledt and S. Kim (2019). 'Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control'. In: *arXiv preprint arXiv:1909.06586* (cit. on pp. 71, 112).

Kim, D., S. J. Jorgensen, J. Lee, J. Ahn, J. Luo and L. Sentis (2020). 'Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control'. In: *The International Journal of Robotics Research* 39.8, pp. 936–956 (cit. on p. 6).

Kim, D., J. Lee, O. Campbell, H. Hwang and L. Sentis (2018). *Computationally-Robust and Efficient Prioritized Whole-Body Controller with Contact Constraints*. arXiv: 1807.01222 [cs.RO] (cit. on p. 4).

Kockelman, J. and M Hubbard (2004). 'Bungee jumping cord design using a simple model'. In: *Sports Engineering* 7.2, pp. 89–96 (cit. on p. 48).

Koenig, N. and A. Howard (2004). 'Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator'. In: *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2149–2154 (cit. on p. 83).

Kormushev, P., B. Ugurlu, D. G. Caldwell and N. G. Tsagarakis (2019). 'Learning to exploit passive compliance for energy-efficient gait generation on a compliant humanoid'. In: *Autonomous Robots* 43.1, pp. 79–95 (cit. on p. 45).

Kormushev, P., B. Ugurlu, S. Calinon, N. G. Tsagarakis and D. G. Caldwell (2011). 'Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization'. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 318–324 (cit. on p. 46).

Kryczka, P., P. Kormushev, N. G. Tsagarakis and D. G. Caldwell (2015). 'Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing'. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3352–3357 (cit. on p. 89).

Kuindersma, S., R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion and R. Tedrake (2016). 'Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot'. In: *Autonomous robots* 40.3, pp. 429–455 (cit. on pp. 4, 62, 122).

LLC, tmotor (2021). *t-motor official website*. URL: https://store.tmotor.com/ (visited on 15/11/2021) (cit. on p. 24).

LOLWOT (2021). *Dog Catch Frisbee*. URL: https://www.lolwot.com/10-hilarious-pictures-of-dogs-catching-frisbees/ (visited on 19/02/2022) (cit. on p. 3).

Lab, M. L. (1995). *3D Biped*. URL: http://www.ai.mit.edu/projects/leglab/robots/robots.html (visited on 19/02/2022) (cit. on p. 5).

Lee, J., J. Hwangbo, L. Wellhausen, V. Koltun and M. Hutter (2020). 'Learning quadrupedal locomotion over challenging terrain'. In: *Science Robotics* 5.47, eabc5986. DOI: 10.1126/scirobotics.abc5986. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abc5986 (cit. on p. 2).

Lee, S.-H. and A. Goswami (2007). 'Reaction Mass Pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots'. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4667–4672. DOI: 10.1109/ROBOT.2007.364198 (cit. on pp. 113, 116).

Levine, S., C. Finn, T. Darrell and P. Abbeel (2016). 'End-to-end training of deep visuomotor policies'. In: *The Journal of Machine Learning Research* 17.1, pp. 1334–1373 (cit. on p. 8).

Levine, S., P. Pastor, A. Krizhevsky, J. Ibarz and D. Quillen (2018). 'Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection'. In: *The International Journal of Robotics Research* 37.4-5, pp. 421–436 (cit. on p. 8).

Li, W. and E. Todorov (n.d.). 'Iterative linear quadratic regulator design for nonlinear biological movement systems.' In: Citeseer (cit. on p. 7).

Lin, W.-S., Y.-P. Shih and J.-J. Lee (2014). 'Design of a two-stage cycloidal gear reducer with tooth modifications'. In: *Mechanism and Machine Theory* 79, pp. 184–197 (cit. on p. 27).

Liu, Y., P. M. Wensing, D. E. Orin and Y. F. Zheng (2015). 'Trajectory generation for dynamic walking in a humanoid over uneven terrain using a 3D-actuated Dual-SLIP model'. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 374–380. DOI: 10.1109/IROS.2015.7353400 (cit. on p. 70).

Liu, Y., P. M. Wensing, J. P. Schmiedeler and D. E. Orin (2016). 'Terrain-blind humanoid walking based on a 3-D actuated dual-SLIP model'. In: *IEEE Robotics and Automation Letters* 1.2, pp. 1073–1080 (cit. on p. 71).

Lu, C., B. Huang, K. Wang, J. M. Hernández-Lobato, K. Zhang and B. Schölkopf (2020). 'Sample-efficient reinforcement learning via counterfactual-based data augmentation'. In: *arXiv preprint arXiv:2012.09092* (cit. on p. 14).

Lynch, K. M. and F. C. Park (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press (cit. on p. 46).

Manchester, Z., N. Doshi, R. J. Wood and S. Kuindersma (2019). 'Contact-implicit trajectory optimization using variational integrators'. In: *The International Journal of Robotics Research* 38.12-13, pp. 1463–1476 (cit. on p. 7).

Mastalli, C., R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar and N. Mansard (2020a). 'Crocoddyl: An efficient and versatile framework for multi-contact optimal control'. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2536–2542 (cit. on pp. 2, 7, 89, 146).

Mastalli, C., W. Merkt, J. Marti-Saumell, H. Ferrolho, J. Sola, N. Mansard and S. Vijayakumar (2020b). 'A Direct-Indirect Hybridization Approach to Control-Limited DDP'. In: *arXiv preprint arXiv:2010.00411* (cit. on p. 7).

Mazumdar, A., S. J. Spencer, C. Hobart, J. Salton, M. Quigley, T. Wu, S. Bertrand, J. Pratt and S. P. Buerger (2017). 'Parallel elastic elements improve energy efficiency on the STEPPR bipedal walking robot'. In: *IEEE/ASME Transactions on Mechatronics* 22.2, pp. 898–908 (cit. on p. 45).

Mordatch, I., M. De Lasa and A. Hertzmann (2010). 'Robust physics-based locomotion using low-dimensional planning'. In: *ACM SIGGRAPH 2010 papers*, pp. 1–8 (cit. on pp. 70, 112).

Motoi, N., T. Suzuki and K. Ohnishi (2009). 'A Bipedal Locomotion Planning Based on Virtual Linear Inverted Pendulum Mode'. In: *IEEE Transactions on Industrial Electronics* 56.1, pp. 54–61 (cit. on p. 88).

Naranjo-Lozada, J., H. Ahuett-Garza, P. Orta-Castañón, W. M. Verbeeten and D. Sáiz-González (2019). 'Tensile properties and failure behavior of chopped and continuous carbon fiber composites produced by additive manufacturing'. In: *Additive Manufacturing* 26, pp. 227–241 (cit. on p. 26).

Naveau, M., M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Souères (2016). 'A reactive walking pattern generator based on nonlinear model predictive control'. In: *IEEE Robotics and Automation Letters* 2.1, pp. 10–17 (cit. on p. 7).

Neunert, M., M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter and J. Buchli (2018). 'Whole-body nonlinear model predictive control through contacts for quadrupeds'. In: *IEEE Robotics and Automation Letters* 3.3, pp. 1458–1465 (cit. on p. 7).

Neunert, M., M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter and J. Buchli (2017). 'Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds'. In: *CoRR* abs/1712.02889. arXiv: 1712.02889. URL: http://arxiv.org/abs/1712.02889 (cit. on p. 89).

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization*. second. New York, NY, USA: Springer (cit. on p. 93).

Odrive (1st Mar. 2019). *Odrive Robotics*. URL: https://odriverobotics.com/ (visited on 01/03/2019) (cit. on pp. 34, 50).

Orin, D. E., A. Goswami and S.-H. Lee (2013). 'Centroidal dynamics of a humanoid robot'. In: *Autonomous robots* 35.2, pp. 161–176 (cit. on pp. 61, 121, 123).

OutRunner (2021). *IHMC robot sets world speed record*. URL: https://www.ihmc.us/ihmc-robot-sets-world-speed-record/ (visited on 29/12/2021) (cit. on p. 130).

POWELL, M. J. D. (1969). 'A method for nonlinear constraints in minimization problems'. In: *Optimization*, pp. 283–298. URL: https://ci.nii.ac.jp/naid/20000922074/en/ (cit. on p. 89).

Patla, A. and J. Vickers (Feb. 2003). 'How far ahead do we look when required to step on specific locations in the travel path during locomotion?' In: *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale* 148, pp. 133–8. DOI: 10.1007/s00221-002-1246-y (cit. on p. 108).

Playter, R. and M. Raibert (1992). 'Control Of A Biped Somersault In 3D'. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 1, pp. 582–589. DOI: 10.1109/IROS.1992.587396 (cit. on p. 4).

Posa, M., C. Cantu and R. Tedrake (2014). 'A direct method for trajectory optimization of rigid bodies through contact'. In: *The International Journal of Robotics Research* 33.1, pp. 69–81 (cit. on p. 7).

Powell, M. J., E. A. Cousineau and A. D. Ames (2015). 'Model predictive control of underactuated bipedal robotic walking'. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5121–5126 (cit. on p. 6).

Pratt, G. A. and M. M. Williamson (1995). 'Series elastic actuators'. In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. Vol. 1. IEEE, pp. 399–406 (cit. on p. 45).

Pratt, J., J. Carff, S. Drakunov and A. Goswami (2006). 'Capture point: A step toward humanoid push recovery'. In: *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, pp. 200–207 (cit. on p. 6).

Pratt, J., T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson and P. Neuhaus (2012). 'Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid'. In: *The International Journal of Robotics Research* 31.10, pp. 1117–1133 (cit. on p. 45).

Radkhah, K. and O. Von Stryk (2014). 'A study of the passive rebound behavior of bipedal robots with stiff and different types of elastic actuation'. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5095–5102 (cit. on p. 45).

Raibert, M. H. (1986). *Legged robots that balance*. MIT press (cit. on pp. 6, 71, 111).

Rasmussen, C. E. (2003). 'Gaussian processes in machine learning'. In: *Summer School on Machine Learning*. Springer, pp. 63–71 (cit. on p. 51).

Robotics, A. (2017a). *Cassie - Next Generation Robot*. URL: https://www.youtube.com/watch?v=Is4JZqhAy-M\&ab_channel=OregonStateUniversity (visited on 04/03/2021) (cit. on pp. 71, 145).

Robotics, A. (2017b). *Cassie Robot*. URL: https://spectrum.ieee.org/agility-robotics-introduces-cassie-a-dynamic-and-talented-robot-delivery-ostrich (visited on 19/02/2022) (cit. on p. 5).

Saputra, R. P., N. Rakicevic, D. Chappell, K. Wang and P. Kormushev (2021). 'Hierarchical Decomposed-Objective Model Predictive Control for Autonomous Casualty Extraction'. In: *IEEE Access* 9, pp. 39656–39679. DOI: 10.1109/ACCESS.2021.3063782 (cit. on p. 14).

Schaft, G. (2016). *Schaft Robot*. URL: https://www.youtube.com/watch?v=bTKUkCT7Fmk (visited on 07/08/2022) (cit. on pp. 4, 5).

Sreenath, K., H.-W. Park, I. Poulakakis and J. W. Grizzle (2011). 'A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL'. In: *The International Journal of Robotics Research* 30.9, pp. 1170–1193 (cit. on p. 45).

Stanford Artificial Intelligence Laboratory et al. (23rd May 2018). *Robotic Operating System*. Version ROS Melodic Morenia. URL: https://www.ros.org (cit. on pp. 32, 35, 100).

Stasse, O., T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux et al. (2017). 'TALOS: A new humanoid research platform targeted for industrial applications'. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, pp. 689–695 (cit. on p. 35).

Stouraitis, T., L. Yan, J. Moura, M. Gienger and S. Vijayakumar (2020). 'Multi-mode Trajectory Optimization for Impact-aware Manipulation'. In: *2020 IEEE/RSJ International Conference*

on Intelligent Robots and Systems (IROS), pp. 9425–9432. DOI: 10.1109/IROS45743.2020.9341246 (cit. on p. 113).

Wächter, A. and L. T. Biegler (2006). 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming'. In: *Mathematical programming* 106.1, pp. 25–57 (cit. on pp. 96, 124).

Wang, K., H. Fei and P. Kormushev (2021a). *Fast Online Optimization for Terrain-Blind Bipedal Robot Walking with a Decoupled Actuated SLIP Model*. arXiv: 2109.09373 [cs.RO] (cit. on pp. 12, 13, 146).

Wang, K., D. Marsh, R. P. Saputra, D. Chappell, Z. Jiang, A. Raut, B. Kon and P. Kormushev (2020a). 'Design and Control of SLIDER: An Ultra-lightweight, Knee-less, Low-cost Bipedal Walking Robot'. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3488–3495. DOI: 10.1109/IROS45743.2020.9341143 (cit. on pp. 10, 11, 47, 72, 103, 123).

Wang, K., D. Marsh, R. P. Saputra, D. Chappell, Z. Jiang, A. Raut, B. Kon and P. Kormushev (2020b). 'Design and Control of SLIDER: An Ultra-lightweight, Knee-less, Low-cost Bipedal Walking Robot'. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3488–3495. DOI: 10.1109/IROS45743.2020.9341143 (cit. on p. 13).

Wang, K., R. P. Saputra, J. P. Foster and P. Kormushev (2022). 'Improved Energy Efficiency via Parallel Elastic Elements for the Straight-Legged Vertically-Compliant Robot SLIDER'. In: *Robotics for Sustainable Future*. Cham: Springer International Publishing, pp. 129–140. ISBN: 978-3-030-86294-7 (cit. on pp. 11, 13).

Wang, K., A. Shah and P. Kormushev (2018). 'SLIDER: A Bipedal Robot with Knee-less Legs and Vertical Hip Sliding Motion'. In: *Proc. 21st International Conference on Climbing and Walking Robots and Support Technologies for Mobile Machines (CLAWAR 2018)*. Panama. URL: http://kormushev.com/papers/Wang_CLAWAR-2018.pdf (cit. on pp. 10, 11, 13, 27, 123).

Wang, K., G. Xin, S. Xin, M. Mistry, S. Vijayakumar and P. Kormushev (2021b). *A Unified Model with Inertia Shaping for Highly Dynamic Jumps of Legged Robots*. arXiv: 2109.04581 [cs.RO] (cit. on pp. 12, 14).

Wensing, P. M. and D. E. Orin (2013). 'High-speed humanoid running through control with a 3D-SLIP model'. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5134–5140 (cit. on p. 112).

Wensing, P. M. and D. E. Orin (2014). 'Development of high-span running long jumps for humanoids'. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 222–227 (cit. on p. 112).

Wieber, P.-B. (2006). 'Trajectory free linear model predictive control for stable walking in the presence of strong perturbations'. In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 137–142 (cit. on pp. 71, 88).

Winkler, A. W., C. D. Bellicoso, M. Hutter and J. Buchli (2018). 'Gait and trajectory optimization for legged systems through phase-based end-effector parameterization'. In: *IEEE Robotics and Automation Letters* 3.3, pp. 1560–1567 (cit. on p. 2).

Wu, J.-c. and Z. Popović (2010). 'Terrain-adaptive bipedal locomotion control'. In: *ACM Transactions on Graphics (TOG)* 29.4, pp. 1–10 (cit. on p. 90).

Xie, Z., P. Clary, J. Dao, P. Morais, J. Hurst and M. van de Panne (2020). 'Learning Locomotion Skills for Cassie: Iterative Design and Sim-to-Real'. In: *Proceedings of the Conference on Robot Learning*. Ed. by L. P. Kaelbling, D. Kragic and K. Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, pp. 317–329. URL: https://proceedings.mlr.press/v100/xie20a.html (cit. on p. 90).

Xie, Z., P. Clary, J. Dao, P. Morais, J. Hurst and M. van de Panne (2019). 'Iterative reinforcement learning based design of dynamic locomotion skills for cassie'. In: *arXiv preprint arXiv:1903.09537* (cit. on p. 9).

Xin, S., R. Orsolino and N. Tsagarakis (2019). 'Online Relative Footstep Optimization for Legged Robots Dynamic Walking Using Discrete-Time Model Predictive Control.' In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS 2019*. Macau, China (cit. on pp. 6, 71, 77).

Xiong, X. and A. D. Ames (2018). 'Bipedal hopping: Reduced-order model embedding via optimization-based control'. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3821–3828 (cit. on pp. 6, 71).

Yang, T., E. R. Westervelt, J. P. Schmiedeler and R. A. Bockbrader (2008). 'Design and control of a planar bipedal robot ERNIE with parallel knee compliance'. In: *Autonomous robots* 25.4, p. 317 (cit. on p. 45).

Yang, W. and M. Posa (2021a). *Impact Invariant Control with Applications to Bipedal Locomotion*. arXiv: 2103.06907 [cs.RO] (cit. on p. 113).

Yang, Y., T. Zhang, E. Coumans, J. Tan and B. Boots (2021b). *Fast and Efficient Locomotion via Learned Gait Transitions*. arXiv: 2104.04644 [cs.RO] (cit. on p. 9).