Towards Humanoids Operating Mobility Devices Designed for Humans

by

Vidyasagar Rajendran

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

© Vidyasagar Rajendran 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:
Gordon Cheng
Professor, Cognitive Systems
School of Computation, Information and Technology
Technical University of Munich

Supervisor(s):
Katja Mombaur
Professor, Systems Design Engineering &
Mechanical and Mechatronics Engineering, University of Waterloo
Professor, Institute for Anthropomatics and Robotics,
Karlsruhe Institute of Technology

Internal Members:
Nasser Lashgarian Azad
Professor
Systems Design Engineering
University of Waterloo

John McPhee
Professor
Systems Design Engineering
University of Waterloo

Internal-External Member:
William Melek
Professor
Mechanical and Mechatronics Engineering
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Vidyasagar Rajendran authored all chapters which were written under the supervision of Katja Mombaur.

This thesis contains content from several first-authored publications:

- Chapter 3: **V. Rajendran**, J. F. -S. Lin and K. Mombaur, "Towards Humanoids Using Personal Transporters: Learning to Ride a Segway from Humans," 2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob), Seoul, Korea, Republic of, 2022, pp. 01-08, doi: 10.1109/BioRob52689.2022.9925458. [198][1]
- Chapter 4: **V. Rajendran**, F. J. A. Chavez and K. Mombaur, "Whole-body Stabilization of the REEM-C Humanoid on Actuated Balance Boards in Sagittal and Frontal Directions," 2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids), Austin, TX, USA, 2023, pp. 1-8, doi: 10.1109/Humanoids57100.2023.10375223. [197]
- Chapter 5: W. Thibault*, **V. Rajendran*** and K. Mombaur, "Bimanual Manipulation Workspace Analysis of Humanoid Robots with Object Specific Coupling Constraints," 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Ginowan, Japan, 2022, pp. 141-148, doi: 10.1109/Humanoids53995.2022.10000183. *Equal first authorship. [224]
- Chapter 5: **V. Rajendran***, W. Thibault*, K. Mombaur, "Motion Planning for Bimanual Manipulation of Humanoid Robots based on a Novel Manipulability-Stability Metric and Object-Specific Constraints," in IEEE Transactions on Robotics, 2024, Revise and Resubmit. *Equal first authorship. [201]
- Chapter 6: **V. Rajendran** and K. Mombaur, Advancing Humanoid Locomotion: Enhancing Versatility Through the use of Personal Transporters. Science Robotics, 2024, To Be Submitted. [200]

The content from these publications has been used verbatim, save for minor formatting changes to fit the thesis template. Chapter 5 contains content from a co-first authored paper with William Thibault. I lead the code development, simulation, experimental validation and manuscript writing for the work as the major contributor. The work in [201] is an evolved paper, extending the work in [224]. All other works were first authored with all major code development, simulation, experiments and manuscript writing done by myself, Vidyasagar Rajendran.

[1] In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the University of Waterloo's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Abstract

Humanoid robotics is advancing rapidly, with significant potential to address challenges in disaster recovery, manufacturing, and healthcare. Despite great progress, current humanoid capabilities remain limited, particularly in terms of efficient mobility over long distances. Integrating humanoid robots with personal transporters (PTs) like Segways, offers a promising solution, enabling them to operate more efficiently in human-centric environments such as factories, malls, and airports. This approach not only preserves the humanoid's ability to navigate complex, uneven terrain with its legs, but also enhances versatility, allowing for faster, more energy-efficient movement on flat surfaces.

This thesis explores methods for enabling bipedal humanoids to operate PTs, focusing on the REEM-C humanoid riding a Segway x2 SE. The research begins by analyzing human interactions with Segways to reverse-engineer their internal controllers, leading to a high-fidelity simulation model. This model informs the development of control algorithms for the REEM-C, enabling successful simulation-based demonstrations of humanoid-driven Segway motions, including translational, rotational, and mixed maneuvers. Building on this, balance stabilization strategies are devised for actuated balance boards, addressing both frontal and sagittal plane control through an integration of admittance control strategies.

A comprehensive analysis of bimanual manipulation is also conducted, emphasizing manipulability and stability within a constrained workspace. Using a combined manipulability-stability metric, collision-free bimanual trajectories are generated, demonstrating improved stability during dynamic tasks such as manipulating objects of varying shapes and masses. This analysis underpins the implementation of bimanual manipulation strategies needed for operating the Segway's LeanSteer handlebar.

The final contribution consolidates all findings, presenting a whole-body control strategy that enables the REEM-C to ride a Segway safely and effectively. A stack-of-tasks quadratic program is utilized to ensure stability, balance, and bimanual control in dynamic conditions. Experimental validation demonstrates the feasibility of this approach, showcasing the REEM-C's ability to operate a Segway under real-world conditions. This research provides a step towards more versatile and adaptable humanoid mobility solutions for everyday human environments.

Acknowledgements

I would like to begin by expressing my heartfelt gratitude to my supervisor, Katja Mombaur, for her unwavering support, mentorship, and guidance over the past four years. Her insightful ideas and thoughtful feedback have been a constant source of inspiration, profoundly shaping my development as a researcher. Under her mentorship, I was fortunate enough to present my work in Spain, Germany, the Netherlands, South Korea, Japan, and the United States, experiences for which I am deeply grateful.

I would also like to extend a sincere thanks to my committee members, John McPhee, Nasser Lashgarian Azad, William Melek, and Gordon Cheng, for their invaluable feedback on my thesis. It has been a privilege to engage with each of them over the years, whether through courses or discussions about my research. Every interaction has enriched my academic journey.

I am sincerely appreciative to all the members of the HCRMI lab for making these past four years an exciting and collaborative journey. Our discussions, whether in the lab or elsewhere, were always productive and enjoyable. A special thanks to Will Thibault for the collaborations, travel, and friendship. It has been amazing working with you, and I look forward to continuing our joint efforts in the future. To Peter Q. Lee: we did it! Completing our PhDs together has been an incredible journey. Thank you for the collaborations and friendship along the way. Thank you to Branca da Costa for your great support with administrative tasks, especially managing expenses. Jonathan Lin, your insightful advice and encouragement since my Master's days in Dana's lab have been invaluable. Francisco Javier Andrade Chavez, I deeply appreciate your advice, discussions, and collaborations on humanoids. Anas Mahdi, thank you for being a fantastic office mate and for our work together at the IDEAs clinic. Pranav Barot, thank you for the engaging discussions and our time spent working together in the lab. Marie Charbonneau, your help and insights into humanoid control were greatly appreciated. Alyson Colpitts, thank you for your support in the final stages of my PhD. Jonas, our discussions over the years have been incredibly enriching, thank you. Finally, I want to express my gratitude to Jan Lau, Kareem Mostafa, Mennatallah Rihan, Christian Mele, Jiwon Yang, and Mahsa Parsapour. Collaborating and exchanging ideas with all of you was both inspiring and rewarding.

I would also like to express my appreciation to the IDEAs Clinic, where I had the privilege of contributing to course development, hackathons, and mentoring the next generation of engineers. In particular, I am grateful to Chris Rennick, Sanjeev Bedi, and Silas Ifeanyi for their close collaboration on these endeavors. Through the IDEAs Clinic, I also had the opportunity to collaborate with the Student Success Office, which was a great experience that allowed me to help deliver several in-class modules on teamwork and soft skills. It was a pleasure working with Derek Rayside to mentor students in the 2024 interdisciplinary capstone course, which was another connection I made through the IDEAs Clinic. Lastly, thanks to Andrew Gryguc for bringing me on to help teach and contribute to the ECE 699 project course.

The RoboHub has been an invaluable resource, and I am grateful for all the support, whether it was borrowing equipment or seeking expert advice. I extend my thanks to Alexander Werner and Brandon DeHart for the insightful discussions on humanoids and robotics.

I am also deeply appreciative of Robert Wagner, who assisted me with numerous tasks throughout my PhD, including building the gantry used for the final REEM-C testing.

I want to express deep gratitude to my friends for their camaraderie throughout this journey. Your support has been my driving force. To Sunny, Jay, and Harsh: thank you for always standing by me and lifting me up. Our enduring friendship kept me grounded throughout my academic career (and thank you for the Prada bag!). To Çagla, thank you for the collaborations in the lab and the incredibly fun board game nights that brought so much joy. Shaundell, I deeply appreciate the good times we shared and the fun dance classes we attended together. A special thanks to Shubham, Anavir, Laurence, and Ayush for their encouragement along the way. To the many others who go unnamed here, please know that your support and presence have left a lasting impact on me, and I am forever grateful for the role you played in this journey.

A heartfelt thank you to my parents, Raj and Shanthi, for their steadfast support and for seeing me through to this day. I could not have done it without you. I was a few months old around the time my Mom was defending her PhD. It's funny how life comes full circle. Thank you to my uncle, Mahen, for his encouragement and valuable life advice whenever we met. I am also truly indebted to my grandparents, Mani and Uma, for their constant support. To those no longer with us, Bhuvaneshwari, Rajasekaran, and Sundar, thank you for always supporting me throughout my educational journey. To the many relatives who will go unnamed, thank you for all of your support and love over the years.

I am deeply grateful to NSERC for the funding that made this research possible, as well as to the Canada Excellence Research Chair program and the Hector Foundation (Germany). A special thank you to Segway and Segway of Ontario for the initial collaboration which played a pivotal role in shaping my research.

Finally, I am genuinely thankful to the University of Waterloo for an incredible 11 years of post-secondary education (BASc, MASc, and now PhD). It has been an unforgettable journey, filled with growth, learning, and countless memorable experiences.

Dedication

In memory of my Grandmother, Bhuvaneshwari. She was my greatest cheerleader and supporter, seeing me through all of my educational milestones. I miss you, dearly.





Table of Contents

List of Figures xiv

List of Tables xxii

1 Introduction 1

1.1 Problem Definition and Approach . . . 3
1.2 REEM-C Humanoid Robot Platform . . . 6
1.3 Thesis Contributions . . . 8
1.4 Thesis Organization . . . 9

2 Literature Review and Background Theory 10

2.1 Current State of Humanoid Robots . . . 10
2.2 Humanoid Modelling . . . 11
2.2.1 Whole-body Dynamics . . . 11
2.2.2 Centroidal Dynamics . . . 12
2.3 Whole-body Control . . . 13
2.3.1 Whole-body Controllers from Literature . . . 13
2.3.2 Whole-body Control Formulation . . . 13
2.4 Model-based Motion Generation for Humanoids . . . 15
2.4.1 Template Models . . . 16
2.5 Floating Base Estimation . . . 19
2.5.1 Anchor Point Method . . . 19
2.5.2 Extended Kalman Filter Based Estimation . . . 21
2.6 Inverse Optimal Control . . . 21
2.6.1 Bi-level IOC Problem . . . 22
2.7 Movement Primitives . . . 23

2.8 Model Predictive Control . . . 24
2.9 Reinforcement Learning . . . 25
2.10 Frameworks and Libraries Used . . . 26
2.10.1 Pinocchio Rigid Body Dynamics Library . . . 26
2.10.2 EXOTica Extensible Optimization Library . . . 26
2.10.3 Task Space Inverse Dynamics Library . . . 27

3 Towards Humanoids Using Personal Transporters: Learning to Ride a Segway from Humans in Simulation 28
3.1 Introduction . . . 28
3.2 Experimental Studies of Human-Segway System . . . 30
3.2.1 Marker Sets and Equipment . . . 31
3.2.2 Experimental Protocols . . . 31
3.2.3 Motion Capture Trial Analysis . . . 31
3.3 Modelling and Control of the Segway . . . 35
3.3.1 Pitch Controller . . . 37
3.3.2 Yaw-Rate Controller . . . 37
3.4 Modelling and Control of the REEM-C . . . 44
3.4.1 Linear Inverted Pendulum Model Based Controller . . . 44
3.4.2 Upper Body Impedance and Admittance Controller . . . 45
3.5 Simulation Experiments . . . 46
3.5.1 Driving Straight . . . 46
3.5.2 Rotating In Place . . . 47
3.5.3 Driving in a Small Arc . . . 48
3.6 Conclusions and Perspectives . . . 50

4 Whole-body Stabilization of the REEM-C Humanoid on Actuated Balance Boards in Sagittal and Frontal Directions 52
4.1 Introduction . . . 52
4.2 Analysis of a Human on a Balance Board . . . 55
4.3 Background Theory . . . 56
4.3.1 PID Controller Formulation . . . 56
4.3.2 Balance Board . . . 57
4.3.3 Zero Moment Point Estimation . . . 57

4.4 Humanoid Controller Design . . . 58
4.4.1 Ankle Admittance . . . 58
4.4.2 Base Orientation Admittance . . . 59
4.4.3 CoM Admittance . . . 59
4.4.4 Contact Wrench Distribution . . . 61
4.4.5 Task Space Inverse Dynamics . . . 61
4.5 Experiments . . . 62
4.5.1 Implementation Details . . . 62
4.5.2 Sagittal Plane Balancing in Simulation . . . 63
4.5.3 Sagittal Plane Balancing on Hardware . . . 64
4.5.4 Frontal Plane Balancing . . . 65
4.5.5 Frontal Plane Stabilization against Large Perturbation . . . 65
4.6 Conclusions and Perspectives . . . 66

5 Motion Planning for Bimanual Manipulation of Humanoid Robots based on a Novel Manipulability-Stability Metric and Object-Specific Constraints 67
5.1 Introduction . . . 68
5.1.1 Related Work . . . 68
5.1.2 Contributions . . . 71
5.2 Workspace Metrics . . . 72
5.2.1 Manipulability . . . 72
5.2.2 Stability . . . 73
5.2.3 Combined Manipulability-Stability Metric . . . 73
5.3 Workspace Analysis Methodology . . . 74
5.3.1 Whole-body Inverse Kinematics . . . 74
5.3.2 Generation of a General Bimanual Workspace Map . . . 75
5.3.3 Generation of Bimanual Workspace Maps under Object-induced Constraints . . . 76
5.3.4 Workspace Visualization . . . 77
5.4 Workspace Analysis Results . . . 79
5.4.1 General Workspace Maps . . . 79
5.4.2 Workspace Maps with Varying Center of Mass Velocities . . . 79
5.4.3 Workspace Maps with Different Object Masses . . . 80

5.4.4 Workspace Maps with Different Objects . . . 82
5.4.5 Workspace Maps of TALOS . . . 83
5.5 Motion Generation Methodology . . . 87
5.5.1 Planning Problem Formulation . . . 87
5.5.2 Sampling-Based Planning for Bimanual Manipulation . . . 88
5.5.3 Extension of Bimanual Model to Include Contact Forces . . . 92
5.5.4 Control Formulation . . . 93
5.6 Motion Generation Results for the REEM-C Humanoid Robot . . . 95
5.6.1 Experimental Bimanual Box Motions . . . 96
5.6.2 Experimental Bimanual Broom Motions . . . 97
5.6.3 Experimental Bimanual Rolling Pin Motions . . . 97
5.6.4 Bimanual Box Motions on a Shelf . . . 97
5.7 Conclusions and Outlook . . . 102

6 Advancing Humanoid Locomotion: Enhancing Versatility Through the use of Personal Transporters 103

6.1 Introduction . . . 103
6.2 Materials and Methods . . . 106
6.2.1 Controller Overview . . . 106
6.2.2 Constrained Whole-body Inverse Kinematics . . . 106
6.2.3 Bimanual Model for Operating LeanSteer . . . 107
6.2.4 Bimanual Admittance Control . . . 109
6.2.5 Tasks Implemented in Whole-body Controller and Stabilizer . . . 109
6.2.6 Implementation Details . . . 110
6.3 Results . . . 111
6.3.1 Elementary Motions . . . 111
6.3.2 Mixed Motions . . . 114
6.3.3 Complex Motions . . . 114
6.4 Discussion . . . 119
6.4.1 Limitations and Future Work . . . 120

7 Discussion and Conclusions 121

7.1 Summary of Work . . . 121
7.2 Limitations and Future Work . . . 122

References **124**

Appendices **146**

A Linear MPC for Segway Reference Tracking **147**

A.1 Introduction and Literature Survey 147

A.2 Background 149

A.2.1 General MPC Description 149

A.2.2 Linear Constrained MPC 150

A.2.3 Linear MPC for Tracking 151

A.3 Segway Model 151

A.4 Model Predictive Control Formulation for Segway System 154

A.4.1 Choosing the Weights for Q and R 157

A.4.2 Gazebo/Simulink Co-Simulation 157

A.5 Simulation Experiments 158

A.6 Conclusions 162

B REEM-C Hardware Malfunction and Fix **164**

C REEM-C Energy Expenditure Comparison Between Walking and Personal Transporter Usage **169**

List of Figures

1.1 The personal transporters shown from left to right are: Ninebot Kickscooter E22 [5], Ninebot S-Max (mini self-balancing Segway) [6], Segway x2 SE [13], Segway Drift W1 Electric Skates [11]. The complexity in terms of user control increases from left to right along the arrowhead [48]. This is not an exhaustive list of personal transporters but covers some common designs. For this thesis, we consider the Segway x2 SE as the exemplar PT. 2

1.2 The high-level block diagram of the proposed approach to endow humanoids with the skills to operate PTs. 4

1.3 A high-level overview of the various topics covered in this thesis. They all ultimately contribute to the work in Chapter 6 with the REEM-C operating a Segway personal transporter in real hardware experiments. 5

1.4 The REEM-C humanoid with specifications and degrees of freedom labelled [9]. 7

1.5 REEM-C kinematic structure. 8

1.6 Overview of the chapters in this thesis. The research in Chapter 3-5 contribute to Chapter 6. 9

2.1 NVIDIA CEO Jensen Huang introducing Project GR00T, a general-purpose foundation model for humanoid robots [105]. Several commercial and research humanoids are pictured in the background. 11

2.2 An overview of various legged robot models used for optimization-based control. Adapted from [239]. 16

2.3 3D SEA walker [57]. 18

2.4 The SEROW state estimation method presented in [184]. This method is extended for this thesis. 21

2.5 A categorization of RL agents and the methods used to train them. Adapted from [186]. 25

3.1 A high-level overview of our approach for synthesizing controllers to allow the REEM-C humanoid to drive a Segway. The shaded grey area highlights the focus of this chapter. 30

3.2 Segway x2 SE with motion capture markers. The markers were placed in a way that would allow for easy reconstruction of the LeanSteer and wheel kinematics. 31

3.3 Motion capture trial with the human driving in a straight line. 32

3.4 Representations of the human model based on De Leva anthropometrics [66] for kinematic analysis. On the right, the meshes of the model are shown and on the left, the axes frames for each joint. The x, y, z frames corresond to the colors red, green and blue respectively. The root frame of the human model is at the pelvis (denoted with a larger frame in the right figure). The blue dots on the right figure correspond to marker placement during motion capture. . 33

3.5 A graph showing the human's CoM translation in the x-direction (pointing forward with respect to the Segway) versus the Segway's pitch angle. 34

3.6 Graphs showing the human's arm joint angles and the Segway LeanSteer joint angle for a "rotate in place" trial. x, y and z correspond to roll, pitch and yaw respectively. The oscillations seen in the graphs are artefacts from the inverse kinematics process. 35

3.7 A block diagram showing the general procedure for PT control synthesis. The pipeline starts with motion capture and model design of the real PT. Kinematic and dynamic analysis is performed using RBDL and BIORBD to retrieve the PT joint angles, CoM trajectories etc. Finally, this data is used to perform control parameter identification with the PT model, joint angle data and the Simulink parameter identification toolbox. 37

3.8 Block diagram of the pitch controller parameter estimation procedure. A point mass trajectory excites the system and the Segway's pitch (θ_p) and linear velocity are monitored for the estimation procedure. τ_{yaw} is set to zero for this identification process. 38

3.9 The Segway simulated pitch and linear velocity response, as compared to measured experiment data, after parameter estimation. 39

3.10 a) An exemplar "rotate in place" motion capture trial showing the LeanSteer angle and corresponding yaw rate. b) The polynomial fit (Equation 3.3) of all the Segway "rotate in place" motion capture trials to map the relationship between the LeanSteer input angle to the Segway yaw-rate. 41

3.11 Block diagram of the yaw rate controller parameter estimation procedure. A LeanSteer trajectory excites the system and ψ, the Segway's yaw-rate is monitored. P denotes the relationship between the LeanSteer angle and the desired yaw rate as in Equation 3.3. 42

3.12 The simulated yaw-rate response of the Segway system. 43

3.13 REEM-C shifting its COM back and forth. The force-torque sensors at the wrists of the REEM-C allow the upperbody to actively switch between compliance and stiff control. 47

3.14 REEM-C operating the LeanSteer mechanism. 47

3.15 REEM-C CoM and Segway linear velocity as compared to a corresponding "straight" motion capture trial. The desired CoM trajectory CoM_{des} comes from human motion capture experiments. 48

3.16 Segway LeanSteer angle, yaw-rate, pitch and REEM-C CoM position for a rotate in place simulation. 49

3.17 REEM-C performing a small arc. The yellow parking stop keeps the Segway stable before starting the motion. 50

3.18 The global $x-$ and $y-$ position of the REEM-C in the World frame along with the desired arc trajectory. 51

4.1 The REEM-C Humanoid on a balance board. The black steel bar was retrofitted to actuate the board manually. Sagittal and frontal plane setups are shown in the background, being actuated by two humans. 54

4.2 A high-level overview of the stabilization controller developed in this work to balance on an actuated balance board. 55

4.3 a) One of the authors balancing in the sagittal plane on a balance board. The Theia3D system reconstructs all limb segments. b) Pressure insoles showing foot center of pressure estimates. Here, the human is leaning backwards, concentrating their center of pressure near the heals. 56

4.4 The balance board model used in this work. 58

4.5 a) Unmodelled ankle flexibility b) The modified cart-table model taking into account flexibility (figure taken and adapted from Benallegue et al. [30]). . . 60

4.6 Frames (from left to right) of the REEM-C humanoid balancing on an actuated sagittal board rotating at 20 °/s, with a maximum inclination of 7°. On the second to last frame (on the right), the robot is pushed with a force of 55 N for 0.3 s and successfully rejects the disturbance by generating momentum in the opposite direction of the push with its arms. 62

4.7 Graphs showing the CoM, ZMP and left ankle wrench measurements for the simulation sagittal board experiment. The references for the CoM and ZMP are shown using dotted black lines. 62

4.8 Frames (from left to right) of the REEM-C humanoid balancing on a sagittal board actuated by a human. The maximum inclination is 7°. We note that the robot has no knowledge of the board and it is treated as a disturbance in this case. 63

4.9 Graphs showing the CoM, ZMP and left ankle wrench measurements for the real robot sagittal board experiment. The references for the CoM and ZMP are shown using dotted black lines (at $y = 0$). 63

4.10 Frames (from left to right) of the REEM-C humanoid stabilizing on an actuated frontal plane balance board. 65

4.11 Frames (from left to right) of the REEM-C humanoid stabilizing in the frontal plane after a large step-like disturbance is applied to the balance board by a human foot. 66

5.1 Examples of objects introducing tightly coupled bimanual manipulation scenarios addressed in this work: a) a small 1 kg box b) a large 7 kg box c) a sweeping broom (with a left-handed grasp) d) a rolling pin. 69

5.2 a) The REEM-C humanoid with $World$ frame and left and right TCP frames labeled. Here x is in red, y is in green and z is in blue. b) For the general workspace generation case, a single point on the workspace grid encodes the center between the two end effectors of the robot. 75

5.3 The metric color range used throughout this work. 77

5.4 Sensitivity visualizations of the workspace considered for the general case for α and β. Low quality voxels are blue and high quality voxels are yellow/orange. Voxels that are unreachable or unstable are translucent grey. The first row shows the workspace that was considered and the second row shows a horizontal cut through. a) and d) show $\alpha = 1$ and $\beta = 0$ (the manipulability metric), b) and e) show $\alpha = 0$ and $\beta = 1$ (the stability metric), and c) and f) show $\alpha = 0.5$ and $\beta = 0.5$ (the combined manipulability-stability metric) as detailed in Equation 5.9. 80

5.5 Visualizations of different CoM velocity (m/s) cases for a subset of the general workspace using the stability metric. a) $\dot{r}_{CoM} = [0.0, 0.0, 0.0]$ b) $\dot{r}_{CoM} = [0.2, 0.0, 0.0]$ box c) $\dot{r}_{CoM} = [0.0, 0.6, 0.0]$ d) $\dot{r}_{CoM} = [0.2, 0.6, 0.0]$. 81

5.6 Visualization of the bimanual manipulability-stability metric for the REEM-C manipulating a 1 kg box. a) A cut through at $z = 1.0$ m b) A cut through at $z = 1.2$ m. 82

5.7 Visualization of the bimanual manipulability-stability metric for the REEM-C manipulating a 7 kg box. a) A cut through at $z = 1.0$ m b) A cut through at $z = 1.2$ m. 83

5.8 Visualization of the bimanual manipulability-stability distribution for REEM-C manipulating a rolling pin. a) A cut through at $z = 0.9$ m b) A cut through at $z = 1.1$ m. 84

5.9 Visualization of the bimanual manipulability-stability distribution for REEM-C manipulating a broom. a) A cut through at $z = 0.9$ m b) A cut through at $z = 1.0$ m. 85

5.10 Visualization of the bimanual manipulability-stability distribution for the TALOS humanoid, for the general case with spherical coordinates as described in Subsection 5.3.2. a) A cut through at $z = 0.9$ m b) A cut through at $z = 1.1$ m. 85

5.11 Visualization of the bimanual manipulability-stability metric for the TALOS manipulating a 1 kg box. a) A cut through at $z = 1.0$ m b) A cut through at $z = 1.2$ m. 86

5.12 The planning and control block diagram highlighting the various components of the motion generation pipeline. As a note, $\boldsymbol{q}_{cmd}$ is tracked with the "Joint Reference Tracking" task and is updated at each controller time step (0.005s). 91

5.13 The dynamic model for bimanual manipulation with the robot rigidly grasping an object. 92

5.14 Force-torque data from the left wrist force-torque sensor for one trial of the box motion shown in Figure 5.17. 96

5.15 Manipulability-Stability metric along real hardware trajectories. 10 trials were done for each object and the graphs show the average and standard deviation along the trajectories for the box, broom and rolling pin. Exemplar paths generated with the manipulability-stability cost are shown for each object. . 98

5.16 The left and right hand end effector error along all real hardware trajectories (with manipulability-stability cost). As can be seen, all the errors are bounded to sub centimeter and sub 0.03 radians. 99

5.17 Experimental box trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. The red dot indicates the start of the trajectory and the green indicates the goal. Both arms are kept in manipulable configurations along the path as compared to the path constructed without the cost in Figure 5.18. 100

5.18 Experimental box trajectory without manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. The red dot indicates the start of the trajectory and the green indicates the goal. At the end of the trajectory the left arm is more extended and near joint limits, resulting in lower path quality as was shown in the graph in Figure 5.15. . . 100

5.19 Experimental broom trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. 101

5.20 Experimental rolling pin trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. 101

5.21 Experimental shelf box manipulation, with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. 101

5.22 Experimental shelf box manipulation, without manipulability-stability cost. The arms are closer to singularity and the whole-body configurations are less stable along the trajectory as compared to Figure 5.21. A video of this motion (along with the one from Figure 5.21) is shown in this video: https://youtu.be/q8r3p5O4F6s. 101

6.1 The REEM-C humanoid operating a Segway. a) The control framework in this work is deployed for various test scenarios on real hardware. b) Operation of the Segway is depicted in simulation, showing handlebar manipulation and forward and backward CoM shifting. 105

6.2 The stack-of-tasks whole-body controller (WBC) developed and used in this work shown in block diagram form. 107

6.3 A diagram of the LeanSteer handlebar modeling. The REEM-C operates the LeanSteer mechanism of the Segway with a closed grasp. While turning the LeanSteer, the upperbody uses admittance control to minimize internal forces. 108

6.4 REEM-C driving straight on a Segway. a) Snapshots of the REEM-C driving a Segway. The humanoid starts from rest, drives straight and then stops after 2 m. b) An x-y plot of the path travelled by REEM-C during this trial. c) The local CoM trajectory in the x-direction (c_x^l) traversed by the REEM-C. Red indicates the start of the trajectory, blue is when the direction changes to start to slow down and green is when the robot re-centers itself to bring the system to a stop. d) The local CoM trajectory traversed by the REEM-C plotted against the desired local CoM trajectory. The forces felt at the ankles e) of the REEM-C and the wrists f). The start of the decelerations are indicated with a vertical red line. 112

6.5 The REEM-C humanoid rotating in place by turning the LeanSteer handlebar. a) Real hardware trial snapshots of the REEM-C rotating in place. b) Simulation trial snapshots of the REEM-C rotating in place. c) Plot of the floating base yaw in red and floating base signed linear velocity in green. As can be seen, the linear velocity is maintained close to zero and the robot does a full rotation of 2π. 113

6.6 REEM-C driving a serpentine path. a) Real hardware snapshots of the REEM-C humanoid driving a serpentine path. b) The path travelled by the REEM-C in the x-y plane. The c) forces and d) torques measured at the right ankle. The spikes in the graphs indicate points that the robot turned the LeanSteer; they settled at the end of the trajectory. 115

6.7 REEM-C driving over a bump. a) Snapshots of the experiment where REEM-C drove over a bump (exercise mats). b) The path taken by the REEM-C plotted in 3D. The bump is denoted by vertical dashed lines. c) The left ankle forces during the trial. Climbing the bump, the forces are contained but descending the bump causes large spikes that settles after a few seconds. d) The forces measured at the left wrist. We see that similar to the ankle forces, the bump is clearly noted in the plots. 116

6.8 REEM-C self-balancing on a Segway and rejecting a push disturbance. a) The REEM-C humanoid rejecting a disturbance from an instrumented push stick (the push happens on frame ii, t=1.2s). The push was 113N and lasted for 0.7s. b) The desired local CoM in the x-direction ($c_{x_{\text{des}}}^l$) plotted against the measured local CoM in the x-direction (c_x^l). c) The measured local CoM in the x-direction plotted against the measured global CoM in the x-direction (c_x^g). d) The measured local CoM velocity in the x-direction $\dot{c}_x^l$ plotted against the measured global CoM velocity in the x-direction $\dot{c}_x^g$. 118

6.9 The REEM-C humanoid driving over a set of randomly strewn bike tires. In these snapshots, the REEM-C drives over a set of bike tires that act as disturbances or uneven terrain. The robot can easily drive over the tires with the proposed controller. 119

A.1 High-fidelity Segway model created in URDF format, simulated in ROS/Gazebo. 149

A.2 Controller action using MPC [255]. $u(t+k|t)$ is the predicted control action at $t+k$ given $u(t)$. $y(t+k|t)$ is the predicted output at $t+k$ given $y(t)$. . . 150

A.3 Segway diagram of forces and moments acting on the body, pendulum, and wheels [63]. As a note, a few parameters and symbols are different than the ones presented in this section. 152

A.4 Decoupled MPC control scheme to track reference trajectories with the Segway system. 155

A.5 Decoupled MPC control scheme to track reference trajectories with the Segway system. 156

A.6 Decoupled MPC control scheme to track reference trajectories with the Segway system using Gazebo/Simulink co-simulation. 158

A.7 Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs. The model used here is the linearized equations of motion, and the simulation was done in Simulink as shown in Figure A.5. 159

A.8 θ response for Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs. 160

A.9 Simulation of a) 5 Nm ramp disturbance acting on Segway handlebar b) system response using decoupled LQR control c) system response using decoupled MPC. 161

A.10 Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs. The model used here is the high fidelity model in Gazebo, and the simulation was performed using Gazebo/Simulink co-simulation [163]. As a note, the x-axis is labelled incorrectly and should read $Time[steps]$. 162

A.11 Segway model with a human point mass rider. 163

B.1 The REEM-C taken apart for debugging purposes. 165

B.2 My colleagues and I debugging the robot's logic box and compute module, which had been removed from its torso. 166

B.3 The REEM-C control computer board is shown here. Highlighted in red is the mSATA SSD connector. Once the SSD was removed from this connector, the motherboard was able to complete its POST (Power-On Self-Test). While this may seem like a simple fix, the board is located deep within the robot's torso, requiring the entire compute logic box to be disassembled (see Figure B.2 and Figure B.1). 167

B.4 The final solution to operationalize the robot involved mounting an external SSD onto the robot, running the control computer over USB 3.0. This configuration was sufficient to perform all tests described in Chapter 6. 168

C.1 The REEM-C walking. The REEM-C humanoid walking 0.2 m/s over flat ground for 0.75 m (from left to right). 170

C.2 REEM-C driving a Segway compared against walking. To travel a distance of 0.75 m, the REEM-C used 72% less energy when driving a Segway, as compared to walking. Using the internal 48v 1225 Wh battery, the REEM-C would last 5.4 hours riding a Segway and only 1.535 hours walking. 171

List of Tables

1.1 REEM-C degrees of freedom in each body part. 6

1.2 REEM-C mass distribution. 6

1.3 The associated videos for each chapter. These videos detail the methods and experiments conducted and serve as a quick summary of the work. 9

3.1 Motion capture trials performed by human riding the Segway. 33

3.3 Segway dynamic parameters. 36

3.4 Segway controller gains, before and after parameter estimation. 37

3.5 Segway yaw-rate controller gains, before and after parameter estimation. . . 40

4.1 The TSID Task Gains for the balance problem. The same gains are used for sagittal and frontal boards. 64

5.1 EXOTica Task Maps used for whole-body inverse kinematics. 78

5.3 The TSID task gains for the bimanual manipulation whole-body controller. . 95

5.4 The start and goal configurations across different tasks. 97

5.5 Comparison of BiRRT_n (baseline, no cost) and BiRRT_c (with manipulability-stability cost) across different tasks. 98

6.1 Nominal joint angles used for whole-body IK. The nominal joint angles for the REEM-C humanoid standing in double support with hands on the LeanSteer handlebar. 107

6.2 The TSID Task Gains for driving a personal transporter. 110

A.1 The parameters for the Segway system. 154

A.2 The parameters for the Segway balance and position MPC, controlling the system described in Equation A.15. 156

A.3 The parameters for the Segway yaw MPC, controlling the system described in Equation A.16. 156

Chapter 1

Introduction

Humanoid robotics is a burgeoning field which has received significant attention in the past years and is predicted to play a central role in the advancement of society as a whole in the 21st century [24]. Humanoids are typically designed to mimic human-like topology, structure, and motion capabilities, enabling them to seamlessly integrate into existing infrastructure built for humans [244]. Disaster recovery, manufacturing and healthcare are a few of the areas where humanoids can takeover dangerous, repetitive and labourious tasks to ease the workload on humans. However, much work remains to elevate humanoid capabilities to a level that allows them to be seamlessly integrated into our daily lives.

Bipedal humanoid walking was a difficult task at the DARPA Robotics Challenge (DRC), highlighting the gap between current state-of-the-art humanoid performance, and the level of performance required for real-world operation of humanoids. The humanoid robots that competed in the DRC required close to an hour to complete tasks that would typically take a healthy adult 10 minutes to do [215]. Few of the most successful humanoids at the competition used wheeled-motion to their advantage. However, bipedal walking is still an important skill to have, especially for uneven terrain [94]. How can we maintain bipedal gait while also taking advantage of wheeled locomotion for humanoids?

In recent years, multi-modal locomotion has been implemented in various robots. The DRC-HUBO+ is a humanoid robot that has wheeled attachments at its knees [149] enabling both wheeled and bipedal locomotion. A variant of the popular ANYmal quadruped is wheeled, with the ability to locomote efficiently and perform many different tasks [33]. The LEONARDO robot is able to fly and walk, the first bipedal robot able to do so [128]. Ascento is a compact wheeled bipedal robot that can jump to avoid obstacles [131]. These robots offer multi-modal locomotion capabilities but their design is neither optimized for legged or wheeled motion, making their implementations heavy and complex [48].

Implementing methods to endow bipedal humanoids with the skills to operate personal transporters (PTs), such as scooters, Segways (Segway Inc., New Hampshire) and electric skates is an option with several advantages. First, it keeps the humanoid's form-factor close to a human's and maintains bipedal walking capabilities. Secondly, it allows humanoids to conserve their energy while efficiently travelling longer distances.

Operating PTs with humanoids is challenging because it requires the ability to control

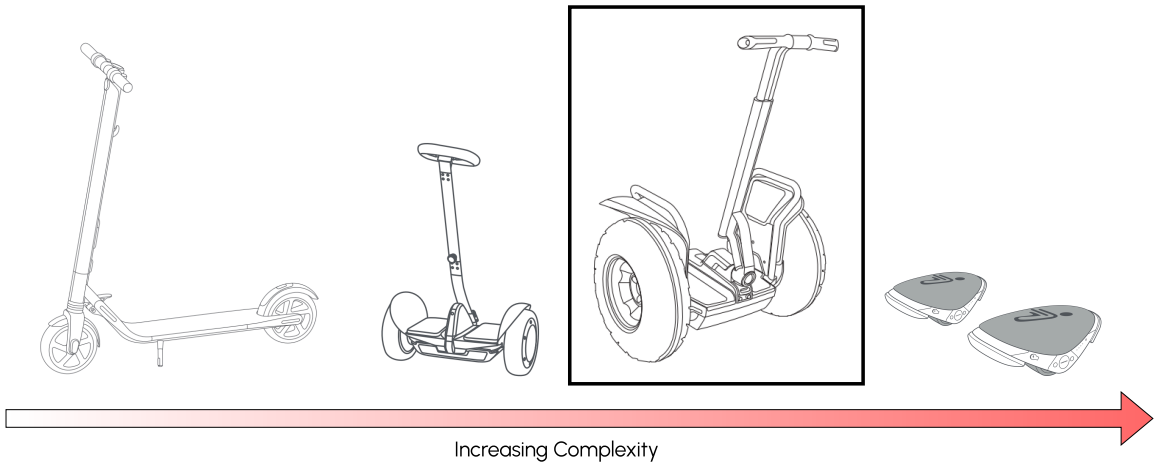


Figure 1.1: The personal transporters shown from left to right are: Ninebot Kickscooter E22 [5], Ninebot S-Max (mini self-balancing Segway) [6], Segway x2 SE [13], Segway Drift W1 Electric Skates [11]. The complexity in terms of user control increases from left to right along the arrowhead [48]. This is not an exhaustive list of personal transporters but covers some common designs. For this thesis, we consider the Segway x2 SE as the exemplar PT.

devices that have their own complex dynamics. However, like many other humanoid tasks, it can be broken down into subtasks such as environment recognition, state estimation, path planning and whole body motion computation [57]. To operate a PT, humanoids should be able to mount and dismount the PT, sit or stand on the PT stably and control its command inputs (e.g., handlebar for steering and pedals for linear velocity). Figure 1.1 shows example PTs from left to right in increasing complexity. In this thesis, we tackle the problem of riding a Segway x2 SE [13] as it offers sufficient complexity while also being practical for usage in factories, malls, airports and other large campuses. The other PTs shown are either too small for the humanoid platform used in this thesis or impractical for daily usage (Segway Drift W1 Electric Skates).

There has been limited previous work in this area. Xin et al. [243] tackle the problem of a humanoid riding a small two-wheeled mobile platform. The COMAN (COmpliant HuMANoid Platform) successfully balances on and drives a RoboSavvy self-balancing robotic platform [10] in simulation, showing that the COMAN can actively balance on the two-wheeled platform using quadratic optimization to generate whole-body torques. While the arms are taken into consideration for balancing, they are left unactuated as the specific two-wheeled platform used had no physical handle for steering. Gong et al. [90] successfully enabled the Cassie bipedal robot to balance and drive a Segway miniPro by modifying the nominal standing controller and shifting its centre of mass (CoM) for accelerating and turning. Hyungjik et al. [144] implement center of gravity control on miniature humanoid robots and show that they are able to ride a two-wheeled mobile platform. Similar to other prior work, the arms are used purely for balancing. While these prior works have successfully balanced and driven two-wheeled mobile platforms, all works employed lightweight robots, and used arms only for balance and not platform control.

The work in [129, 130] demonstrated a method for the HRP-2 humanoid to ride a self-balancing hoverboard. The authors devised separate controllers for riding and speed-governing of the hoverboard. This paper proposed the Sequential Online Learning Control (SOLC) method composed of the cascade connection of SGD-based open-loop Learning Control (SLC) and Mini-batch-based closed-loop Learning Control (MLC). The SLC contributes the damping gain online tuning for the foot torque to mount the hoverboard, and the MLC handles speed governing. The method is tested in real life with the HRP2-JSK humanoid.

The work in [48] proposes a method towards achieving multi-modal locomotion with the bipedal robot Cassie. The authors develop a framework for Cassie to ride a pair of hovershoes across various terrains. The framework allows Cassie to balance, regulate forward and rotational velocities, achieve fast turns and move over flat terrain, slopes, stairs and rough outdoor terrain. The method uses the built in sensors of Cassie along with a depth camera to perform visual SLAM for navigation in an obstacle course. The method augments the dynamic model of Cassie with a model of the hovershoes, has a tight perception + planning loop and uses simple PD control for balance. The method is demonstrated in simulation and in real world experiments.

While great efforts have been made in this domain, there are several shortcomings of existing literature. First, the methods proposed are highly handcrafted and tied to specific hardware. Dynamic models are augmented to include the PT dynamics and controllers are kept simple, raising questions on stability margins outside the examples shown. Second, other than the work in [129, 130], most works deal with relatively small bipedal robots as compared to a full humanoid model (which has challenging dynamics due to heavy limbs). Finally, most works so far have only considered the lower body for motion as none of the PTs tested had handlebars. In the next section, we outline our more general approach for the problem of operating PTs with (full-sized) humanoids.

1.1 Problem Definition and Approach

The objective of this research is to endow humanoids with the skills to operate PTs. More precisely, by "operate", we mean control the command inputs of the PT, realize desired linear and angular velocities and stably balance on the PT. Our approach is unique with respect to current literature in that we analyze how humans operate PTs qualitatively and use this to aid in controller design as pictured in Figure 1.2.

Our proposed approach relies solely on the sensors internal to the humanoid platform and Vicon motion capture for global base estimation, making it independent of the PT. Forces measured at the wrists and feet are used in the control methods to balance and operate steering mechanisms. The PT control system is not modified in any way for this research.

Some simplifying assumptions were introduced to capture the essential structure of the problem without losing generality:

1. The humanoid is already in its driving posture with hands on the handlebar. The grasp is predefined and unchanged for the duration of operation.

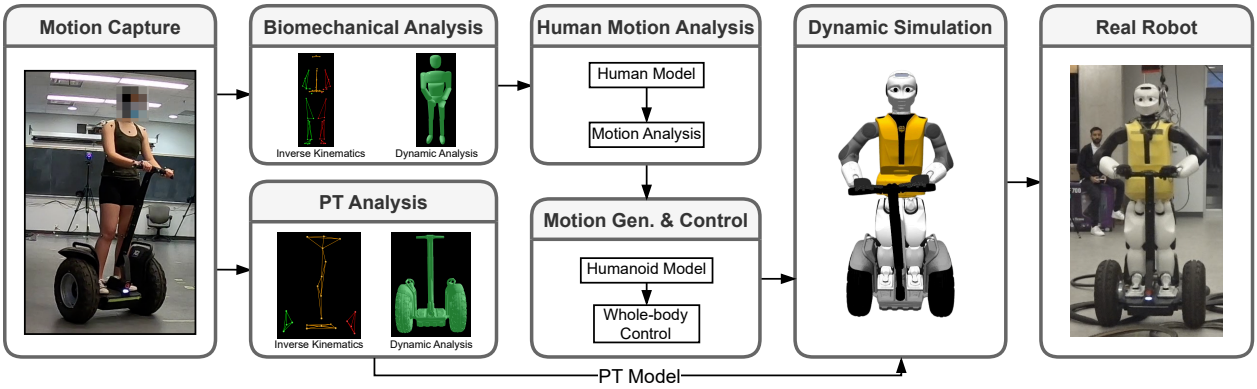


Figure 1.2: The high-level block diagram of the proposed approach to endow humanoids with the skills to operate PTs.

2. Friction is not taken into consideration for the PT's wheels and it is assumed that the wheels do not slip. However, friction is taken into consideration for the humanoid feet when standing on the PT.

Given these assumptions, we present our general framework to operate PTs with humanoids in Figure 1.2. The flow is from left to right and starts with human and PT motion capture experiments. The PT experiments are used to identify the control and dynamic structure of the PT while the human experiments aid in the identification of control and motion strategies for balance and handlebar manipulation. Once the PT is characterized, a dynamic model is constructed and tested in simulation. With the knowledge of how humans ride Segways, a whole-body controller is synthesized that takes into account feet contacts, hand contacts, bimanual constraints to operate the handlebar and whole-body stabilization through centre of mass motion. This whole-body controller is deployed onto the REEM-C humanoid to drive a Segway. The same pipeline can be used for various other PTs with small modifications to gains, and modelling. However, for this thesis, only the Segway is considered as an exemplar PT.

While developing the controller for the REEM-C to drive a Segway, various seemingly orthogonal works were investigated and make up the various chapters of this thesis. This is highlighted in the diagram shown in Figure 1.3. First, a high-fidelity simulation of the Segway was developed along with a basic controller to test the dynamic feasibility of a humanoid riding a Segway. Next, whole-body stabilization on dynamic platforms (balance boards) was investigated. This is arguably a harder problem than operating a PT because the platform shifts to steeper angles than a PT would. Next, a motion generation pipeline for bimanual manipulation was investigated. Various household items like brooms, boxes and rolling pins were used as test objects. This same pipeline was used to operate the LeanSteer handlebar of the Segway. All of these works were combined and used to enable the REEM-C humanoid to operate a Segway. We note that each of these individual works also spawned contributions of their own and did not only contribute to the research in Chapter 6. For example, during the development of the bimanual manipulation pipeline, a new metric for gauging the quality of bimanual grasps was developed.

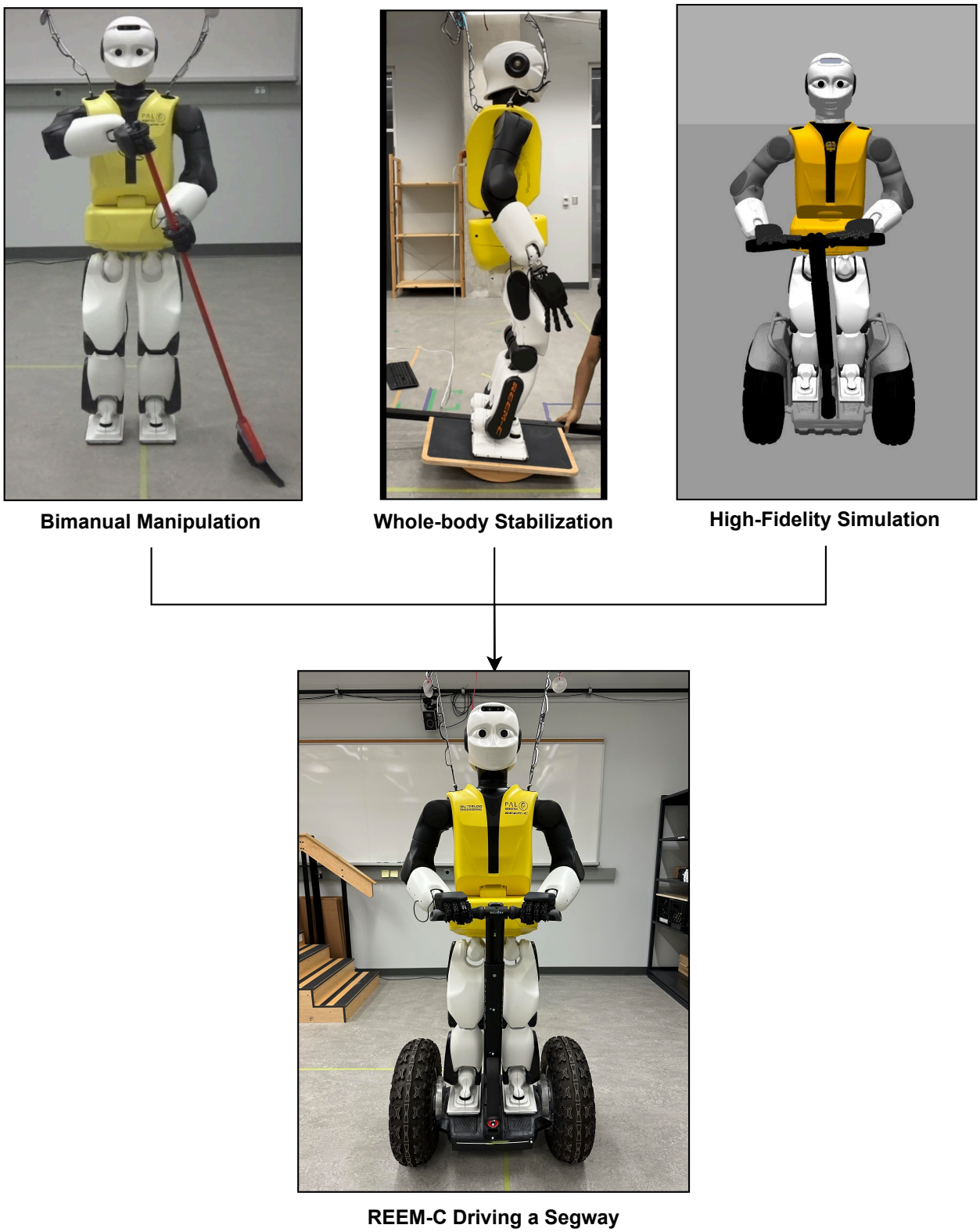


Figure 1.3: A high-level overview of the various topics covered in this thesis. They all ultimately contribute to the work in Chapter 6 with the REEM-C operating a Segway personal transporter in real hardware experiments.

1.2 REEM-C Humanoid Robot Platform

For this thesis, the humanoid platform, REEM-C, is used for all testing in simulation and real life. REEM-C is a bipedal humanoid [16] designed and manufactured by PAL Robotics. It has a height of 165 cm, weighs 80 kg and has 68 degrees of freedom (DoF), 36 actuatable and all revolute. The DoF summary is given in Table 1.1. The humanoid has the ability to stand on 1 foot and carry a payload of 1 kg in each hand separately, or 10 kg together. The hands are tendon driven with 3 actuators controlling 19 degrees of freedom. This allows the hands to grasp cylindrical objects (like handlebars) easily. For sensors, the robot has force-torque sensors at the ankles and wrists, lasers on each foot and an IMU in its torso. All joints have encoders, and the robot is position-controlled. It is also fully integrated with ROS (Robot Operating System) making it an easy-to-use platform for research. The robot is shown in Figure 1.4 and its kinematic structure in Figure 1.5. The REEM-C dynamic model is described in URDF format. ROS, Gazebo and rviz was used to simulate the dynamics of the REEM-C.

Table 1.1: REEM-C degrees of freedom in each body part.

Body Part	DoF
Head	2
Arm	7
Hand	19 (3 actuatable)
Pelvis (Waist)	2
Leg	6

The mass distribution of the REEM-C is summarized in Table 1.2. For an average-sized human, the torso and pelvis account for 45% of the total mass, and a single leg is usually 17% [241]. From Table 1.2, we see that for the REEM-C, the distribution is 37% for the torso and pelvis, and 18% for a single leg. While close, this is slightly different than an averaged sized-human. When standing upright, the CoM of the REEM-C is roughly at the center of the pelvis, about 85 cm off the ground.

Table 1.2: REEM-C mass distribution.

Body Part	Mass (kg)	% of Total Mass
Head	1.57	2%
Torso	12.76	19%
Arm (+hand)	8.37	12%
Pelvis	12.32	18%
Leg (+foot)	12.12	18%

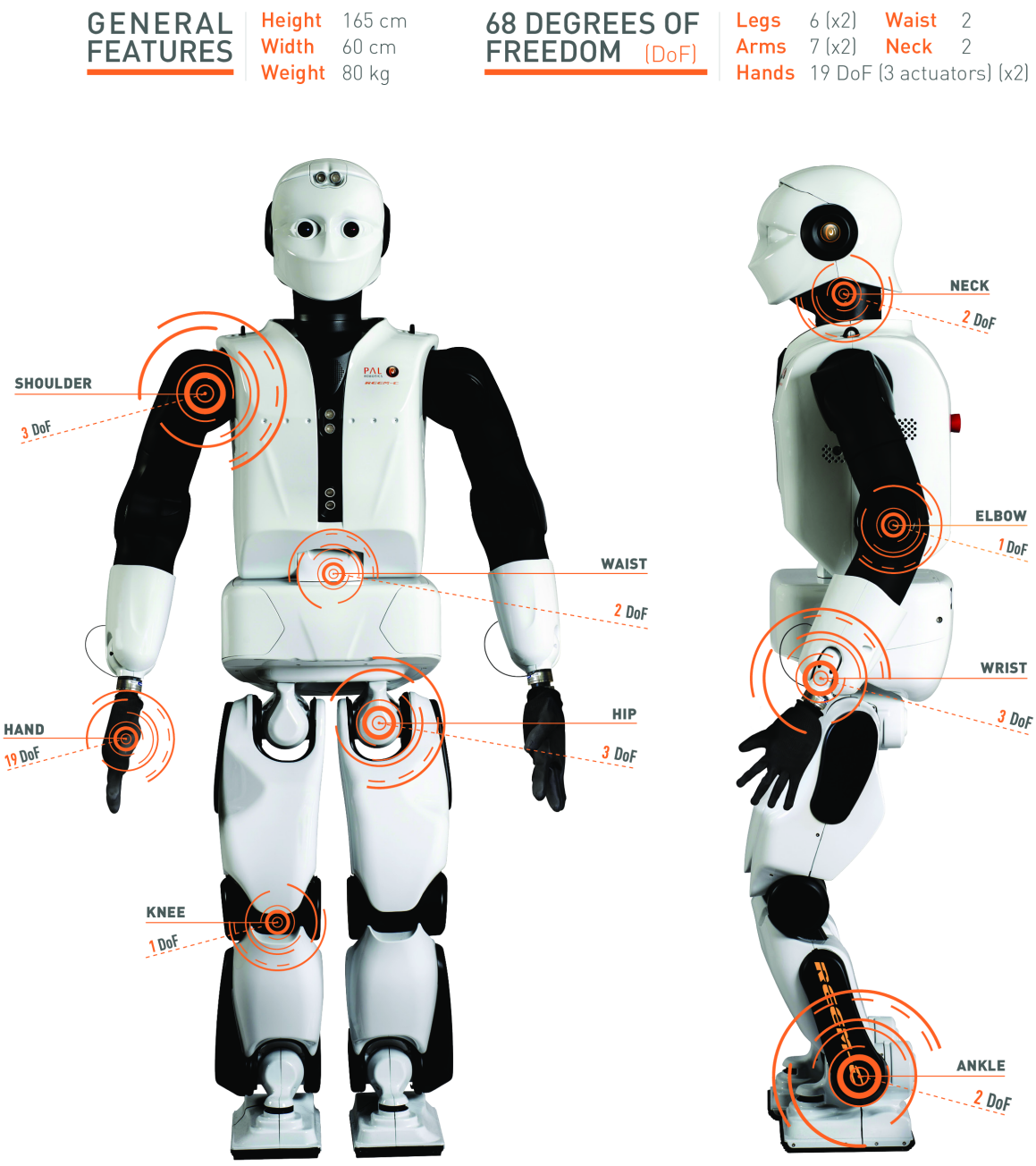
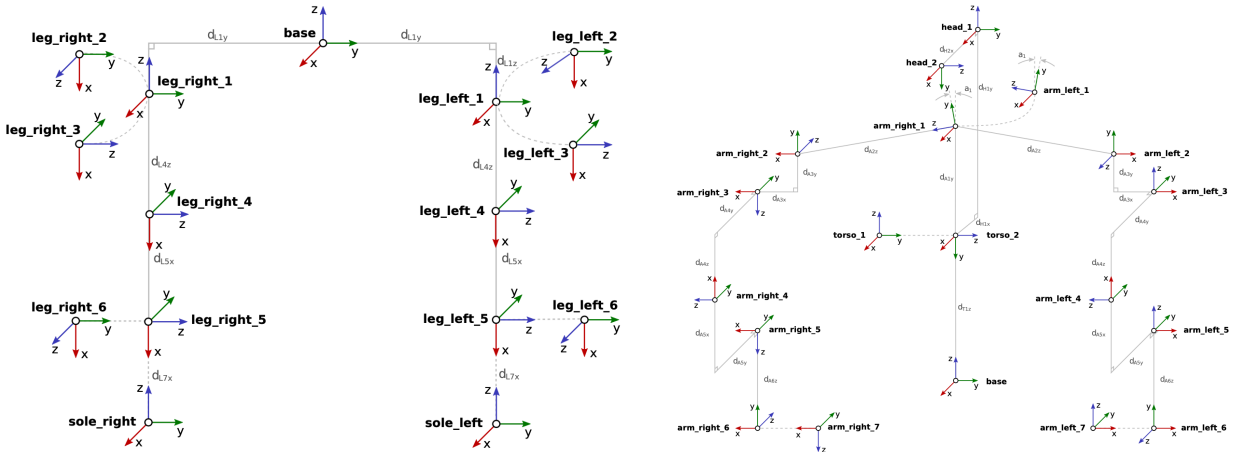


Figure 1.4: The REEM-C humanoid with specifications and degrees of freedom labelled [9].



(a) Lower body kinematic structure.

(b) Upper body kinematic structure.

Figure 1.5: REEM-C kinematic structure.

1.3 Thesis Contributions

In this section, we highlight the various contributions of this thesis.

1. Relating to PTs
   - *Control:* the internal controllers of the Segway x2 SE PT were reverse engineered through human motion capture in order to develop a high-fidelity simulation model.
2. Relating to Humanoids
   - *Bimanual Manipulation*: a novel manipulability-stability metric is developed for humanoid robots. This metric is used to perform a workspace analysis for humanoids manipulating everyday objects. The metric was also used in a motion generation pipeline to plan trajectories to manipulate household objects. This pipeline was used to operate the Segway handlebar.
   - *Whole-body stabilization*: an integration of various existing methods from literature along with novel controller components aided in developing a general whole-body stabilization method for humanoids. The whole-body controller developed through this investigation was used to stabilize the humanoid while driving a Segway.
   - *Real-world Experiments*: rigorous experiments were performed with the REEM-C humanoid for each chapter in this thesis. Ultimately, this culminated into the experiments done in the final chapter with the REEM-C successfully operating a Segway in various test scenarios. This showed, to our knowledge, the first demonstration of a full-sized humanoid operating a Segway with a handlebar.

1.4 Thesis Organization

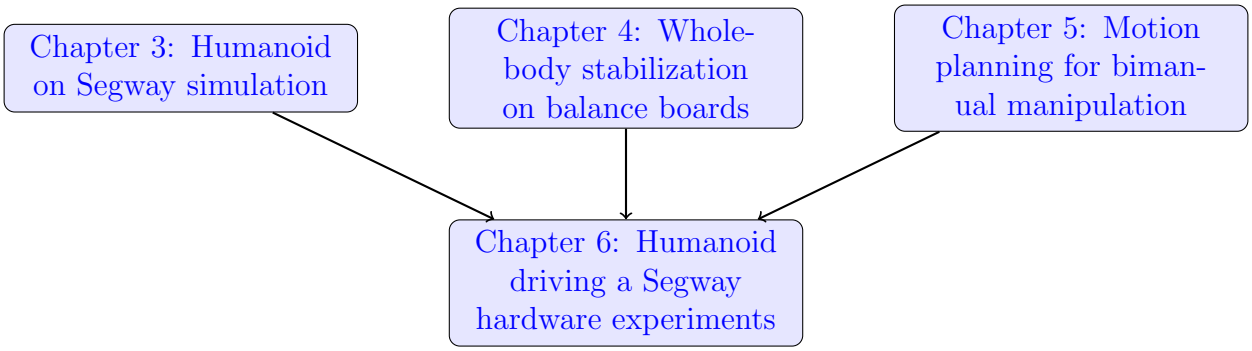


Figure 1.6: Overview of the chapters in this thesis. The research in Chapter 3-5 contribute to Chapter 6.

The rest of this thesis is organized as follows. Chapter 2 provides a literature review of the key subtopics that are explored in this thesis along with theoretical background. Chapter 3 details a method to model PTs to develop a high-fidelity simulation. An initial basic controller is tested in simulation that allows the REEM-C to drive a Segway. After initial testing, it was realized that more sophisticated controllers would be needed to stably operate PTs with humanoids. To this end, Chapter 4 investigates whole-body stabilization of humanoids on actuated balance boards. Chapter 5 investigates bimanual manipulation for humanoid robots and proposes a novel manipulability-stability metric and motion generation pipeline. Chapter 6 brings together the work in all previous chapters and shows extensive real hardware experiments of a humanoid driving a Segway. In Chapter 7 we conclude and provide outlook on future work. Each chapter has an associated video that explains the method and experiments presented. They are listed in Table 1.3.

	Link
Chapter 3	https://youtu.be/cVHbf4fwzcs
Chapter 4	https://youtu.be/X61gbXRYg50
Chapter 5	https://youtu.be/dALPqitEdOA
Chapter 6	https://youtu.be/eZxXSlk1v0w

Table 1.3: The associated videos for each chapter. These videos detail the methods and experiments conducted and serve as a quick summary of the work.

Chapter 2

Literature Review and Background Theory

In this chapter, we provide a brief literature review along with necessary background theory relevant to this thesis.

2.1 Current State of Humanoid Robots

The humanoid robotics landscape has seen significant growth since 2020 (the start of my PhD). In 2020, the field of humanoids was still mainly research focused, with robots like the Boston Dynamics (hydraulic) Atlas, Kawada HRP-2, DLR ToRo, PAL TALOS, PAL REEM-C and Honda ASIMO leading the state of the art. Fast forward four years and the landscape has changed dramatically. There is a large commercial interest in the humanoid robotics space with new companies debuting humanoids every month. A few are pictured in Figure 2.1 from companies such as Figure AI, Apptronik, Agility Robotics, Unitree and many more. Tesla has also entered the race with their Optimus robot, performing various tasks in Tesla factories [221]. NVIDIA is also playing a key role, enabling large foundation models to be trained on their hardware. These robots are expected to become part of the labour force and assist with household tasks. Time will tell if this vision becomes reality.

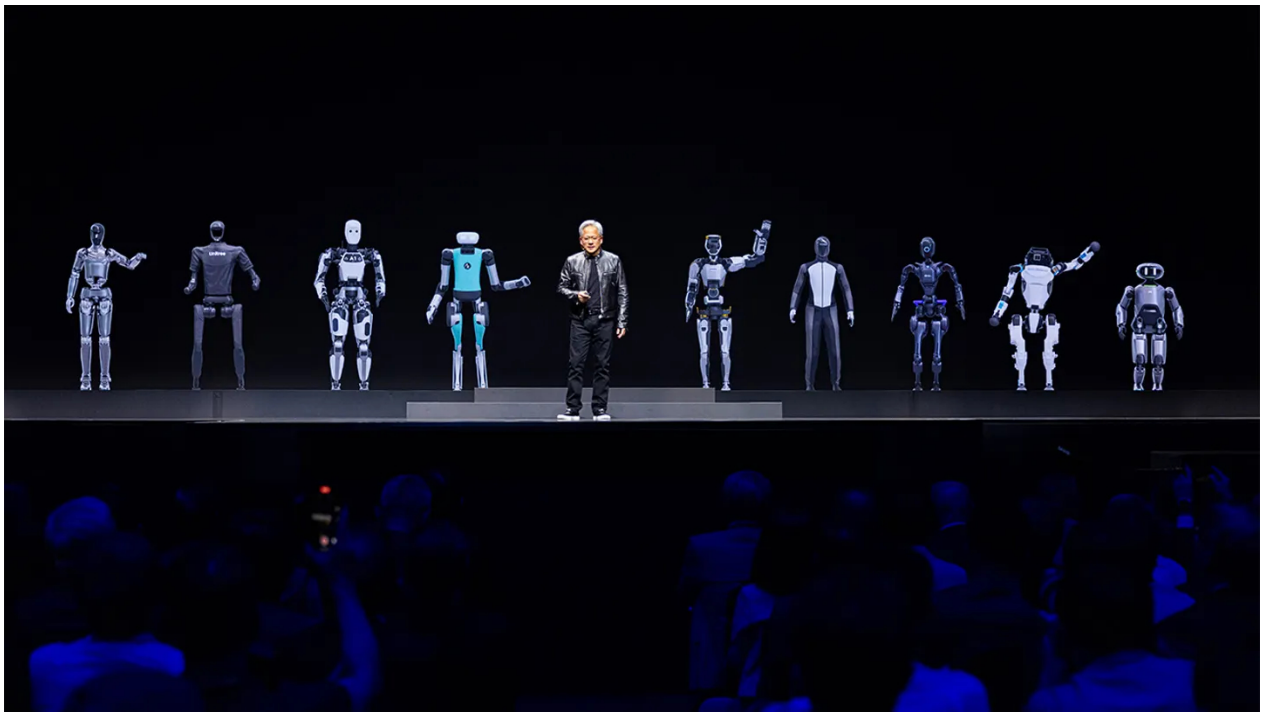


Figure 2.1: NVIDIA CEO Jensen Huang introducing Project GR00T, a general-purpose foundation model for humanoid robots [105]. Several commercial and research humanoids are pictured in the background.

2.2 Humanoid Modelling

To begin the technical discourse, in this section we summarize humanoid whole-body dynamics and centroidal dynamics. The modelling here is used throughout this thesis.

2.2.1 Whole-body Dynamics

A humanoid robot is modeled as a floating base system with n_q joints and $n_q + 6$ degrees of freedom, accounting for both the joints and the 6 generalized coordinates of the floating base. The configuration vector $\boldsymbol{q} \in \mathbb{R}^{n_q+6}$ includes the joint positions as well as the floating base's position and orientation. The robot's dynamics can be described by:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{S}^T\boldsymbol{\tau} + \boldsymbol{\tau}_g(\boldsymbol{q}) + \sum_{k=1}^{n_c}\boldsymbol{J}_k^T\boldsymbol{w}_k, \tag{2.1}$$

$$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{2.2}$$

where:

- $\boldsymbol{S} \in \mathbb{R}^{n_q\times(n_q+6)}$ is the selection matrix for actuated joints,

- $\boldsymbol{\tau} \in \mathbb{R}^{n_q}$ represents the torque input vector,
- $\boldsymbol{\tau}_g \in \mathbb{R}^{n_q+6}$ is the gravitational term,
- $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{(n_q+6)\times(n_q+6)}$ is the mass matrix,
- $\boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{(n_q+6)\times(n_q+6)}$ accounts for centrifugal and Coriolis forces,
- $\boldsymbol{J} = [\boldsymbol{J}_1^\top, \ldots, \boldsymbol{J}_k^\top]^\top \in \mathbb{R}^{6n_c\times(n_q+6)}$ is the contact Jacobian, and
- $\boldsymbol{w}_k = (\boldsymbol{f}_k, \boldsymbol{\tau}_k) \in \mathbb{R}^6$ denotes the wrench at the k^{th} contact point.

For a humanoid standing on two feet, $n_c = 2$ (this is the case for Chapter 4) and for a humanoid carrying or manipulating an object with both hands, $n_c = 4$ (this is the case for Chapter 3, Chapter 5 and Chapter 6). Equation 2.2 arises from the no-sliding constraint at the contact points, assuming unilateral ground contact forces constrained by friction cones, described as:

$$\|\boldsymbol{f}_i^{x,y}\| \leq \mu_0 \boldsymbol{f}_i^z, \quad \forall \boldsymbol{f}_i, \tag{2.3}$$

with $\mu_0 > 0$ as the coefficient of friction.

Additionally, when the hands of the humanoid are in contact with an object, they are not treated as fixed contacts in the world frame; they move with the robot and are bilateral, meaning they cannot detach and the object being carried or manipulated is essentially considered attached to the robot.

2.2.2 Centroidal Dynamics

Centroidal dynamics describe the underactuated part of the robot's whole-body dynamics. Using the Newton-Euler equations, they relate changes in centroidal momentum to contact forces [178]:

$$\begin{aligned} m\ddot{\boldsymbol{c}} &= \sum_i \boldsymbol{f}_i + m\boldsymbol{g} = \dot{\boldsymbol{l}}_c, \\ \dot{\boldsymbol{h}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \sum_i (\boldsymbol{p}_i - \boldsymbol{c}_i) \times \boldsymbol{f}_i + \boldsymbol{\tau}_i = \dot{\boldsymbol{k}}_c, \end{aligned} \tag{2.4}$$

where $\boldsymbol{c}$ is the center of mass (CoM) position, m is the robot's mass, $\boldsymbol{f}_i \in \mathbb{R}^3$ is the contact force at point i, $\boldsymbol{p}_i \in \mathbb{R}^3$ is the position vector, and $\boldsymbol{\tau}_i \in \mathbb{R}^3$ is the contact torque. Linear and angular momentum about the CoM are denoted as $\boldsymbol{l}_c$ and $\boldsymbol{k}_c \in \mathbb{R}^3$, respectively, while $\boldsymbol{g} \in \mathbb{R}^3$ represents gravitational acceleration.

The centroidal angular momentum is given by [178]:

$$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.5}$$

where $\boldsymbol{A} \in \mathbb{R}^{3\times(n_q+6)}$ is the centroidal momentum matrix.

2.3 Whole-body Control

In this section, we detail whole-body control techniques that are popularly used on humanoid robots before detailing the one specifically used in this thesis.

2.3.1 Whole-body Controllers from Literature

Whole-body controllers often take the form of quadratic programs (QPs) that are solved in real-time at high frequencies (100Hz to 1kHz). They are said to be instantaneous since they find a control input $\boldsymbol{u} \in \mathbb{R}^n$ (where n is the number of actuators) to apply at the next controller timestep with no horizon, using an optimization method. Typically, $\boldsymbol{u}$ will be velocity, acceleration or torques (the free variables of the optimization). Forces are also typically optimized in whole-body controllers. For the case of a velocity or acceleration input, the robot is usually position-controlled (such as the PAL REEM-C) and the control generated by the optimization needs to be integrated before being sent to the low-level joint position controllers (which are usually PID controllers).

Many research humanoids have used (and still use) whole-body control as the final layer in the control pipeline even if more sophisticated controllers precede them. A popular framework that has been adopted by the community is the stack-of-tasks approach [161]. There are many variants of this approach but it is essentially a control approach for humanoids that prioritizes multiple simultaneous tasks by organizing them hierarchically, ensuring that higher-priority tasks (e.g., balance) are achieved first, while lower-priority tasks (e.g., arm movement) are executed without disrupting higher-priority objectives. It uses optimization techniques to manage constraints, enabling coordinated and safe motion in dynamic environments. For many scenarios, whole-body controllers are enough to perform complex maneuvers, manipulation and locomotion (running, walking). In the work presented in this thesis, we solely make use of a whole-body controller.

TORO, developed by DLR, is a humanoid robot operated in torque control [74]. Over the years, much work has been presented using TORO, including walking, balancing, multi-contact scenarios [99], centroidal angular momentum generation for balancing [208] and much more. Central to all of these works is a passivity based whole-body controller implementation [72].

In [42], the authors develop a whole-body admittance controller that allows the HRP-4 robot to climb stairs in an Airbus manufacturing plant. The controller has a strict hierarchy of tasks. CoM admittance, foot force difference control, and feet wrench distribution are a few of the tasks implemented to stabilize the system.

2.3.2 Whole-body Control Formulation

The whole-body controller implemented for this work is a weighted Quadratic Program (QP), implemented with the TSID (Task Space Inverse Dynamics) library [3]. It sums task functions into a general cost function using user-defined weights for their priorities. The formulation has two layers, costs and constraints, whose trade-off can influence controller performance.

Formulation of task functions

Adapting the formulation from [203], the task errors, $\boldsymbol{e} \in \mathbb{R}^l$ (with l being the dimension of the task error), in this controller formulation are implemented as acceleration-based tracking laws in task space and are expressed as follows:

$$\begin{aligned} \ddot{\boldsymbol{e}}(\boldsymbol{q}, t) &= \ddot{\boldsymbol{x}}(\boldsymbol{q}) - \ddot{\boldsymbol{x}}^*(t) \\ \ddot{\boldsymbol{e}}(\boldsymbol{q}, t) &= (\boldsymbol{J}\ddot{(\boldsymbol{q})} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) - \ddot{\boldsymbol{x}}^*(t) \end{aligned} \tag{2.6}$$

where $\boldsymbol{x}^*$, $\ddot{\boldsymbol{x}}^*$ are the desired position and acceleration for the task, and $\boldsymbol{J}$ is the Jacobian.

The dynamics imposed on these task errors are as follows:

$$\begin{aligned} \ddot{\boldsymbol{e}}(\boldsymbol{q}, t) &= \boldsymbol{K}_P \left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}^*\right) + \boldsymbol{K}_D \left(\dot{\boldsymbol{x}}(\boldsymbol{q}) - \dot{\boldsymbol{x}}^*(t)\right) \\ \Leftrightarrow \quad \ddot{\boldsymbol{x}}(\boldsymbol{q}) &= \ddot{\boldsymbol{x}}^*(t) + \boldsymbol{K}_P \left(\boldsymbol{x}(\boldsymbol{q}) - \boldsymbol{x}^*(t)\right) + \boldsymbol{K}_D \left(\dot{\boldsymbol{x}}(\boldsymbol{q}) - \dot{\boldsymbol{x}}^*(t)\right) \end{aligned} \tag{2.7}$$

with $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ being the task gains.

Using Equation 2.6 and Equation 2.7, we can write a generic formulation for a task-function as follows:

$$\underbrace{\boldsymbol{J}}_{\boldsymbol{O}} \underbrace{\ddot{\boldsymbol{q}}}_{\boldsymbol{y}} = \underbrace{\ddot{\boldsymbol{x}}^* - \boldsymbol{J}\dot{\boldsymbol{q}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_D\dot{\boldsymbol{e}}}_{\boldsymbol{o}} \tag{2.8}$$

Formulation of contacts

When solving the inverse dynamics for the robot, we assume that it is in rigid contact with the environment. Therefore, these contacts constrain the motion. They are implemented as nonlinear functions, differentiated twice, leading to zero contact point acceleration as follows:

$$\begin{aligned} c(\boldsymbol{q}) = 0 \quad &\Leftrightarrow \quad \text{Contact point does not move} \\ \boldsymbol{J}_c^T\dot{\boldsymbol{q}} = 0 \quad &\Leftrightarrow \quad \text{Contact point velocity is zero} \\ \boldsymbol{J}_c^T\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} = 0 \quad &\Leftrightarrow \quad \text{Contact point acceleration is zero} \end{aligned} \tag{2.9}$$

In this work we use a unilateral rectangular plane contact for the feet, which is defined as a 6D motion task, as in Equation 2.8. This is an inequality on the force bounds and the QP is allowed to freely optimize the forces at the contact.

Formulation of quadratic program

As mentioned previously, TSID sums all task functions using user defined scalar weights λ_i for all $i \in N$ tasks. The constraints are the dynamics and rigid contacts. The formulation is given as follows:

$$
\begin{aligned}
\min_{\boldsymbol{y}=[\ddot{\boldsymbol{q}},\boldsymbol{f}]} \quad & \sum_{i=0}^{N} \lambda_i \left\| \boldsymbol{O}_i \boldsymbol{y} - \boldsymbol{o}_i \right\|^2 \\
\text{s.t.} \quad & \textit{Equation } 2.1, 2.2, 2.3 \\
& \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \\
& \boldsymbol{q}_j^{\min} \leq \boldsymbol{q}_j \leq \boldsymbol{q}_j^{\max} \\
& \dot{\boldsymbol{q}}_j^{\min} \leq \dot{\boldsymbol{q}}_j \leq \dot{\boldsymbol{q}}_j^{\max} \\
& \boldsymbol{f}_{\min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{\max}
\end{aligned}
\tag{2.10}
$$

The free variables that the QP solves for are the acceleration, $\ddot{\boldsymbol{q}}$ and the forces, $\boldsymbol{f}$. Of note, the QP has two strict layers, constraints and costs. The next section details all of the tasks implemented for this work and highlights whether they are costs or constraints.

2.4 Model-based Motion Generation for Humanoids

Humanoid robots have been around for a few decades with a major research focus on balancing and dynamical walking. Robots like ASIMO [205], HRP-2 [101], ToRo [180], Atlas [1] and REEM-C [16] have been used in research capacities and many walking, balancing and whole-body controllers have been developed. However, even though these robots have proven they can walk, fast and efficient motion is still an open topic for research. These robots are relatively slow as compared to humans (save for Atlas) and their motions in general could be more human-like to assimilate better into human environments. Along this line of research, Cheng et al. explored more human-like sensing and motion control with the CB (computational brain) humanoid, developed in collaboration with SARCOS [50]. The work presented a whole-body compliance controller, perception and visual ocular-motor responses close to those of humans. In a similar vain, this thesis aims to make humanoids more human-like by enabling them to use PTs designed for humans.

Figure 2.2 shows an overview of various modelling techniques that are used to simplify control. Since humanoid robots have many degrees of freedom, optimization-based control can be very computationally expensive. Reduced order models, as will be discussed in the rest of this section, offer a tradeoff between accurate dynamics and efficient computation.

Existing motion generation methods can be classified into three broad categories: motion remapping, reduced-order model-based methods and optimization-based methods [104]. A quick summary and non-exhaustive list of works in these three categories is presented in Table 2.1. As a note, we mainly consider classical model-based approaches in this section. A summary of reinforcement learning (RL) is provided in Section 2.9.

Of these three methods, model-based is the most popular due to the ease of implementation and real-time control capabilities (especially with reduced models). However, in this thesis, going by the definitions provided here, the approach taken is a hybrid of model-based and optimization-based. In order to make the optimization problems more tractable, we also use a reduced model specific for the task we want to accomplish (operating PTs). As

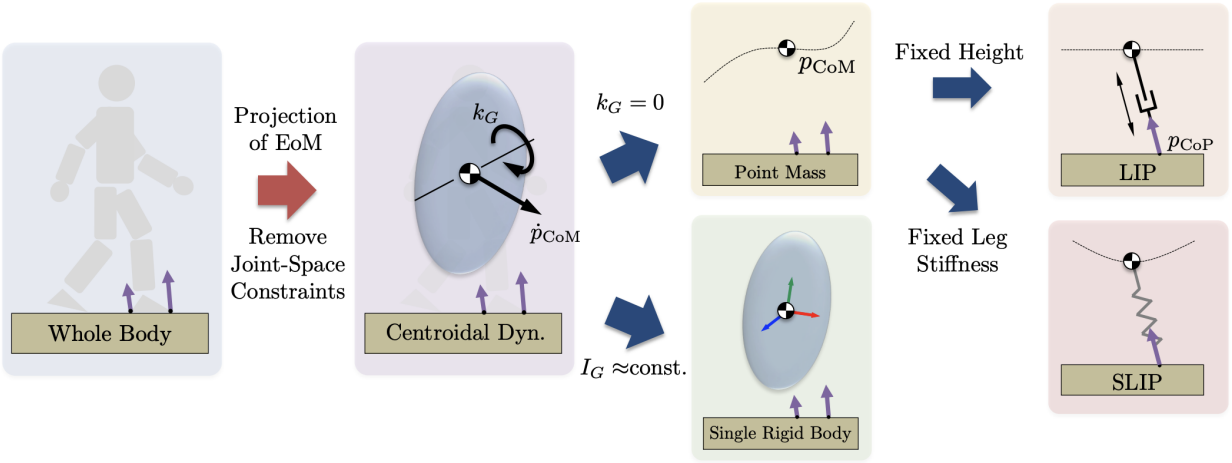


Figure 2.2: An overview of various legged robot models used for optimization-based control. Adapted from [239].

Method and Works	Definition
Motion Remapping [36, 171]	Mapping trajectories from human motion to humanoids (directly). Special consideration needs to be made to scale the motions and augment them with physical constraints and stability criteria before transferring to the humanoid.
Reduced-order Model-based Methods [120, 123, 86, 158, 190, 73]	Reduced or whole-body models are used to generate motions based on stability criteria such as ZMP and capture point theory.
Optimization-based Methods [166, 57, 132, 55]	Optimal control is used to learn optimal human motions. This can be from demonstration data (IOC) or from technical specification of an objective function. While computationally expensive, especially for whole-body models, a major advantage of this method is that feasibility constraints for the specific humanoid are taken into account during optimization.

such, the next subsection details the literature on reduced template models and their usage in humanoid control.

2.4.1 Template Models

While working with full dynamic models of humanoids would be ideal and allow for accurate sim to real translation, it is generally computationally expensive due to the number of degrees of freedom in a humanoid. An alternative modelling approach is to form what are called

"template" models [82]. These models usually abstract away the large number of degrees of freedom in favor of keeping the most essential parts for dynamic motion.

Practically in control, using the template model usually has two layers [158]. First, a proper template model is developed and its trajectories are regulated through some control strategies or optimization techniques. In the second layer, the template model trajectories are "tracked" by the full humanoid model. As an example, template models will often produce CoM and ZMP trajectories which are then tracked by the full humanoid through inverse kinematics and inverse dynamics routines at an actuator level.

We will now discuss several template models from the literature.

Linear Inverted Pendulum Model

The Linear Inverted Pendulum Model (LIPM) is a well-known and widely used template model for walking. The model was first proposed in 2D [121] before being extended to 3D [120]. In this model, the CoM movement is constrained to a horizontal plane allowing for a simplification and relation between the position of the Center of Pressure (CoP) and the CoM. Defining $[p_x, p_y]$ as the position of the CoP and $[x_c, y_c, z_c]$ as the position of the CoM (and noting that z_c is constant), the relation between the CoP and CoM is given as:

$$p_x = x_c - \frac{z_c}{g}\ddot{x}_c$$
$$p_y = y_c - \frac{z_c}{g}\ddot{y}_c$$

The LIPM has been used successfully for many dynamic walking tasks. Kajita et al. proposed ZMP Preview Control [118] based on Model Predictive Control [126, 164]. The work proposed a walking pattern generator that took into account future ZMP locations and produced stable CoM trajectories by solving an optimal control problem that minimizes ZMP tracking error and control effort while also maximizing CoM trajectory smoothness.

Several improvements have been made to ZMP Preview Control including an efficient quadratic programming (QP) [240] formulation to better handle disturbances, adjustment of the optimization weights, costs and variables to realize adaptive foot positioning and automatic footstep placement [69, 100] and loosening constraints on the ZMP to allow it to deviate from the reference trajectory on uneven ground [123].

Capture point theory is another formulation that uses the LIPM. Originally proposed for push recovery [190], the capture point is the point on the ground that the robot can step to in order to come to a complete stop. For a given initial CoM state of the LIPM, in 2D there is a unique capture point but not in the 3D case. In 3D, the capture point moves over time, making it *instantaneous*. Several works have exploited this property and formulated walking methods/controllers based on the instantaneous capture point, later coined Divergent Component of Motion (DCM) [189, 219, 102, 73, 103, 192].

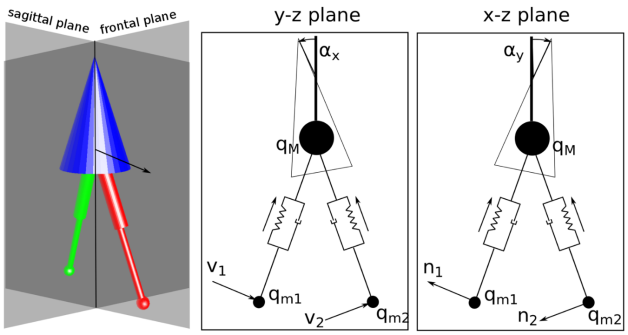


Figure 2.3: 3D SEA walker [57].

Spring Loaded Inverted Pendulum Model

The Spring Loaded Inverted Pendulum (SLIP) model was first introduced in [82]. The SLIP model assumes a linear spring during stance with a rest length equal to the touchdown length of the spring. When the leg spring reaches its rest length, stance terminates and is followed by ballistic CoM flight dynamics [238]. The model uses a point mass with massless legs that can arbitrarily reposition in flight, in preparation for touchdown. The SLIP model has been used as a motion template for running and hopping in bipedal robots [108, 238].

Dual Spring Loaded Inverted Pendulum Model

The Dual Spring Loaded Inverted Pendulum (Dual-SLIP) was introduced in [86]. The authors found that this model could better reproduce the double-peak ground reaction force pattern and CoM vertical and horizontal oscillations while walking. Since it is bipedal in nature, it can more readily address the double support phase in gait with no extra hand-designed mechanism [158]. There are a few works that develop walking controllers using the Dual-SLIP template, most in 2D [235, 85], and recently in 3D [158].

3D Serial Elastic Actuator Walker

The 3D Serial Elastic Actuator (SEA) walker is a recent template model proposed in [58, 57]. The model has an upper body, two actuated legs and point masses for feet. Actuation is done through SEAs in the legs and torques between the upper trunk and legs. The model is a suitable candidate to reproduce and predict realistic 3D CoM and swing foot trajectories. The template model is shown in Figure 2.3.

The 3D SEA walker was used to demonstrate a computational chain going from human motion capture experiments to experiments on real robots [57]. The template was successfully adapted for use with the HeiCub bipedal robot, and the HRP-2.

2.5 Floating Base Estimation

An important quantity to estimate for humanoids is the position and orientation of the floating base (typically close to the pelvis of the robot). In this section, we present two methods for floating base estimation, an anchor point method [41] which was used for stair climbing in [42] and another based on an EKF [14]. The EKF version is implemented with the SEROW library and is extended with ground truth Vicon data to estimate the floating base position and velocity of a humanoid (in double support) driving on a Segway. Since the robot is moving in freespace without making and breaking contact with the environment, the anchor point method does not suffice for this scenario. However, the anchor point method is useful for estimating the floating base location with respect to the PT being driven.

2.5.1 Anchor Point Method

The formulation presented here is adapted from [41].

Orientation Estimation

We assume that our legged robot is equipped with an inertial measurement unit (IMU) in its pelvis link, providing measurements of angular velocity $\boldsymbol{\omega}_k$ and linear acceleration $\boldsymbol{a}_k$. The IMU's orientation $\boldsymbol{R}_k$ and position $\boldsymbol{p}_k$ at time step k evolve according to:

$$\dot{\boldsymbol{R}}(t) = \boldsymbol{R}(t)[\boldsymbol{\omega}(t)\times] \tag{2.11}$$

where:

- $\boldsymbol{R}(t)$ is the rotation matrix representing the orientation of the floating base.
- $\boldsymbol{\omega}(t)$ is the angular velocity vector.
- $[\boldsymbol{\omega}(t)\times]$ is the skew-symmetric matrix of the angular velocity vector.

The frames of the robot are defined like so:

- B: The body frame, which is fixed to the robot's floating base and moves with it (the IMU is mounted here).
- W: The world frame, which is the inertial frame of reference.
- A: The anchor frame, representing a fixed contact point on the ground.

The orientation estimation relies on integrating angular velocity measurements, while translation estimation uses linear acceleration and known contact points. This approach

estimates both orientation and translation of the floating base frame in relation to the inertial frame.

To update the rotation matrix $\boldsymbol{R}_{k+1}$ at time t_{k+1} from the prior time t_k, we use angular velocity integration:

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k \exp([\boldsymbol{\omega}_k \times] \Delta t) \tag{2.12}$$

where:

- $\Delta t = t_{k+1} - t_k$
- $[\boldsymbol{\omega}_k \times]$ is the skew-symmetric matrix of the angular velocity vector.

The matrix exponential can be approximated using Rodrigues' rotation formula:

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k \left(\boldsymbol{I}_3 + \sin(\theta_k)[\boldsymbol{\omega}_k \times] + (1 - \cos(\theta_k))[\boldsymbol{\omega}_k \times]^2 \right) \tag{2.13}$$

where $\theta_k = \|\boldsymbol{\omega}_k\| \Delta t$.

Gyroscope-based integration of angular velocity tends to drift over time, so accelerometers are used to estimate orientation based on the gravity vector, which helps correct for roll and pitch. However, this approach cannot estimate yaw. A combined solution using a Kalman filter mitigates both short-term noise and long-term drift.

Translation Estimation

The translation estimation uses an anchor point A that remains in contact with the ground. The translation $\boldsymbol{p}_B^W$ of the floating base in the world frame is calculated as:

$$\boldsymbol{p}_B^W = \boldsymbol{p}_A^W - \boldsymbol{R}_B^W \boldsymbol{p}_A^B \tag{2.14}$$

where:

- $\boldsymbol{p}_A^W$ is the anchor point position in the world frame.
- $\boldsymbol{R}_B^W$ is the rotation matrix from the floating base to the world frame.
- $\boldsymbol{p}_A^B$ is the anchor point's position in the floating base frame.

The anchor-point method effectively estimates translation by assuming a fixed contact point, updated continuously during double-support phases to avoid discontinuities. This simple approach performs well in practice.

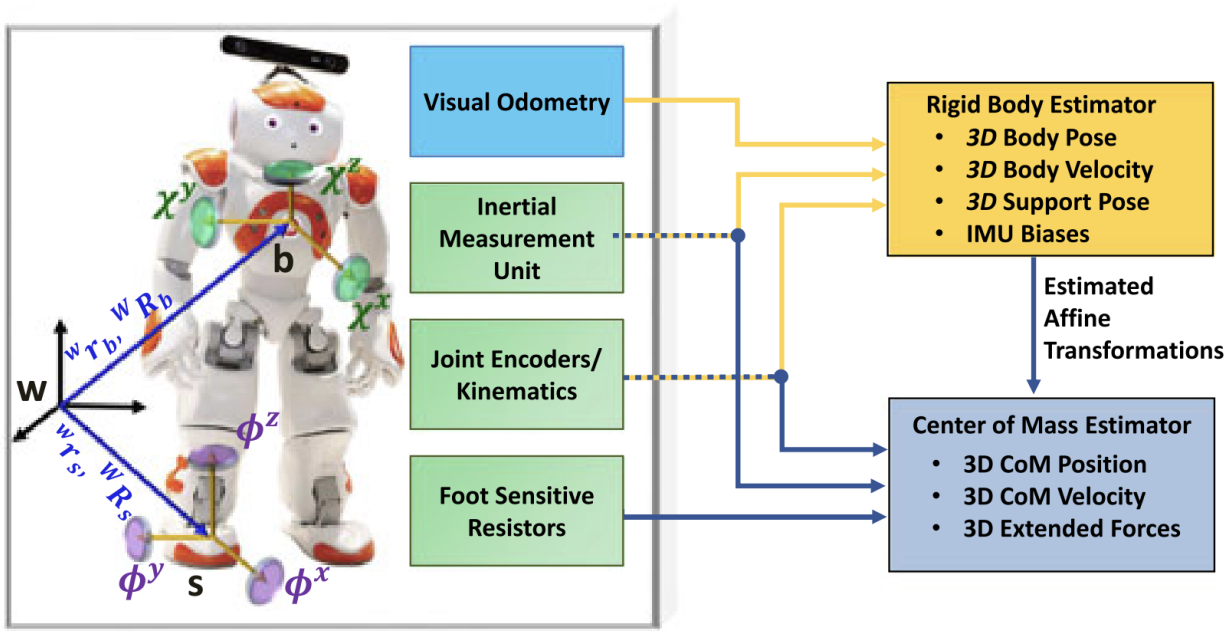


Figure 2.4: The SEROW state estimation method presented in [184]. This method is extended for this thesis.

2.5.2 Extended Kalman Filter Based Estimation

The work in [184, 14] introduces a cascade state estimation framework, SEROW, for estimating the 3D center of mass (CoM) of walking humanoid robots (detailed in Figure 2.4). SEROW fuses joint encoder, inertial, feet pressure, and visual odometry data, incorporating Newton-Euler dynamics to derive a nonlinear CoM estimator that handles uneven terrain and body angular momentum. It extends a floating mass estimator to account for support foot pose, forming a comprehensive cascade estimation scheme. Evaluations, both simulated and on a real NAO robot walking outdoors, demonstrate the framework's accuracy and robustness. The implementation is provided as an open-source ROS/C++ package and used in this thesis. Vicon motion capture data is fed into this estimator as ground truth data to retrieve accurate estimation of the base position and velocity while the robot operates a PT.

2.6 Inverse Optimal Control

Human motion is hypothesized to be optimal. Humans have evolved and learned to produce motor commands that optimize their behaviour with respect to a set of well-defined but unknown criteria. In the literature, these criteria have been developed into composite cost functions that can be grouped into task-related and body-related sub-costs [32]. Producing trajectories by optimizing against these cost functions has been shown to closely resemble human motion [32, 113, 151, 152].

Inverse optimal control (IOC) to identify objective functions from measurements was first

proposed by Kalman [125]. The method has been extended to various domains such as human motion analysis [18, 209] and robotics [166, 113, 57].

An important prerequisite to solve inverse optimal control problems is establishing a set of independent base functions $\psi_i(t)$ for the objective function [167]. Broadly speaking, there are four types of features commonly used in human motion analysis: kinematic features such as velocity, acceleration and jerk, dynamic features such as torque and torque derivatives, geodesic features such as path length, and energy features such as kinetic energy, work, positive work and total absolute work [32, 152].

2.6.1 Bi-level IOC Problem

In the bi-level IOC approach proposed by Mombaur et. al [166, 167], there are two nested optimizations, an "upper-level" which minimizes the error between the measurement data and the simulated trajectory, and a "lower-level" optimization that solves the forward optimal control problem to generate the simulated trajectory. Here we assume a set of base functions $\psi_i(t)$ with weight factors α_i that are unknown. As such, the inverse optimal control problem is to determine the weight factors α that results in the best possible fit to the measurement data. Mathematically, the problem is written as:

$$\min_{\alpha} \left\{ \sum_{j=1}^{m} ||x^{'*}(t_j;\alpha) - x^{'}{}_M(t_j)||^2 \right\} \tag{2.15}$$

where $x^{'*}(t_j;\alpha)$ is the result from the solution of

$$\min_{x,u,p,T} \left\{ \sum_{i=1}^{n} \gamma_i \int_0^T \psi_i(x(t),u(t),p)\,dt \right\} \tag{2.16}$$

such that

$$\dot{x}(t) = f(t,x(t),u(t),p) \tag{2.17}$$
$$g(t,x(t),u(t),p) \geq 0 \tag{2.18}$$
$$r_{eq}(x(0),...,x(T),p) = 0 \tag{2.19}$$
$$r_{ineq}(x(0),...,x(T),p) \geq 0 \tag{2.20}$$

Equation 2.15 is the upper-level optimization where the aim is to minimize the distance between the measured motion and the simulated trajectory by optimizing over the vector of weight parameters α. It should be noted that the notation x' signifies that only a subset of the states is measured or the measurements concern variables that are not directly a part of the state variables but allow us to draw conclusions about the state. Equation 2.16 is the lower-level optimization where the forward optimal control problem for the current α iteration is solved to compute x^* which is used in the upper-level optimization. Equation 2.17 summarizes the full or reduced vector of states, controls and free parameters (system dynamics). Equation 2.18 describes all continuous inequality constraints such as box constraints

on all variables or more complex relations between several variables. The coupled and decoupled pointwise equality and inequality constraints are described in Equation 2.19 and Equation 2.20 respectively. These could be start and end point constraints on states, phase switching conditions or periodicity constraints.

On generating numerical solutions to this bi-level IOC problem, we note that the lower-level optimal control problem has to be solved first and the upper-level function cannot be expected to satisfy any conditions on smoothness or differentiability. As such, we wish to apply a derivative free optimization for the upper-level, i.e., only requiring function evaluations and no explicit gradient information. In previous works [57, 166], the quadratic-approximation-based method BOBYQA (Bound Optimization BY Quadratic Approximation) [188] and COBYLA (Constrained Optimization BY Linear Approximations) [187] from the NLopt library [114] have been used successfully. For the lower-level problem, one can use a derivative-based approach, which relies on state parametrization by multiple shooting, a discretization of the controls and a structure that exploits the SQP algorithm to solve the resulting discrete nonlinear programs. It is important for this method to be efficient since it is solved on every iteration of the upper-level problem.

2.7 Movement Primitives

Learning through observation is an important paradigm for humanoids as it enables the robot to take advantage of the similarity in topology between humans and humanoids. A fundamental hypothesis in human movement is that motions are comprised of movement primitives which are combined and sequenced to generate complex behaviours [207].

Movement primitives have many definitions in the literature but can be grouped roughly into four main categories: spatial, temporal, spatio-temporal [132] and dynamical. Table 2.2 details the different types and lists some related work for each.

Primitive Type and Works	**Definition**
Spatial [174, 97]	Patterns of instantaneous co-activation that are described as vectors in the activation-space, and can be extracted from motion data via Principal Component Analysis (PCA).
Temporal [52, 177, 53, 55]	Time-courses of activity that are linearly superimposed to give rise to complex movements such as human whole-body motion. Identification approaches: PCA, Independent Component Analysis or Anechoic Demixing.
Spatio-temporal [65]	Combination of spatial and temporal primitives where instantaneous patterns of co-activation are combined with time-courses. These can be shifted in time and linearly combined to generate new movements.

Dynamical [110]	Trajectories are generated by transforming the output of a dynamical system via a learned kinematics mapping. They can be easily modulated based on task demand and sequenced to generate complex movements.

Kinematic motion primitives (temporal, spatio-temporal) [65, 168] are advantageous because they are *modular*: novel movements can be generated by a weighted combination of previously-learned primitives. Dynamical Movement Primitives (DMPs) are typically not modular, meaning there aren't many approaches to combine them across different effectors [132].

In the area of dynamical MPs, a modular approach was developed using coupled Gaussian Process Dynamical Model (GPDM) [233]. This method can learn movement primitives from human movement data. The approach allows for controllable couplings between primitives, driving different parts of the body.

In [132, 55, 56], the flexibility of the full GPDM framework [233] was not needed and instead, a simplified approach using weight morphing was used, inspired by earlier works in computer graphics and animation [87, 60].

2.8 Model Predictive Control

One realization of model-based optimal control is called model predictive control (MPC). It is an approach that relies on real-time trajectory optimization over a finite-horizon. The method has become widely adopted in the robotics community because of its ability to handle nonlinearity, constraints and system dynamics in an online fashion [107]. MPC has been used in many robotics and industrial applications including path planning for autonomous vehicles [154], quadrotor control [27], industrial processes [80] and humanoid robot control [76, 64].

Being more specific, at every iteration of the MPC loop, the current state of the robot is measured and a trajectory optimization algorithm is applied to obtain a locally optimal solution from the current state for a particular (user-set) prediction horizon. The initial part of the policy is applied as a control and the rest of the solution is discarded. The optimization is repeated continuously in this manner.

MPC is very commonly used in the chemical process industry [21]. In this domain, the natural dynamics of the systems are generally quite slow (on the order of minutes) and thus the real-time optimization is not too much of a computational challenge. When the dynamics of the system are low-dimensional, it also eases computational load. This is why MPC is quite popular in the control of autonomous vehicles.

In contrast, humanoid robots are high-dimensional and require control loops to run on the order of milliseconds [76]. As such, the online optimization process is computationally expensive. This can be circumvented by using a reduced order model (as we do in this thesis) but recent works have also used the full robot model [64]. There are several challenges with using MPC for humanoids. Firstly, we need a computationally efficient optimizer that is able to work at high frequencies. Recently, shooting methods like Differential Dynamic Programming (DDP) have been considered to be more suitable for MPC. Efficient solvers based on DDP have been implemented and used for whole-body control [64]. Secondly, we need to ensure the number of iterations needed by the solver for convergence is reasonable (as low as possible). This can be done in several ways: formulating the DDP as a more convex problem [64] or by providing a good initial guess to the solver. A common approach is to simply provide the solution computed at the previous control cycle. This is a good heuristic but may fail in the face of external disturbances [64]. A more recent approach, proposed in the context of humanoid locomotion, is to warm start the MPC iterations with a motion from a precomputed motion database containing optimal motions for the given task [147].

2.9 Reinforcement Learning

Reinforcement learning (RL) is an approach which aims to find a control policy π to maximize a reward function. A categorization of RL agents and the methods used to train them is given in Figure 2.5. In robotics, a model is typically used to find an appropriate control input, although a parametric functional space can also be used [218]. The main mechanism to learn a policy is to execute an action in an environment and evalute the reward. This is, in general, a time-consuming method and hard to do on a real robot without damaging it. Because of this, simulation is often used to pre-explore the space and then transfer onto a robot to further tune the policy. Many techniques like domain-randomization [225] and accurate modelling allow for "zero-shot" transfer of policies trained in simulation onto real hardware.

The work in [153] compares Deep Reinforcement Learning (DRL) and Model Predictive

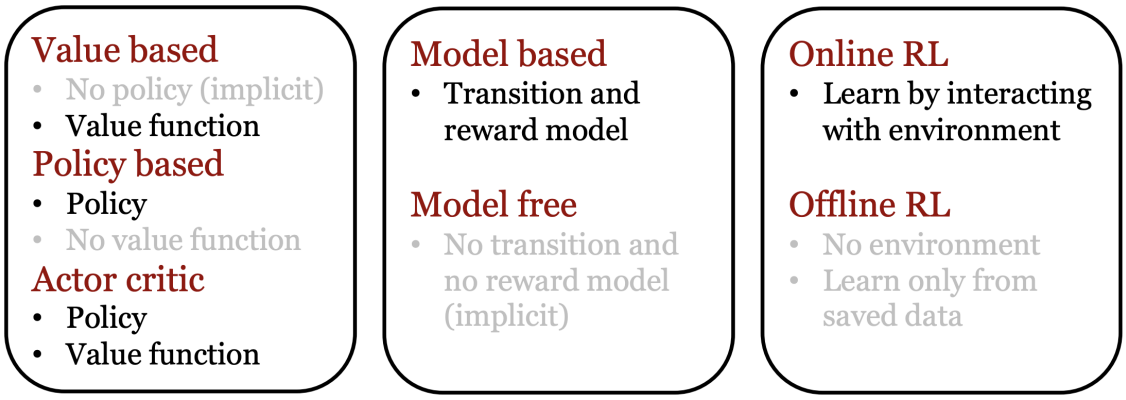


Figure 2.5: A categorization of RL agents and the methods used to train them. Adapted from [186].

Control (MPC) for adaptive cruise control in car-following scenarios. Using a first-order model to approximate vehicle dynamics, it finds that DRL achieves better generalization and handles modeling errors more effectively, while MPC offers consistent performance but struggles with modelling error. Overall, DRL can achieve higher control accuracy in varying conditions, making it a viable control scheme for adaptive cruise control. This shows the generalization capabilities of learning-based control.

There have been several works that have used reinforcement learning for locomotion [213, 212, 210, 204, 222], with some works even applying it to tasks like skateboarding [223]. Another recent development is the usage of transformers (originally developed for large language models) for humanoid locomotion [194].

2.10 Frameworks and Libraries Used

A number of libraries were used to realize the controllers, modelling and motion generation pipelines presented in this thesis. A few of the major ones are detailed in this section.

2.10.1 Pinocchio Rigid Body Dynamics Library

Pinocchio [43] is a rigid body dynamics library that implements many of the algorithms from Featherstone [77]. The library efficiently computes the dynamics of a robot model and provides analytical derivatives of the main rigid body dynamics algorithms such as the Recursive Newton-Euler Algorithm and the Articulated-Body Algorithm. The library was validated against another very popular library, RBDL [78]. Pinocchio is optimized for real-time performance and can handle complex multi-body systems like humanoid robots. The library is open-source, written mostly in C++ with Python bindings for ease of use. Pinocchio was used extensively throughout this thesis for all the computations of rigid body dynamics and kinematics.

2.10.2 EXOTica Extensible Optimization Library

EXOTica (Extensible Optimization Toolset for Robotics) is a modular, open-source optimization library designed for motion planning, trajectory optimization, and inverse kinematics in robotics. It provides a flexible framework that allows users to define, combine, and solve various optimization problems for robot control and motion planning. With its extensible architecture, EXOTica supports different robot models, cost functions, constraints, and solvers, making it suitable for a wide range of robotic applications, from industrial manipulators to humanoids. It is used in this work to perform a task-based whole-body inverse kinematics for humanoids (in Chapter 5 and Chapter 6). SNOPT [88] is used to solve the non-linear optimization problem.

2.10.3 Task Space Inverse Dynamics Library

TSID (Task Space Inverse Dynamics) is an optimization-based inverse dynamics library designed for whole-body control of legged robots, humanoids, and manipulators [3]. It allows users to define tasks directly in the task space (e.g., end-effector positions, orientations, center of mass) and compute joint torques (or accelerations) that fulfill these tasks while satisfying constraints, such as joint limits. It implements a non-strict two-layer hierarchy: constraints and costs. It is used in this thesis as the main whole-body controller.

Chapter 3

Towards Humanoids Using Personal Transporters: Learning to Ride a Segway from Humans in Simulation

Foreword

The contents of this chapter was accepted and published in the the 2022 IEEE International Conference for Biomedical Robotics and Biomechatronics with the following citation:

V. Rajendran, J. F. -S. Lin and K. Mombaur, "Towards Humanoids Using Personal Transporters: Learning to Ride a Segway from Humans," 2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob), Seoul, Korea, Republic of, 2022, pp. 01-08, doi: 10.1109/BioRob52689.2022.9925458. [198]

Minor edits involving formatting were made to adhere to thesis guidelines, but otherwise, the content was used verbatim.

This work describes the simulation model development of the Segway PT along with an initial basic controller for humanoids to operate a Segway. The video associated with this chapter is at the following link: https://youtu.be/cVHbf4fwzcs.

3.1 Introduction

Legged bipeds are becoming increasingly advanced but their mobility is still restricted. Generating walking motions for humanoids still lags behind the capabilities of humans due to the number of actuators and mechanisms required for stable gait. Beyond the hardware, complex algorithms are required to ensure time and energy efficient walking motions. Some recent works have focused on humanoids with wheels instead of feet to allow for faster mobility [256], [257]. While performance is good for continuous surfaces, wheeled locomotion may not be generalizable to other human-centric environments or uneven terrain, where legged motion reigns supreme [220]. To achieve the best of both worlds, designing control algorithms for

humanoids to operate PTs could be a viable option. This allows the humanoid to maintain the advantages of bipedal walking while also allowing it to be more mobile and efficient for long distance travel.

Xin et al. [243] tackle the problem of a humanoid riding a small two-wheeled mobile platform. The COMAN (COmpliant HuMANoid Platform) successfully balances on and drives a RoboSavvy self-balancing robotic platform [10] in simulation, showing that the COMAN can actively balance on the two-wheeled platform using quadratic optimization to generate whole-body torques. While the arms are taken into consideration for balancing, they are left unactuated as the specific two-wheeled platform used had no physical handle for steering.

Gong et al. [90] successfully enabled the Cassie bipedal robot to balance and drive a Segway miniPro by modifying the nominal standing controller and shifting its centre of mass (CoM) for accelerating and turning. Hyungjik et al. [144] implement center of gravity control on miniature humanoid robots and show that they are able to ride a two-wheeled mobile platform. Similar to other prior work, the arms are used purely for balancing.

The work in [129, 130] demonstrated a method for the HRP-2 humanoid to ride a self-balancing hoverboard. These papers proposed the Sequential Online Learning Control (SOLC) method composed of the cascade connection of SGD-based open-loop Learning Control (SLC) and mini-batch-based closed-loop Learning Control (MLC). The SLC contributes the damping gain for online tuning of the foot torque to mount the hoverboard, and the MLC handles speed governing.

While these prior works have successfully balanced and driven two-wheeled mobile platforms, most works employed lightweight robots, and used arms only for balance and not platform control.

This chapter aims to evolve a human-sized biped humanoid's mobility by synthesizing control strategies that allow it to drive a Segway (two-wheeled inverted pendulum). We identify the controller of a Segway x2 SE via motion capture experiments and then formulate controllers for the REEM-C humanoid to drive the Segway in simulation. Using motion capture data, we estimated the Segway's proportional-integral-derivative (PID) control parameters used to drive and turn the vehicle. We also estimate the human's whole-body behaviour while riding the Segway to motivate our humanoid control approach. Mimicking a human model is important for applications where humanoids need to integrate within existing human environments. Moreover, it is a common assumption that human locomotion and motion is optimal [20], as such, we exploit the fact that humanoids are anthropometrically similar to humans in controller design.

To drive the Segway forward and backward, we model the REEM-C humanoid as an inverted pendulum on top of the Segway and control the CoM position through linear-quadratic regulator control (LQR). The REEM-C uses the Segway's LeanSteer handlebar for turning in the same fashion as a human. Since the REEM-C is position-controlled, the arms holding the LeanSteer mechanism pose a problem when shifting the humanoid's CoM because, in general, they are too stiff to accurately track position commands. To alleviate this issue, an admittance controller was introduced for the upperbody to allow for compliance while the humanoid shifts its CoM when holding the LeanSteer handlebar. We show that through

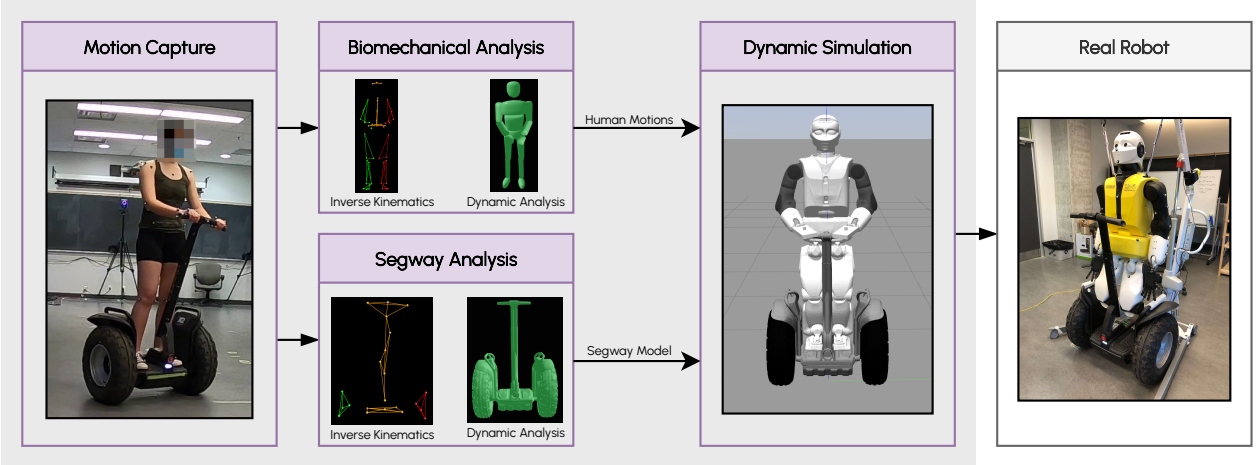


Figure 3.1: A high-level overview of our approach for synthesizing controllers to allow the REEM-C humanoid to drive a Segway. The shaded grey area highlights the focus of this chapter.

these simple control strategies, we can effectively balance on and drive the Segway with a humanoid robot (Figure 3.1).

The contributions of this work are:

- the reverse engineering and identification of the internal controllers of a Segway (from human experiments) in order to develop a high-fidelity simulation model,
- biomechanical analysis of a human rider operating a Segway,
- controller synthesis for a humanoid robot to operate a Segway, motivated by human riding data,
- simulation experiments showing that the proposed humanoid controllers allow for the operation of a Segway.

This chapter is organized as follows. In Section 4.2, we share results and analysis of a motion capture study where a human drove a Segway through various motion scenarios. Section 3.3 details the development of a simulation model for a Segway x2 using the motion capture data. Section 3.4 details the control scheme for the REEM-C humanoid to operate a Segway. Section 3.5 presents the simulation experiments run to validate the proposed humanoid controller design for operating the Segway. Finally, Section 4.6 summarizes the chapter and provides an outlook for future work.

3.2 Experimental Studies of Human-Segway System

We first model the human-Segway system by examining motion capture data of a human riding a Segway to estimate the Segway and human kinematics and dynamics.

3.2.1 Marker Sets and Equipment

Motion capture markers were placed on both the Segway and human rider. For the human, the IOR marker set [143] was used, consisting of 36 markers covering each limb, torso, and head. For the Segway, 4 markers were placed on each wheel, 5 on the base platform, and 9 on the LeanSteer (Figure 3.2). All marker data was collected through the Vicon Motion Capture system at 100Hz with 12 cameras.



Figure 3.2: Segway x2 SE with motion capture markers. The markers were placed in a way that would allow for easy reconstruction of the LeanSteer and wheel kinematics.

3.2.2 Experimental Protocols

The experiments were done with the first author (male, 63 kg, 170cm) who had prior experience driving a Segway. He performed various trials (Table 3.1, Figure 3.3) which consisted of short motions under 30 seconds, repeated 6 times each. The goal of this experiment was to collect a variety of data that was representative of typical Segway usage.

3.2.3 Motion Capture Trial Analysis

A detailed kinematic and dynamic analysis of the human on the Segway was performed for all trials. To estimate the kinematics, De Leva anthropometrics [66] were used to form the human kinematic model as seen in Figure 3.4. The kinematic model was combined with the motion capture marker position data in a Kalman filter inverse kinematics estimator, where expected marker position relative to the kinematic model and the actual marker position are minimized. The inverse kinematics estimation was carried out using RBDL [78] and biorbd [165].

Figure 3.5 shows data from a "straight" trial. We see that as the human starts to shift their CoM forward in the x-direction, the Segway starts to pitch forward and start moving. When the human shifts their CoM backward to stop (starting at 2.5 s), the Segway similarly reacts to this and pitches backward. The close coupling between the human rider CoM shifting and the Segway will be exploited in Section 3.3 to reverse engineer a pitch controller



Figure 3.3: Motion capture trial with the human driving in a straight line.

Table 3.1: Motion capture trials performed by human riding the Segway.

Motion	Description
Straight	Start from rest and drive in a straight line before coming to a complete stop. Three speeds collected.
Rotate In Place	Turn on the spot using the LeanSteer without moving forward or backward.
Figure Eight	Trace a figure eight shape that was marked on the ground.
Small Arc	Drive in a small arc shape by driving forward and then making a smooth arc-shaped turn.

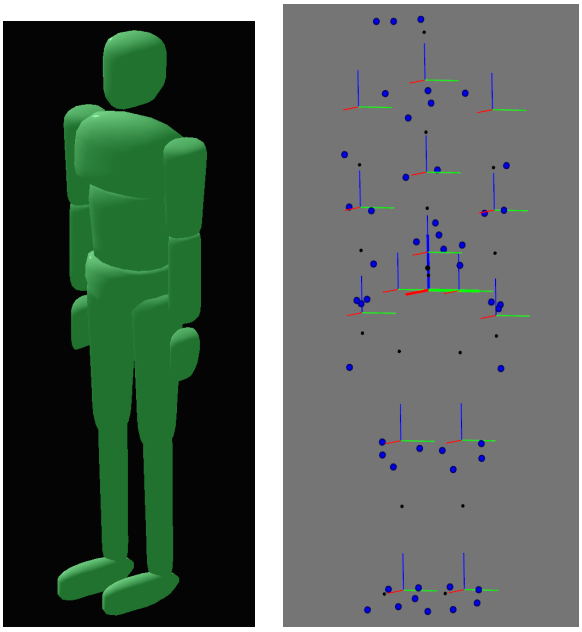


Figure 3.4: Representations of the human model based on De Leva anthropometrics [66] for kinematic analysis. On the right, the meshes of the model are shown and on the left, the axes frames for each joint. The x, y, z frames coresond to the colors red, green and blue respectively. The root frame of the human model is at the pelvis (denoted with a larger frame in the right figure). The blue dots on the right figure correspond to marker placement during motion capture.

for the Segway. We note that the y and z-direction are not shown as they do not contribute as significantly to the motion of the Segway.

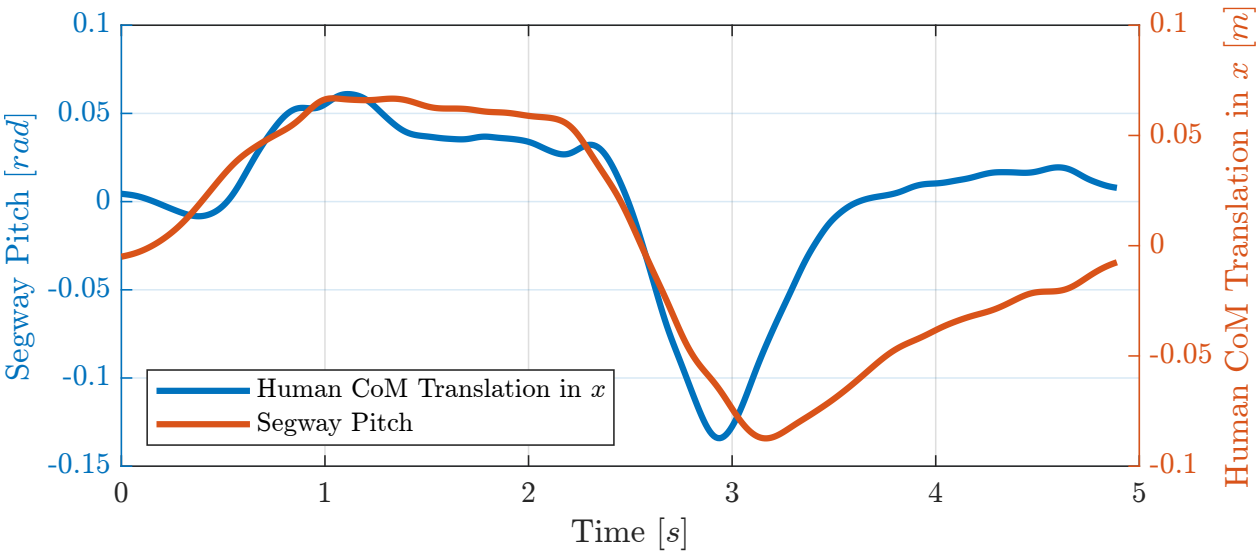


Figure 3.5: A graph showing the human's CoM translation in the x-direction (pointing forward with respect to the Segway) versus the Segway's pitch angle.

From Figure 3.6, we see the human's arm joint angles plotted for a "rotate in place" trial. "Right Arm Shoulder y" and "Right Arm Elbow x" contribute the most to turning the LeanSteer as seen by the similar shaped graphs in blue. The angles are different in sign as compared to the LeanSteer due to the difference in axes definition. This knowledge is exploited in Section 3.4 where an upper body controller is developed for the REEM-C humanoid to operate the Segway LeanSteer.

From all of the trials, some general observations about the human motions were deduced. First, the human keeps their feet flat on the Segway's platform and does not move them (with respect to the Segway) while riding. When pitching forward and backward, the human uses the LeanSteer for support to allow for more rapid CoM shifting. The human also keeps their torso upright, effectively shifting their CoM through their leg joints.

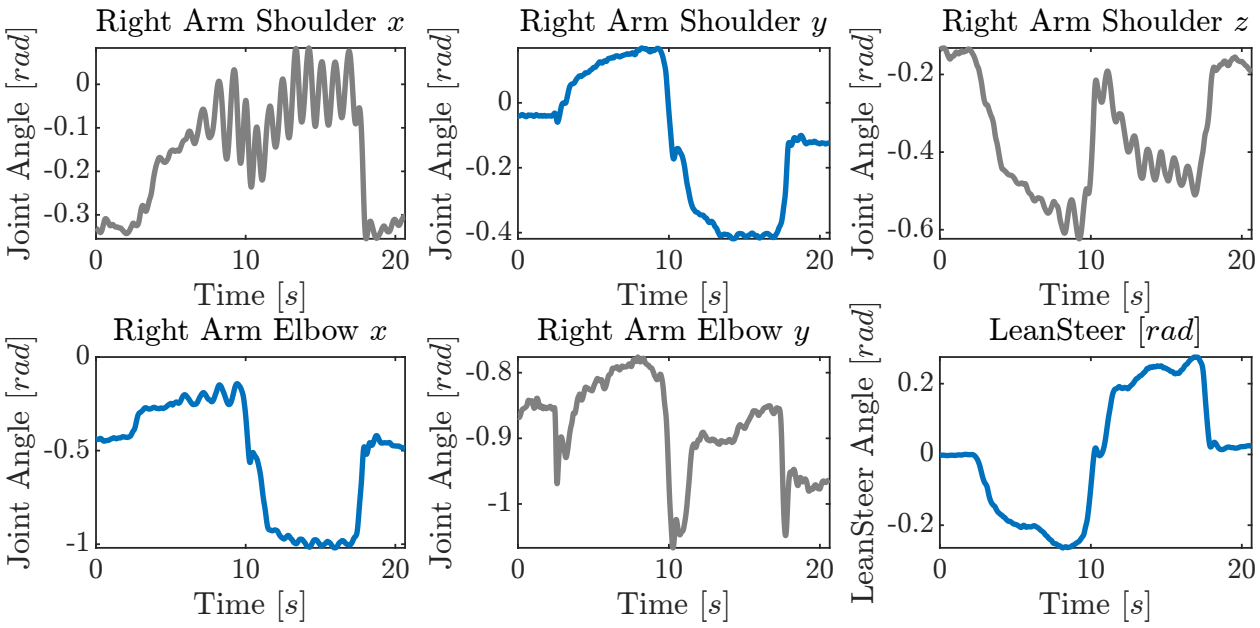


Figure 3.6: Graphs showing the human's arm joint angles and the Segway LeanSteer joint angle for a "rotate in place" trial. x, y and z correspond to roll, pitch and yaw respectively. The oscillations seen in the graphs are artefacts from the inverse kinematics process.

3.3 Modelling and Control of the Segway

The Segway model used in simulation is developed in this section (Figure 3.8). The system is approximated as a two-wheeled inverted pendulum. While a full human model was used for biomechanical analysis Figure 3.4, as noted in Subsection 3.2.3 the x-direction of CoM translation contributes most significantly to the motion of the Segway. As such, for the parameter identification process outlined in this section, the human rider is approximated as a point mass (centred at the CoM of the human) connected to the base platform of the Segway by a massless rod. The point mass is allowed to shift forward and backward (in the x-direction) via a prismatic joint to simulate a human rider shifting their CoM to drive the Segway. This corresponds to a linear inverted pendulum attached to the Segway. The simulation model was developed in URDF format to be simulated with the Gazebo dynamics simulator. It is assumed that the tires do not slip and the LeanSteer mechanism is modelled as a torsional spring joint with equilibrium at $0rad$.

Our Segway controllers were motivated by prior PID controller models [44, 169, 253], where independent self-balancing and yaw-rate controllers are developed to produce wheel torques at the left and right wheels of the Segway. The self-balancing controller governs the forward and backward motion by closing the loop on the pitch angle error whereas the yaw-rate controller acts upon the heading error of the Segway. While many works have presented more sophisticated approaches for controlling PTs similar to Segways, simple PD and PI controllers present a smaller parameter set for tuning, and have even been used in commercial Segway products [169]. In our approach, a PD controller was used for the self-balancing controller (Equation 3.1) and a PI controller with feedforward gain was used

for the yaw-rate controller (Equation 3.2). The overall controller outputs left and right wheel torques. To achieve direction control, the yaw-rate controller torque, τ_{yaw} is applied in opposite directions to each wheel. The desired pitch, θ_{des}, was set to 0 so that the base platform of the Segway would stay level while being driven or standing in place [169].

$$\tau_l, \tau_r = K_{\theta_p}(\theta_p - \theta_{des}) + K_{\theta_d}\dot{\theta}_p \pm \tau_{yaw} \tag{3.1}$$

$$\tau_{yaw} = K_{ff}\dot{\psi}_{des} + K_{\psi_p}(\dot{\psi} - \dot{\psi}_{des}) + K_{\psi_i}\int_0^t (\dot{\psi} - \dot{\psi}_{des})dt \tag{3.2}$$

The yaw-rate was estimated from the individual wheel rotation speeds as $\dot{\psi} = \frac{r(\omega_r - \omega_l)}{l}$ where ω_r and ω_l are the right and left wheel angular velocities. Segway dynamic parameters (Table 3.3) were estimated from our collected data, the Segway manual [13], and previous studies [44].

Table 3.3: Segway dynamic parameters.

Parameter	Description	Value
r	Wheel radius	0.27 m
m_w	Wheel mass	4.6 kg
m_l	LeanSteer mass	2 kg
m_b	Platform mass	37 kg
h_l	Point mass rod height	0.85 m
h_m	Point mass weight	65 kg
g	Acceleration due to gravity	-9.81 m/s^2

While the form of the controller and parameters are approximations, in the next subsections we will see that the parameter identification process yields a Segway response that is reasonably close to the collected motion capture data. This was deemed sufficient for our simulation experiments with the REEM-C humanoid.

Many works in the literature have suggested control schemes for various different PTs. Some have relied on model-based controllers like LQR or MPC and others have used traditional PID controllers. In this work, we aim to exclusively use PID controllers to keep the control synthesis simple and also to keep the number of tunable parameters low.

Controller tuning is an important aspect to take into consideration to ensure the simulated PT behaves similar to its real-life model. While manual gain tuning or like Ziegler-Nichols [71] can be used to attain good system response, it is hard to closely match the response of the real system using these methods. As such, we use the motion capture data of human riders operating the PTs in a lab setting from Section 4.2 to perform controller parameter identification. This process is detailed in Figure 3.7.

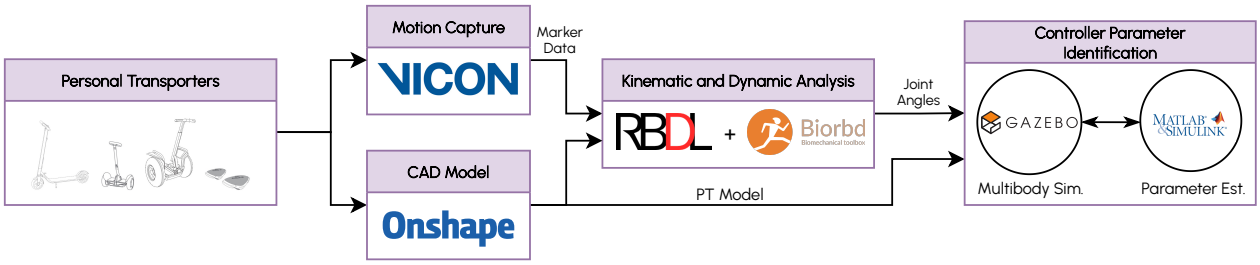


Figure 3.7: A block diagram showing the general procedure for PT control synthesis. The pipeline starts with motion capture and model design of the real PT. Kinematic and dynamic analysis is performed using RBDL and BIORBD to retrieve the PT joint angles, CoM trajectories etc. Finally, this data is used to perform control parameter identification with the PT model, joint angle data and the Simulink parameter identification toolbox.

Table 3.4: Segway controller gains, before and after parameter estimation.

Parameter	Description	Initial Values	Final Values
K_{θ_p}	Pitch controller proportional gain	1000	209.95
K_{θ_d}	Pitch controller derivative gain	20	9.72

3.3.1 Pitch Controller

To command a certain linear velocity on the Segway, the user has to shift their CoM forward to create a pitch error and the Segway will compensate by applying torques at the wheels, thus moving the vehicle forward. If we look at this controller independently, the controller takes the form of Equation 3.1 with $\tau_{yaw} = 0$. The parameters that need to be estimated are K_{θ_p} and K_{θ_d}. For the parameter estimation procedure, we take as input the human CoM trajectories for all of the straight trials. This CoM trajectory is used to shift the point mass sitting on top of the Segway to simulate a human's motion. The Segway pitch angle and linear velocity are used as outputs that need to be matched by the parameter estimation (Figure 3.8). The MATLAB Parameter Estimation Toolbox was used, applying trust-region nonlinear least squares to minimize the sum of squared errors (Table 3.4). The initial gains were set manually such that the system was balanced in the upright position stably. Our estimated Segway model matches well with the experimental motion capture pitch and linear velocity response (Figure 3.9). The response when pitching backwards slightly differs in the simulation model likely due to dynamic parameter inaccuracies as compared to the real Segway. Nonetheless, the controller is able to balance the Segway and achieves a similar linear velocity response to the motion capture data making this a viable model for use in simulation.

3.3.2 Yaw-Rate Controller

The Segway yaw-rate is commanded by the user through the LeanSteer mechanism. The LeanSteer has a plate spring on its joint to allow it to come back to its equilibrium point.

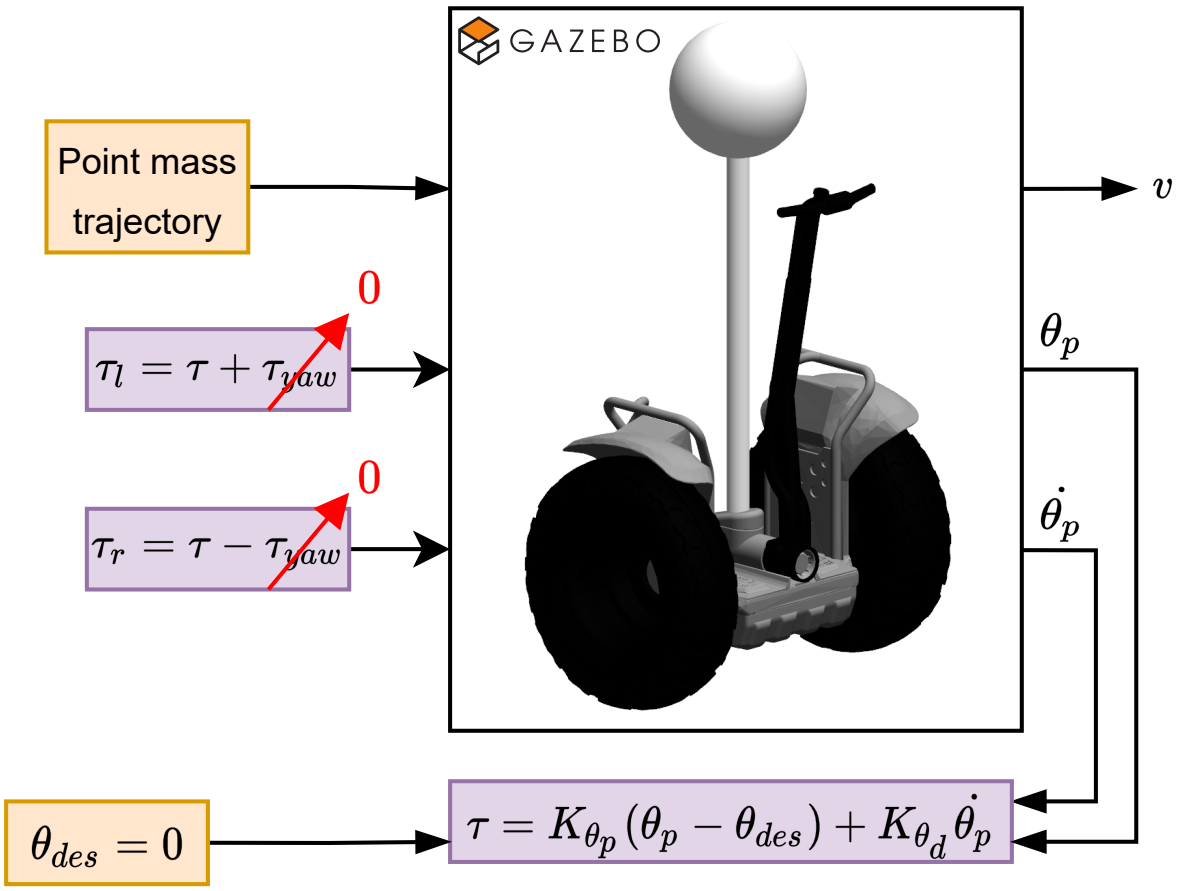


Figure 3.8: Block diagram of the pitch controller parameter estimation procedure. A point mass trajectory excites the system and the Segway's pitch (θ_p) and linear velocity are monitored for the estimation procedure. τ_{yaw} is set to zero for this identification process.

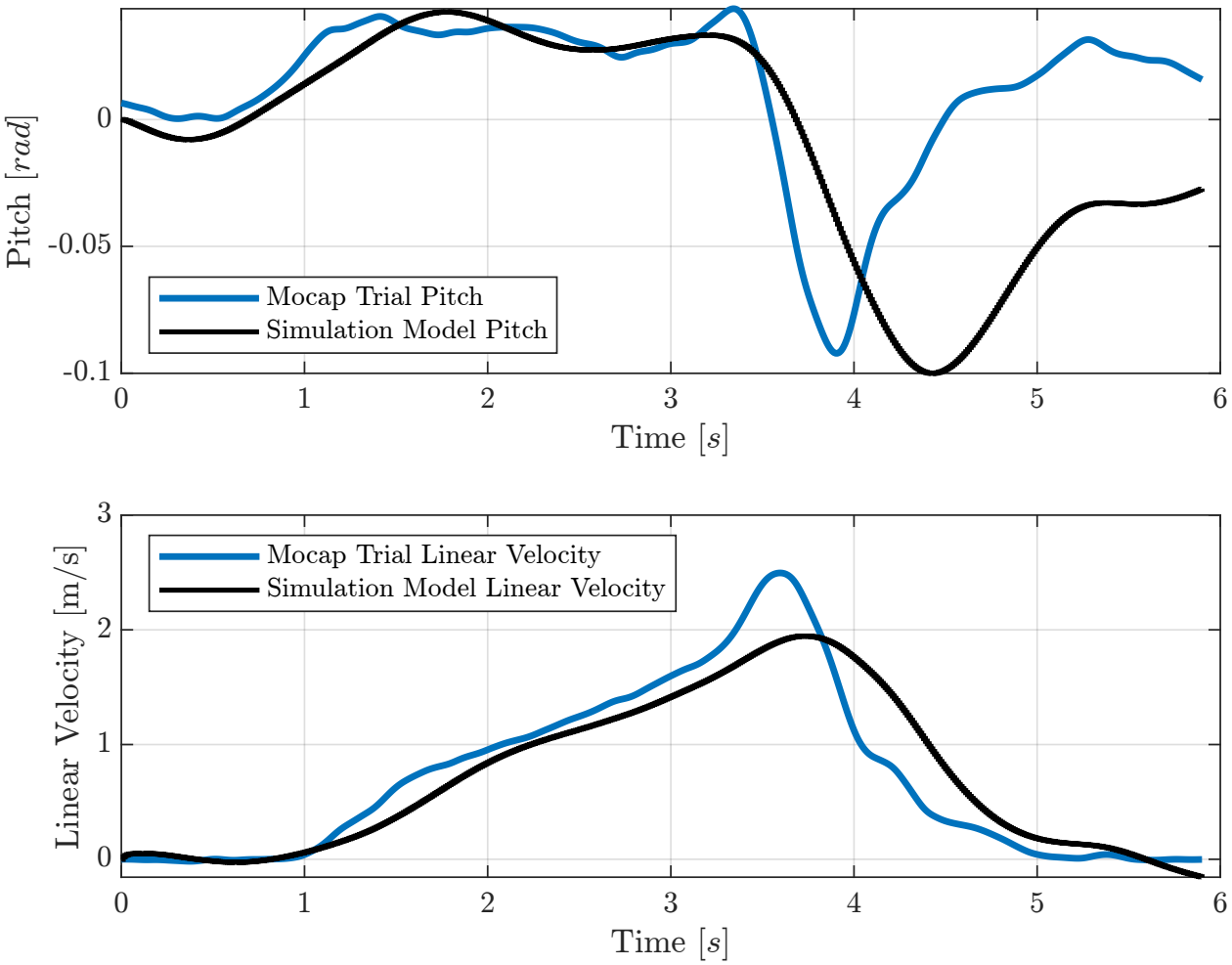


Figure 3.9: The Segway simulated pitch and linear velocity response, as compared to measured experiment data, after parameter estimation.

When the user moves the LeanSteer to a certain angle, a commanded yaw-rate is sent to the Segway's controller and opposite torques are applied to both wheels (Equation 3.1).

We first need to find the relation between the LeanSteer input and commanded yaw-rate for the Segway. For this, several trials were conducted where the human rotated the Segway in place by moving the Leansteer left and then right, turning a full clockwise, then counterclockwise circle by moving the LeanSteer mechanism to the maximum comfortable position (±0.28 rad), which is limited by the high stiffness of the LeanSteer springs. The data show a clear inverse relationship between the LeanSteer angle and Segway yaw-rate (Figure 3.10). Curve fitting was applied to the LeanSteer angle against the yaw-rate, resulting in a 7th-order polynomial (Equation 3.3). While the data could be fitted with a lower order polynomial, we use knowledge from previous studies [44] [253] that show that the region close to 0 rad is roughly linear but beyond 0.1 rad the behaviour becomes nonlinear. This is due to the torsional spring used in the LeanSteer. The use of more data will make this effect more pronounced.

$$\begin{aligned} y = &-19375.57x^7 + 875.05x^6 + 3331.07x^5 - 105.10x^4 \\ &-159.65x^3 + 2.26x^2 - 2.45x + 0.0087 \end{aligned} \tag{3.3}$$

The parameter estimation procedure for identifying the yaw-rate controller gains involves using the previously estimated pitch controller to maintain balance and the LeanSteer to yaw-rate polynomial relationship (Figure 3.11, Equation 3.3). From Equation 3.2, we note that this controller is assumed to be a PI controller with feedforward gain on the yaw-rate, requiring three parameters to be identified: K_{ff}, K_{ψ_p}, and K_{ψ_i}. The input to the estimation procedure are the LeanSteer trajectories, using the "rotate in place" motion capture trials. The output being monitored by the estimation procedure is the Segway's yaw-rate. The resultant parameters (Table 3.5) matched the experimental data well (Figure 3.12).

Table 3.5: Segway yaw-rate controller gains, before and after parameter estimation.

Parameter	Description	Initial Values	Final Values
K_{ff}	Yaw rate controller feedforward gain	0	-1.2411
K_{ψ_p}	Yaw rate controller proportional gain	0	249.88
K_{ψ_i}	Yaw rate controller integral gain	0	51.477

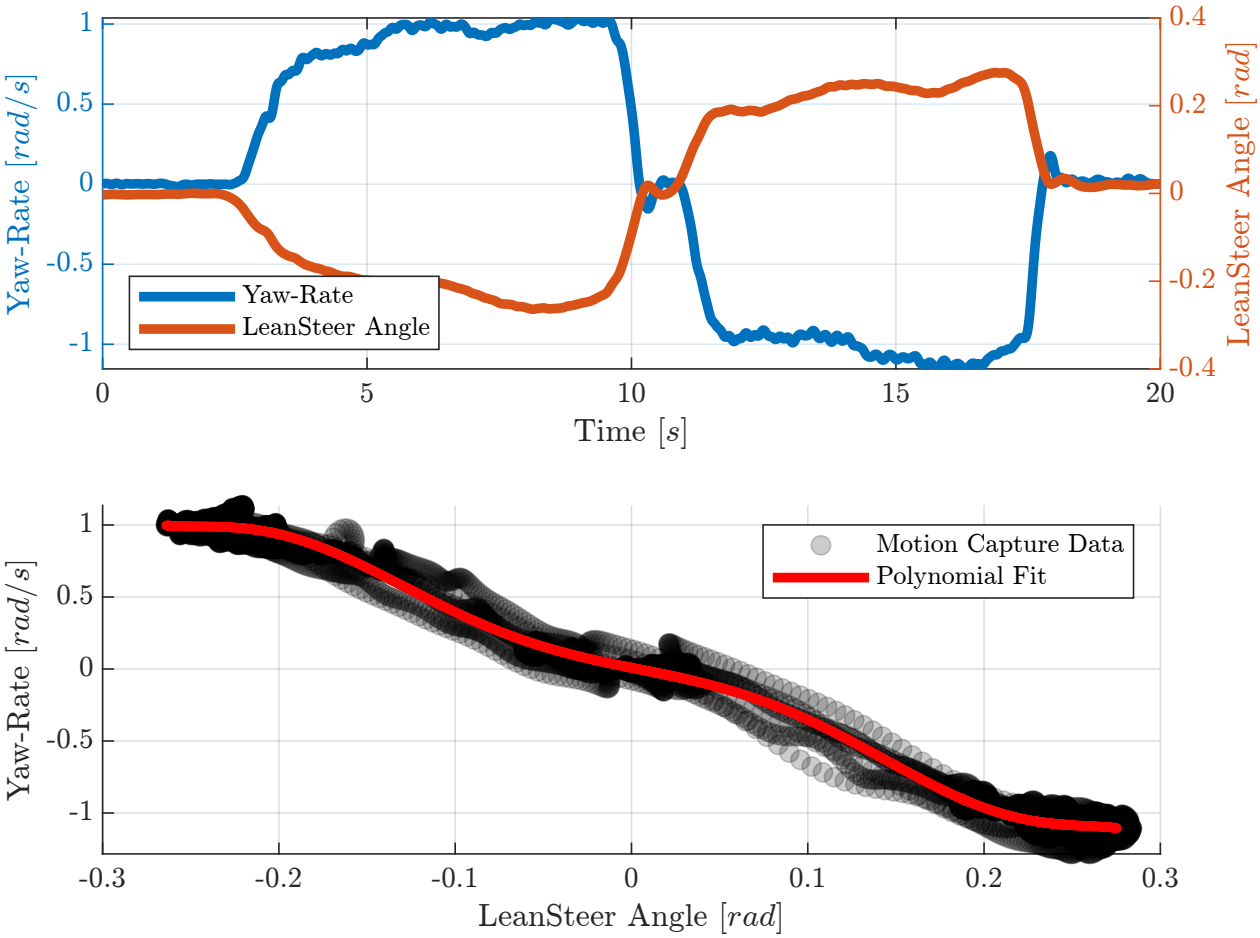


Figure 3.10: a) An exemplar "rotate in place" motion capture trial showing the LeanSteer angle and corresponding yaw rate. b) The polynomial fit (Equation 3.3) of all the Segway "rotate in place" motion capture trials to map the relationship between the LeanSteer input angle to the Segway yaw-rate.

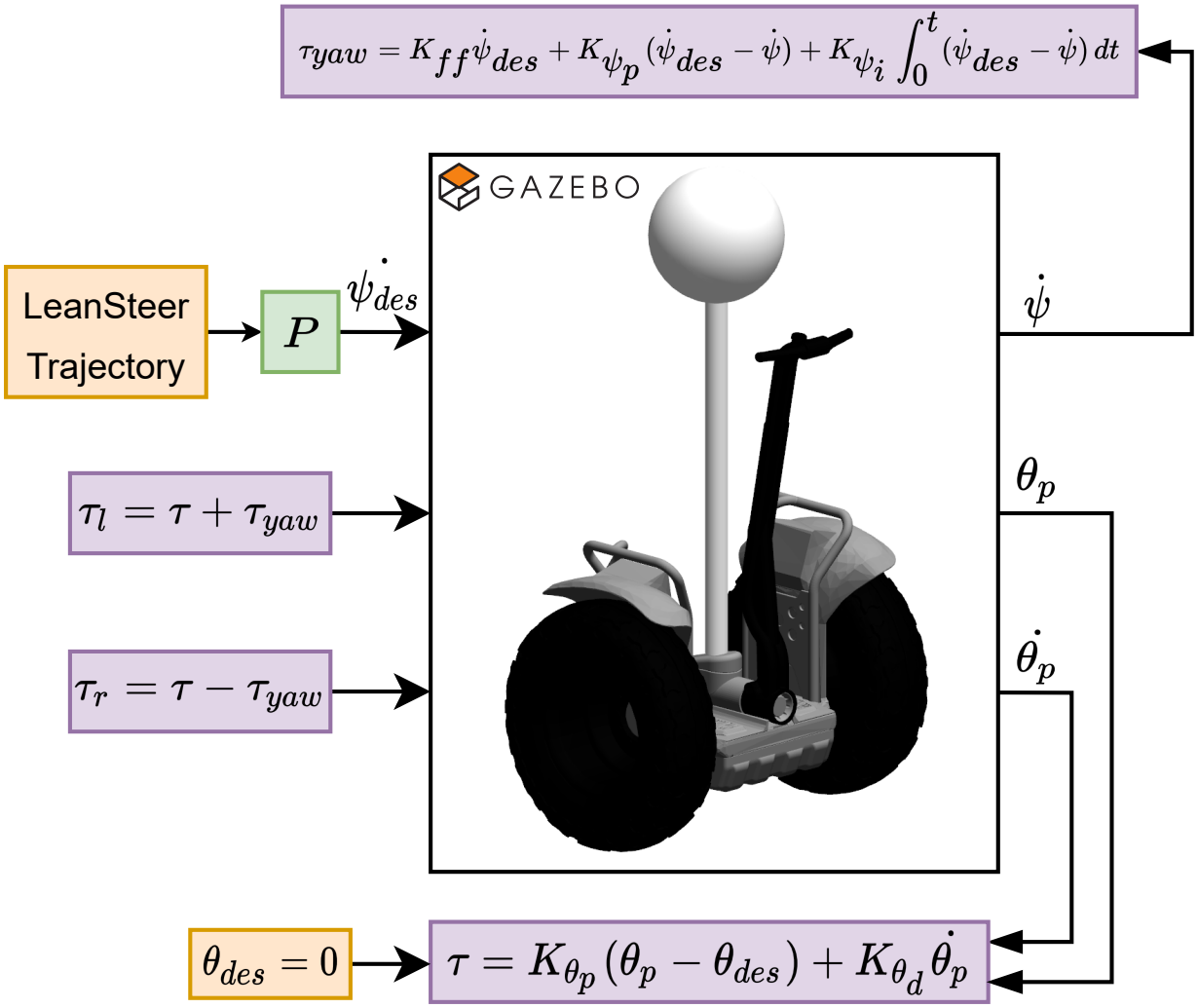


Figure 3.11: Block diagram of the yaw rate controller parameter estimation procedure. A LeanSteer trajectory excites the system and ψ, the Segway's yaw-rate is monitored. P denotes the relationship between the LeanSteer angle and the desired yaw rate as in Equation 3.3.

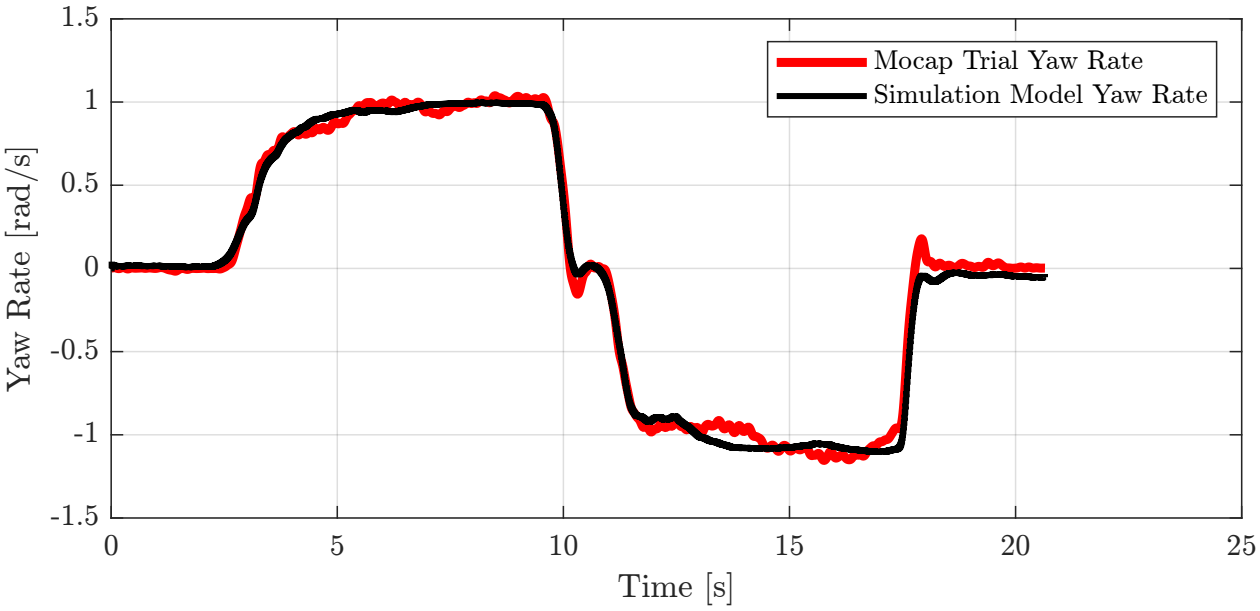


Figure 3.12: The simulated yaw-rate response of the Segway system.

3.4 Modelling and Control of the REEM-C

With the Segway system modelled, we can now synthesize controllers for the REEM-C humanoid to drive the Segway in simulation. The REEM-C humanoid has 36 actuatable degrees of freedom, weighs 67 kg, and is 165 cm tall. It has force-torque sensors in the ankle and wrists.

From our knowledge of the Segway control synthesis and human motion capture trials, we synthesize two independent controllers: one focused on shifting the CoM to pitch the Segway forwards and another to operate the LeanSteer to control the yaw-rate. First, we describe the simplified linear inverted pendulum model (LIPM) along with the controller used to shift the humanoid's CoM. Then we discuss the upper body impedance and admittance control used to operate the LeanSteer and also be compliant when the humanoid is actuating its lower body to shift its CoM. We note that the LIPM is not used for walking or foot placement as is typically done [118]. Instead, our aim is to control and command Segways to a vertical position to drive and operate them.

3.4.1 Linear Inverted Pendulum Model Based Controller

To simplify the control synthesis for pitching the Segway, we employ the LIPM. The body of the humanoid is approximated as a point mass at the CoM, connected to the ground contact point via a massless rod and maintains a fixed height from the ground. This reduces the system to a linear uncoupled system in the x- and y-directions. From Subsection 3.2.3, human data showed that the x-component of the CoM shifting attributed the most to commanding a velocity on the Segway. Thus, we simplify our LIPM model to only the x-direction CoM:

$$\ddot{c}_x = \frac{g}{z_c} c_x + \frac{1}{m z_c} u_x, \tag{3.4}$$

where m is the mass, z_c is the height of the plane on which the point is constrained and u_x is the control torque. We used the manufacturer provided URDF for the REEM-C where $m = 76$ kg and $z_c = 0.83$ m in resting posture as pictured in the leftmost figure in Figure 3.14. A common criterion for stability in biped robots is the zero moment point (ZMP), the point on the plane where no horizontal torque components exist. For the LIPM point mass, the x-direction ZMP coordinate is defined as:

$$p_x = c_x - \frac{z_c}{g} \ddot{c}_x. \tag{3.5}$$

We then apply a Laplace transform [118, 75] and a low pass filter with filter time constant T to get the following state-space representation of our LIPM:

$$\begin{bmatrix} \dot{p}_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} = \begin{bmatrix} -1/T & 1/T & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ c_x \\ \dot{c}_x \end{bmatrix} + \begin{bmatrix} -z_c/(gT) \\ 0 \\ 1 \end{bmatrix} \ddot{c}_x \tag{3.6}$$

where the control input is the CoM acceleration in the $x-$ direction. With a linear model, we can use LQR (Linear Quadratic Regulator) to control the CoM acceleration $\ddot{c}_x$, calculated by:

$$\ddot{c}_x = -K_1 * (p_x - c_x) - K_2(c_x - c_{x_{des}}) - K_3 * \dot{c}_x, \tag{3.7}$$

where K_1, K_2, and K_3 are the computed LQR gains and $c_{x_{des}}$ is the desired CoM position. We note that the first term involving the ZMP and CoM aims to bring the LIPM to an equilibrium configuration by driving the ZMP towards the CoM. As long as $c_{x_{des}}$ is commanded to positions that are within the stability boundary of the humanoid (i.e., the support polygon), we can ensure the humanoid stays balanced. From the motion capture trials in Subsection 3.2.3, the human only needed to translate their CoM between ± 0.1 m which is within the support polygon of our particular humanoid. Since the REEM-C is position controlled, when applying this control to the robot, two discrete time integrations are performed in order to produce a reference CoM position. The KDL library [214] is used to calculate the humanoid's CoM and the two leg sub-chain inverse kinematics to move the CoM to the desired location.

We note that the LIPM (with fixed CoM height) is most accurate on flat surfaces. From the motion capture analysis presented in Subsection 3.2.3, and the fact that we are considering the case where the Segway is operating on flat ground, the Segway tilts at most a few degrees. To keep the control structure and model simple, we did not consider the tilt angle in this work.

3.4.2 Upper Body Impedance and Admittance Controller

From the motion capture experiments, we found that when the human rider shifted their CoM to command a velocity on the Segway, they tended to lean their body onto the LeanSteer using their arms to prevent falling. This is quite obvious when observing a human rider on a Segway but to achieve this with a humanoid requires some special attention. To this end, we model the upper body joints with admittance controllers, focusing on the torso and arms. The REEM-C has force-torque sensors at its wrist, used to measure any forces that act on the hands and are further projected to other joints in the arms to achieve compliance. This is useful when the REEM-C is shifting its CoM forward and backward because the upper body can yield to the forces felt at the wrist from the LeanSteer and allow for a smooth CoM shifting motion. The method presented in this section is adapted from [181].

The arms are also required to operate the LeanSteer, and this requires a level of tracking accuracy, hence the controller becomes stiffer, not yielding to external forces. This allows the REEM-C to turn the LeanSteer and command a desired yaw-rate on the Segway.

To form the Cartesian admittance law for the upper body we take the hand position with respect to the base frame as $x = \begin{bmatrix} Q^\top & t^\top \end{bmatrix}^\top$ with translation t and rotation quaternion Q. The desired pose of the hand is taken as x_d such that a Cartesian impedance law that tracks the desired hand pose is given as,

$$W_i = -K_p\tilde{x} - K_d\dot{\tilde{x}}, \tag{3.8}$$

where $\tilde{x}$ is the pose error, $\tilde{\dot{x}}$ is the velocity error, W_i is the resultant impedance wrench and K_p and K_d are diagonal proportional and derivative gain matrices. This control allows for adequate tracking of Cartesian hand poses.

Analogously, we can define a Cartesian admittance law as,

$$W_c = w_i * W_i + K_{ft} * W_{ft}. \tag{3.9}$$

Here, W_{ft} is the wrench as measured at the wrist of the arm, K_{ft} is a gain to amplify the wrench, and w_i a weight that allows us to vary the effect of the Cartesian impedance control. When W_{ft} is greater than a set threshold, w_i is set to 0 so that the hand can yield to any forces or torques without interference from the pose tracking Cartesian impedance control law, W_i.

The wrench W_c is then projected onto each joint of the upperbody through the Jacobian associated with the force-torque sensor frame $\tau_c = J_{ft}^{\top} W_c$.

Since the REEM-C is position-controlled, we must transform this torque to a position command. This is done by first defining a velocity command,

$$\dot{q}_{cmd} = M_j^{-1}\tau_c - M_j^{-1}K_{j_p}(q - q_d), \tag{3.10}$$

where M_j is the mass matrix related to the controlled joints. K_{J_p} is set to 0 for each joint in the arm but to large values for the torso joints so that they stay stiffer and have less influence on CoM shifting of the humanoid. $\dot{q}_{cmd}$ is then integrated using the controller's time period to generate a position command q_{cmd}.

3.5 Simulation Experiments

In this section we show several simulation results of the REEM-C riding the Segway. For all test scenarios, the REEM-C grasps the LeanSteer with both hands but is compliant when shifting its center of mass due to the upperbody admittance controller (Figure 3.13). When required to make turns, w_i, the Cartesian impedance gain, is set to 1.0 to allow for adequate pose tracking. This allows the REEM-C to operate the LeanSteer mechanism and command a desired yaw-rate on the Segway.

3.5.1 Driving Straight

First, a straight trial where the humanoid was required to follow a CoM trajectory was conducted. The desired CoM, CoM_{des}, was a CoM trajectory of the human from a motion capture experiment. This trajectory was not used during the parameter estimation procedure detailed in Section 4.2. The REEM-C humanoid is able to track the desired CoM trajectory adequately via LQR control (Figure 3.15). Although there is a slight error, as can be seen from the Segway velocity graph, the output velocity of the Segway based on the REEM-C's CoM motion tracks very closely to the motion capture trial. This shows that the REEM-C is able to easily regulate its CoM while holding on to the LeanSteer.

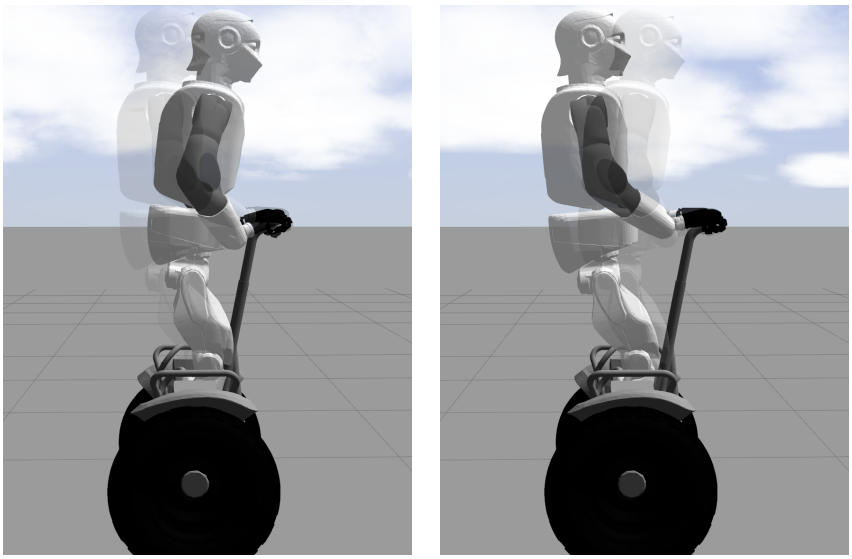


Figure 3.13: REEM-C shifting its COM back and forth. The force-torque sensors at the wrists of the REEM-C allow the upperbody to actively switch between compliance and stiff control.

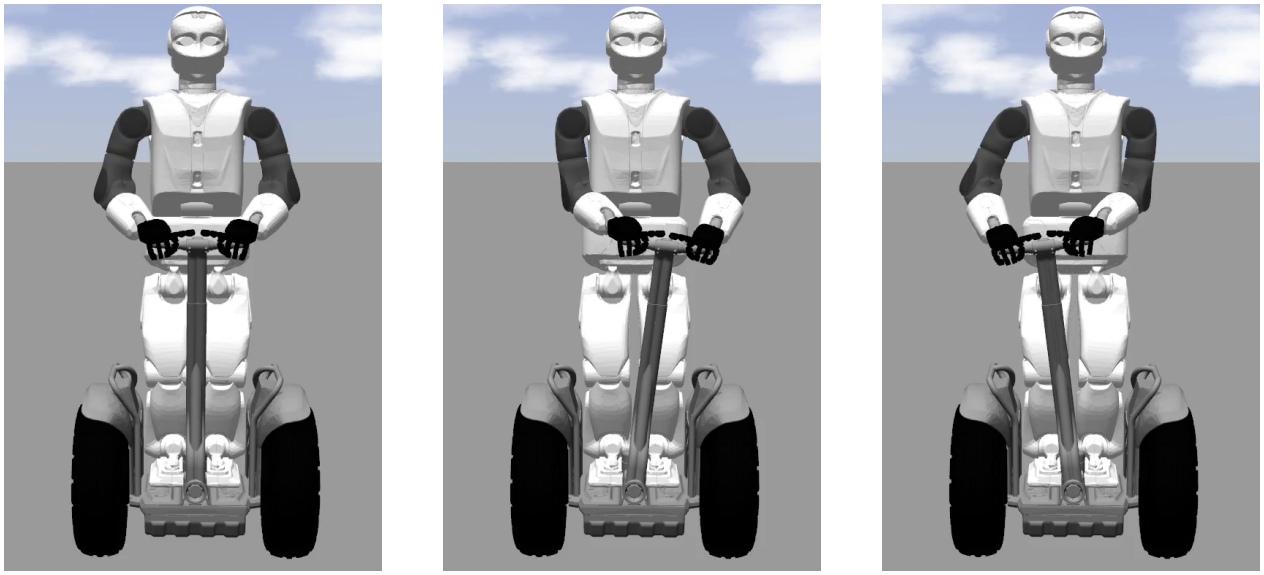


Figure 3.14: REEM-C operating the LeanSteer mechanism.

3.5.2 Rotating In Place

In this experiment, the REEM-C moved the LeanSteer to ± 0.28 rad similar to the rotate in place trials by the human. This was achieved by commanding equal and opposite joint position commands to the shoulder pitches of the REEM-C's two arms. First, the REEM-C turned the LeanSteer left and after a set amount of time, brought its arms back to resting pose (Figure 3.14) before turning the LeanSteer right. This approach of commanding only the shoulder pitch was deduced from the "rotate in place" human trial shown in Figure 3.6, where it was shown that the shoulder pitch was a significant contributor to turning the LeanSteer. The elbow also contributed to turning the Segway for the human but this was

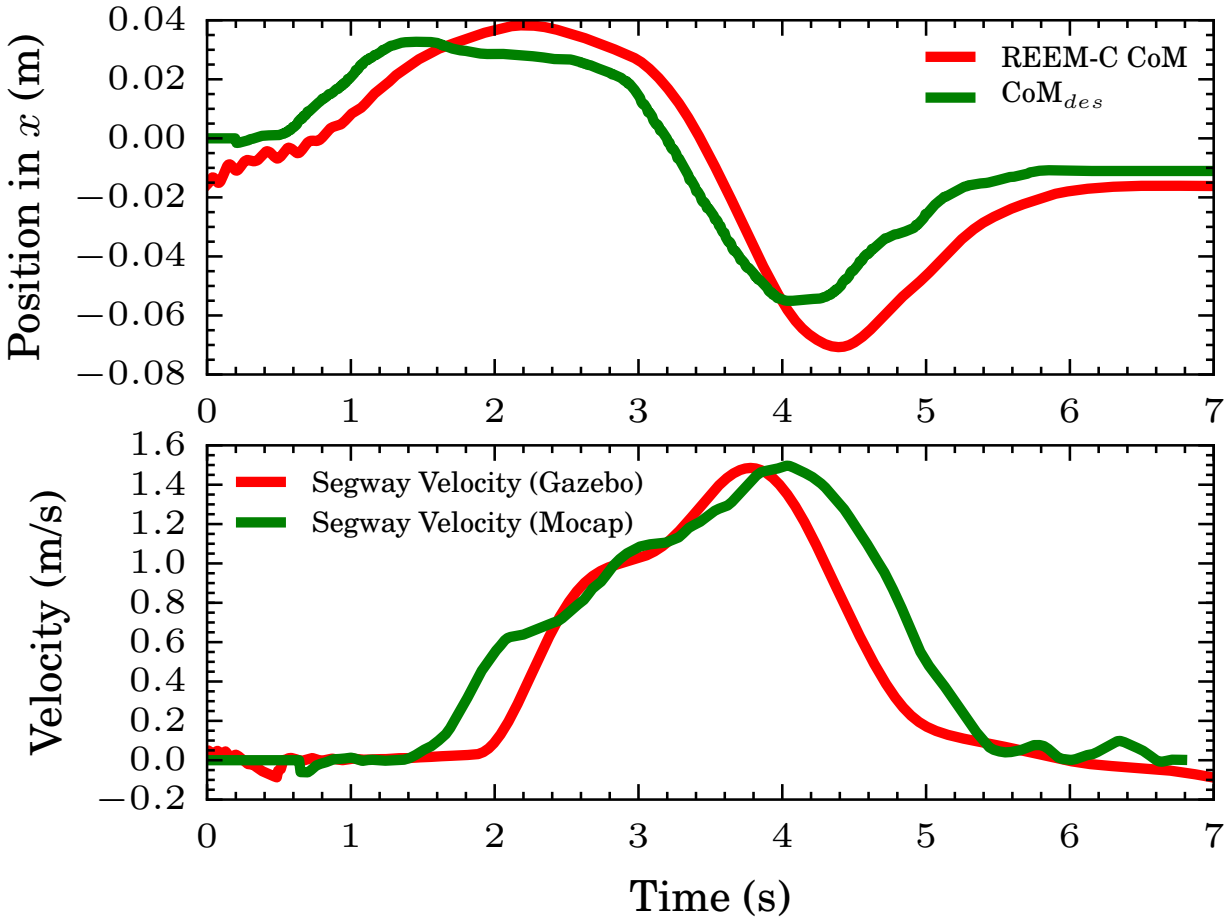


Figure 3.15: REEM-C CoM and Segway linear velocity as compared to a corresponding "straight" motion capture trial. The desired CoM trajectory CoM$_{des}$ comes from human motion capture experiments.

neglected for the humanoid as the shoulder was able to easily move the LeanSteer to the desired joint angle.

For this maneuver, the REEM-C regulated its CoM position to 0 m in the $x-$ direction, aiming to stay in place while the Segway rotated (Figure 3.16). As can be seen, the REEM-C effectively moved the LeanSteer to the desired position and commanded a yaw-rate on the Segway. The Segway pitched back and forth very little during this motion and the REEM-C also maintained its CoM very closely to 0m in the x-direction.

3.5.3 Driving in a Small Arc

The last task for the REEM-C was to drive a small arc. This was to show that the REEM-C can actively pitch forward and backward while also operating the LeanSteer (Figure 3.17). As can be seen, the REEM-C is able to regulate its CoM and command a yaw-rate through the LeanSteer effectively. The REEM-C was able to follow the prescribed trajectory only overshooting near the top of the arc where the direction of motion switches (Figure 3.18).

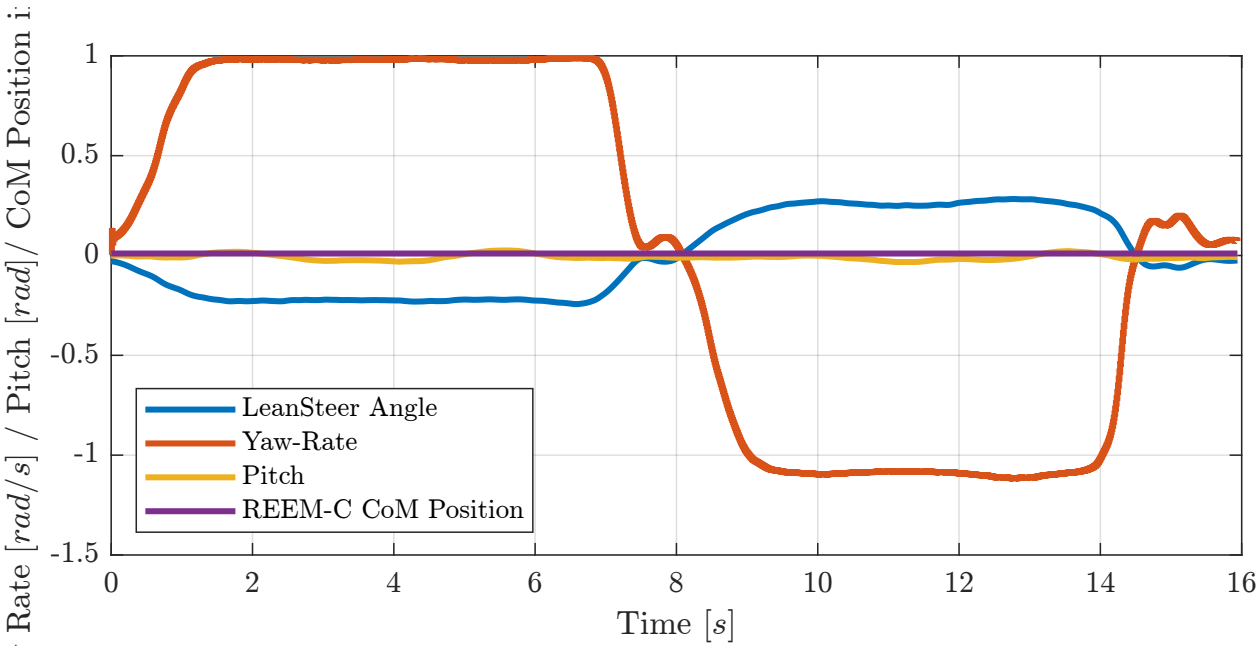


Figure 3.16: Segway LeanSteer angle, yaw-rate, pitch and REEM-C CoM position for a rotate in place simulation.

The prescribed trajectory was designed by driving the Segway without the REEM-C and applying a constant yaw-rate and pitch angle. As such the error in the trajectory tracking is likely due to the delays induced by the REEM-C arm motion and time taken to pitch forward from rest.

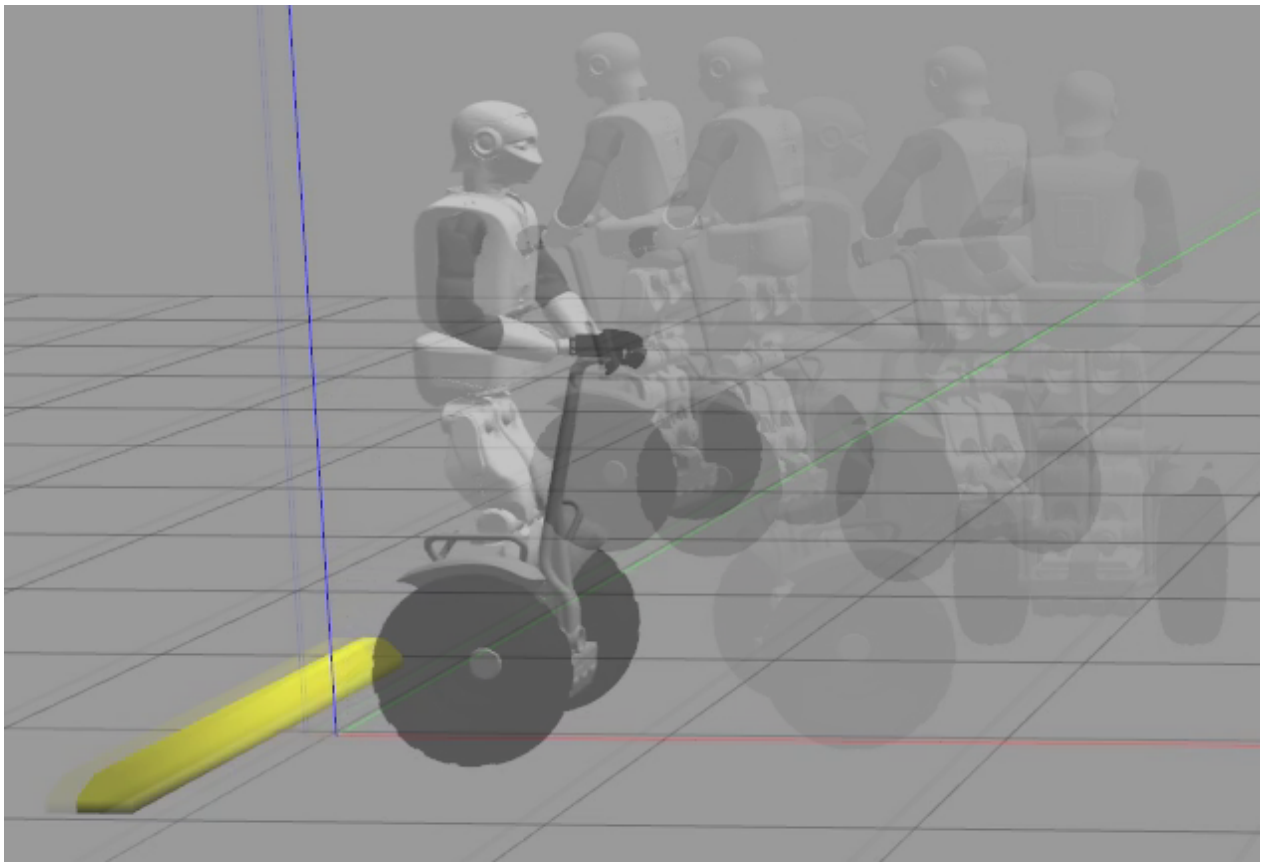


Figure 3.17: REEM-C performing a small arc. The yellow parking stop keeps the Segway stable before starting the motion.

3.6 Conclusions and Perspectives

In this work we presented an approach to go from motion analysis of a human riding a Segway to a humanoid riding a Segway. First we presented and analyzed the motion capture experiments conducted with a human. Next, we developed a simulation model of the Segway that matched the motion capture experiments closely. Finally, we presented a decoupled control approach for the REEM-C humanoid to balance on and operate the Segway. The approach involved modelling the REEM-C humanoid as a LIPM and using LQR control to shift its CoM in order to command a velocity on the Segway. The upperbody was able to hold and command the LeanSteer mechanism via an impedance/admittance control scheme. The controller was compliant during CoM shifting and stiff when commanding a LeanSteer position. A limitation of the work presented in this chapter is the robustness of the method to parameter uncertainties (models and controller gains). Analysis of the coupling effects between the upper and lower body controllers of the REEM-C is also important to study.

For future work we will be testing the proposed control scheme with the real Segway and REEM-C humanoid. We will also be performing more extensive motion capture trials to collect data from many participants. Extending the approach to directly learn from the human motion capture trials using inverse optimal control is another approach which

could yield better performance. Comparing foot reaction forces between the Segway and the human/REEM-C could also help develop a more sophisticated model. Developing a wholebody controller instead of decoupling the humanoid into a two part system will also help to remove any coupling effects between the independent controllers (which were neglected in this work). Further, we would like to extend this initial work into a larger framework that would allow humanoids to use other transportation devices meant for humans such as scooters and electric skates.

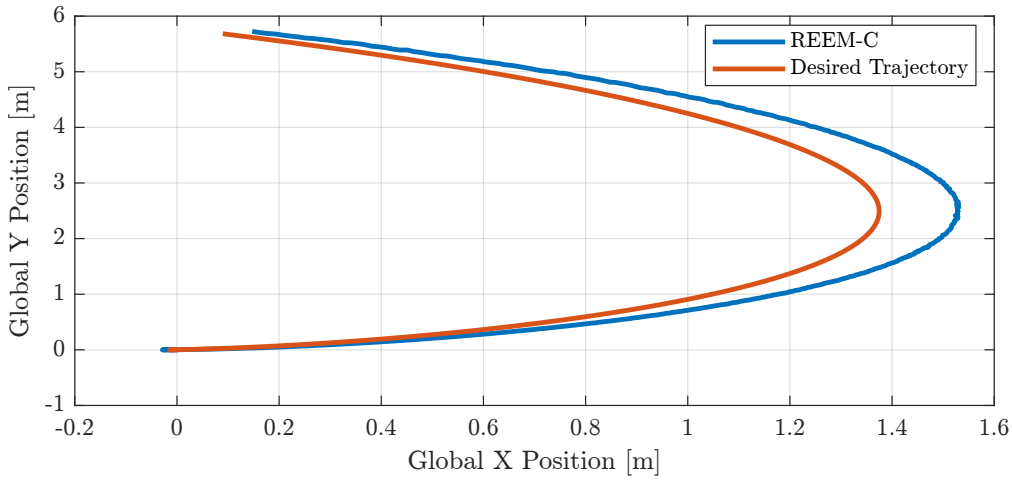


Figure 3.18: The global $x-$ and $y-$ position of the REEM-C in the World frame along with the desired arc trajectory.

Chapter 4

Whole-body Stabilization of the REEM-C Humanoid on Actuated Balance Boards in Sagittal and Frontal Directions

Foreword

The contents within this chapter was accepted and published at the 2023 IEEE International Conference on Humanoid Robots with the following citation:

V. Rajendran, F. J. A. Chavez and K. Mombaur, "Whole-body Stabilization of the REEM-C Humanoid on Actuated Balance Boards in Sagittal and Frontal Directions," 2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids), Austin, TX, USA, 2023, pp. 1-8, doi: 10.1109/Humanoids57100.2023.10375223. [197]

Minor edits involving formatting were made to adhere to thesis guidelines, but otherwise, the content was used verbatim.

This work develops a whole-body stabilization controller for the REEM-C to balance on actuated balance boards. This work was a stepping stone to driving and balancing on PTs. The controller developed in this work was adapted and extended in Chapter 6 to enable the REEM-C humanoid to drive a Segway. The video associated with this chapter is at the following link: https://youtu.be/X61gbXRYg50.

4.1 Introduction

The challenge of achieving stable balance control is a critical issue in the field of legged robotics, particularly for humanoid robots. For these robots to become fully integrated into human society, they must be able to navigate various terrains, including unstable and dynamic environments, with a high level of safety. Many works in the literature focus on

stabilization for walking and uneven terrain [124, 42, 51, 40, 93, 112]. However, it is also interesting for humanoids to balance on and interact with human-centred environments and devices which offer uncertain dynamics such as personal transporters [198]. Most stabilizers made for position-controlled walking are very damped, aiming to reject as many disturbances as possible. However, this strategy does not work for other challenging environments such as on moving platforms or vehicles. In these cases, humanoids should be more compliant and adapt to the deviations and disturbances. Torque controlled robots are inherently compliant but to achieve this same behaviour for position controlled robots, admittance-based control schemes are required.

Humanoid balance and stabilization has been a topic of interest over the years. In the work by Kajita et al. [124], a stabilization approach incorporating posture control, foot force control, foot torque control and zero moment point (ZMP) distribution is used to stabilize the walking gait of the HRP-4C humanoid. A similar approach is used in [42] to climb stairs with the HRP-4 humanoid. In [246], the authors propose a stabilizer that takes into account full 6D force measurements to balance a UBTECH Walker robot in the MuJoCo simulator. Simulation experiments on unstable platforms validate the approach.

In this work, we focus on stabilizing a humanoid robot on actuated balance boards. In the literature, there has been some interest in stabilizing humanoids on unstable platforms and seesaws. Some seminal work in this area is presented in [109]. The authors introduce a control technique for full-body balancing in humanoid robots, enabling safe and compliant physical interaction without requiring force measurements, inverse kinematics, or inverse dynamics. The method, validated through experiments on a bipedal humanoid, optimally distributes forces across contact points and adapts to uneven ground, showcasing its effectiveness in replicating human-like responses during interaction. In [22], an adaptive torque-based controller is designed for the Sarcos Primus humanoid. The controller used an augmented model-based policy-mixing controller to balance on a frontal plane seesaw. In [176], a momentum-based balancing controller is developed for the iCub humanoid and simulation results are shown to stabilize a seesaw. Recently, a stabilization approach is proposed for the HRP-4 to balance on a frontal plane seesaw based on super-twisting sliding mode control [179]. The controller aims to apply a torque on the seesaw to command it to any angle by sliding its foot along the seesaw. The controller is tested in a physics based simulation but it is unclear whether it would work on real hardware.

In this work, we tackle humanoid balancing on actuated balance boards in the frontal and sagittal planes. Sagittal seesaw balancing has not been studied as well as frontal plane balancing in literature. This perspective on the traditional (frontal) seesaw balancing problem is particularly significant because of the potential challenges that may arise when considering the range of motion of humanoid robots. The actuated balance board provides a tractable problem with enough similarities to a fully unactuated balance board to serve as an efficient bridge between simulation results and real robot experiments while minimising risks to the robot. It further allows the opportunity to provide challenging rapidly changing scenarios to stress test the proposed method. For humanoids to be useful in the real world, they should be able to balance in dynamic environments (e.g., riding buses or using personal transporters). This directly motivates our work in this chapter.

Our approach first starts with qualitative analysis of a human balancing on a balance

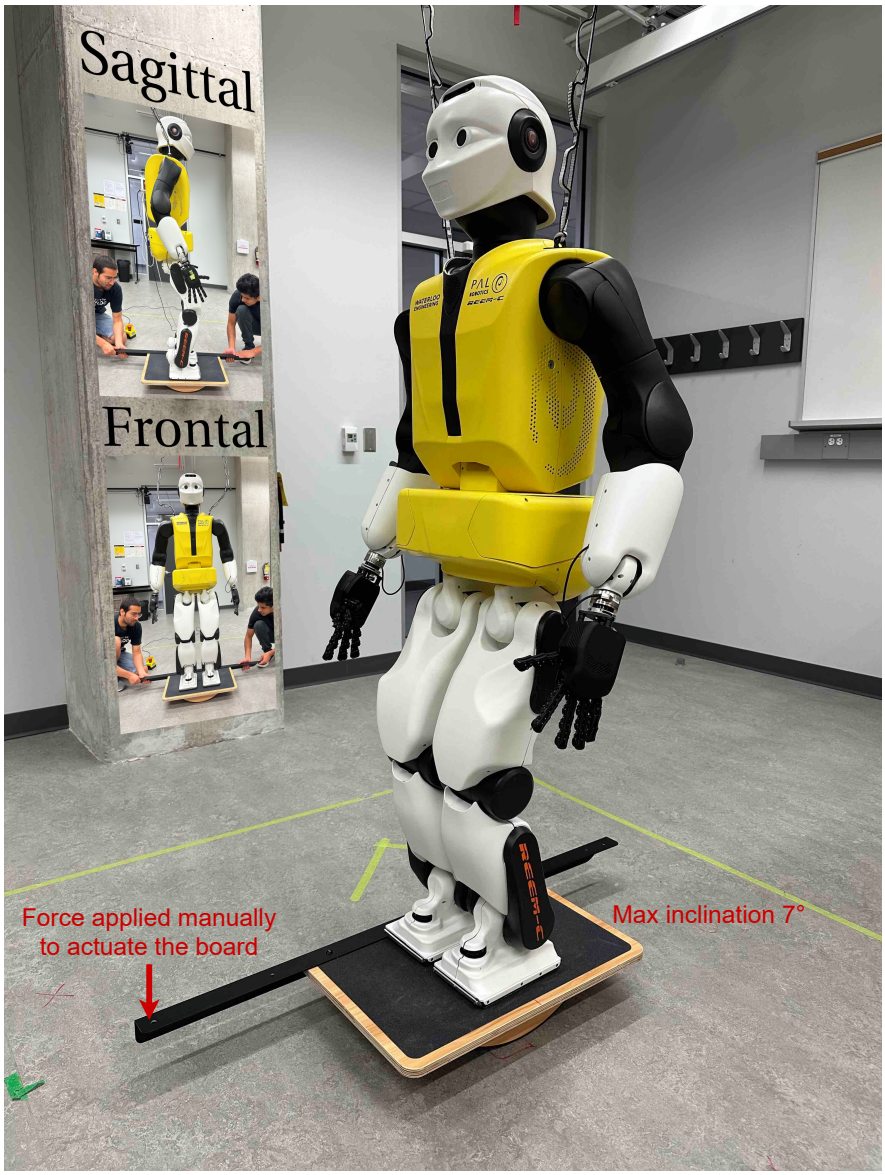


Figure 4.1: The REEM-C Humanoid on a balance board. The black steel bar was retrofitted to actuate the board manually. Sagittal and frontal plane setups are shown in the background, being actuated by two humans.

board. From this, we deduce key takeaways to inform our control design. We then take an approach similar to [124, 42] but modify portions of the approach to form a new integration to balance on a sagittal plane balance board. The method includes ankle admittance to correct the roll and pitch of the ankle based on ZMP measurements, base orientation admittance to correct the roll and pitch of the robot based on IMU measurements and foot force difference control, center of mass (CoM) admittance that takes into account a model of flexiblity in the x direction based on the work in [30] and contact wrench distribution. All of these methods produce task-space references which are consumed by a whole-body task-based QP

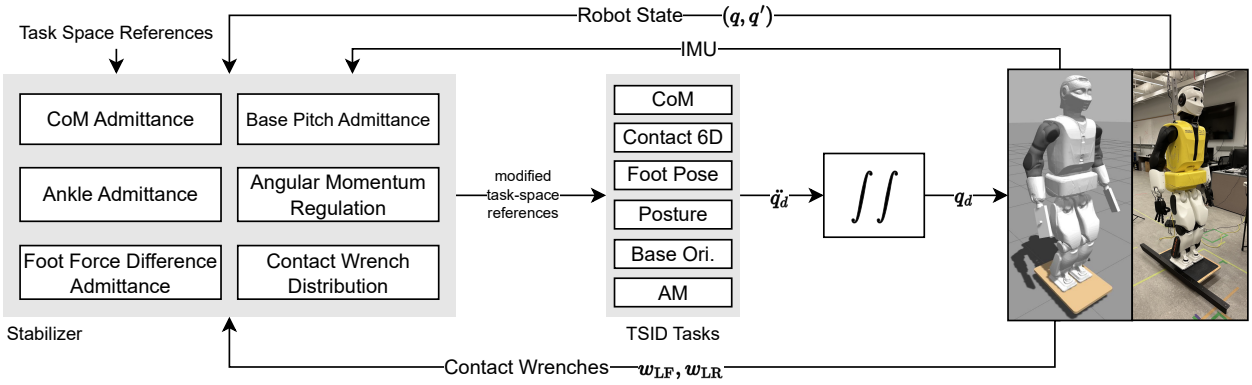


Figure 4.2: A high-level overview of the stabilization controller developed in this work to balance on an actuated balance board.

(quadratic program) controller to track the references. The contributions of our work are as follows:

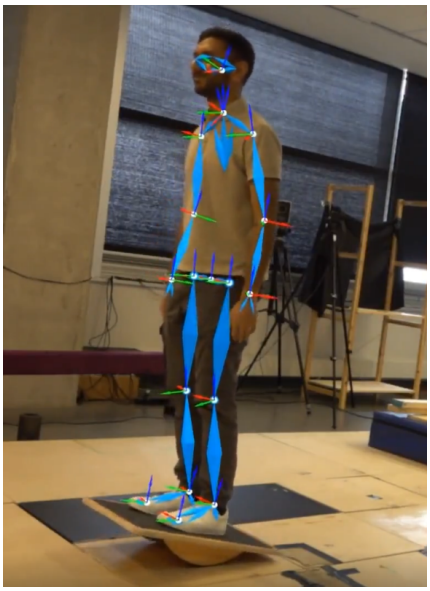
- We propose a method for balancing on an actuated balance board in the frontal and sagittal planes.
- We develop a controller integrating multiple strategies found in literature for a humanoid robot to balance on an actuated balance board, which is inspired by human motion capture data. Results show that the controller is able to withstand large perturbations.
- We conduct simulation and **real hardware experiments** to test and validate the effectiveness of our control approach.

This chapter is organized as follows. In Section 4.2 we share qualitative insights of motion capture studies of a human on a balance board to motivate the control approach for a humanoid. Section 4.3 introduces background theory used throughout the rest of the work. In Section 4.4 we lay out the full control approach to stabilize a humanoid on an actuated balance board. Section 4.5 shows experimental results in simulation and on hardware. We conclude in Section 4.6 with a summary of the work, limitations and future directions.

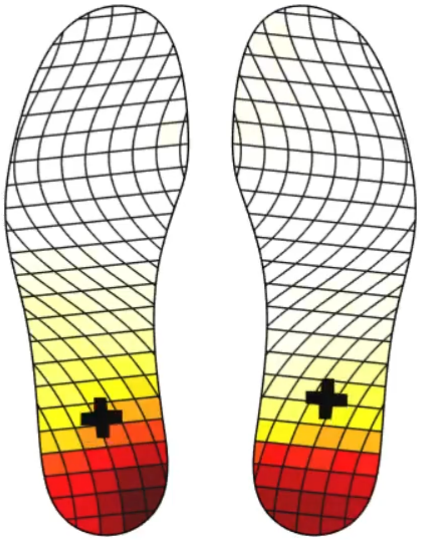
4.2 Analysis of a Human on a Balance Board

To motivate our humanoid control design, we took inspiration from human motion capture trials where one of the authors balanced themselves on an off-the-shelf balance board. The Theia3D system (Theia Markerless, Inc.; Kingston, Ontario, Canada) was used to obtain 3D subject pose estimates of all limb segments. Two trials were conducted, one where the human stood on the board and balanced in the sagittal plane and another where the human balanced in the frontal plane. The authors also used pressure insoles to measure the center of pressure of each foot while balancing on the balance board as shown in Figure 4.3.

From the human motion studies we made a few qualitative observations:



(a)



(b)

Figure 4.3: a) One of the authors balancing in the sagittal plane on a balance board. The Theia3D system reconstructs all limb segments. b) Pressure insoles showing foot center of pressure estimates. Here, the human is leaning backwards, concentrating their center of pressure near the heals.

- The human keeps both feet fixed and in contact with the balance board at all times,
- The human keeps their torso upright and CoM roughly in place while balancing,
- The human lets their ankles be compliant to the board's motion and they can also command a torque on the board.

These observations informed our choice of model and control design for humanoid balancing (on balance boards).

4.3 Background Theory

In this section, we introduce notation and formulations related to PID controllers, modelling of a balance board and ZMP estimation to set the groundwork for the rest of the chapter.

4.3.1 PID Controller Formulation

In this work, several *proportional-integral-derivative* (PID) controllers are used to track task-space references. A vectorized formulation is given as:

$$\text{PID}(t, s(t), i(t)) = K_p(s - i) + K_i \int_0^t (s - i)d\tau - K_d \frac{di}{dt}, \tag{4.1}$$

where $s \in \mathbb{R}^{r\times 1}$ and $i \in \mathbb{R}^{r\times 1}$ are the *setpoint* and *input* (measurement) column vectors, and K_p, K_i and K_d are matrices of proportional, integral and derivative gains. We note that this formulation is different than the classical version. Specifically, for the derivative term, we use the *derivative on measurement* form [79]. In practice, this does not change the performance of the controller substantially. It has the added benefit of removing output spikes due to derivative kick on setpoint changes in contrast to performing *derivative on error* control.

We note that the integral term is implemented as a leaky integrator, similar to the formulation presented in [42]; it has the benefit of not winding up by design.

A discretized version of Equation 4.1 is used in this work. With an abuse of notation, we drop the t from here onwards and write the equation as $\text{PID}(s, i)$.

4.3.2 Balance Board

For this chapter, we focus on balancing on actuated balance boards. A sagittal plane board (along the world x-axis) is shown in Figure 4.4. By actuated, we mean that the board is powered/moved externally rather than as a passive mechanism. As such, in this work, we don't explicitly take the (dynamic) model of the balance board into account but do feed the angle of inclination and board dimensions into the robot controller. However, this information was only used in simulation tests; for real robot experiments, no information about the balance board is fed into the controller.

The balance board used for this work is off-the-shelf, made of wood and has a semi-circular base. To actuate it, we added an extended steel bar (as shown in Figure 4.1) to move it manually by hand. To aid the controller in simulation, we estimate the contact point on the ground through Equation 4.2:

$$d_b = r\theta_b, \tag{4.2}$$

where r is the radius of the circular base, calculated from its chord length l and segment height h. The contact follows the arc traversed by the circular base, and d_b is the arc length between the original contact point (shown with a green dot) and the current contact point as a result of tilting the board by angle θ_b.

4.3.3 Zero Moment Point Estimation

From the previous section, we can think of the balance board as a moving slope with inclination angle θ_b. As such, the humanoid's ZMP needs to be projected onto the platform. We use the well-known cart-table model [122] and the formulation in [106]:

$$p_x(\theta_b) = \begin{cases} c_x - \frac{z_c}{g\cos\theta_b}\ddot{c}_x + z_c\tan\theta_b & \theta \geq 0 \\ c_x - \frac{z_c}{g\cos\theta_b}\ddot{c}_x - z_c\tan\theta_b & \theta < 0 \end{cases} \tag{4.3}$$

$$p_y(\theta_b) = \begin{cases} c_y - \frac{z_c}{g\cos\theta_b}\ddot{c}_y & \theta \geq 0 \\ c_y - \frac{z_c}{g\cos\theta_b}\ddot{c}_y & \theta < 0 \end{cases} \tag{4.4}$$

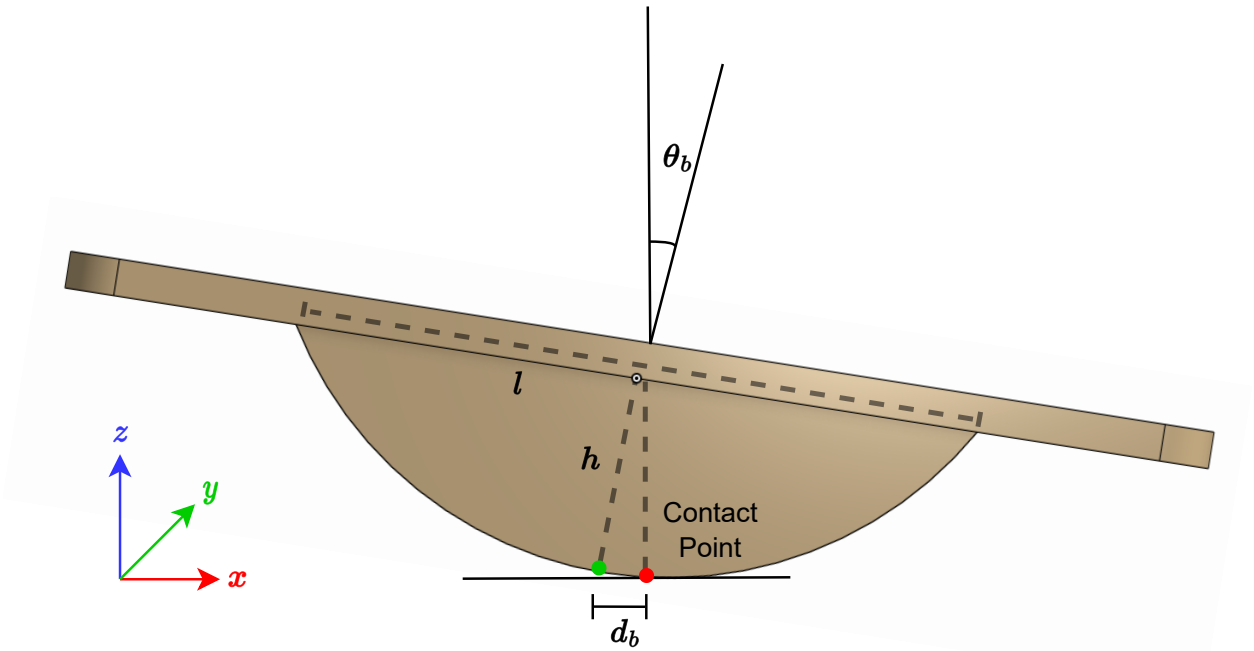


Figure 4.4: The balance board model used in this work.

where θ_b is the balance board inclination as depicted in Figure 4.4, c_x and c_y denote the center of mass position of the robot in the x and y directions, p_x and p_y are the ZMP in the x and y directions, z_c is the constant height of the cart-table model and g is gravity.

4.4 Humanoid Controller Design

We now describe the balancing controller design. The approach is depicted at a high-level in Figure 4.2 and this section will dive deeper into each of the blocks shown. Task-space references are generated by the stabilizer and fed into a TSID-based whole-body QP controller which produces the desired accelerations for each joint of the humanoid. By integrating twice, we get the desired joint positions to send to the robot position controllers.

4.4.1 Ankle Admittance

First we describe the ankle admittance controller. When balancing on an actuated balance board (or moving platform), it is important for the robot's ankles to be compliant and yield to the forces felt at the foot in order to keep its ZMP within its support polygon. In this work, we implement the following ankle admittance control:

$$\begin{bmatrix} \dot{\omega}_r^c \\ \dot{\omega}_p^c \end{bmatrix} = \text{PID}(p_d, p) \tag{4.5}$$

where p and p_d denote the ZMP position and ZMP reference position respectively and $(\dot{\omega}_r^c, \dot{\omega}_p^c)$ are the commanded ankle roll and pitch velocities. For our particular humanoid, there are actuators directly at the ankle roll and pitch and these commands are sent directly to these motors (after being integrated to get desired positions). The gains for the PID controller are used to tune the responsiveness of this controller. In particular, for balancing on an actuated balance board, a combination of Kp and K_i can yield good tracking performance with human-like ankle compliance to the board's motion.

4.4.2 Base Orientation Admittance

Maintaining the orientation of the base is important to ensure whole-body stability. In many humanoid controllers, the objective is to keep the base and torso upright but it can also be used to counteract force differences at the feet [124]. In this work, we implement a base upright task for the base pitch and a modified foot force difference admittance control for the base roll as an overall base admittance control.

The formulation for foot force difference control is:

$$\zeta_r^c = \zeta_r + \text{PID}((f_{Lz}^{qp} - f_{Rz}^{qp}), (f_{Lz}^{m} - f_{Rz}^{m})), \tag{4.6}$$

where ζ_r^c denotes the base roll angle command, and f_{Lz}^{qp}, f_{Rz}^{qp} denote the normal forces produced by a force distribution quadratic program and the superscript m denotes measured quantities. As noted in [42], an implicit side effect of this control strategy is that it increases CoM compliance. This control strategy is useful to shift the base when balancing on a frontal plane balance board. This control is also useful for humanoids without a torso roll joint (such as the REEM-C humanoid used in this thesis). The formulation to keep the base pitch upright is given as:

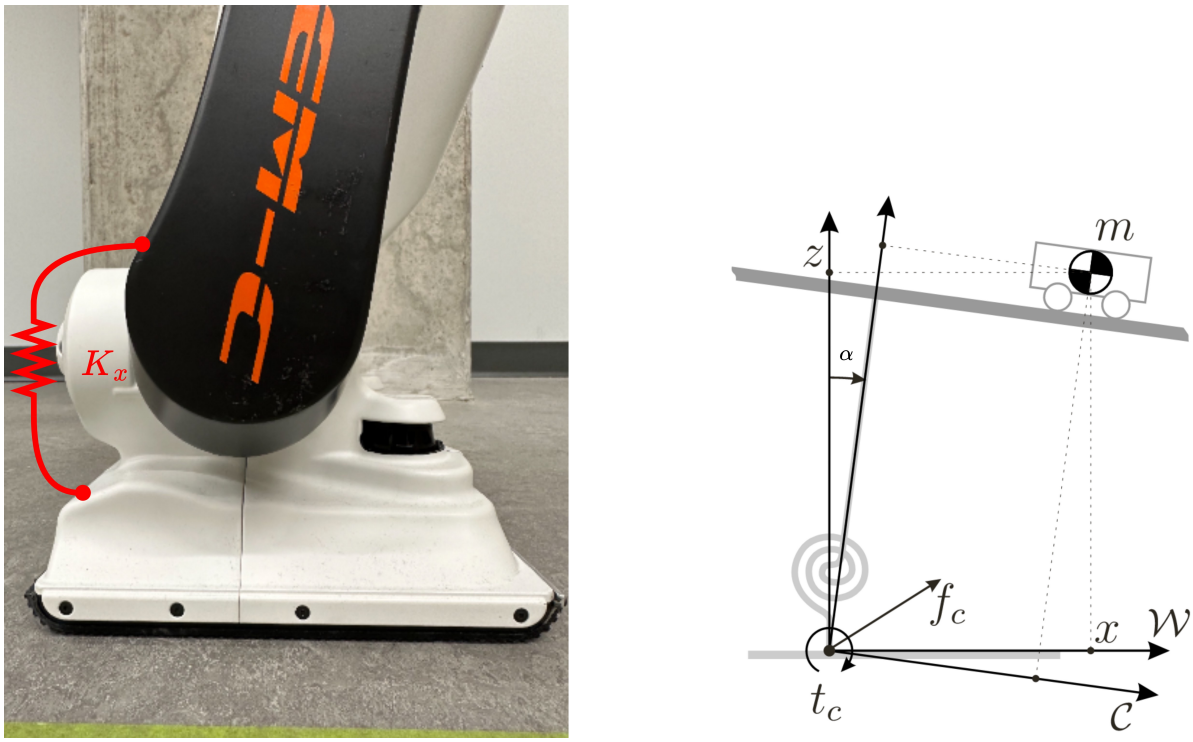
$$\zeta_p^c = \zeta_p + \text{PID}(\zeta_p^d, \zeta_p), \tag{4.7}$$

where ζ_p^c denotes the base pitch angle command and superscript d denotes a desired quantity. In this work, the desired quantity was set to keep the base pitch upright but this can be varied based on the application. For balancing on a balance board in the sagittal plane, we found that humans generally keep their torso and pelvis upright. This control strategy aims to emulate this.

4.4.3 CoM Admittance

For the center of mass admittance control we take a modified approach that incorporates model flexibility into the table cart model, as in [30]. We do this only for the x direction and use a formulation similar to the one proposed in [42] for the y direction.

For the x direction CoM control, we use a modified table-cart model which takes into account ankle flexibility as an offset angle α as in Figure 4.5. If we assume α is small the system equations can be linearized and we can write the state space equations of the system



(a) (b)

Figure 4.5: a) Unmodelled ankle flexibility b) The modified cart-table model taking into account flexibility (figure taken and adapted from Benallegue et al. [30]).

as:

$$\begin{bmatrix} \dot{c_x} \\ \dot{\alpha} \\ \ddot{c_x} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-g}{z_c} & \frac{-K_x}{mz_c} & 0 & 0 \\ \frac{-g}{z_c^2} & \frac{-K_x}{mz_c^2} & 0 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ \alpha \\ \dot{c_x} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{z_c} \end{bmatrix} \ddot{c}_x \tag{4.8}$$

We control this system using LQR to obtain the optimal control u. The spring constant K_x was not directly identified but tuned experimentally starting with the model identified for HRP-2 in [30]. In this work, α and $\dot{\alpha}$ are measured with an IMU on board the humanoid. This control works in tandem with the base orientation admittance which keeps the torso upright. This allows for accurate estimation of the angle α to reject disturbances (pushes) and to track a desired CoM reference.

For the y direction CoM control, the formulation is:

$$\ddot{c}_y = \ddot{c}_y^d - \text{PID}(z^*, z), \tag{4.9}$$

where z^* and z are the ZMP reference and ZMP position respectively. The reference z^* is generated from a divergent component of motion (DCM) feedback control law as reported in [42].

4.4.4 Contact Wrench Distribution

For the contact wrench distribution problem, we follow a similar approach to [234] using Newton and Euler equations to consider desired values c^d, $\ddot{c}^d$, $\dot{L}^d$ (angular momentum rate) along with z^*, p^d to obtain ground contact wrenches w^d_{LF}, w^d_{RF}. The redundancy from being in double support is handled by formulating a quadratic program to minimize each foot's wrench magnitude. We also consider unilaterality and friction cone equality constraints for this problem.

4.4.5 Task Space Inverse Dynamics

To briefly review the details from Subsection 2.3.2, TSID [3] is a whole-body quadratic program which sums task functions into a general cost function using user-defined weights for their priorities. The formulation has two layers, costs and constraints, whose trade-off can influence controller performance. The task errors in this controller formulation are implemented as acceleration-based tracking laws in task space. The whole-body QP is solved assuming the robot is in rigid contact with its environment.

In this work, we implement several tasks to track the task space references generated in previous sections:

- **Center of Mass:** to track the output of Equation 4.8 and Equation 4.9,
- **Contacts (6D):** to track the ground contact wrenches w^d_{LF}, w^d_{RF} from Subsection 4.4.4,
- **Feet Pose:** to track the output of Equation 4.5 to control the ankle roll and pitch of each foot,
- **Joint Posture:** to regulate the joint posture in a "half-sitting" stance with knees bent and arms to the side,
- **Trunk Orientation:** to track the output of Equation 4.6 and Equation 4.7 to control the roll and pitch of the floating base,
- **Angular Momentum (AM):** to regulate the angular momentum of the system as in Equation 4.10.

For the angular momentum task as formulated in TSID (reference set to $\mathbf{0}$), the dynamics are expressed as follows:

$$\begin{aligned} \dot{l}_c &= m[\ddot{c}^d + K_{D_{com}}(\dot{c}^d - \dot{c}) + K_{P_{com}}(c^d - c)] \\ \dot{k}_c &= k^d_c + K_{P_{AM}}(K^d_c - k_c) \end{aligned} \tag{4.10}$$

4.5 Experiments

In the simulation and real robot experiments, Pinocchio [43] is used to compute dynamic quantities and TSID [67] [3] is used to compute the task-space inverse dynamics whole-body quadratic program (Equation 2.10). We also make use of SEROW (State Estimation RObot Walking) [14] for base estimation. The performance of the stabilizaton approach is validated on the REEM-C simulation model in Gazebo and with real experiments on hardware. The REEM-C is a 1.65 m tall, 80 kg position controlled humanoid with 30 actuated joints (excluding the hands). In addition to the experiments shown, the attached video includes sagittal and frontal plane balancing, as well as perturbation recovery tests.

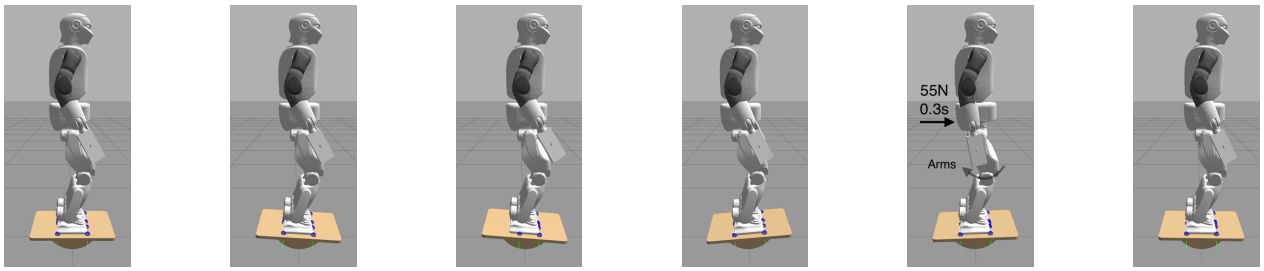
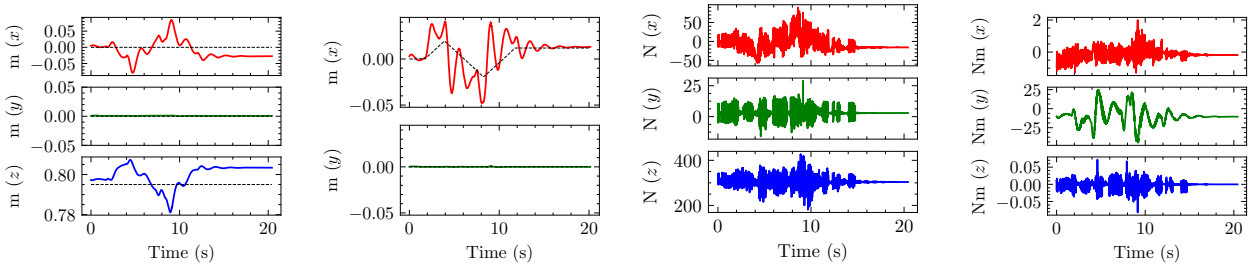


Figure 4.6: Frames (from left to right) of the REEM-C humanoid balancing on an actuated sagittal board rotating at 20 °/s, with a maximum inclination of 7°. On the second to last frame (on the right), the robot is pushed with a force of 55 N for 0.3 s and successfully rejects the disturbance by generating momentum in the opposite direction of the push with its arms.



(a) CoM Position (b) ZMP Position (c) Left Foot Forces (d) Left Foot moments

Figure 4.7: Graphs showing the CoM, ZMP and left ankle wrench measurements for the simulation sagittal board experiment. The references for the CoM and ZMP are shown using dotted black lines.

4.5.1 Implementation Details

The controller's cycle time was measured to be on average, 2.5 ± 0.5 ms, which is within the 5 ms control loop time for the REEM-C humanoid. This time is spread almost evenly between solving the wrench distribution quadratic program and the TSID quadratic program (using `eiquadprog` [2]). The wrench distribution is solved using the `qpsolvers` [8] interface and `quadprog` [7] [89].

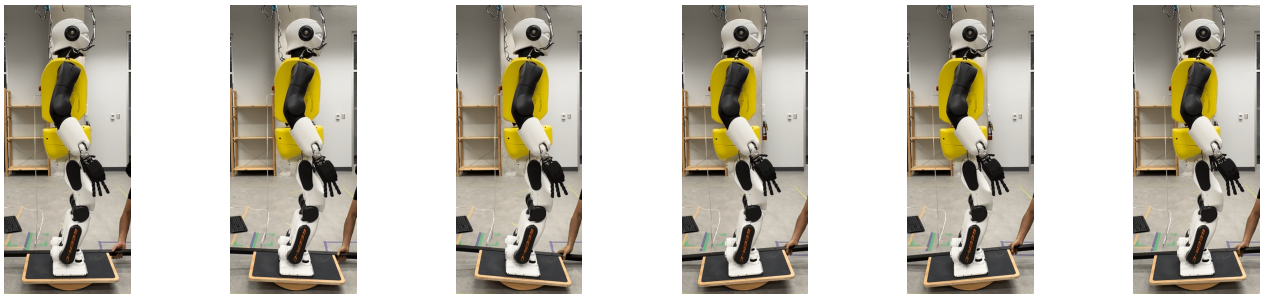
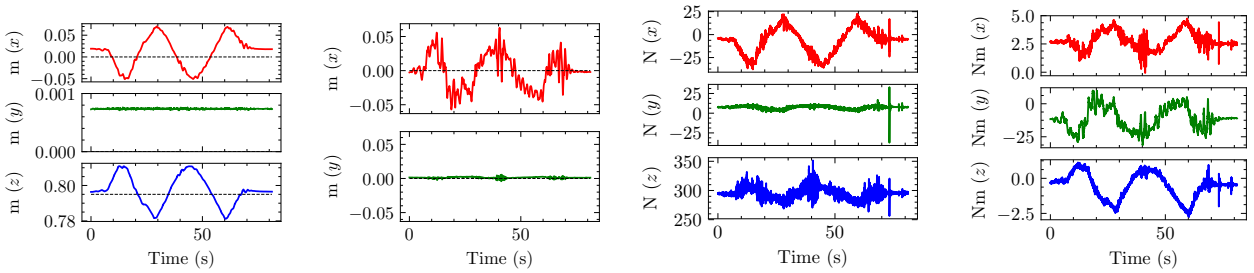


Figure 4.8: Frames (from left to right) of the REEM-C humanoid balancing on a sagittal board actuated by a human. The maximum inclination is 7°. We note that the robot has no knowledge of the board and it is treated as a disturbance in this case.



(a) CoM Position (b) ZMP Position (c) Left Foot Forces (d) Left Foot moments

Figure 4.9: Graphs showing the CoM, ZMP and left ankle wrench measurements for the real robot sagittal board experiment. The references for the CoM and ZMP are shown using dotted black lines (at $y = 0$).

The controller was written in Python using the ROS Melodic middleware (Ubuntu 18.04) and ran on a desktop Intel Core i7-10700F with 16GB of RAM. Whole-body position commands were sent to the REEM-C over a ROS topic.

For the TSID whole-body QP, the task gains were set as in Table 6.2. All task damping gains (K_{D}) were set to their critical value $2\sqrt{K_{\mathrm{P}}}$. For the priority levels listed, 0 indicates a constraint and 1 indicates a cost. The "Contact" and "Foot Pose" tasks are imposed on both feet of the humanoid independently.

With regards to state feedback, we note that since we are using a position-controlled robot, sensor data is only used to estimate the floating base and to calculate references for the various tasks in the TSID whole-body QP as shown in Figure 4.2. Feeding the measured state into the TSID whole-body QP can lead to instabilities in position-control [202].

4.5.2 Sagittal Plane Balancing in Simulation

The first experiment to test the stabilization method was a sagittal plane balancing problem on an actuated balance board. The problem setup starts with the balance board completely horizontal before gradually changing inclination about its pivot point up to 7° one direction. It then goes the opposite direction declining 7° before returning to a near horizontal state.

Table 4.1: The TSID Task Gains for the balance problem. The same gains are used for sagittal and frontal boards.

Task	Weight	Priority	K_P gain
Torque Bounds	1	0	-
Joint Bounds	1	0	-
Center of Mass	1	0	20
Contact (6D)	1	0	30
Foot Pose	1	1	300
Joint Posture (30 DoFs)	1×10^{-2}	1	all equal (1)
Trunk Orientation	1	1	20

The speed of the motion during this test was 20°/s. Additionally, at the 9 s mark, there is a perturbation at the humanoid's waist of 55 N for 0.3 s.

Figure 4.6 shows snapshots of the simulation results where the robot successfully balances itself on top of the actuated balance board. The ankle admittance control (Equation 4.5) allows the robot to be compliant to the board's motion and track the contact point of the balance board with the ground. We see from Figure 4.7a that the center of mass stays bounded within the support polygon limits of the humanoid (which are roughly 10 cm from the center to each foot edge for the REEM-C in the initial configuration). The ZMP also tracks the reference, with some noise and oscillation (Figure 4.7b). This is expected behaviour given the sudden start and direction switching of the balance board. The forces and moments at the left ankle are regulated by the controller as seen in Figure 4.7c and Figure 4.7d. The right ankle forces and moments are very similar to the left ankle measurements and are left out for clarity. It is interesting to notice that the momentum task which allows for arm swinging, damps the 55 N push at around 9 s with little deformation in the CoM position. This shows the importance of controlling momentum (both linear and angular) and is something that should be explored further in future work for different types of balance boards and stabilization situations. The CoM height also shifts slightly from the reference which is expected since we allow this to be free for the QP solver to choose.

4.5.3 Sagittal Plane Balancing on Hardware

In this scenario, we run the controller on the real REEM-C hardware. With minimal changes to the gains, we could test the actuated board scenario in the sagittal plane. For this test, we moved slower but at a constant rate of 10 °/s to validate the controller. The snapshots in Figure 4.8 show that the REEM-C can successfully balance on the board and also perform all task objectives such as ankle admittance, CoM stabilization and base orientation correction. As can be seen in Figure 4.9a, the CoM stays bounded in the x direction and similar to simulation trials, there is CoM height variation as an additional stabilization strategy. One thing we note is that for the real hardware trial, we did not estimate the contact point of the board and the ground and thus the balance board is treated as an unmodelled disturbance. The references for the CoM and ZMP were set to $\mathbf{0}$ to keep the ZMP at the center of the

foot. This strategy worked to stabilize the system including bounded forces and torques as measured by the ankle force-torque sensors (Figure 4.9c, Figure 4.9d).

4.5.4 Frontal Plane Balancing

The control approach described in Section 4.4 also works for frontal plane balancing. The snapshots shown in Figure 4.10 show the REEM-C successfully balancing as the board is actuated manually. All task objective were satisfied and notably, foot force difference control came into play to equalize the normal force between the two legs as the board oscillated back and forth. This helped keep the base upright as can be seen in frames 2 and 3 from the left, in Figure 4.10.

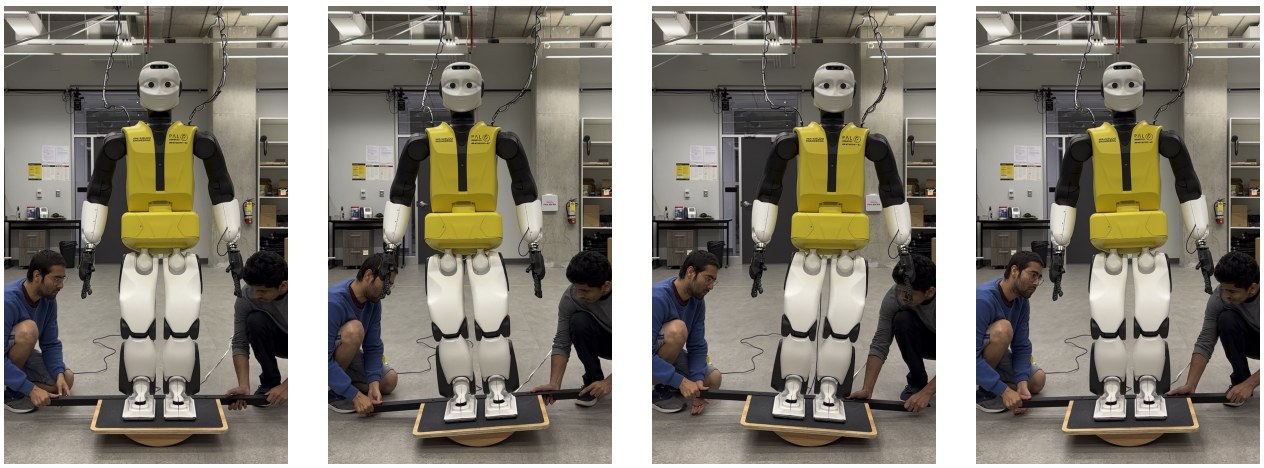


Figure 4.10: Frames (from left to right) of the REEM-C humanoid stabilizing on an actuated frontal plane balance board.

4.5.5 Frontal Plane Stabilization against Large Perturbation

In this last test, the robot started on the frontal plane board leaning to one side before a large disturbance was applied (by a human foot) to the board's steel bar. This briefly sent the robot airborne before it landed back on the board. Here, the ankle admittance and force difference controller was able to stabilize the robot by admitting to the forces/torques felt upon landing back in double support. The peak force felt at the right ankle force-torque sensor was **1750N**. The video associated with this chapter shows this experiment in slow motion for better clarity: https://youtu.be/X61gbXRYg50.

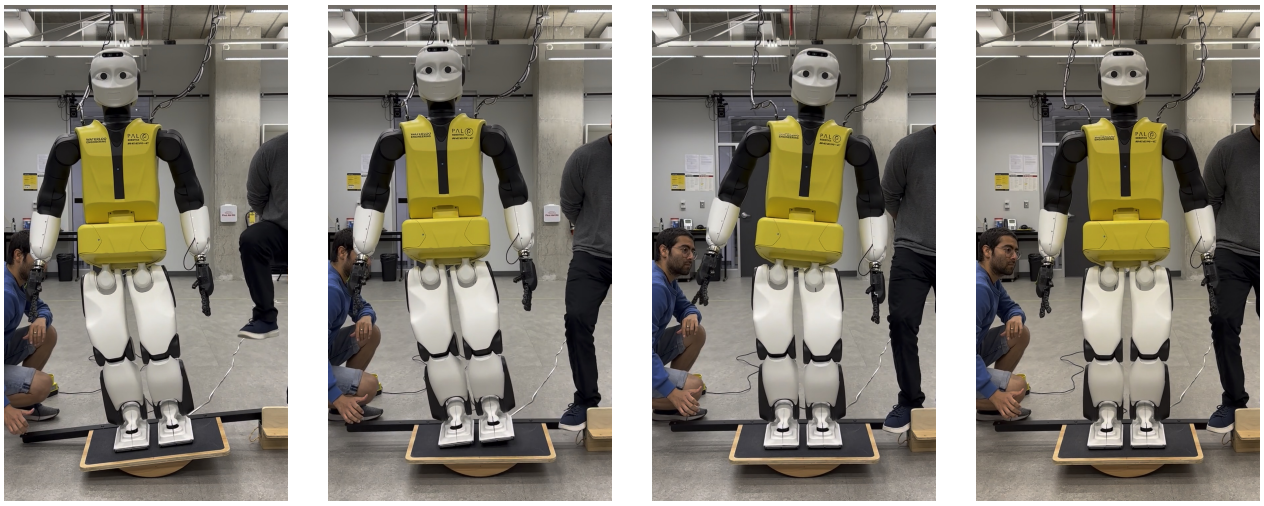


Figure 4.11: Frames (from left to right) of the REEM-C humanoid stabilizing in the frontal plane after a large step-like disturbance is applied to the balance board by a human foot.

4.6 Conclusions and Perspectives

In this work, we describe a formulation for balancing on actuated balance boards in the frontal and sagittal plane not only in simulation, but also on real humanoid hardware. The proposed stabilizer incorporates various strategies from the literature (ankle admittance, CoM admittance, base pitch admittance, foot force difference admittance, contact wrench distribution and angular momentum regulation) and integrates them in a new formulation. TSID is used to unify the strategies in a whole-body QP to realize the control approaches at the acceleration level.

By utilizing an actuated balance board, we were able to intentionally create disturbances to the robot in a controlled environment. This enabled us to validate our proposed approach and assess the effectiveness of this type of controller in scenarios where the environment is rapidly changing. In future developments, it could be used to provide challenging disturbance profiles based on real-world scenarios such as maintaining balance on a moving boat.

Future work will involve improving the performance of the controller to balance on an unactuated balance board. While simplified models are tractable and elegant to formulate, our focus will shift to considering the whole-body to generate high-fidelity references for whole-body QP solvers. This work will also be extended to a general purpose stabilizer to allow humanoids to balance in more challenging scenarios, such as while operating personal transporters [198].

Chapter 5

Motion Planning for Bimanual Manipulation of Humanoid Robots based on a Novel Manipulability-Stability Metric and Object-Specific Constraints

Foreword

A portion of this chapter was accepted and published at the 2022 IEEE International Conference on Humanoid Robots. The other half is currently being revised for resubmittal at the IEEE Transactions on Robotics journal. The citations are as follows:

- W. Thibault*, **V. Rajendran*** and K. Mombaur, "Bimanual Manipulation Workspace Analysis of Humanoid Robots with Object Specific Coupling Constraints," 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Ginowan, Japan, 2022, pp. 141-148, doi: 10.1109/Humanoids53995.2022.10000183. *Equal first authorship. [224]
- **V. Rajendran***, W. Thibault*, K. Mombaur, "Motion Planning for Bimanual Manipulation of Humanoid Robots based on a Novel Manipulability-Stability Metric and Object-Specific Constraints," in IEEE Transactions on Robotics, 2024, Revise and Resubmit. *Equal first authorship. [201]

Minor edits involving formatting were made to adhere to thesis guidelines but otherwise, the content was used verbatim.

Within the context of this thesis, this work introduces a novel manipulability-stability metric and motion generation pipeline for bimanual manipulation. This work was extended and adapted in Chapter 6 to enable a humanoid to operate the handlebar of a Segway. The video associated with this chapter is at the following link: https://youtu.be/dALPqitEdOA.

5.1 Introduction

With the integration of robots into our daily lives, bimanual or dual-arm robotic manipulation has become increasingly important to manipulate large, heavy and complex shaped objects. Krebs et al. [135] identified a number of common bimanual tasks in their dataset created from human motion capture studies including sweeping with a broom, opening juice containers, mixing ingredients for cooking and rolling with a rolling pin. Other tasks requiring bimanual manipulation include lifting boxes, pushing carts and carrying trays [250, 232, 142]. These tasks are well suited for dual-arm robots, especially humanoids, when operating in real-world, human settings (e.g., in a home or factory). However, the complex nature of these high degree of freedom, coordinated, multi-contact tasks result in many challenges in motion planning and control. Bimanual manipulation tasks can be distinguished in terms of hand coordination, coupling and symmetry/asymmetry of the two hands with respect to the object, as suggested in a recent taxonomy developed by Krebs and Asfour [134]. In this work, we focus on coordinated, tightly coupled bimanual manipulation cases in which the object imposes clear constraints between the position and orientation of the two hands. We consider objects that introduce symmetric as well as asymmetric grasps, as shown in Figure 5.1.

5.1.1 Related Work

Recent Works

The field of bimanual manipulation covers a wide breadth of tasks of varying goals, objects and constraints. Identifying and classifying these tasks, that generally are based on human bimanual tasks, has been studied to better understand them and inform the development of such motions on bimanual robots [70, 135, 134]. In many recent works for bimanual manipulation learning has been a dominant direction of research as these motions are very complex to define. Imitation learning and learning from human demonstrations has been used for tasks like stir-fry cooking [155], pouring and box manipulation [156], and furniture manipulation [242]. Other learning based techniques have been used in bimanual manipulation including learning from configuration and operational space constraints [211], perception based learning [47, 46], multi-agent deep reinforcement learning [157], combined offline and online reinforcement learning [216], and neural network control [245]. Due to the human-like nature of bimanual manipulation teleoperation methods are another popular method of motion generation allowing humans to control the arm positions with position tracking and EMG sensors [185, 139]. Human motions can also be used to develop movement primitives used in bimanual tasks [84]. Traditionally, bimanual manipulation research includes work on reachability and workspace analysis, planning and control, which are reviewed in the following sections.

Reachability and Workspace Analysis

When dealing with multiple manipulators or kinematic trees, such as with humanoids and dual-arm robots, reachability is often considered. Reachability maps have been shown to be

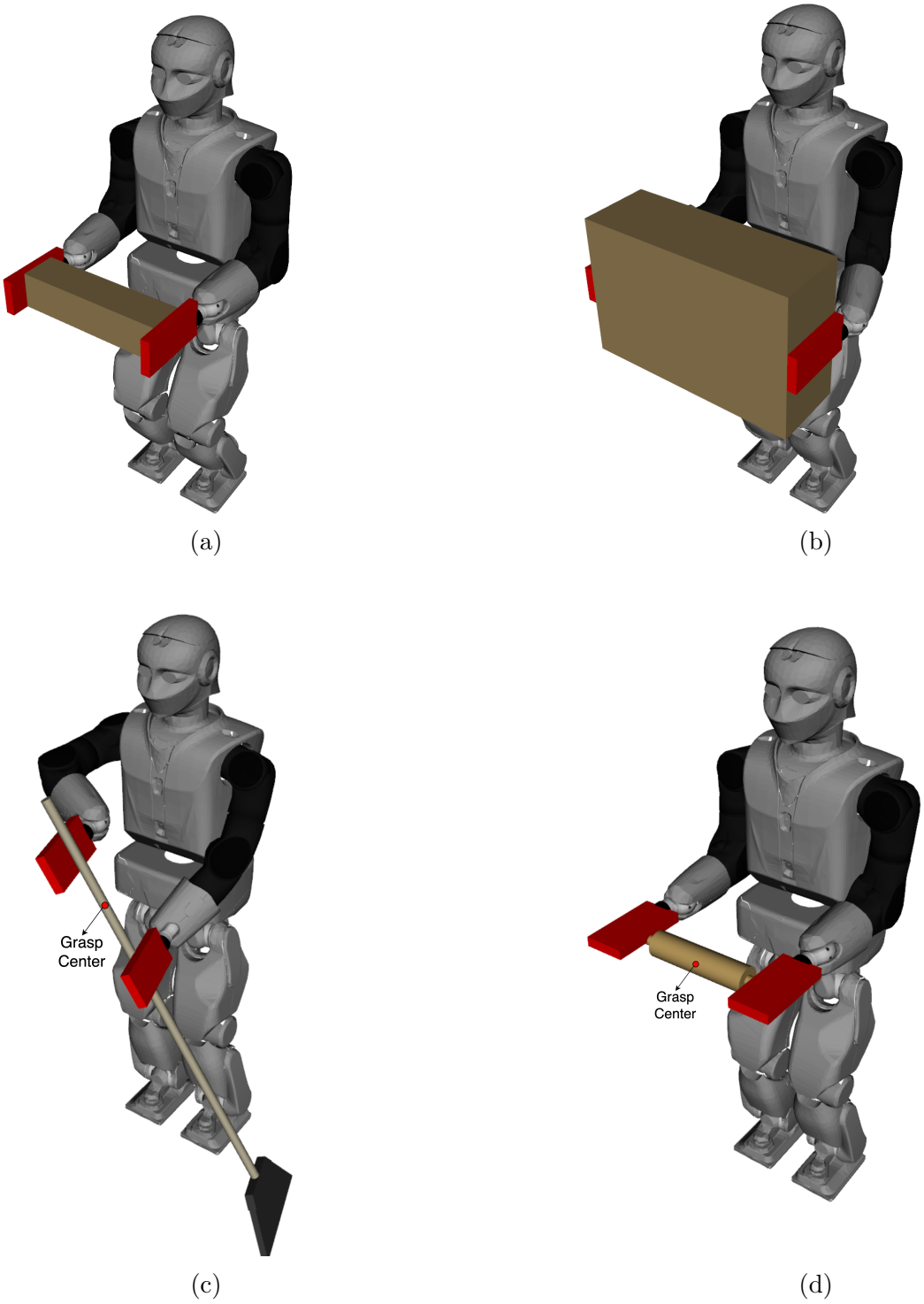


Figure 5.1: Examples of objects introducing tightly coupled bimanual manipulation scenarios addressed in this work: a) a small 1 kg box b) a large 7 kg box c) a sweeping broom (with a left-handed grasp) d) a rolling pin.

useful when determining valid poses planning manipulation related tasks for bimanual or humanoid robots [172] including inverse reachability maps allowing for determining base poses for specific hand poses [247, 249, 229]. Another perspective on reachability is the capability map developed by Zacharias et al. that allows for the visual representation of valid grasps at reachable points, particularly useful when the robot has a fixed base [254]. Workspace analysis extends reachability maps by evaluating the quality of poses that are reachable. A common metric considered is manipulability, a measure of how close a robot manipulator is to singularity [251]. An alternative metric proposed by Leibrandt et al. measures dexterity using joint-space configurations to evaluate the task-space for a more global metric [146]. In the case of bimanual workspace analysis, manipulability becomes more challenging to consider due to each arm having its own manipulability and it is often desirable for both arms being able to operate in a similar workspace. Bimanual manipulability has been studied as the intersection of the two manipulability ellipsoids [145]. Other bimanual manipulability approaches include using individual arm manipulabilities for search-based approaches to optimize dual arm grasping manipulabilities [231] and an extended manipulability metric based on the Yoshikawa manipulability metric using an augmented Jacobian matrix to include both arms with cost terms for joint limits and obstacle avoidance [230, 227]. However, bimanual manipulability is much easier to consider with robots that have a wheeled base design as stability is not a concern. Static stability considerations for humanoids have been shown in reachability maps in terms of a ratio of the distance of the center of mass to the edge of the support polygon [116] and have been fused with extended manipulability metrics as a product for a combined metric when determining whole-body affordances [117].

Planning

Bimanual, or dual arm, planning is an important part of motion generation when manipulating larger objects or more challenging manipulation is required. Orientation constraints are often important when tipping objects or spilling contents is of concern like manipulating trays leading to planning approaches that focus on object orientation [236, 59, 141, 142, 140] and wrench balancing through sliding gripper motions [62]. Other works focus on large objects like furniture [148] or objects with more challenging shapes to grasp leading to grasp oriented planning [228, 217, 183]. Bimanual planning works have used a variety of planning approaches including motion primitive search-based planning [59], trajectory optimization and analytic inverse kinematics approaches [61, 236], sampling based techniques [228, 148] and reactive planning [142] to achieve feasible plans these constrained manipulations necessitate. In the case of semi-deformable bimanual manipulation a benchmark for the performance of planners was proposed [45]. However, much of the work still focuses around fixed manipulators and wheeled based robots. Manipulation planning with legged humanoids generally centers on successful end effector planning rather than a bimanual object focused sense [115].

Control

Bimanual manipulation scenarios are often used because the manipulation task involves a large, heavy object or a manipulation too complex for a single manipulator. This motivation

with the need to perform tasks in cluttered environments and near or with humans leads to many control frameworks leveraging impedance or admittance control to have compliant motions [83, 237, 150]. These frameworks focus on compensation for the load of the object being manipulated along with disturbance and interaction from humans. Other control methods include simpler torque control for optimization based trajectories [98], adaptive master-slave control for hybrid force-position control [29] and cooperative task-space control methods using a dual quaternion representation [17]. Overall, force and constraint considerations are key components of bimanual control methods. Another important consideration for control is the prioritization of tasks, which is often resolved using a sequential approach based on the prescribed priority. Sathya et al. propose the an MPC approach where weighted exact penalties are used to ensure strict priority [206]. This method produced equivalent prioritization to a sequential approach with a reduced computational burden for a dual arm pick and place task.

5.1.2 Contributions

Despite the extensive research already performed on bimanual manipulation, we have identified a few missing pieces from the perspective of dynamically moving bipedal humanoid robots. The limitations that tight bimanual object grasps impose on the reachable workspace and manipulability of humanoids, the constraints introduced by the object on the hands and collision avoidance have to be explicitly considered to clearly assess bimanual manipulation. In order to be able to plan, optimize and control dynamic (loco-)manipulation motions or to evaluate key points along such motions, the weight and motion of the robot and the manipulated load have to be considered along with whole-body stability beyond the static case. A planning and control pipeline using these considerations would enable highly manipulable and stable humanoid motions. Knowing this, the contributions of our work presented here are:

C.1 A new bimanual manipulability-stability metric that considers the manipulability of a bimanual task and its static and dynamic whole-body stability,

C.2 A workspace analysis with object specific considerations such as mass, collisions and end effector constraints,

C.3 Computations and visualizations of the metric in a variety of workspace scenarios with different example objects and velocities,

C.4 Workspace comparison of two kinematically and dynamically different humanoids, REEM-C and TALOS, in the same manipulation scenario,

C.5 Cost-based planning with proposed bimanual manipulability-stability metric using BiRRT for object constrained, collision-free whole body inverse kinematic planning,

C.6 A task space inverse dynamics whole body quadratic programming control strategy to consider object dynamics during manipulation,

C.7 Simulation and real robot evaluation of planning and control pipeline on the REEM-C humanoid.

5.2 Workspace Metrics

In this section the metrics used to analyze the workspace of a humanoid are developed, specifically related to manipulability and whole-body stability. Using these individual metrics, a combined metric is defined to analyze the quality of each pose/voxel in the workspace volume of a humanoid.

5.2.1 Manipulability

A frequently used metric for measuring manipulability is the volume of the manipulablility ellipsoid, also referred to as the Yoshikawa manipulability [251]. For a robotic manipulator with n degrees of freedom in a task space of $q \in \mathbb{R}^m$ with $m \leq n$ and Jacobian J, a square, symmetric and positive definite matrix A can be defined as:

$$A = JJ^T \in \mathbb{R}^{m \times m}. \tag{5.1}$$

The eigenvalues λ_i of A correspond to the squares of the singular σ_i values of J. A describes the mapping of a unit sphere in the joint velocity space to end effector velocities, and serves to define the manipulability ellipsoid. The lengths of the half-axes of this ellipsoid are $\sqrt{\lambda_i}$, and the manipulability measure μ for a single manipulator is computed based on its volume as:

$$\mu(A) = \sqrt{\lambda_1 \lambda_2 ... \lambda_m} = \sqrt{\det A}, \tag{5.2}$$

$$\mu(J) = \sqrt{\det JJ^T} = \sigma_1 \sigma_2 .... \sigma_m, \tag{5.3}$$

where larger values indicate higher manipulability [159]. When considering manipulability in a bimanual manipulation situation, this becomes more challenging, since both end effectors have to be manipulable enough to support a given motion. We therefore extend the metric to select the less manipulable arm which will be the limiting factor:

$$c_m = \min\left(\mu(A_{\text{LeftArm}}), \mu(A_{\text{RightArm}})\right). \tag{5.4}$$

For non-anthropomorphic robots (with more than two arms), this metric can be extended as such:

$$c_{m_{\text{multi-arm}}} = \min\left(\mu(A_{\text{Arm}_i}), i = 1, \ldots, n\right). \tag{5.5}$$

5.2.2 Stability

A whole-body stability metric is another important quality value to consider for workspace analysis of humanoids. A reachable and manipulable pose is only useful if the humanoid is also stable. Often, a simple confirmation of stability for a static pose is checking if the ground projected center of mass (GPCM) lies within the support polygon. In [116], a quality metric for stability was proposed based on the distance of the GPCM to the edge of the support polygon. For non-static poses, the capture point can be used as it considers the orbital energy of the linear inverted pendulum model (LIPM) when the center of mass is moving at a velocity to determine if a step needs to be taken [191].

For a 3D LIPM at position r_{CoM}, moving at a velocity of $\dot{r}_{\text{CoM}}$, the instantaneous capture point, r_{ICP}, can be written as follows:

$$r_{\text{ICP}} = P\,(r_{\text{CoM}} + \frac{\dot{r}_{\text{CoM}}}{\omega_0}), \tag{5.6}$$

where $P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is a projection matrix and $\omega_0 = \sqrt{\frac{g}{z_0}}$, where z_0 is a constant center of mass height and g is acceleration due to gravity. Note that if the center of mass is not moving, static stability is preserved [193].

Using the capture point of the robot, a stability metric can be calculated in situations where the center of mass is experiencing a velocity. This metric can be defined as the minimum distance to the edge of the support polygon and is denoted c_s in this work. In the situation where the capture point does not lie within the support polygon, the pose is considered unstable for the current stance and not a valid reachable pose without taking another step. Note that in the case of static poses, the capture point criterion automatically reduces to the static stability criterion on the GPCM.

5.2.3 Combined Manipulability-Stability Metric

In order to have a single metric to evaluate different reachable poses in a workspace, a combined manipulability-stability metric must be defined. To create such a metric, Equation 5.4 and c_s should be normalized by the largest value found for each metric in the workspace analysis to allow for comparison on a similar scale. The normalization is written as:

$$c'_{m_i} = \frac{c_{m_i}}{\max(c_{m_0}, c_{m_1}, ..., c_{m_n})}, \tag{5.7}$$

$$c'_{s_i} = \frac{c_{s_i}}{\max(c_{s_0}, c_{s_1}, ..., c_{s_n})}, \tag{5.8}$$

where n is the number of reachable poses/voxels in a workspace volume and i indicates the i^{th} reachable voxel from 0 to n. Normalization is a crucial step to ensure a standardized metric can be used when optimizing the workspace for better stability and manipulability, as each metric has its own unique scale. To maintain consistency, only feasible configurations are considered, ensuring the metric stays within a reasonable range. Although some feasible poses

might have poor manipulability or stability, the resulting metrics do not become unreasonably small in practice.

With the normalized metrics in Equation 5.7 and 5.8, a combined manipulability metric at a reachable voxel can be defined as a weighted sum of the two metrics as follows:

$$c_{v_i} = \alpha c'_{m_i} + \beta c'_{s_i}, \tag{5.9}$$

where α and β are weights between 0 and 1 that together sum to 1. Equal weighting for the two metrics is used for the initial evaluation of the metric in this work. In the implementation of this metric, detailed in Section 5.3, we perform a hard check and discard any reachable poses that are unstable.

This combination of metrics to represent the stability and manipulability for a humanoid has not been explored previously and can offer insight into quality of a whole-body bimanual manipulation pose. High values would indicate if the pose of the robot is highly stable while the arms also remain very manipulable for a given bimanual grasp.

5.3 Workspace Analysis Methodology

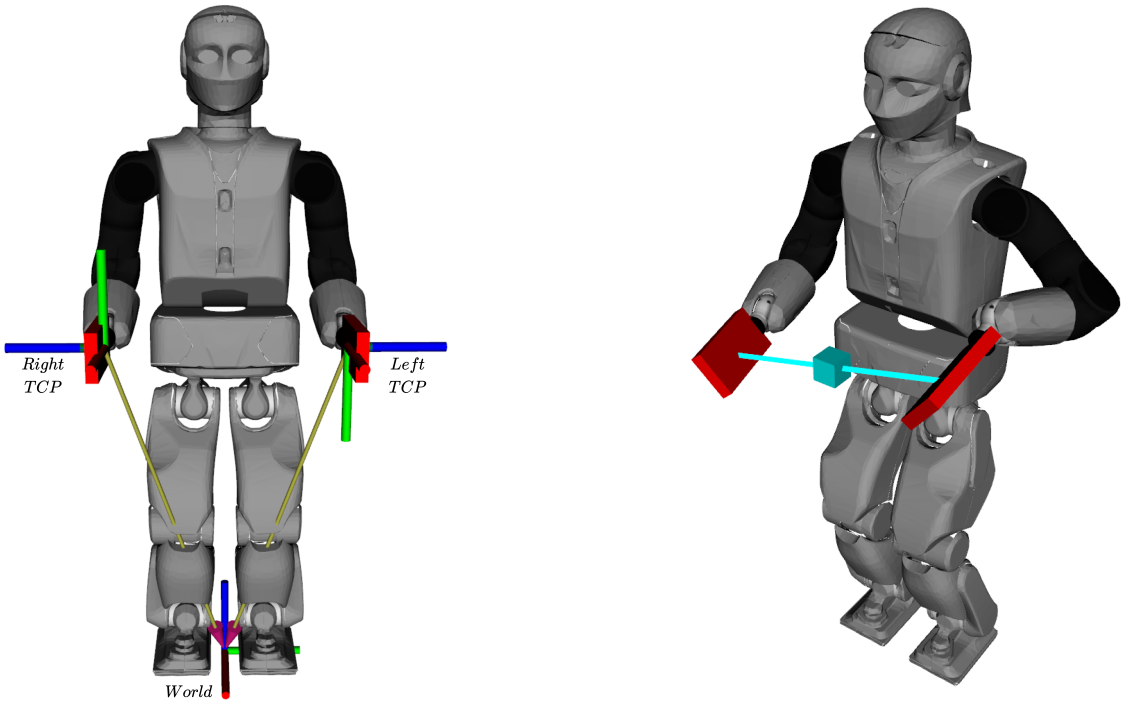
5.3.1 Whole-body Inverse Kinematics

This work considers humanoid robots, which kinematically have a tree like structure, unlike the chain structure of typical articulated manipulators. This requires a different approach to solving the inverse kinematics (IK) problem of reaching a certain pose in space, especially when considering multiple end effectors and whole-body stability. In this work, we use the EXOTica (**Ext**ensible **O**ptimization **T**oolset) framework by Ivan et al. [111] to solve the whole-body inverse kinematics problem for the humanoid using an optimization approach. The optimization problem is set up using task maps that build the cost function and constraints. The task maps include considerations for joint limits, end effector positions and center of mass position, among others. The task maps used for the whole-body IK optimization are described in Table 5.1. To formally define the whole-body IK problem, we refer to [248]:

$$q^* = IK(q_{seed}, C), \tag{5.10}$$

where C is the set of constraints and tasks (as described above) and q_{seed} is a given seed configuration (or start configuration).

Using these task maps, queries for poses of the tool center point (TCP) of the left and right hand of the humanoid are performed. In this work, we predominantly use the REEM-C humanoid "Seven" by PAL Robotics, from the University of Waterloo. Figure 5.2a shows the model of Seven with the left and right TCPs labeled. The task map weights were set with joint limit avoidance as highest priority, followed by stability considerations (restricting the position of the feet, avoiding unnecessary CoM motion) and finally achieving desired end



(a) (b)

Figure 5.2: a) The REEM-C humanoid with $World$ frame and left and right TCP frames labeled. Here x is in red, y is in green and z is in blue. b) For the general workspace generation case, a single point on the workspace grid encodes the center between the two end effectors of the robot.

effector poses. For the IK procedure, we used the `ExoticaWholeBodyIKConfig` problem type with the `SciPyEndPoseSolver` (using the L-BFGS-B optimization method).

5.3.2 Generation of a General Bimanual Workspace Map

For a general bimanual workspace analysis, that will serve as a reference for the results to be presented, the method applied is similar to that of [227]. No specific object is defined in this case, so constraints on distance and orientation of the hands are not applied. For each end effector center point, a range of hand distances and positions around the point are evaluated. This workspace generation procedure discretizes a task space in front of the robot, the area where humanoids perform the majority of their manipulation related motions, into a 6D voxelized workspace. A general workspace analysis is performed for points in a 3D grid and for a series of spherical coordinates around each point. A 3D position vector describes the center between the two end effectors and a spherical coordinate (r, θ, ϕ) encodes the distance between the end effectors as shown in Figure 5.2b. This description does not store any information about the relative end effector orientation. By removing the orientation of the end effectors, a significant speedup is achieved due to the simplicity of

position-only inverse kinematics. We define $j = [v_0, v_1, v_2]$ as the position of a 6D voxel v that encodes the relation between the two end effectors of the humanoid, represented in the $World$ frame (as pictured in Figure 5.2a). From the spherical coordinates $s = [r, \theta, \phi]$ a Cartesian position $j' = [x', y', z']$ can be derived as $x' = r\,sin(\theta)\,cos(\phi)$, $y' = r\,sin(\theta)\,sin(\phi)$ and $z' = r\,cos(\theta)$. The Cartesian positions of the left and right TCPs are then given as: $TCP_{left} = j + j', TCP_{right} = j - j'$. Therefore, the spherical coordinates allow the hands to be positioned around the point at different distances, where the TCPs are always considered to be across from each other. During generation, the IK solution is considered valid if the bimanual configuration is reachable, the capture point is within the support polygon, the configuration is collision-free and it respects all joint limits. These hard checks are performed because the cost-based optimization approach for whole-body inverse kinematics, detailed in Subsection 5.3.1, does not have any hard constraints and may converge to infeasible solutions. It is important to note that these checks are independent of any manipulability-stability metric that is measured, as a result every logged solution is a stable and realizable pose.

5.3.3 Generation of Bimanual Workspace Maps under Object-induced Constraints

The workspace maps generated by the approach presented in the previous section give an overly optimistic picture of the situation when it comes to bimanual manipulation of a given object. An object introduces different kinds of constraints:

- the distance of the hands are fixed to the grasping width of the object,
- the orientation of the hands with respect to each other is determined by the grasp required for the object,
- for certain objects, there may be additional constraints on their orientation, e.g. crates that have to be carried upright (fixed roll and pitch).

In this section, we therefore consider an object specific workspace analysis for humanoid bimanual manipulation. In a similar manner, the workspace generation produces a discretized area in front of the robot; however, the 6D voxelized parameters now consider the position and orientation. The object specific workspace requires specific offsets for each TCP based on the grasp for the object rather than using spherical coordinates with position-only IK. In this case, full position and orientation IK is required to meet the grasp requirements. The addition of orientation then allows for the consideration of the object rotation at each position in the workspace, which may be useful depending on the task the robot is performing.

Algorithm 1 provides the details on generating the object specific workspace maps. The algorithm takes as input the range of each voxel dimension, the discretization, the velocity of the center of mass and the object geometry. It then computes the metric c_v (Equation 5.9) for each voxel in the defined workspace. Note that in this workspace generation, the object influences the kinematic grasp, the dimensions of the object are considered for collision-checking and the mass of the object is considered for stability. The hands are simplified end effectors for faster computation, and we assume simplified fixed contact grasps.

Algorithm 1 Object Bimanual Workspace Generation

Input: x, y, z, $roll$, $pitch$, yaw ranges and discretization steps, $\dot{r}_{CoM}$, `object_geometry`
Parameters:

for $x \in \{x_{min}, \ldots, x_{max}\}$ **do**
  for $y \in \{y_{min}, \ldots, y_{max}\}$ **do**
    for $z \in \{z_{min}, \ldots, z_{max}\}$ **do**
      for $roll \in \{roll_{min}, \ldots, roll_{max}\}$ **do**
        for $pitch \in \{pitch_{min}, \ldots, pitch_{max}\}$ **do**
          for $yaw \in \{yaw_{min}, \ldots, yaw_{max}\}$ **do**
            $j \leftarrow$ Vector(x, y, z, roll, pitch, yaw)
            $j_l, j_r \leftarrow$ calcGraspOffsets(`object_geometry`)
            $q \leftarrow$ calcBothArmsIK($j + j_l$, $j + j_r$)
            stateValid $\leftarrow$ isStateValid(q)
            if stateValid **then**
              $c'_m \leftarrow$ calcNormManipulability(q)
              $c'_s \leftarrow$ calcNormStability(q, $\dot{r}_{CoM}$)
              $c_v \leftarrow \alpha c'_m + \beta c'_s$
              writeVoxelToDatabase($x, y, z, roll, pitch, yaw, c_v$)

5.3.4 Workspace Visualization

The generated results for c_v at each voxel in the workspace are visualized by a color spectrum ranging from low quality in blue to high quality in yellow/orange (going through green as medium quality). Any unreachable or unstable voxel is plotted as gray. As multiple results appear at each 3D voxel, the average result is displayed through an overlap of voxels by applying a transparency.



Figure 5.3: The metric color range used throughout this work.

Table 5.1: EXOTica Task Maps used for whole-body inverse kinematics.

Task Map	Formulation
Joint position	$\Phi_{\text{Ref}}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x_{ref}}$
Joint limits	$\Phi_{\text{Bound}}(x) = \begin{cases} x - x_{min} - \epsilon, & \text{if } x < x_{min} + \epsilon \\ x - x_{max} + \epsilon, & \text{if } x > x_{max} - \epsilon \\ 0, & \text{otherwise} \end{cases}$
End effector frame	$\Phi_{\text{EffFrame}}(x) = \boldsymbol{M}_A^B$ where $M_A^B \in SE(3)$
End effector position	$\Phi_{\text{EffPos}}(x) = \boldsymbol{P}_A^B$ where $P_A^B \in \mathbb{R}^3$
End effector orientation	$\Phi_{\text{EffRot}}(x) = \boldsymbol{R}_A^B$ where $R_A^B \in SO(3)$
Center of mass	$\Phi_{\text{CoM}}(\boldsymbol{x}) = \left(\sum_i (\boldsymbol{P}_{\text{CoM}_i}^{\text{world}} m_i)\right) \Big/ M$

5.4 Workspace Analysis Results

In this section we present various workspace scenarios to illustrate our proposed metric in the context of bimanual manipulation with humanoids.

5.4.1 General Workspace Maps

For the general case, generated as described in Subsection 5.3.2, a $0.8\,\mathrm{m}^3$ workspace with 5 cm voxel discretization is considered. The workspace is predominantly in front of the robot where most manipulation tasks take place and no CoM velocity is considered. Figure 5.4 shows a simple sensitivity analysis of the full workspace considered in the first row and a horizontal cut through in the second row as α and β are varied. Figure 5.4a and 5.4d show the manipulability distribution (i.e., $\alpha = 1$ and $\beta = 0$ in Equation 5.9). As can be seen, voxels closer to the REEM-C are of higher manipulability while voxels near the edge of the workspace are of lesser manipulability. This is to be expected given that the further the arms extend, the closer they are to singularity (and the less manipulable the configuration will be). Figure 5.4b and 5.4e shows the stability distribution of the workspace ($\alpha = 0$ and $\beta = 1$ in Equation 5.9). The REEM-C is relatively stable for this cut through with voxels near the edges being of lower quality. This is expected given that the REEM-C would need to attempt to shift its CoM in order to reach some of these poses, reaching the edge of the support polygon. Figure 5.4c and 5.4f show the combined manipulability-stability metric with weightings, $\alpha = 0.5$ and $\beta = 0.5$. Configurations nearer to the REEM-C are of higher quality. This visualization gives an overview of the most stable and manipulable configurations in the considered workspace. Of all the queried voxels, REEM-C was able to reach 22 % of the workspace. Other combinations of manipulability-stability could be obtained with different α and β weights to prioritize manipulability or stability more depending on the application.

5.4.2 Workspace Maps with Varying Center of Mass Velocities

As described in our metric formulation in Subsection 5.2.2, we consider the capture point for our stability metric which allows us to account for CoM velocity. Encoding this data into the workspace visualization gives a sense of the robot's capabilities while in motion. To this end, we present exemplary workspaces to show how this consideration affects voxel reachability.

For this example, we consider a small subset of the general workspace presented in Subsection 5.4.1 and show four workspaces of varying CoM velocities. Here the metric considered is stability (i.e., $\alpha = 0.0$ and $\beta = 1.0$ in Equation 5.9). First, in Figure 5.5a, we show a case where no CoM velocity is applied. As can be seen, most of the workspace can be reached and voxels close to the REEM-C are of higher quality. When a CoM velocity is applied in the x-direction, a portion of the workspace becomes unreachable due to the capture point exiting the support polygon, as shown in Figure 5.5b. Similarly, Figure 5.5c shows the scenario when a CoM velocity in the y-direction is applied. A portion of the workspace in the y-direction again, becomes unreachable. Applying a CoM velocity with both x and y components shows

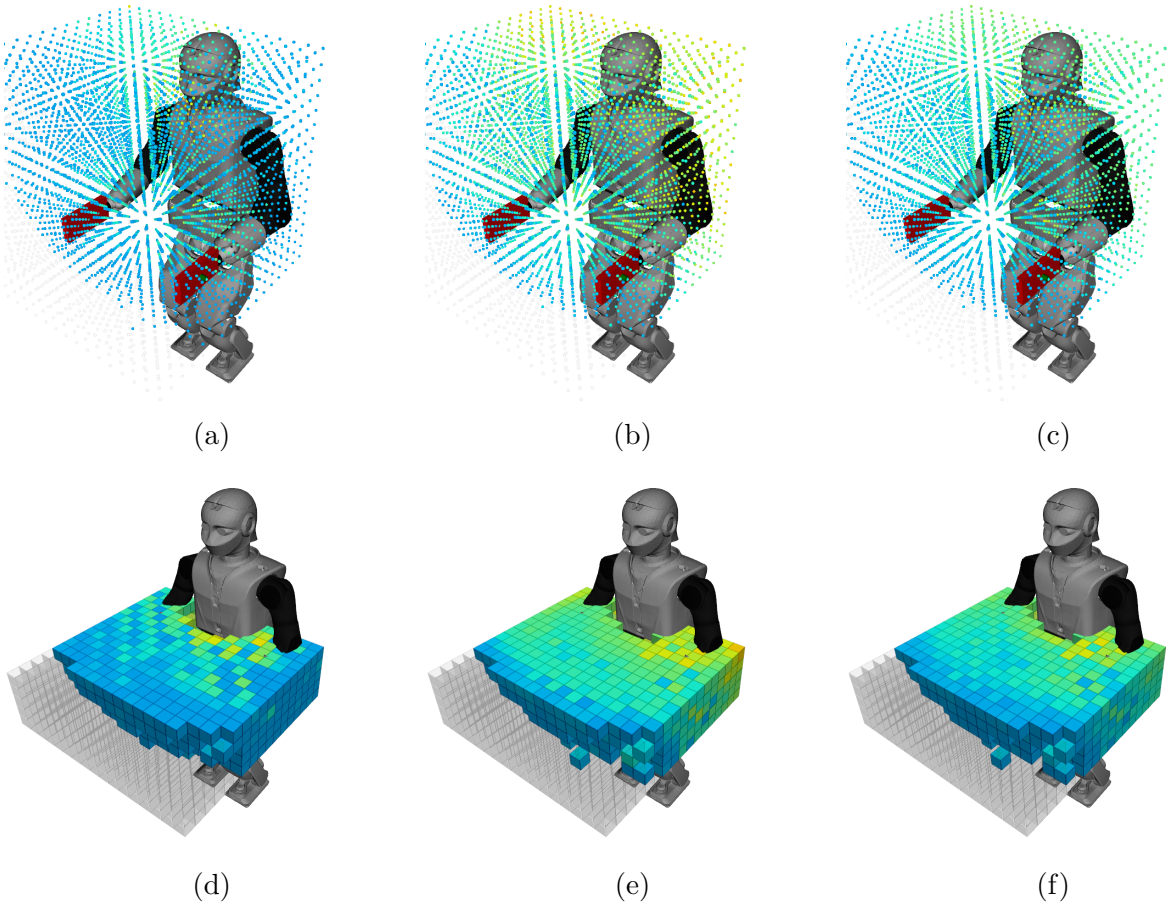


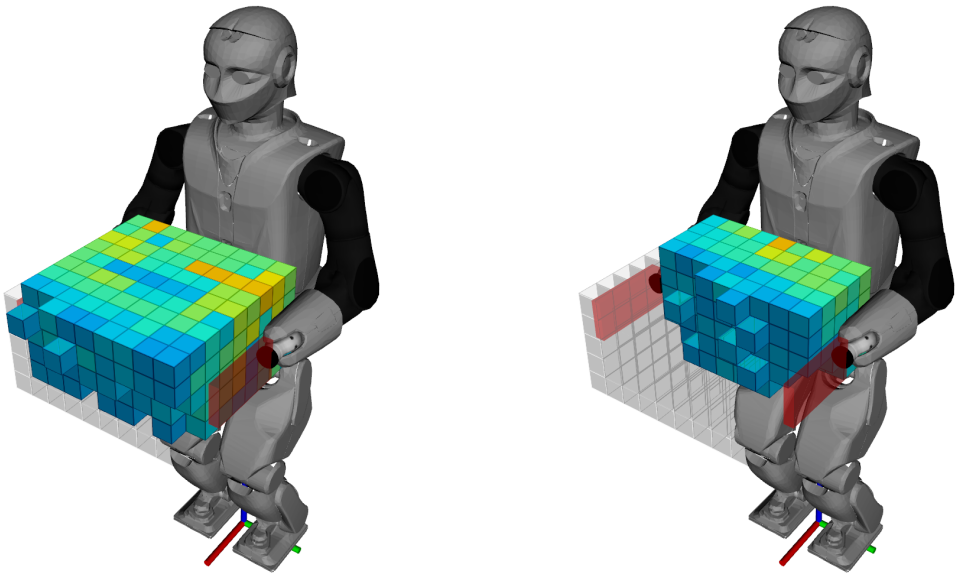
Figure 5.4: Sensitivity visualizations of the workspace considered for the general case for α and β. Low quality voxels are blue and high quality voxels are yellow/orange. Voxels that are unreachable or unstable are translucent grey. The first row shows the workspace that was considered and the second row shows a horizontal cut through. a) and d) show $\alpha = 1$ and $\beta = 0$ (the manipulability metric), b) and e) show $\alpha = 0$ and $\beta = 1$ (the stability metric), and c) and f) show $\alpha = 0.5$ and $\beta = 0.5$ (the combined manipulability-stability metric) as detailed in Equation 5.9.

that only a small corner of the workspace is now reachable. This information could be useful when considering CoM velocity in humanoid motion planning scenarios.

5.4.3 Workspace Maps with Different Object Masses

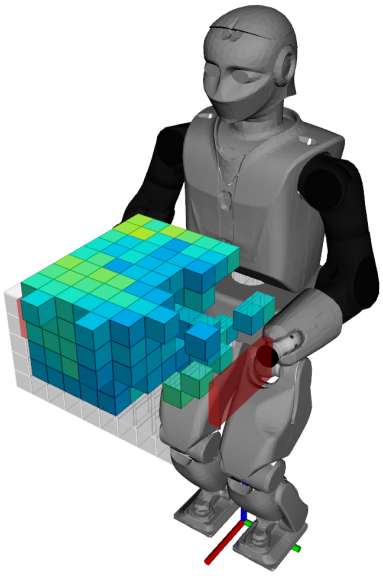
In this section we show two workspace scenarios where the REEM-C manipulates boxes. Box dimensions are taken into consideration for collision-checking and box mass is taken into account for the stability metric.

The first scenario involves a light 1 kg box of dimensions 0.1 m · 0.4 m · 0.1 m. The workspace checked for this scenario was 0.8 m^3 with a translational discretization of 5 cm and a rotational discretization of 15°. Figure 5.6 shows cut throughs of the workspace. We can see that the workspace is fairly symmetric which is expected given that the box being

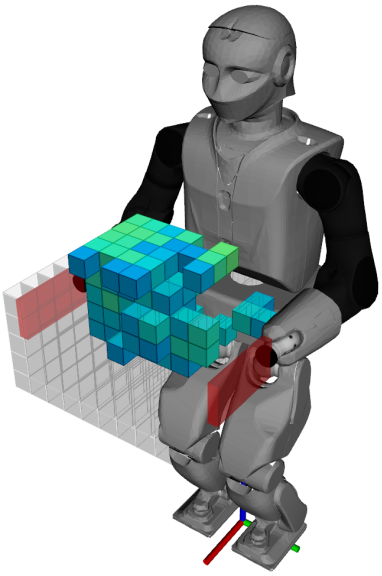


(a)

(b)



(c)



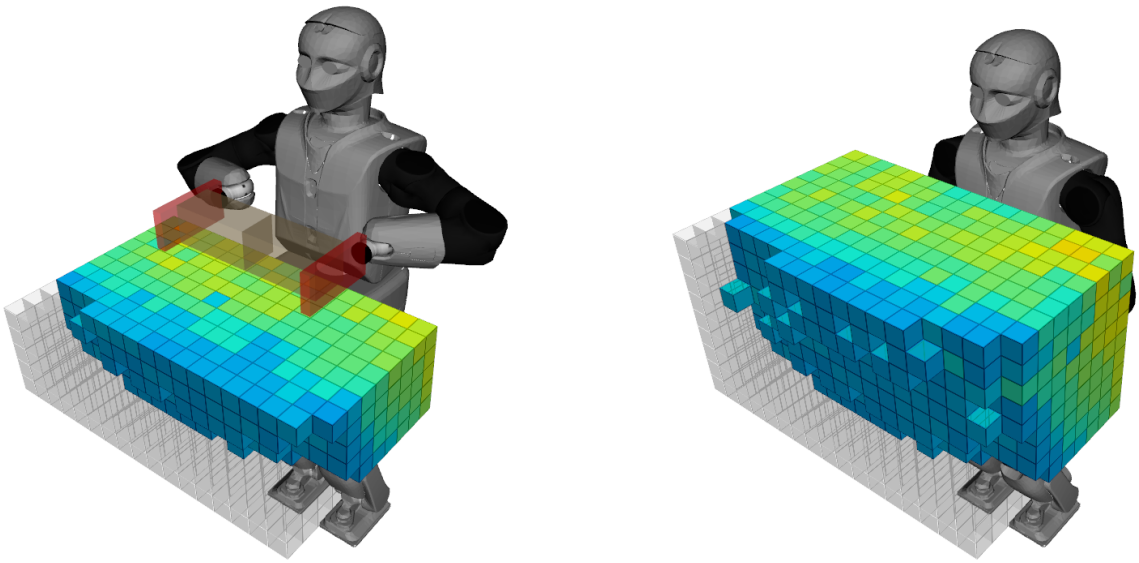
(d)

Figure 5.5: Visualizations of different CoM velocity (m/s) cases for a subset of the general workspace using the stability metric. a) $\dot{r}_{CoM} = [0.0, 0.0, 0.0]$ b) $\dot{r}_{CoM} = [0.2, 0.0, 0.0]$ box c) $\dot{r}_{CoM} = [0.0, 0.6, 0.0]$ d) $\dot{r}_{CoM} = [0.2, 0.6, 0.0]$.

manipulated is held with a symmetric grasp. The impact of carrying even this light box can be seen in the reduction of the feasible workspace in the x-direction. The outer voxels are infeasible due to the capture point exiting the support polygon which violates our currently

developed metric. For this case, the feasible area of the checked workspace was 18 % which is a reduction from the general workspace shown in Subsection 5.4.1.

The second scenario is manipulating a bigger box with dimensions 0.2 m · 0.6 m · 0.4 m, weighing 7 kg. From Figure 5.7 we can see that the effect of the increased box mass has a significant impact on the feasible workspace volume. For this case, the feasible area of the checked workspace was 10 %, a reduction from the 1 kg box case. We can see in Figure 5.7a, even holding the box at a distance of 0.2 m to 0.3 m from the torso brings the capture point (shown as a green dot between the feet) close to the edge of the support polygon (shown in blue outline).



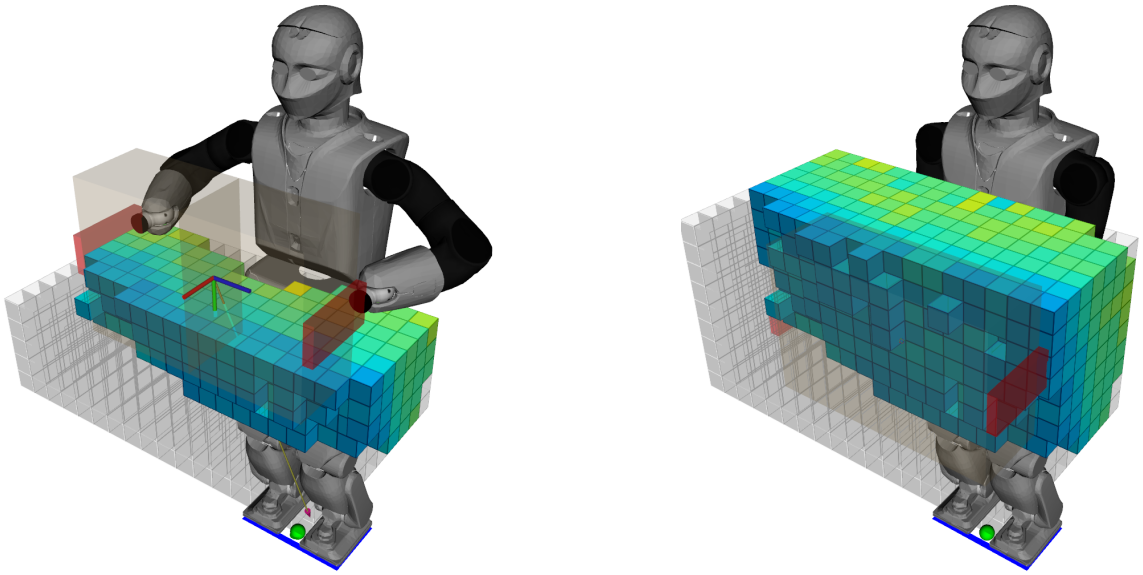
(a) (b)

Figure 5.6: Visualization of the bimanual manipulability-stability metric for the REEM-C manipulating a 1 kg box. a) A cut through at $z = 1.0$ m b) A cut through at $z = 1.2$ m.

5.4.4 Workspace Maps with Different Objects

In this section we consider two other objects with different grasping configurations, a rolling pin and a broom.

For the rolling pin, we considered a small workspace in front of the robot and restricted the orientation of the rolling pin to 7.5° around all axes. Pictured in Figure 5.8a, is the rolling pin workspace at a height of $z = 0.9$ m from the ground. The middle portion of the workspace in the x-direction is reachable but the edges closer to the robot and away from the robot are unreachable. The edges further away being unreachable is due to the fact that the arms and torso must extend and tilt causing stability issues. For the area closer to the robot, reachability is degraded due to self-collisions of the arms with the torso. When visualizing a higher z value, at 1.1 m, we can see that much more of the workspace is reachable due to the elbows pointing outwards from the torso, avoiding self-collisions. Thus, from this workspace,



(a) (b)

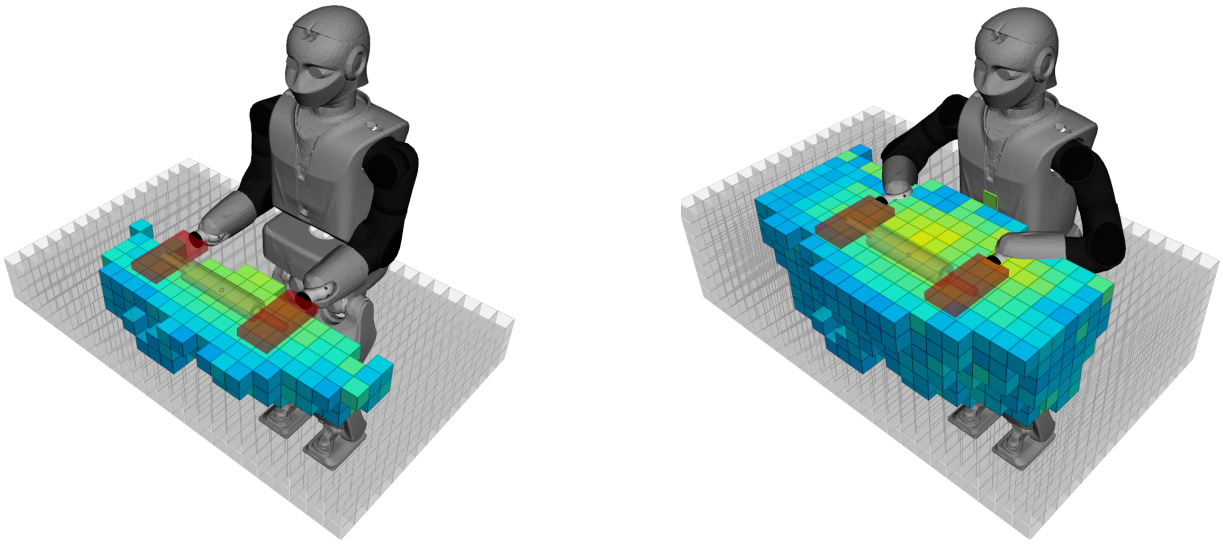
Figure 5.7: Visualization of the bimanual manipulability-stability metric for the REEM-C manipulating a 7 kg box. a) A cut through at $z = 1.0$ m b) A cut through at $z = 1.2$ m.

we can see the effects of having this grasp constraint and how it restricts motion at lower heights while standing in one place. We note that contacts and force are not considered in this workspace, and we are purely evaluating the kinematics of this constrained bimanual task.

For the broom, we considered a left-handed grasp with the right hand of the REEM-C above the left hand. Figure 5.9 shows two cut throughs of the workspace. Importantly, from Figure 5.9b, we see that the area towards the left side of the REEM-C is of higher quality and reachable where as much of the right side is unreachable. This is expected since the robot would have to reach across its body with the left hand to access the right side of the workspace, causing collisions with the torso.

5.4.5 Workspace Maps of TALOS

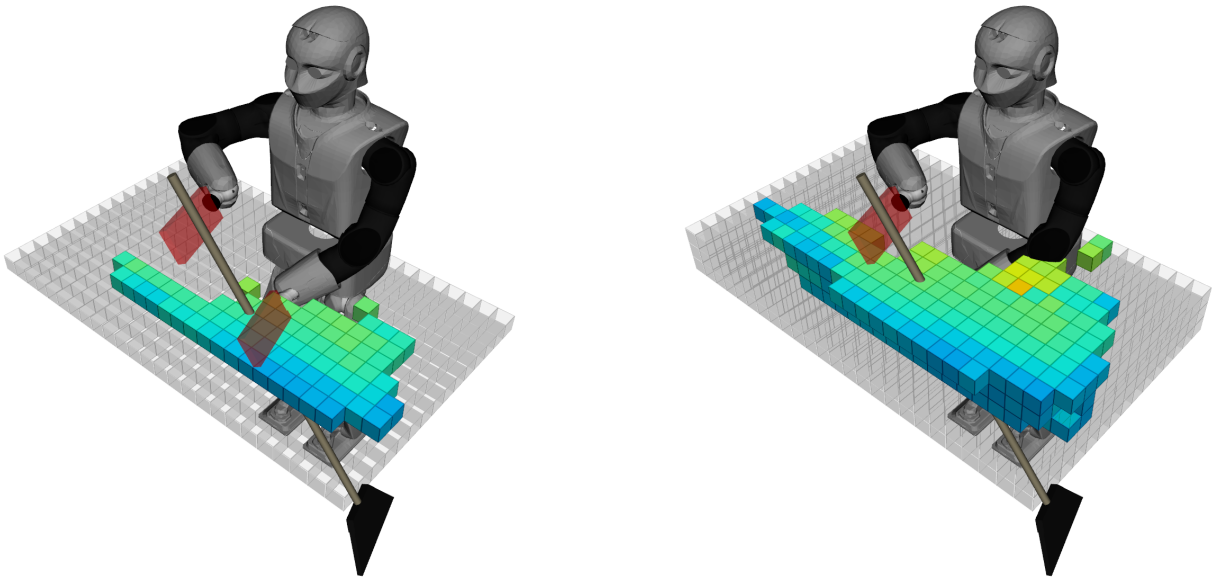
In order to illustrate how specific robot topology and characteristics affect workspace analysis, we show a few workspace scenarios with the TALOS humanoid. Figure 5.10 shows cut throughs of the manipulability-stability distribution for a general workspace, generated with spherical coordinates as described in Subsection 5.3.2. We used the same workspace volume and discretization parameters as was done for the REEM-C in Figure 5.4. We see that voxels nearer to TALOS are of higher quality while the edges of the generated workspace are of lower quality. Notably, as compared to the REEM-C, TALOS was able to reach 34 % of the queried voxels as compared to the REEM-C's 22 %. This is expected given that TALOS has longer arms and a shoulder topology that allows more motion than the REEM-C.



(a) (b)

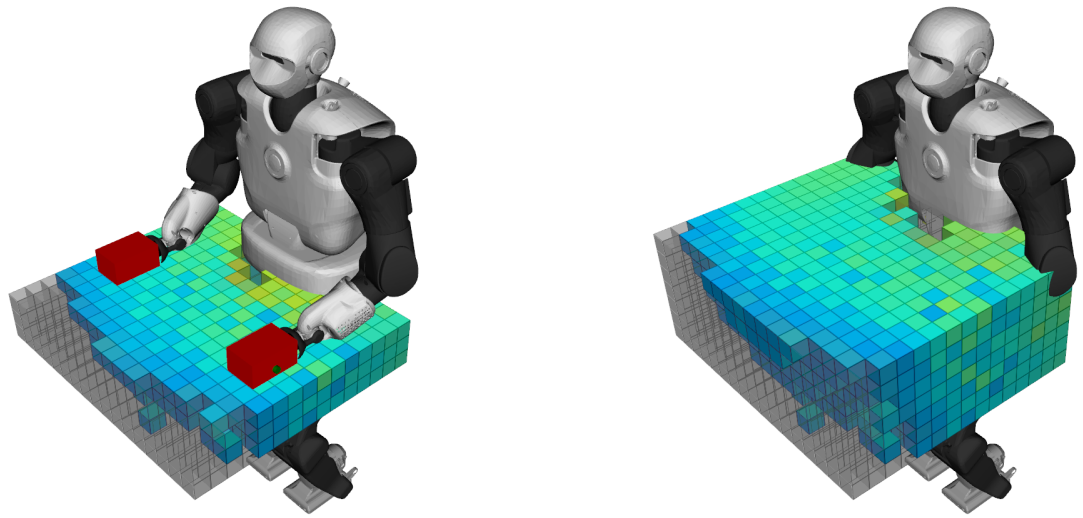
Figure 5.8: Visualization of the bimanual manipulability-stability distribution for REEM-C manipulating a rolling pin. a) A cut through at $z = 0.9\,\mathrm{m}$ b) A cut through at $z = 1.1\,\mathrm{m}$.

We also consider the same 1 kg box that was considered for the REEM-C in Subsection 5.4.3 for the TALOS, as shown in Figure 5.11. Notably, TALOS was able to reach 26% of the checked volume whereas the REEM-C was only able to reach 18 %. Again, this is due to the differences in limb lengths and topology.



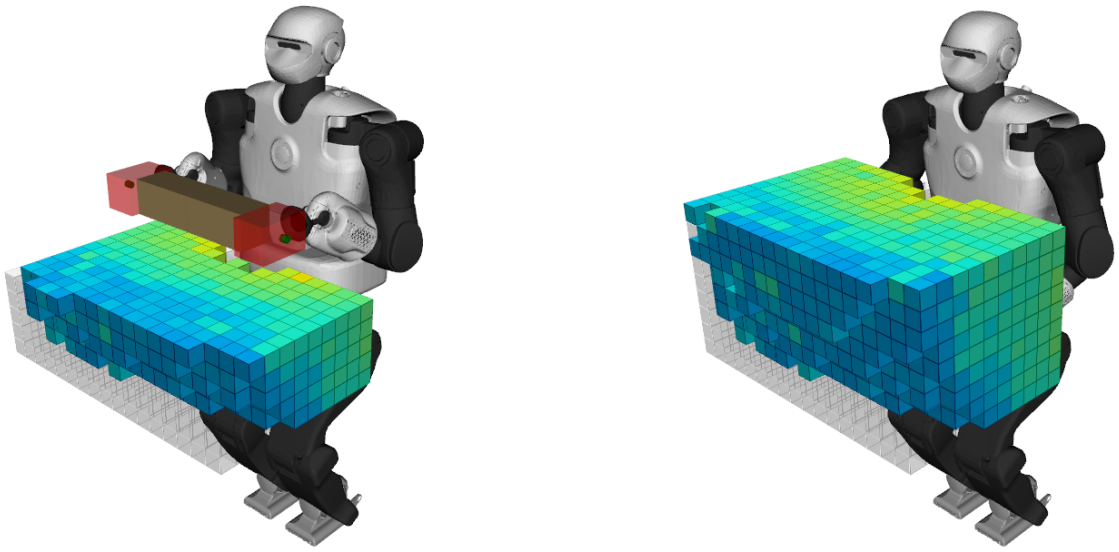
(a) (b)

Figure 5.9: Visualization of the bimanual manipulability-stability distribution for REEM-C manipulating a broom. a) A cut through at $z = 0.9\,\mathrm{m}$ b) A cut through at $z = 1.0\,\mathrm{m}$.



(a) (b)

Figure 5.10: Visualization of the bimanual manipulability-stability distribution for the TALOS humanoid, for the general case with spherical coordinates as described in Subsection 5.3.2. a) A cut through at $z = 0.9\,\mathrm{m}$ b) A cut through at $z = 1.1\,\mathrm{m}$.



(a) (b)

Figure 5.11: Visualization of the bimanual manipulability-stability metric for the TALOS manipulating a 1 kg box. a) A cut through at $z = 1.0\,\text{m}$ b) A cut through at $z = 1.2\,\text{m}$.

5.5 Motion Generation Methodology

In this section, we outline the motion generation methodology to generate bimanual motions using the metric proposed in Section 5.2. A bimanual variant of BiRRT is proposed which produces a kinematic path. This is then fed to a task space inverse dynamics controller to generate dynamically feasible trajectories.

5.5.1 Planning Problem Formulation

In this section we follow and adapt the formulations presented in [248, 196]. A humanoid robot is modelled as a floating base system with n_q joints and $n_q + 6$ degrees of freedom, including the joints and the 6 generalized coordinates of the floating base. The vector $q \in \mathbb{R}^{n_q+6}$ contains the joint positions of the robot and the position and orientation of the floating base. Given the the set of all possible configurations of the robot, i.e., *configuration space* or *C-space* (C), a *configuration* $q \in C$ is a minimal set of variables that specifies the position and orientation of each rigid body composing the robot along with the position and orientation of the floating base. The subset of C inducing collisions with obstacles in the environment is denoted C_{obs}. Assuming that $C \setminus C_{obs}$ is an open set, we denote the *obstacle-free space* as $C_{free} = cl(C \setminus C_{obs})$, where $cl(\cdot)$ denotes the closure of a set. In addition, since we are dealing with a humanoid robot, we introduce $C_{balanced} \subset C$ as the manifold that contains statically balanced configurations. Therefore, the set of *obstacle-free* and *statically balanced* configurations is denoted $C_{valid} \equiv C_{free} \cap C_{balanced}$.

For a given start and goal configurations, $q_{init}, q_{goal} \in C_{valid}$ respectively, the humanoid path planning problem can be defined as a triplet $(C_{valid}, q_{init}, q_{goal})$. A geometric path over the *configuration space* is a continuous function $\sigma : [0, 1] \to C$ and it is said to be collision-free and statically balanced if for all $\tau \in [0, 1]$, $\sigma(\tau) \in C_{valid}$. Let Σ denote the set of all paths, and Σ_{valid} denote the set of all collision-free paths. The *feasible path planning problem* is defined as:

Problem 1. ***Feasible path planning problem.*** *Given a path planning problem* $(C_{valid}, q_{init}, q_{goal})$*, find a feasible path* $\sigma(\tau) \in C_{valid}$ *such that* $\sigma(0) = q_{init}$ *and* $\sigma(1) = q_{goal}$*, if one exists, or report failure otherwise.*

For planning on a a continuous-cost space, we define a cost-function $c : C \to R_{\geq 0}$ such that a positive real value is assigned to all configurations in C. Then the *cost-space path planning problem* is:

Problem 2. ***Cost-space path planning problem.*** *Given a path planning problem defined as the quadruplet* $(C_{valid}, q_{init}, q_{goal}, c)$*, solve the feasible path planning problem while accounting for the cost function* c *during exploration of the C-space.*

Methods that aim to solve the *cost-space path planning problem* often perform a rejection sampling of configurations in the *C-space* by imposing specific cost constraints evaluating each configuration on its cost alone, or on the cost variation to move locally between configurations. Specifically for humanoids, generating collision-free samples is straightforward

using random sampling and off-the-shelf collision checking libraries. However, generating samples from $C_{balanced}$ is nontrivial since random sampling from a low-dimensional manifold is not straightforward. In this work, we propose a planning method to solve Problem 2 for humanoids by employing whole-body inverse kinematics (Equation 6.1) to generate statically balanced samples, with the cost being the proposed manipulability-stability metric. In addition, as detailed later, we impose kinematic bimanual constraints to ensure that each sample generated also maintains a grasp on the object that the humanoid is manipulating.

5.5.2 Sampling-Based Planning for Bimanual Manipulation

In this work, we are concerned with bimanual object manipulation and therefore sample in the end-effector space, $SE(3)$. We set bounds on $SE(3)$ to indicated a region of interest in front of the robot where manipulation occurs and the state is equivalent to the end effector's forward kinematics denoted:

$$x = FK(q) \in W, \tag{5.11}$$

where $FK()$ denotes the forward kinematics function, and $W \in SE(3)$ the robot's world. The final paths are represented in the robot's configuration space, thus, we associate an end-effector state with a single full body configuration (using whole-body inverse kinematics) to avoid ambiguity. In this work, the "end-effector" is the mid point between the two hands (as shown in Figure 5.2) and the offsets of the hands are changed for each object to maintain a suitable bimanual grasp.

Algorithm 2 details the bimanual BiRRT planner which is based on the planner from [31] with changes in the ConstrainedExtend function to adapt the task to humanoids performing bimanual manipulation.

In this planner, sampling happens in the task space ($SE(3)$) and a whole-body IK solver is used to find the configuration space joint angles for the humanoid at each valid sampled node. In this way, the object and robot motion are planned together simultaneously and the whole-body IK solver can be tuned to produce whole-body configurations that satisfy costs and constraints. For this work, we used all of the task maps detailed in Table 5.1 to form a constrained IK solver that satisfies bimanual constraints for the various objects studied in this work. Each object has specific geometry which are encoded as grasp offsets for the left and right TCPs of the humanoid as detailed in Subsection 5.3.3. This is also described in Algorithm 4 where the `object_geometry` is taken into account for solving the IK problem.

Algorithm 3 works similarly to the original algorithm [31] by iteratively moving from a configuration x_{near} toward x_{target} with a step size Δx_{step}. After each iteration, there is a check to see whether the new configuration has reached the target and if so, the new configuration x_s is returned. If this condition isn't met, the algorithm takes a step toward the target configuration using the ConstrainedIK function which finds a joint space configuration for the passed in $SE(3)$ configuration, x_s. This joint space configuration adheres to various constraints and costs as set to achieve bimanual motion while holding a specified object. The IK algorithm and costs are similar to those described in Table 5.1 with the exception that we use the SNOPT solver [88] so that we can make some costs constraints without sacrificing computation time. In particular we make the "End effector frame" task map a constraint

and encode bimanual grasp offsets for each object. Additionally, we added a new task map "Manipulability-Stability" where the cost is calculated exactly as in Equation 5.9.

Algorithm 2 Bimanual BiRRT Planner

Input: x_{start}, x_{target}, q_{start}, q_{target}, `object_geometry`
Output: Feasible path or failure
$\mathcal{T}_\mathcal{A}$.Init(x_{start}, q_{start}), $\mathcal{T}_\mathcal{B}$.Init(x_{goal}, q_{goal})
for $n \in \{1, \ldots, max_n_nodes\}$ **do**
    $x_{rand} \leftarrow$ SampleValidSE3()
    $x^a_{near}, q^a_{near} \leftarrow$ NearestNode$(\mathcal{T}_\mathcal{A}, x_{rand})$
    $x^a_{reached} \leftarrow$ ConstrainedExtend($\mathcal{T}_\mathcal{A}$, x^a_{near}, x_{rand}, `object_geometry`)
    $x^b_{near}, q^b_{near} \leftarrow$ NearestNode$(\mathcal{T}_\mathcal{B}, x^a_{reached})$
    $x^b_{reached} \leftarrow$ ConstrainedExtend($\mathcal{T}_\mathcal{B}$, x^b_{near}, $x^a_{reached}$, `object_geometry`)
    if $x^a_{reached} == x^b_{reached}$ **then**
        $P \leftarrow$ ConstructPath($\mathcal{T}_\mathcal{A}$, $\mathcal{T}_\mathcal{B}$)
        return SmoothJointSpacePath(P, $\mathcal{T}_\mathcal{A}$, $\mathcal{T}_\mathcal{B}$)
    Swap ($\mathcal{T}_\mathcal{A}$ and $\mathcal{T}_\mathcal{B}$)
return NULL

Algorithm 3 ConstrainedExtend

Input: $\mathcal{T}$, x_{near}, x_{target}, `object_geometry`
Output: x_{final}
$x_s \leftarrow x_{near}$; $x^{old}_s \leftarrow x_{near}$
while True **do**
    if $x_{target} == x_s$ **then**
        return x_s
    if $\|x_{target} - x_s\| > \|x^{old}_s - x_{target}\|$ **then**
        return x^{old}_s
    $x_s \leftarrow x_s + \min(\Delta x_{step}, \|x_{target} - x_s\|) \cdot \frac{x_{target} - x_s}{\|x_{target} - x_s\|}$
    $x_s, q_s \leftarrow$ ConstrainedIK(x_s, q^{old}_s, `object_geometry`)
    if $q_s \neq$ NULL ***and*** IsSegmentValid($\mathcal{T}$, q^{old}_s, q_s) **then**
        $\mathcal{T}$.InsertNewNode(x_s, q_s)
    else
        return x^{old}_s

After a path P is found, it is passed to SmoothJointSpacePath() which ensures that the path has no large jumps in joint space configurations. This can happen, in particular, at the edge configurations between the two trees $\mathcal{T}_\mathcal{A}$ and $\mathcal{T}_\mathcal{B}$. Since these trees are expanded separately, the configurations where the trees are connected can have jumps in the joint space, for example, with elbow up or elbow down arm configurations given the redundancies in the arms of the humanoid. To alleviate this, the function takes the shorter of the two trees and seeds the first configuration in the tree with the last configuration of the other tree and resolves all the joint space IK solutions along the path. This adds to the overall planning

Algorithm 4 ConstrainedIK

Input: x_s, q_s^{old}, `object_geometry`
Output: x_s, q_s
$j_l, j_r \leftarrow$ calcGraspOffsets(`object_geometry`)
$q_s \leftarrow$ calcBothArmsIK($x_s + j_l$, $x_s + j_r$, q_s^{old})
return q_s

Algorithm 5 IsSegmentValid

Input: $\mathcal{T}$, q_s^{old}, q_s
Output: Boolean (True if the segment is valid)
$q_{diff} \leftarrow$ —(q_s - q_s^{old}) /Δt|
if $q_{diff} >$ `joint_velocity_limits` **then**
    return False
$Q \leftarrow$ linspace (q_s^{old}, q_s, $\|q_s$ - $q_s^{old}\|/\epsilon$)
for i, q in Q **do**
    if IsConfigInCollision(q) **then**
        if $i == 0$ **then**
            return False
        else
            $x_{obj} \leftarrow$ FK($Q[i-1]$)
            $\mathcal{T}$.InsertNewNode(x_{obj}, $Q[i-1]$)
        return False
return True

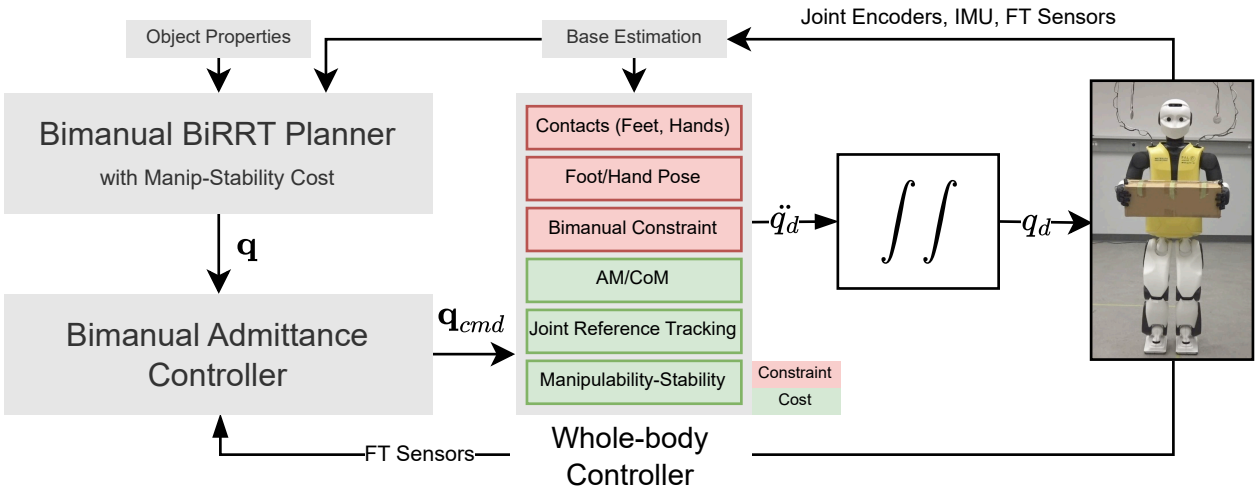


Figure 5.12: The planning and control block diagram highlighting the various components of the motion generation pipeline. As a note, $\boldsymbol{q}_{cmd}$ is tracked with the "Joint Reference Tracking" task and is updated at each controller time step (0.005s).

time but is negligible given that each configuration takes up to 1 ms to solve. If there is no large jump in joint space between the trees, this function exits without modifying the constructed path.

Algorithm 5 checks collisions and velocity limits by stepping along the interval between q_s^{old} and q_s. If any configuration between the interval is in collision or violates joint velocity limits, the last valid configuration is added to the tree and the algorithm exits. Δt is set to 0.005s in this work as the REEM-C robot's internal controller runs at 200Hz.

We note that, for simplicity, during path execution, we set the time step between path nodes to 0.005s. This could also be parametrized using any known algorithm from the literature, such as Iterative Parabolic Time Parameterization (IPTP) algorithm [15].

Additionally, while the manipulability-stability metric is only a local metric this planning procedure ensures a continuous trajectory while optimizing for greater manipulability-stability. As a result, disjoint configurations of high quality for manipulability-stability are not of concern since all nodes of the trajectory are part of the continuous configuration space as determined by the Bimanual BiRRT planner. Ultimately, the proposed metric allows for capturing the key qualities of manipulability and stability at a specific task space point for a more preferential robot configuration.

Since we are only altering the planner to essentially sample configurations on a constrained manifold (C_{valid}) of C_{free}, this planner inherits probabilistic completeness from the original BiRRT/RRT-Connect implementation [136]. That is, if a path exists, it is guaranteed to be found as the number of samples tends to infinity.

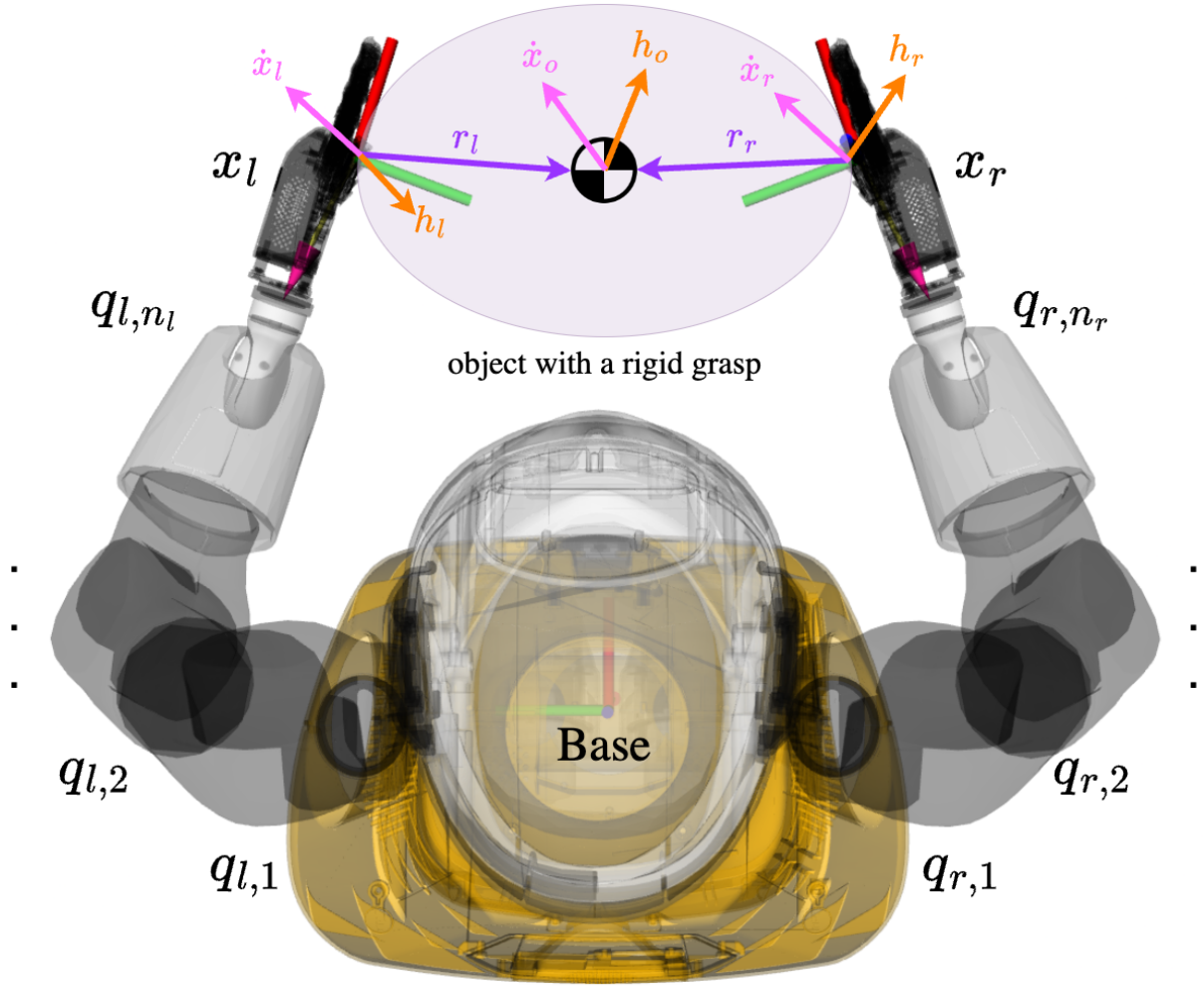


Figure 5.13: The dynamic model for bimanual manipulation with the robot rigidly grasping an object.

5.5.3 Extension of Bimanual Model to Include Contact Forces

So far, we have primarily focused on kinematic modelling and planning to form time-indexed paths for the robot to execute, assuming a fixed, tightly coupled grasp. In this subsection, we introduce the bimanual grasp model used later in our control formulation to execute the paths found during the planning stage.

Consider a system with two hands, tightly grasping an object, as in Figure 5.13. The robot is applying forces on the object with both hands and no relative motion can arise between the grasping points and the grasped object. The generalised forces that each hand applies on the object $\boldsymbol{h}_\mathrm{l}, \boldsymbol{h}_\mathrm{r} \in \mathbb{R}^6$ can be mapped to the net object force $\boldsymbol{h}_\mathrm{o} \in \mathbb{R}^6$ by a grasp matrix, as defined in [173]:

$$\boldsymbol{h}_o = \begin{bmatrix} \boldsymbol{G}_l & \boldsymbol{G}_r \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_l \\ \boldsymbol{h}_r \end{bmatrix}. \tag{5.12}$$

The grasp matrix can be expressed as:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_l & \boldsymbol{G}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{S}(\boldsymbol{r}_l)^\top & \boldsymbol{I}_3 & \boldsymbol{S}(\boldsymbol{r}_r)^\top & \boldsymbol{I}_3 \end{bmatrix}, \tag{5.13}$$

where $\boldsymbol{I}_3$, $\boldsymbol{0}_3$ are three dimensional identity and zero matrices, $\boldsymbol{S}(\cdot)$ is the skew-symmetric matrix operator and $\boldsymbol{r}_l, \boldsymbol{r}_r$ are translation vectors from a point on the object (for example, the center of mass) to the contact point on the hand (these are also known as virtual-sticks in previous works, [38]).

From rigid body kinematics, we can derive the following constraint:

$$\begin{bmatrix} \dot{\boldsymbol{x}}_l \\ \dot{\boldsymbol{x}}_r \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{G}_l^\top \\ \boldsymbol{G}_r^\top \end{bmatrix} \dot{\boldsymbol{x}}_o \tag{5.14}$$

where $\dot{\boldsymbol{x}}_l, \dot{\boldsymbol{x}}_r \in \mathbb{R}^6$ denotes the hand velocities, $\dot{\boldsymbol{x}}_o \in \mathbb{R}^6$ is the object velocity and $\boldsymbol{J} \in \mathbb{R}^{12 \times n_q}$ is the stacked Jacobian matrix associated with all contact points. This constraint, which says that the object and hand velocities are common, is directly used in the control formulation, shown later.

5.5.4 Control Formulation

Bimanual Admittance Control

To avoid high interaction forces with the held object and the environment, admittance control can be employed to allow the trajectory being executed to deform based on forces felt by the robot. Since we are using a position-controlled robot with force-torque sensors at the wrist of the robot, we can form a Cartesian impedance/admittance law that modulates between accurate tracking and compliance based on the forces felt. This controller nominally executes the path planned by the motion planning pipeline and yields to forces felt when needed. A similar controller was used in our previous work for operating a handlebar with a humanoid on a personal transporter [196].

Consider a single n-DOF manipulator with a force torque sensor at its wrist. To form the Cartesian admittance law for the manipulator we take the end-effector position with respect to the base frame as $\boldsymbol{x}$. The desired pose of the hand is taken as $\boldsymbol{x}_d$ such that a Cartesian impedance law that tracks the desired hand pose is given as,

$$\boldsymbol{W}_i = -\boldsymbol{K}_p\tilde{\boldsymbol{x}} - \boldsymbol{K}_d\tilde{\dot{\boldsymbol{x}}}, \tag{5.15}$$

where $\tilde{\boldsymbol{x}}$ is the pose error, $\tilde{\dot{\boldsymbol{x}}}$ is the velocity error, $\boldsymbol{W}_i$ is the resultant impedance wrench and $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are diagonal proportional and derivative gain matrices. This control allows for adequate tracking of Cartesian hand poses.

Analogously, we can define a Cartesian admittance law as,

$$\boldsymbol{W}_c = w_i\boldsymbol{W}_i + \boldsymbol{K}_{ft}\boldsymbol{W}_{ft}. \tag{5.16}$$

Here, $\boldsymbol{W}_{ft}$ is the wrench as measured at the wrist of the arm, $\boldsymbol{K}_{ft}$ is a gain matrix to amplify the wrench, and w_i a weight that allows us to vary the effect of the Cartesian impedance control. When $\boldsymbol{W}_{ft}$ is greater than a set threshold, w_i can be lowered (or set to 0) so that the end effector can yield to any forces or torques while sacrificing pose tracking performance with the Cartesian impedance control law, $\boldsymbol{W}_i$. This can be useful to limit interaction forces due to unexpected disturbances.

Equation 5.16 can be used for multiple independent manipulators, but needs extra attention when the manipulators are carrying a shared load or object. To couple and synchronize the manipulators, another term can be added to reduce the relative error between the two end effectors, assuming a distance and orientation offset between the two is set. Taking $\tilde{\boldsymbol{x}}_b$ and $\dot{\tilde{\boldsymbol{x}}}_b$ as the relative pose error and relative velocity error of the two end effectors, a Cartesian coupling admittance control law to reduce the relative error between the end effectors carrying a shared load is given as,

$$\boldsymbol{W}_r = -\boldsymbol{K}_{p_r}\tilde{\boldsymbol{x}}_b - \boldsymbol{K}_{d_r}\dot{\tilde{\boldsymbol{x}}}_b. \tag{5.17}$$

$\boldsymbol{K}_{p_r}$ and $\boldsymbol{K}_{d_r}$ are diagonal proportional and derivative gain matrices. The total wrench is given as,

$$\boldsymbol{W}_t = \boldsymbol{W}_c + \boldsymbol{W}_r, \tag{5.18}$$

and is projected onto each joint of the two arms through the Jacobian associated with the force-torque sensor frame,

$$\boldsymbol{\tau}_c = \boldsymbol{J}_{ft}^{\top}\boldsymbol{W}_t. \tag{5.19}$$

If using a position-controlled robot (such as the REEM-C humanoid), we must transform this torque to a position command. This is done by first defining a velocity command,

$$\dot{\boldsymbol{q}}_{cmd} = \boldsymbol{M}_j^{-1}\boldsymbol{\tau}_c - \boldsymbol{M}_j^{-1}\boldsymbol{K}_{j_p}(\boldsymbol{q} - \boldsymbol{q}_d), \tag{5.20}$$

where $\boldsymbol{M}_j$ is the mass matrix related to the controlled joints. $\boldsymbol{K}_{J_p}$ can be set depending on how stiff or compliant the arms should be. $\dot{\boldsymbol{q}}_{cmd}$ is then integrated using the controller's time period to generate a position command $\boldsymbol{q}_{cmd}$.

In this work, we deal with scenarios where the object is already grasped with the two hands. For the case where the robot initially grasps the object (comes in contact with a rigid body), gains can be scheduled to reduce the influence of the admittance controller upon inital contact. This would remove instabilities and allow for compliance to come into play once the object is rigidly grasped with both hands. This combined with the force thresholding mentioned earlier allows for stable motion generation with admittance control.

Task Space Inverse Dynamics

In this work, we make use of the whole-body controller developed for whole-body balancing [197] and add some new tasks to more explicitly take into account bimanual manipulation scenarios. The tasks used are listed in Figure 5.12: contacts, feet/hand positions, joint reference tracking, bimanual constraint and the manipulability-stability metric proposed in this work. The joint reference tracking task is used to track the output of the bimanual

admittance controller (which in turn tracks the output of the BiRRT planning stage). We finally impose the bimanual constraint as in Equation 5.14 and the manipulability-stability cost in the whole-body controller (Equation 2.10). We note that the cost considers the CoM velocity at every iteration. This QP is solved in real-time (every 5 ms, the controller timestep) and the final joint accelerations are integrated twice before being sent to the robot as joint position commands.

The task gains were set as in Table 5.3. All task damping gains (K_{D}) were set to their critical value $2\sqrt{K_{\mathrm{P}}}$. For the priority levels listed, 0 indicates a constraint and 1 indicates a cost.

Task	**Weight**	**Priority**	K_{P} **gain**
Torque Bounds	-	0	-
Joint Bounds	-	0	-
Bimanual Constraint	-	0	-
Feet Contacts (6D)	1	0	30
Hand Contacts (6D)	1e-5	0	30
Center of Mass	1e3	1	12.5
Angular Momentum	1e-4	1	10
Feet Pose	1	1	300
Hand Pose	1	1	300
Joint Reference Tracking (30 DoF)	1e2	1	all equal (1)
Manipulability-Stability	1e-2	1	10

Table 5.3: The TSID task gains for the bimanual manipulation whole-body controller.

5.6 Motion Generation Results for the REEM-C Humanoid Robot

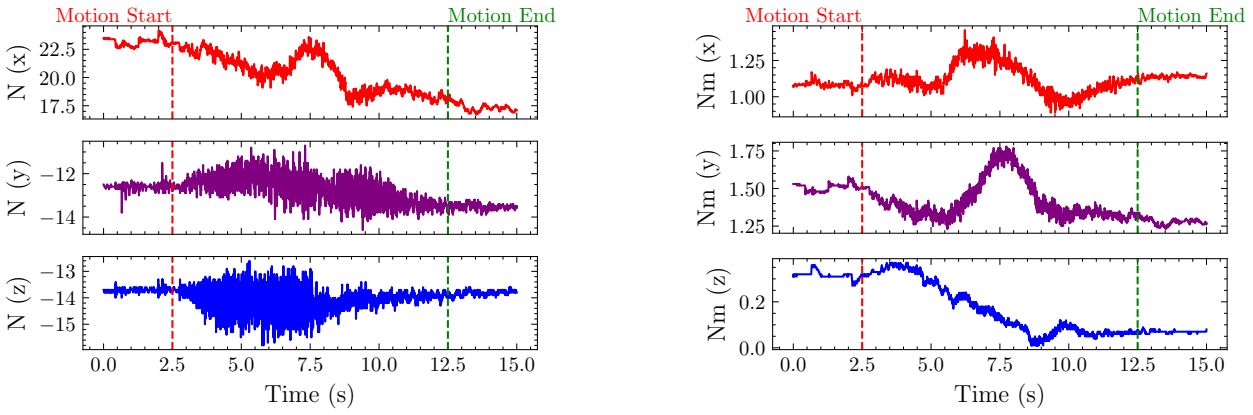
In this section we validate our motion generation pipeline by showing real hardware results. We utilize the PAL REEM-C humanoid as the test platform. Pinocchio [43] is used to compute dynamic quantities and TSID [67] [3] is used to compute the task-space inverse dynamics whole-body quadratic program (Equation 2.10). SEROW (State Estimation RObot Walking) [14] is used for base estimation. All components of the motion generation pipeline are written in Python using the ROS Noetic middleware (Ubuntu 20.04) and ran on a MacBook Pro (M1 Pro processor) with 16GB of RAM. Whole-body position commands were sent to the REEM-C over a ROS topic.

The rest of this section is broken down by each object tested (box, broom, rolling pin) and results are presented and discussed. We note that any potential grasp could be analyzed using the workspace analysis described in Section 5.3. However, many objects are designed to be used with a single prescribed grasp in a fixed manner, as seen in the exemplar cases studied. Therefore, we focus on demonstrating motion generation for these three typical types of grasps.

5.6.1 Experimental Bimanual Box Motions

The first object that was tested was a 50 cm wide box. To rigidly grasp the object, the flat palms of the REEM-C made contact with the sides of the box and the fingers curled in to grasp the corners firmly. Task space start and goal configuration were set with respect to the center of the box and are listed in Table 5.4. A total of 10 trials were run for the start and goal configurations and metrics were collected with and without the proposed manipulability-stability cost. Figure 5.17 shows an exemplar motion with the manipulability-stability cost used during planning. As compared to Figure 5.18 we can clearly see that the elbows are more flared and the robot stays closer to the center of the support polygon when planning with the manipulability-stability cost. This is also confirmed in the box graph in Figure 5.15. We see that the yellow line indicates that along the planned paths, the planner with the proposed cost metric is able to maximize the manipulability-stability metric along the whole path. Table 5.5 also shows some metrics for the planner comparisons, with and without the manipulability-stability metric. As can be seen, the average manipulability-stability cost is much higher when the cost is enabled, producing more manipulable and stable paths. The tradeoff here is planning time which is 5.44 s $\pm$ 1.77. Given the significant increase in manipulability-stability metric, in situations where planning time is not critical, we can produce more high quality paths (e.g., for creating an offline database of trajectories). Figure 5.16 shows the left and right hand end effector errors for the box trajectories. As can be seen, the error in translational components is less than 1 mm and 0.005 rad in orientation. This clearly shows that the full motion pipeline is able to handle all the constraints set in the planning and TSID QP execution stage.

Figure 5.14 shows sensor data from the left wrist force-torque sensor during a box manipulation motion. As can be seen, the force and torque profiles are bounded through the admittance controller, before returning to their nominal values at the end of the trajectory.



(a) Forces

(b) Torques

Figure 5.14: Force-torque data from the left wrist force-torque sensor for one trial of the box motion shown in Figure 5.17.

5.6.2 Experimental Bimanual Broom Motions

The next object that was tested was a standard, off-the-shelf, broom. For this experiment, the REEM-C firmly grasped the cylindrical handle of the broom forming a rigid grasp. Figure 5.19 shows one cycle of the broom motions starting from the left frame where the broom is lightly making contact with the ground, to the final frame where the robot has swept over the ground to the final configuration. As shown in Figure 5.15, the broom motion with the manipulability-stability metric are more manipulable and stable along the trajectory. Portions of the path dip below the baseline planner (shown in blue) but as shown in Table 5.5, the planner with the cost produces paths with higher average manipulability-stability. The planning time only increases by around 1 s when planning with the manipulability-stability metric and this is because the broom motion is already a highly constrained grasp which doesn't allow too much nullspace motion. As such, the difference in cost metrics are less than the box scenario. We see similar results in Figure 5.16 where the left and right hand errors are bounded within 1 mm for translational components and 0.015 rad for orientation.

This grasp configuration is very limiting for the REEM-C due to the wrist design as it can only perform flexion and extension or adduction and abduction. This design limitation along with the short arms, compared to a humanoid platform with a larger workspace such as TALOS with a different arm topology (see Subsection 5.4.5), limit the range of the sweeping motion but nevertheless can perform a reasonable sweeping cycle

5.6.3 Experimental Bimanual Rolling Pin Motions

The last object that was tested was the rolling pin. Figure 5.20 shows the setup where the robot holds a rolling pin with a closed grasp against a table. In this scenario, similar to the broom, the grasping configuration is very constrained as compared to the topology of the REEM-C arms. As such, the difference in metrics between the plan with and without the cost is less pronounced, but nevertheless visible in Figure 5.15. The end effector errors are also well bounded showing very good tracking in Figure 5.16.

Similar to the grasp for the broom, the wrist design and limited workspace size limit the rolling pin's range of motion.

Object	x_{start}	x_{goal}
Box	[0.45, 0.0, 1.1, -1.57, 0, 0]	[0.55, 0.25, 1.3, -1.57, 0.0, 0.0]
Broom	[0.47, 0.17, 1.06, -1.111, 0.047, 0.281]	[0.47, 0.04, 1.14, -1.25, 0.047, 0.281]
Rolling Pin	[0.45, 0.0, 0.96, 0.0, 0, 0]	[0.65, 0.0, 0.96, 0.0, 0, 0]

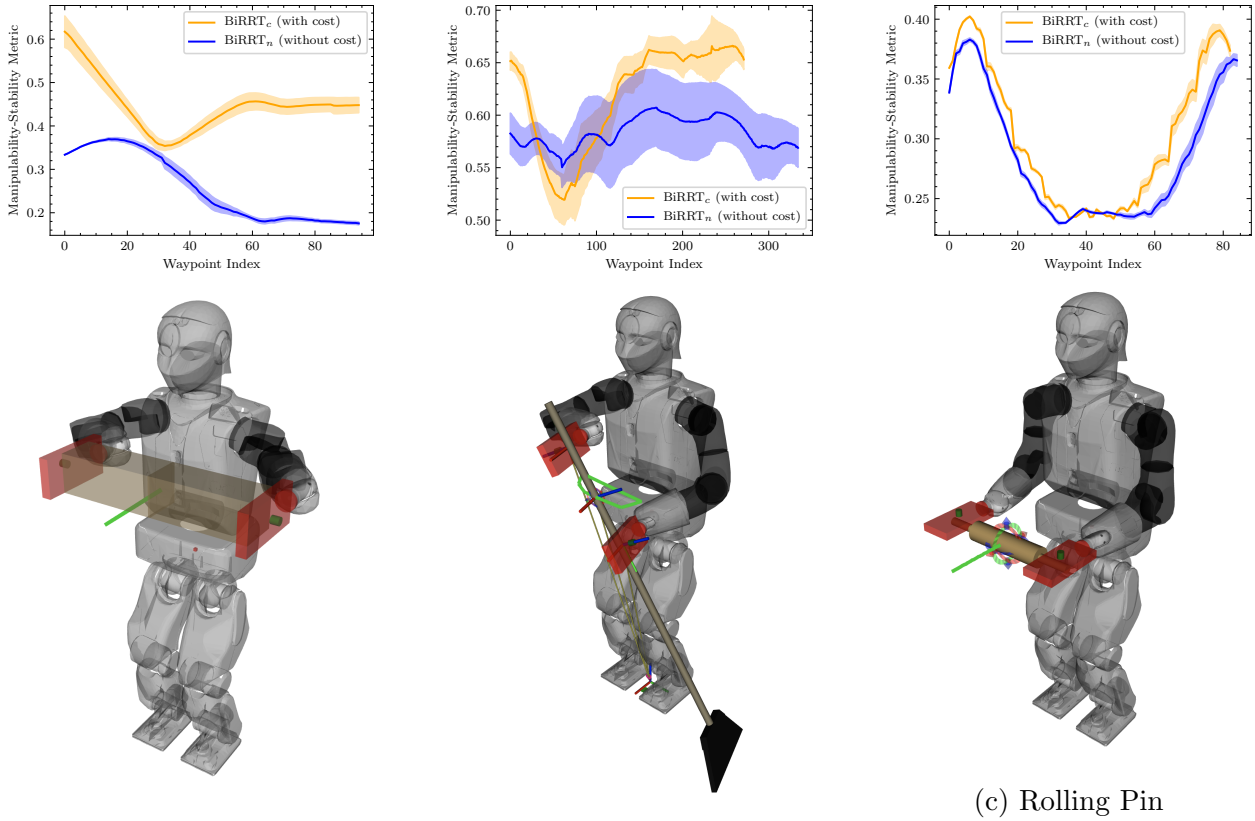
Table 5.4: The start and goal configurations across different tasks.

5.6.4 Bimanual Box Motions on a Shelf

In this final test, we show that the planner can avoid environmental collisions by planning a motion to carry a box from one rung of a shelf to another. In doing this, the robot needs to extend to the edge of its workspace and ensure that its links do not collide with the shelf.

Object	Algorithms	$\mathcal{W}$ Path Length (m)	Planning Time (s)	Nodes Expanded	Avg. MS Cost
Box	BiRRT_n	0.3560 ± 0.0022	0.56 ± 0.04	112.00 ± 10.77	0.25 ± 0.01
	BiRRT_c	0.3323 ± 0.0075	5.44 ± 1.77	163.6 ± 44.53	0.45 ± 0.02
Broom	BiRRT_n	0.3707 ± 0.0295	2.17 ± 0.39	477.6 ± 71.01	0.56 ± 0.02
	BiRRT_c	0.3533 ± 0.0275	3.13 ± 0.98	422.00 ± 56.84	0.60 ± 0.02
Rolling Pin	BiRRT_n	0.198 ± 0.0054	0.95 ± 0.08	46.80 ± 0.91	0.29 ± 0.0
	BiRRT_c	0.199 ± 0.005	7.03 ± 0.16	43.1 ± 0.57	0.31 ± 0.0

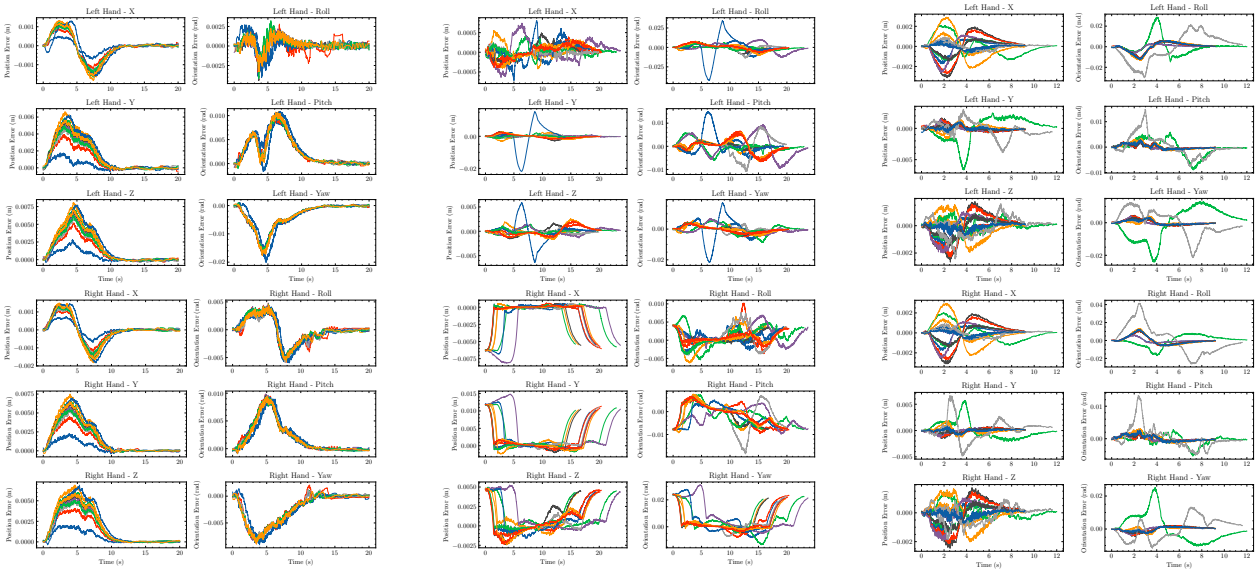
Table 5.5: Comparison of BiRRT_n (baseline, no cost) and BiRRT_c (with manipulability-stability cost) across different tasks.



(a) Box (b) Broom (c) Rolling Pin

Figure 5.15: Manipulability-Stability metric along real hardware trajectories. 10 trials were done for each object and the graphs show the average and standard deviation along the trajectories for the box, broom and rolling pin. Exemplar paths generated with the manipulability-stability cost are shown for each object.

Figure 5.21 shows the trajectory generated with the manipulability stability cost. As can be seen, the path generated avoids the shelf while maximizing the manipulability-stability metric. The arms are slightly flared throughout the trajectory avoiding singularities. Figure 5.22 shows the trajectory generated without the manipulability stability cost. Along the trajectory, there are certain points in which the arms are close to singularity. The average manipulability-stability along the trajectory with the manipulability-stability cost was 1.16, as compared to without the metric, which was 0.68. In these cases, using the metric within



(a) Box (b) Broom (c) Rolling Pin

Figure 5.16: The left and right hand end effector error along all real hardware trajectories (with manipulability-stability cost). As can be seen, all the errors are bounded to sub centimeter and sub 0.03 radians.

the planning framework produces trajectories that are more stable and manipulable, potentially allowing the robot to react to disturbances more easily. For example, if this trajectory was used within a real-time MPC, the trajectory generated with the manipulability-stability cost could more easily be deformed if the robot hits a shelf rung or gets pushed. This is because the arms are away from singularity (allowing the robot to more easily deform its configuration) and is more stable.

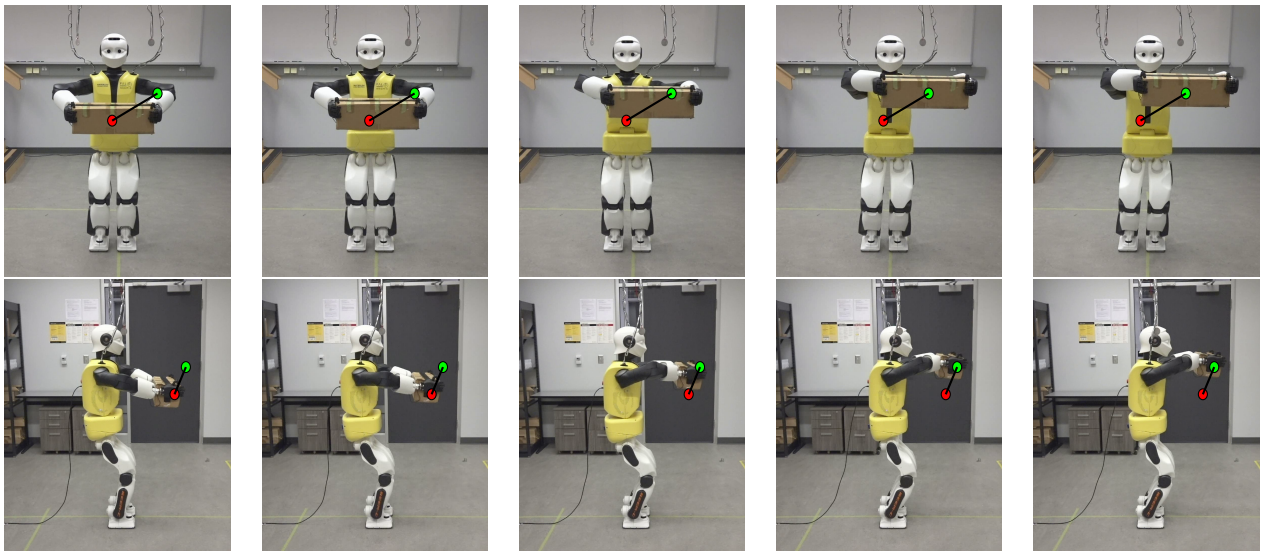


Figure 5.17: Experimental box trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. The red dot indicates the start of the trajectory and the green indicates the goal. Both arms are kept in manipulable configurations along the path as compared to the path constructed without the cost in Figure 5.18.

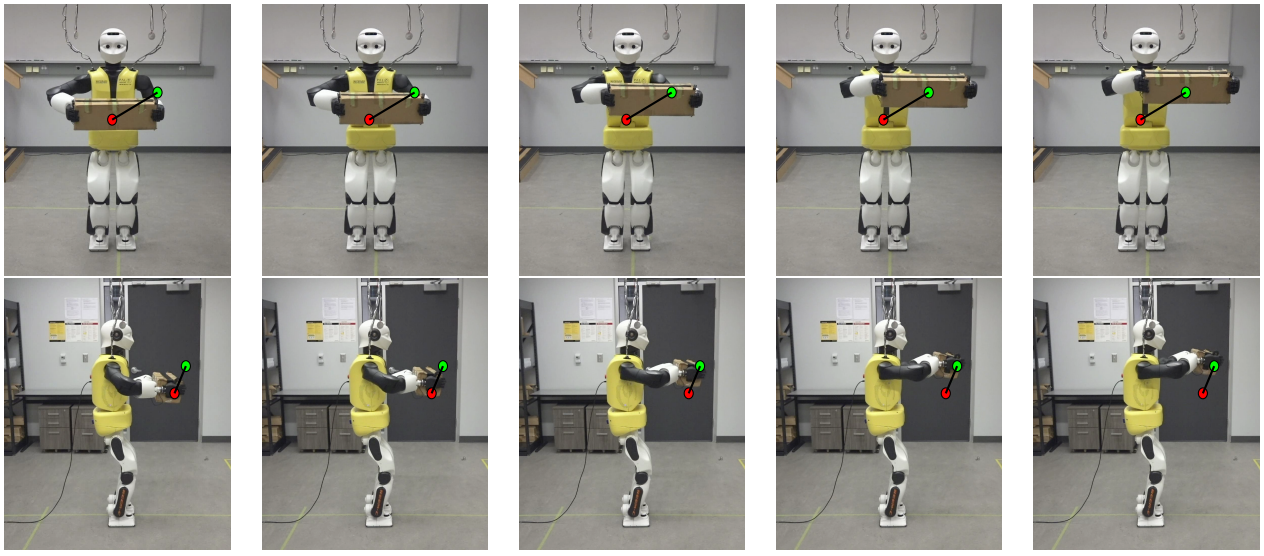


Figure 5.18: Experimental box trajectory without manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost. The red dot indicates the start of the trajectory and the green indicates the goal. At the end of the trajectory the left arm is more extended and near joint limits, resulting in lower path quality as was shown in the graph in Figure 5.15.

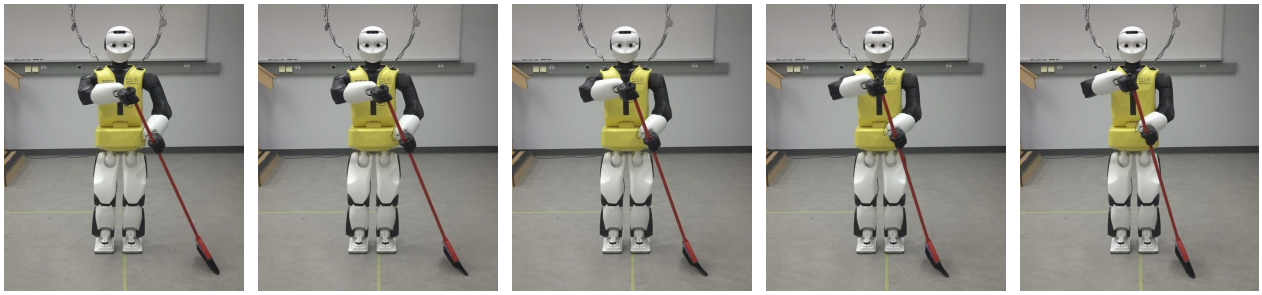


Figure 5.19: Experimental broom trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost.

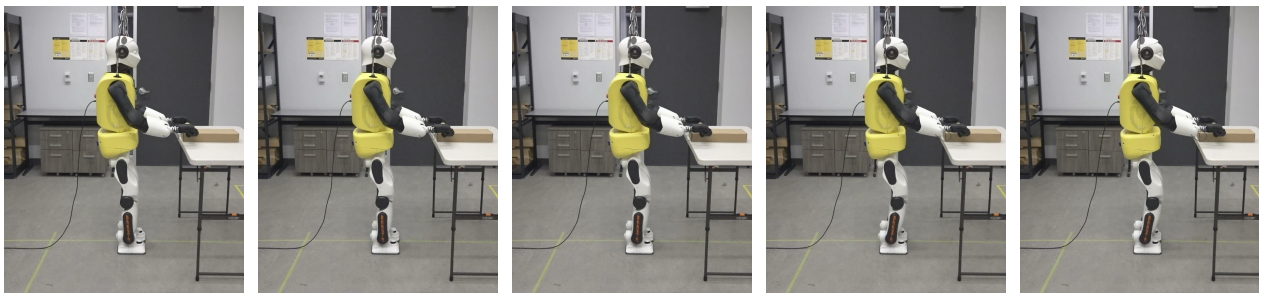


Figure 5.20: Experimental rolling pin trajectory with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost.

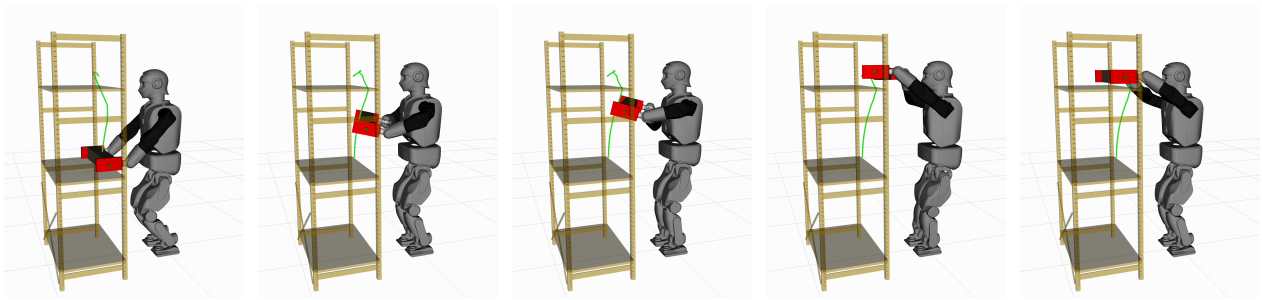


Figure 5.21: Experimental shelf box manipulation, with manipulability-stability cost. The trajectory starts on the leftmost frame and ends on the rightmost.

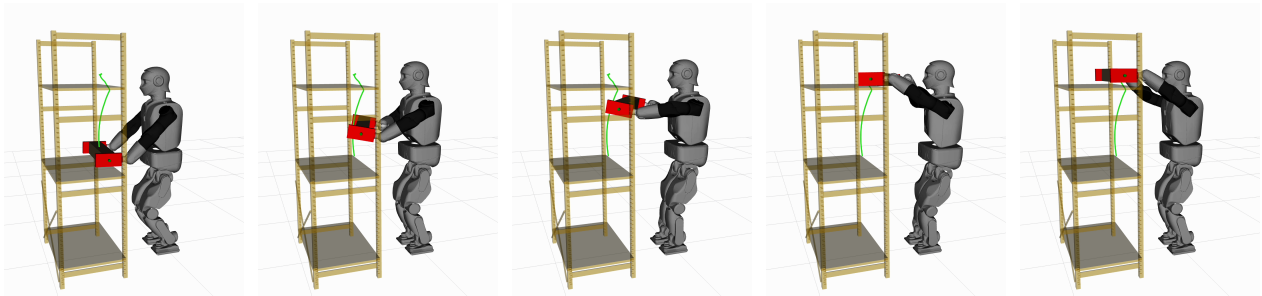


Figure 5.22: Experimental shelf box manipulation, without manipulability-stability cost. The arms are closer to singularity and the whole-body configurations are less stable along the trajectory as compared to Figure 5.21. A video of this motion (along with the one from Figure 5.21) is shown in this video: https://youtu.be/q8r3p5O4F6s.

5.7 Conclusions and Outlook

In this work, a bimanual manipulation workspace analysis was developed for humanoid robots, that considers a combined manipulability-stability metric based on the volume of the manipulability ellipsoid and the distance of the capture point from the edge of the support polygon. A general workspace was generated and visualized for the REEM-C as well as specific object workspace analyses for boxes, a broom and a rolling pin. Additionally, workspaces with varying CoM velocities were compared. These workspace analyses provide valuable insight into how well a humanoid robot can manipulate an object with two hands for a motion planning and control pipeline. Using this bimanual manipulability-stability metric, a BiRRT based motion planner was used to plan a collision-free motion trajectory. These motion plans were then used to generate a dynamically feasible motion using a TSID whole body QP. Motions for a box manipulation, broom sweeping and rolling pin rolling were demonstrated experimentally on the REEM-C humanoid. The motions with the manipulability-stability cost activated during planning were more stable and manipulable with the tradeoff of increased planning time. The trajectories also had very little end-effector error, adhering to the constraints set in the TSID QP.

Current directions for leveraging this work include planning hand trajectories for Segway handlebar motions while dynamically riding a Segway [199] and extending a reinforcement learning based walking controller to perform highly stable and manipulable box locomanipulation [222].

Future directions for this work will focus on exploring learning based approaches of applying the bimanual manipulability-stability metric to motion generation. The bimanual manipulability-stability metric could provide a useful dense reward for massively parallel RL training tools like Isaac Lab [4] and Brax [81]. Also, the pipeline presented in this work offers a convenient way to develop highly manipulable and stable whole body motions that could be used as demonstration data. Demonstrations have proven useful for pre-training learning based approaches and speeding up the learning by overcoming the need for large exploration in what can be a very sparse task to learn [95, 96, 138, 170, 175]. A learning based approach for bimanual manipulation would provide the ability to adapt to more manipulation cases from a single policy and offer a comparison to the optimization based approach presented in this work. An additional area for improvement would be to perform a detailed sensitivity analysis of the motion generation approach with respect to inaccuracies in the robot's kinematic and dynamic model compared to the REEM-C hardware.

Chapter 6

Advancing Humanoid Locomotion: Enhancing Versatility Through the use of Personal Transporters

Foreword

This chapter will be submitted to the Science Robotics journal. The citation is as follows:

V. Rajendran and K. Mombaur, Advancing Humanoid Locomotion: Enhancing Versatility Through the use of Personal Transporters. Science Robotics, 2024, To Be Submitted. [200]

Within the context of this thesis, this work combines and extends the work from previous chapters to enable the REEM-C humanoid to drive a Segway in real life. The video associated with this chapter is at the following link: https://youtu.be/eZxXSlk1v0w.

6.1 Introduction

Humanoid robotics has seen a surge of interest in the past few years due to several commercialization efforts by various companies. These robots are touted to do labour tasks in factories, fine manipulation and even elder care in home settings. While the outlook is ambitious, there are still several technical challenges to overcome. One such challenge is humanoid mobility. The humanoid form factor is versatile, and can integrate into existing environments where humans work or habitate. However, bipedal walking is typically reserved for short distances. Recent research has explored the use of wheeled humanoids as an alternative to legged locomotion, aiming to enhance speed and efficiency on flat, continuous surfaces [256, 257, 25]. While wheeled designs offer advantages in certain scenarios, they face limitations in navigating uneven or complex terrains, where legged locomotion remains more versatile and better suited for human-centric environments [220]. There are several works that show robust bipedal locomotion over various terrains and distances [194, 195, 119, 68, 137],

the walking is typically slower than an adult human gait and more importantly, expends a lot of energy from the system.

Looking to humans, they not only walk but also switch to wheeled modes of transport, using bikes, cars, scooters, or personal transporters (PTs) like Segways, to travel greater distances and carry loads. To address this, we explore the use of PTs to enhance the traversal abilities of humanoid robots. This approach allows robots to maintain their bipedal capabilities while utilizing more efficient wheeled transportation for longer distances.

Current commercial humanoids from the likes of Tesla, Figure, Apptronik and many more have battery lives from 1.5 to 5 hours [133, 19, 23], less than a typical 8 hour shift in a manufacturing environment. Not only this, BMW Group reported that some of their factory and logistics workers cover up to 12 km per day on foot. This spurred a project to introduce an efficient wheeled transportation device to aid in factory floor travel for distances that were too long to walk [34]. The project lead to the development of a statically stable 4 wheel transporter with a handlebar, that human workers could use to travel around the work site. Given BMWs recent partnership with Figure AI to deploy humanoids in their factories [35], would it make sense for humanoids to use similar devices to travel longer distances?

To motivate the usage of personal transporters (PTs) by humanoids, we performed a simple study with the humanoid robot used in this work, the PAL Robotics REEM-C [16], to compare the robot's energy usage to travel a short distance via walking versus using a personal transporter. Calculation details are provided in Appendix C, but to sum up, it was found that to walk a distance of 0.75m the REEM-C used 64677.6 Coulombs while only needing 18271.4 Coulombs while traveling the same distance, riding a Segway. This is a 72% reduction in battery usage to travel the same distance. This is a quantifiable advantage in favor of humanoids using PTs to locomote. It allows them to maintain the versatility of bipedal locomotion while increasing their efficiency for longer distance travel.

As noted in Chapter 3, the works in [129, 130, 243, 90, 144] either tackle the problem of operating a personal transporter in simulation, or use lightweight robots without challenging dynamics. Moreover, none of the works handle the bimanual steering which is required for many PTs.

In this work, we introduce a controller formulation that can readily generate coordinated and constrained whole-body motions to operate a Segway x2 SE. This Segway is a rugged all terrain version with a LeanSteer handlebar. Careful consideration of feet and hand contacts need to be made in order to be able to shift the robot's centre of mass forward and backward to command a linear velocity on the Segway, while also turning the handlebar simultaneously. Through modeling and implementation of a stack-of-tasks whole-body controller, we show that a humanoid can traverse a large area on a Segway, dramatically improving its efficiency. We test the controller rigorously with many empirical trials. First we show that the robot can command a linear velocity on the Segway (Figure 6.4). Next, we show that the robot can reject a small bump (Figure 6.7). In another trial, we show that the robot can reject a large disturbance applied at the torso (Figure 6.8). Several other experiments ranging from driving a serpentine path, rotating in place and driving over uneven terrain, further solidify the adaptability of the controller for personal transporter operation. This work directly uses the Segway model and controller identified in Section 3.3.



Figure 6.1: The REEM-C humanoid operating a Segway. a) The control framework in this work is deployed for various test scenarios on real hardware. b) Operation of the Segway is depicted in simulation, showing handlebar manipulation and forward and backward CoM shifting.

6.2 Materials and Methods

In this section, the overall controller formulation, mathematical modelling and individual controller components are described.

6.2.1 Controller Overview

The overall controller developed in this work is described in this section and detailed in Figure 6.2. Vicon motion capture is used for global (humanoid with respect to inertial frame) state estimation and an anchor point based method [42] is used for local (humanoid with respect to Segway) state estimation. Whole-body inverse kinematics generates a reference q^* which feeds a bimanual admittance controller that deforms the reference based on a projected wrench from the wrist FT sensors. A stabilizer (extended from our previous work [197]) takes in task space references and generates modified task spaces references which are finally optimized by the WBC. Since our REEM-C humanoid is position-controlled, the output joint accelerations are integrated twice using the controller timestep and the resulting q_d is sent to the robot at 200Hz.

6.2.2 Constrained Whole-body Inverse Kinematics

In this work, we use the EXOTica (**Ext**ensible **O**ptimization **T**oolset) framework [111] to solve the whole-body inverse kinematics problem for the REEM-C using an optimization approach. The optimization problem is set up using task maps that build the cost function and constraints. The task maps include considerations for joint limits, end effector positions and center of mass position, among others. The task maps used for the whole-body IK optimization are described in the Supplementary Material. In summary, tasks for CoM, feet position (fixed), hand positions (bimanual) are implemented as equality constraints, while the rest of the tasks are costs (manipulability-stability [224], joint pose regularization).

To formally define the whole-body IK problem, we refer to [248]:

$$\boldsymbol{q}^* = IK(\boldsymbol{q}_{seed}, C), \tag{6.1}$$

where C is the set of constraints and tasks (as described above) and $\boldsymbol{q}_{seed}$ is a given seed configuration (or start configuration). For this work, $\boldsymbol{q}_{seed}$ was chosen as the nominal standing position of the REEM-C with hands on the LeanSteer handlebar (as depicted in the block diagram of Figure 6.3). We use the SNOPT solver [88] to generate the constrained whole-body reference configuration.

Table 6.1 provides the nominal joint angles to seed the whole-body inverse kinematics block of the controller. The joint angles result in a standing configuration with arms holding the LeanSteer handlebar of the Segway (as pictured in Figure 6.1).

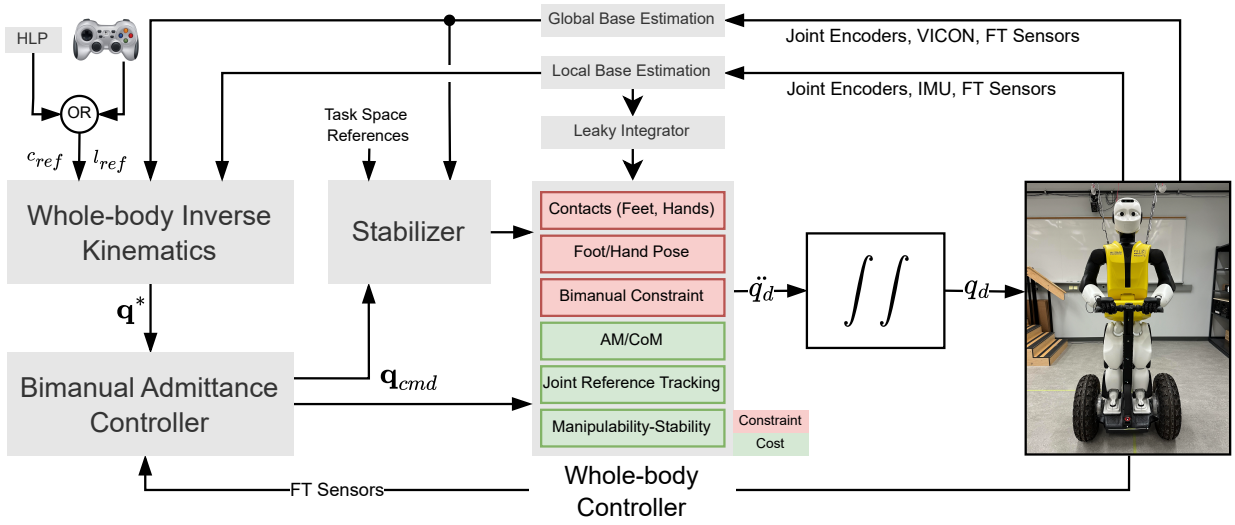


Figure 6.2: The stack-of-tasks whole-body controller (WBC) developed and used in this work shown in block diagram form.

Humanoid Chain	**Joint Angles (in order from index 0)**
Left Arm	$\{0.1566, 0.5616, -0.7356, 1.4812, -0.9243, 0.4709, -0.1941\}$
Right Arm	$\{0.1566, 0.5616, -0.7356, 1.4812, -0.9243, 0.4709, -0.1941\}$
Left Leg	$\{0.0, 0.0706, -0.122, 0.2439, -0.122, -0.0706\}$
Right Leg	$\{0.0, -0.0706, -0.122, 0.2439, -0.122, 0.0706\}$
Torso	$\{0.0, 0.0\}$
Head	$\{0.0, 0.0\}$

Table 6.1: Nominal joint angles used for whole-body IK. The nominal joint angles for the REEM-C humanoid standing in double support with hands on the LeanSteer handlebar.

6.2.3 Bimanual Model for Operating LeanSteer

Consider the bimanual humanoid system with two hands rigidly grasping the LeanSteer handlebar, as illustrated in Figure 6.3. The robot exerts forces with both hands, ensuring no relative motion occurs between the grasping points and the handlebar. The formulation follows exactly as described in Chapter 5 where the full formulation is given in Subsection 5.3.3.

Both hands rigidly grasping the LeanSteer handlebar, ensuring no relative motion between the hands and the handlebar. The forces exerted by the left and right hands, represented as $\boldsymbol{h}_{\mathrm{l}}$ and $\boldsymbol{h}_{\mathrm{r}}$, contribute to the net handlebar force $\boldsymbol{h}_{\mathrm{o}}$, governed by a grasp matrix $\boldsymbol{G}$. This matrix combines individual grasp matrices for each hand and includes identity, zero, and skew-symmetric matrices that relate to the hand positions. The derived constraint, involving the Jacobian $\boldsymbol{J}$, ensures consistent velocities between the hands and the object, serving as a foundation for the control strategy.

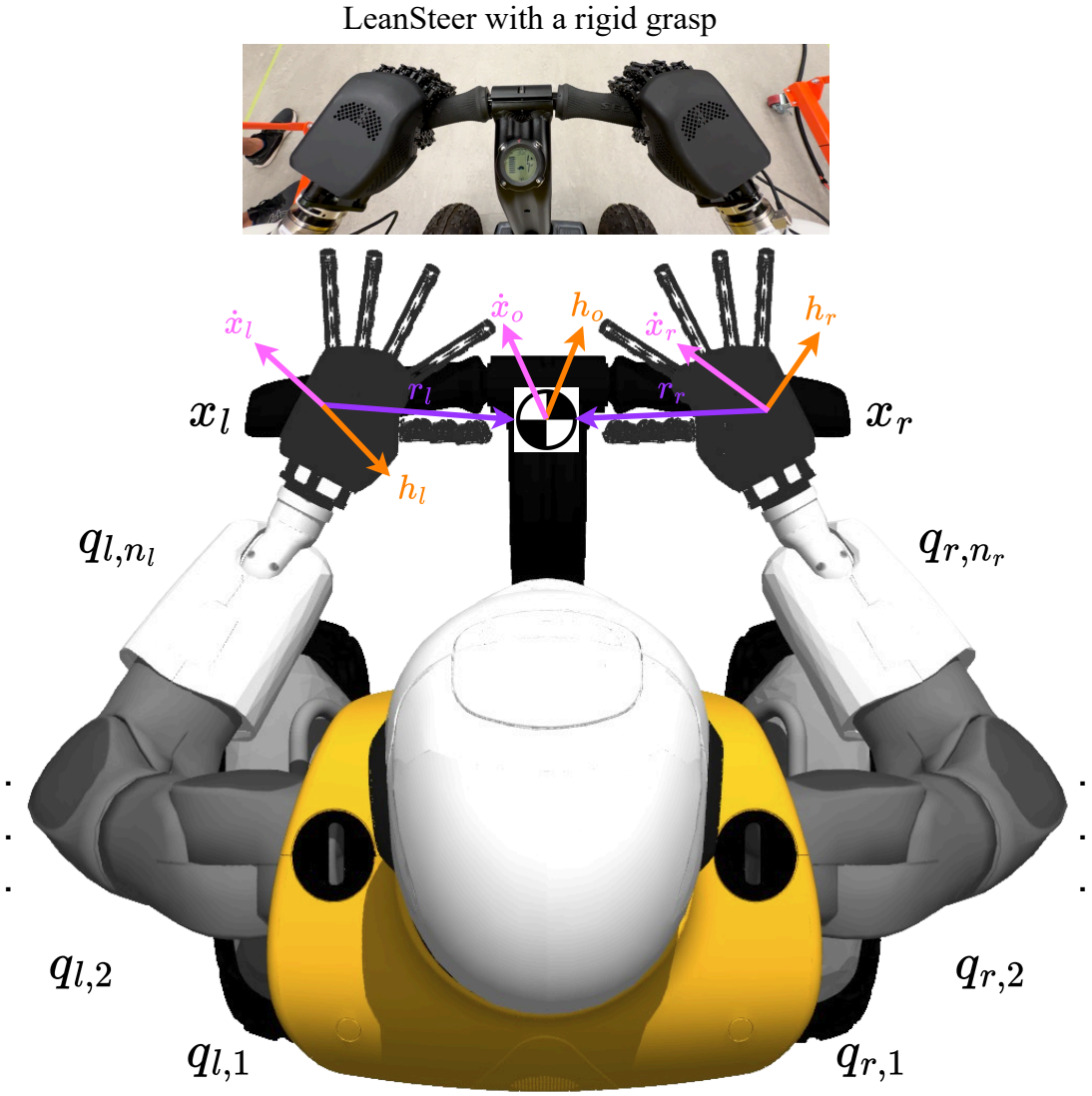


Figure 6.3: A diagram of the LeanSteer handlebar modeling. The REEM-C operates the LeanSteer mechanism of the Segway with a closed grasp. While turning the LeanSteer, the upperbody uses admittance control to minimize internal forces.

6.2.4 Bimanual Admittance Control

Admittance control is used to minimize interaction forces with the environment and the object held by the robot. By allowing trajectory adaptation based on sensed forces, it balances tracking accuracy and compliance. Our position-controlled robot, equipped with wrist force-torque sensors, uses a Cartesian impedance/admittance controller that follows the planned trajectory and adjusts based on the forces felt. The formulation follows exactly as described in Chapter 5 where the full formulation is given in Subsection 6.2.4.

6.2.5 Tasks Implemented in Whole-body Controller and Stabilizer

With the whole-body controller described in Equation 2.10, several task-space reference controllers (collectively, the "Stabilizer" in the block diagram of Figure 6.2) that set the references for the various task functions (Equation 2.8) are implemented for this work. They are extended from our previous work on stabilizing on moving platforms [197].

The stabilizer components are listed below with further mathematical details in our previously published work [197]:

- **Ankle Admittance:** The ankle admittance controller maintains the robot's ZMP (zero moment point) within its support polygon by yielding to forces at the foot through commanded ankle roll and pitch velocities. This is achieved by a PID-based control strategy, where the ZMP error drives the commanded ankle velocities, which are then integrated to obtain desired positions and sent directly to the ankle actuators.
- **CoM Admittance:** The CoM admittance controller uses a modified table-cart model for the x direction, incorporating ankle flexibility as an offset angle α, and is controlled using an LQR approach to maintain stability and disturbance rejection. In the y direction, the CoM control is based on a PID formulation driven by the ZMP error, with the reference ZMP generated through a divergent component of motion (DCM) feedback law.
- **Contact Wrench Distribution:** For the contact wrench distribution problem, we follow a similar approach to [234], using Newton and Euler equations to account for the desired values c^d, $\ddot{c}_{des}$, and $\dot{L}_{des}$ (angular momentum rate), and p_{des} (ZMP reference), to obtain the desired contact wrenches $w_{LF_{des}}$, $w_{RF_{des}}$, $w_{LH_{des}}$, and $w_{RH_{des}}$, where subscripts denote the left foot, right foot, left hand, and right hand, respectively. The redundancy from having four contacts is addressed by formulating a quadratic program (QP) that minimizes the magnitude of each contact wrench while satisfying the system's dynamics.

In terms of tasks, the ones listed in Figure 6.2 were implemented and the task space references for them were generated from the list above. Other than those listed, the "Joint Reference Tracking" task tracks q_{cmd}, the output of the bimanual admittance control. The "Angular Momentum" task tries to regulate the system's angular momentum to zero since the CoM is on a fixed plane for riding a Segway. We can easily generate a centroidal angular

momentum reference for personal transporters that require variations in angular momentum references. The last task implemented is Manipulability-Stability which is a metric proposed in our previous work on workspace analysis for humanoids [224]. The task maximizes a composite cost function that balances the bimanual manipulability of the system along with capture point stability.

For the TSID whole-body QP, the task gains were set as in Table 6.2. All task damping gains (K_D) were set to their critical value $2\sqrt{K_P}$. For the priority levels listed, 0 indicates a constraint and 1 indicates a cost.

Task	**Weight**	**Priority**	K_P **gain**
Torque Bounds	-	0	-
Joint Bounds	-	0	-
Bimanual Constraint	-	0	-
Feet Contacts (6D)	1	0	30
Hand Contacts (6D)	1	0	30
Center of Mass	1e3	1	12.5
Angular Momentum	1e-4	1	10
Feet Pose	1	1	300
Hand Pose	1	1	300
Joint Reference Tracking (30 DoF)	1e2	1	all equal (1)
Manipulability-Stability	1e-2	1	10

Table 6.2: The TSID Task Gains for driving a personal transporter.

6.2.6 Implementation Details

For global floating base state estimation, we utilize Vicon motion capture in the control loop. Markers are placed at the base link of the robot and the 6D pose of the base is given at 200Hz. This is fed into an extended Kalman filter (EKF) via the SEROW package (State Estimation RObot Walking) [14] and we calculate the base position and velocity for use within the controller. This state estimator gives us the global position and velocity of the REEM-C and transporter system together.

We also implemented another *local* state estimator which gives us the floating base estimate relative to the base of the Segway. This allows us to know, for example, how far the centre of mass of the robot is shifted relative to the Segway. This is useful to implement kinematic limits to avoid collisions and to stop the Segway system easily by shifting the CoM forward or backward. This estimator is implemented using the anchor point method. With the robot being in double support (and not making and breaking contacts, such as, when walking), estimating the base position and orientation is trivial with kinematics and the IMU inside the base [42].

The controller was written in Python using the ROS Melodic middleware (Ubuntu 18.04) and ran on a desktop AMD Ryzen 9 7900x with 32GB of RAM. Whole-body position commands were sent to the REEM-C over a ROS topic.

6.3 Results

In this section, we present a series of experiments demonstrating the robustness and adaptability of the proposed controller across diverse scenarios. We cover driving on flat ground, mixed motions of the Segway, driving on uneven terrain and disturbance rejection. We first describe the REEM-C humanoid, hardware used for experiments, controller modelling before detailing each experiment and their results.

6.3.1 Elementary Motions

In this section, we show that the REEM-C can operate a Segway and perform elementary motions. These are driving straight, stopping and rotating in place. Driving straight and stopping requires CoM shifting while rotating in place requires turning of the LeanSteer handlebar.

Driving Straight on Flat Ground

In this first experiment, the robot is commanded to drive straight for 2 metres before coming to rest. It achieves this by shifting it's center of mass forward and then shifting it backward to decelerate and then recovering to roughly it's original CoM position, to stop. We note that this is a challenging task given that 4 contacts (2 at the feet and 2 at the hands) are constraining the motion. Figure 6.4 a) shows snapshots of the real hardware experiment. Figure 6.4 b) shows the path traveled by the REEM-C in the x-y plane. The REEM-C's local center of mass shifting stages are shown in Figure 6.4 c). The red indicates the portion where the robot commands a linear velocity on the Segway, the blue is when the robot starts to decelerate by shifting its CoM backward and green indicates the recovery stage where it re-centers to come to rest. Figure 6.4 d), e) and f) show CoM tracking and forces felt at the ankles and wrists. As can be seen, the most force is felt when the REEM-C is shifting backwards to decelerate.

Rotating in Place

In this experiment, the robot was commanded to maintain a linear velocity of 0 m/s while turning a full 2π radians, effectively rotating in place. This is a challenging task for humans since it is hard to turn the handlebar without shifting one's center of mass forward or backward. Real robot experiments are shown in Figure 6.5 a) and simulation experiments are shown in Figure 6.5 b). As can be seen, from the graph plotted in Figure 6.5 c), the robot is able to turn a full $2pi$ radians (red) while maintaining a linear velocity of 0 m/s. Here, the robot's stand in place controller was activated with a reference velocity of 0 m/s. The robot can easily achieve this precise motion of the Segway.

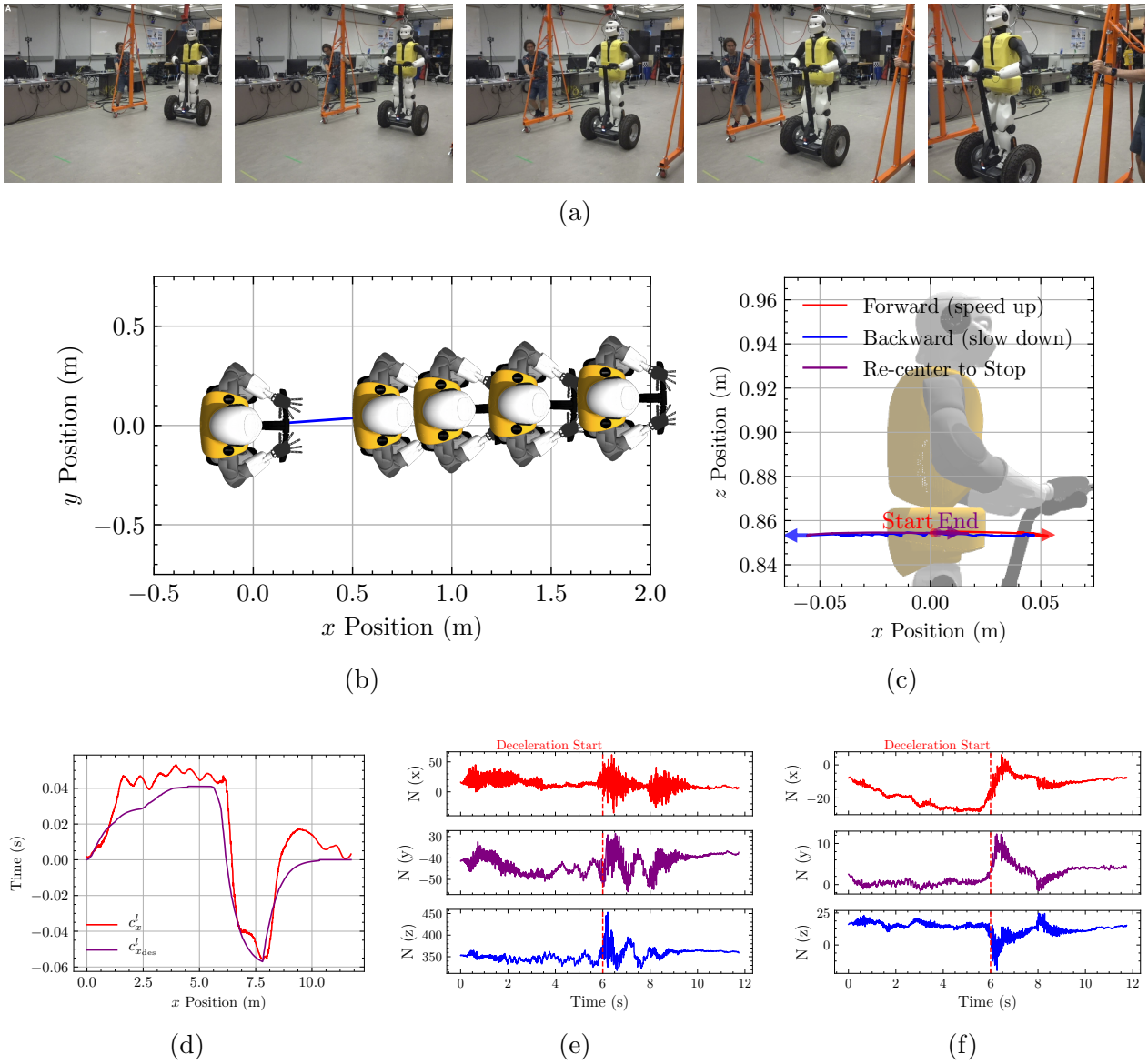
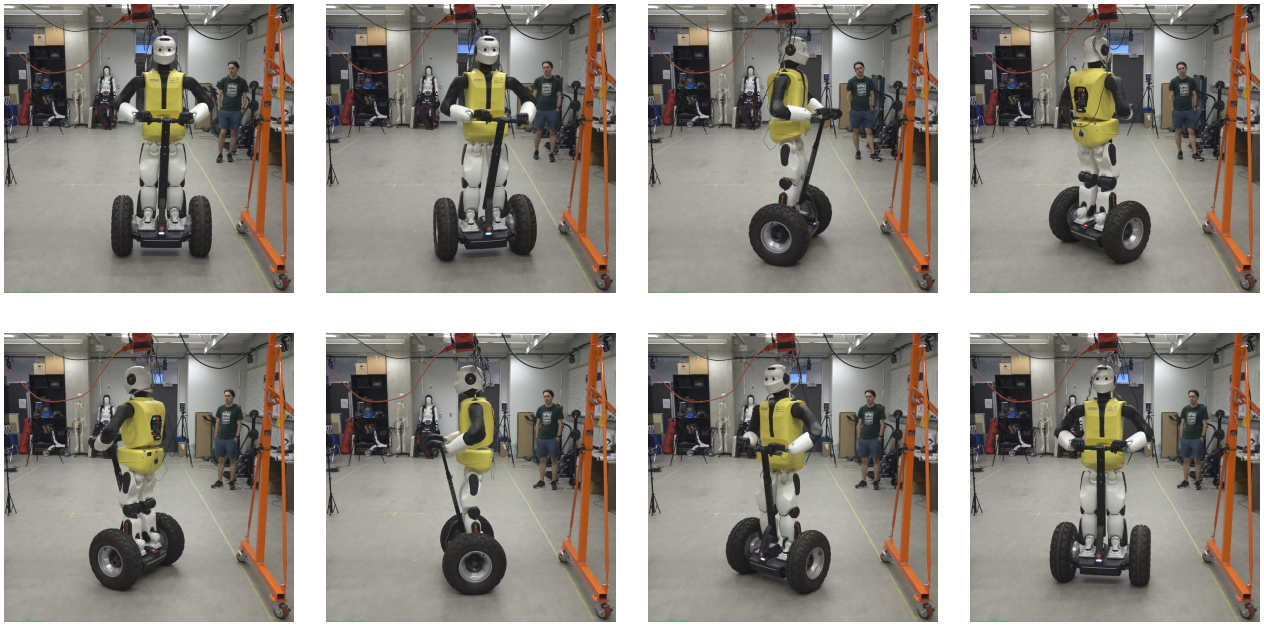
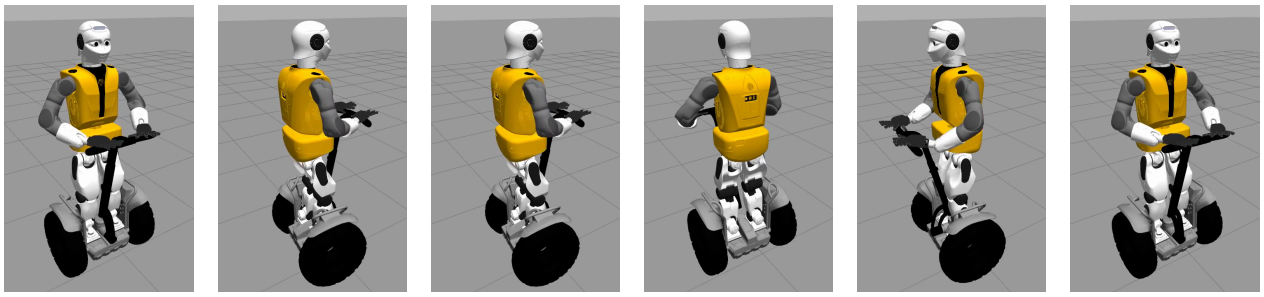


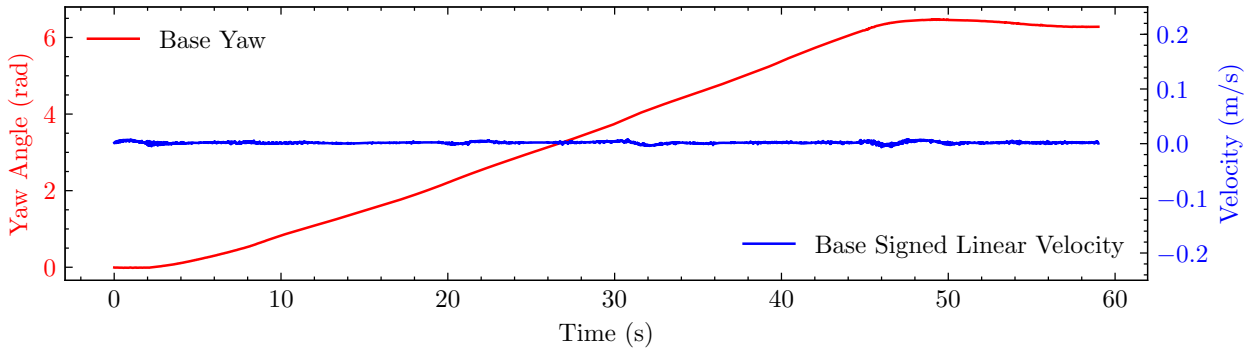
Figure 6.4: REEM-C driving straight on a Segway. a) Snapshots of the REEM-C driving a Segway. The humanoid starts from rest, drives straight and then stops after 2 m. b) An x-y plot of the path travelled by REEM-C during this trial. c) The local CoM trajectory in the x-direction (c_x^l) traversed by the REEM-C. Red indicates the start of the trajectory, blue is when the direction changes to start to slow down and green is when the robot re-centers itself to bring the system to a stop. d) The local CoM trajectory traversed by the REEM-C plotted against the desired local CoM trajectory. The forces felt at the ankles e) of the REEM-C and the wrists f). The start of the decelerations are indicated with a vertical red line.



(a)



(b)



(c)

Figure 6.5: The REEM-C humanoid rotating in place by turning the LeanSteer handlebar. a) Real hardware trial snapshots of the REEM-C rotating in place. b) Simulation trial snapshots of the REEM-C rotating in place. c) Plot of the floating base yaw in red and floating base signed linear velocity in green. As can be seen, the linear velocity is maintained close to zero and the robot does a full rotation of 2π.

6.3.2 Mixed Motions

In this section, we report the results of a mixed motion of the Segway; that is, commanding a linear velocity while turning at the same time. From the robot's perspective, mixed motions refers to CoM shifting while operating the LeanSteer handlebar.

Driving a Serpentine Path

A serpentine maneuver seems simple but requires careful whole-body coordination and force control. The whole-body inverse kinematics module is important here as it gives a good starting configuration for the controller to deform with admittance control and then track with the whole-body controller. Without this module, the optimization in the whole-body controller may take longer to find a suitable configuration that can satisfy all the constraints and costs. In Figure 6.6 a), we see the real hardware experiment. The REEM-C starts at one end of the lab and turns at various points, making a snake like path through the environment. This is shown in the x-y plot in Figure 6.6 b). Figure 6.6 c) and d) shown the right ankle forces and torques, respectively. There is activity in the middle of the trajectory once the REEM-C starts to turn the LeanSteer back and forth. Here, it is aiming to keep the forces among all contacts equal. The forces and torques settle at the end of the trajectory when the robot stops.

6.3.3 Complex Motions

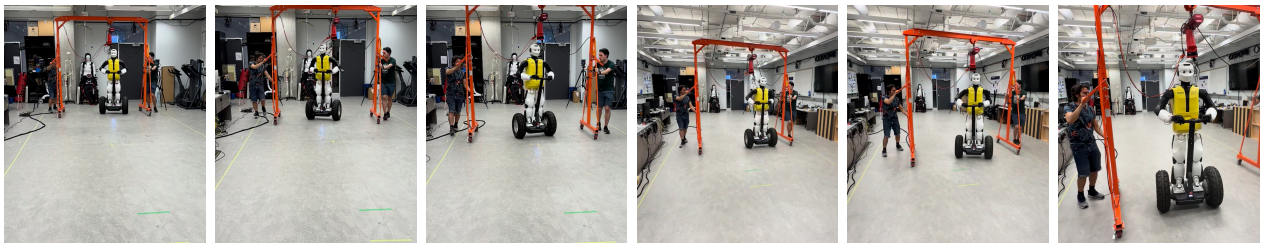
In this section, we detail three experiments where the controller faces unknown disturbances in the form of pushes, bumps and uneven terrain. In all cases, the REEM-C was able to stabilize the system and continue on its path to its final goal.

Driving Straight Over a Bump

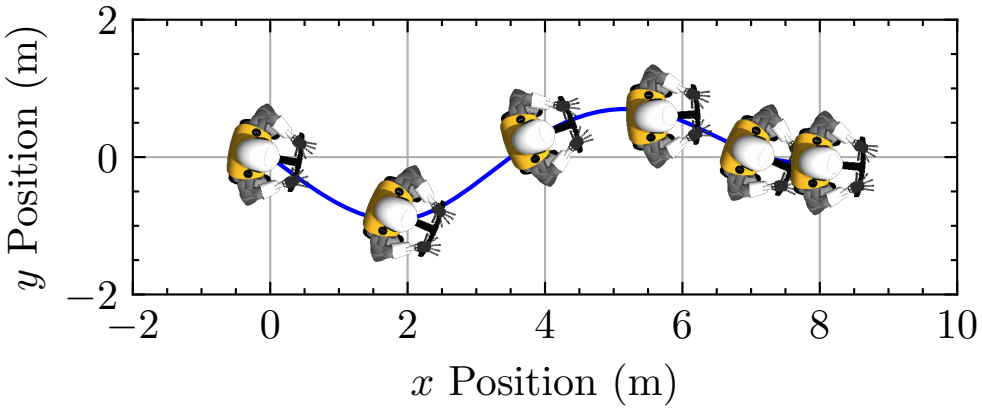
In this next experiment, the REEM-C is commanded to drive straight and over a bump. The bump is constructed using exercise mats. As can be seen in the frames of Figure 6.7 a), the robot starts from rest, climbs the bump and descends from it before coming to a stop. This is also depicted in the 3D plot in Figure 6.7 b). The robot is able to reject this unknown disturbance and the controller stays stable despite the large forces felt during the descent as seen in the left ankle force and torque graphs.

Robustness to External Disturbance

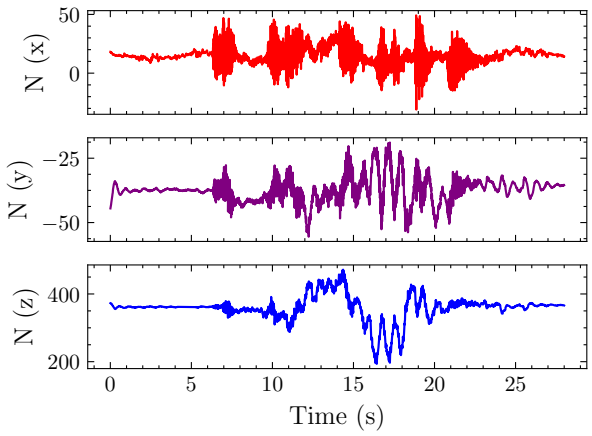
In this experiment, a human pushes the REEM-C while it is standing still, with a push stick. The force applied is 113N. The stand-in-place controller is activated here and the frames from the experiment are shown in Figure 6.8 a). We can see that the robot recieves a push in the second frame, starts travelling backward in the global inertial frame (as can be seen by the blue line in Figure 6.8 c)) but then immediately starts shifting it's CoM forward in



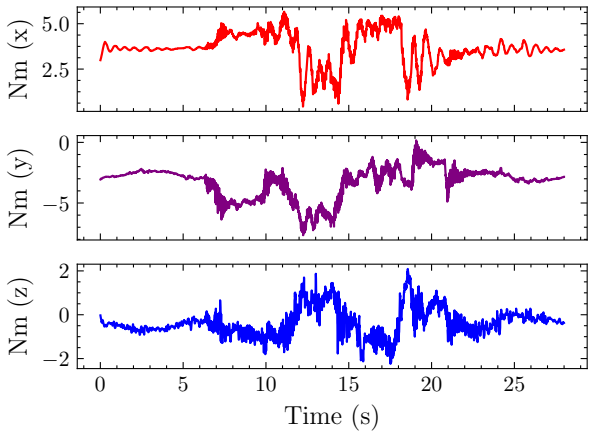
(a)



(b)



(c)

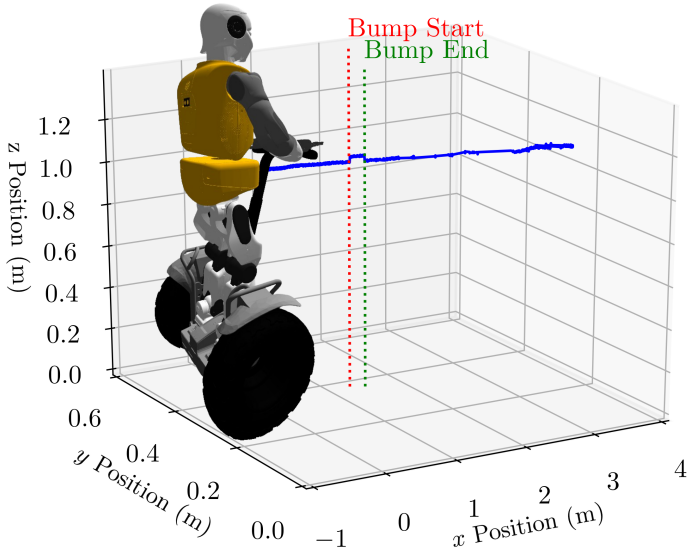


(d)

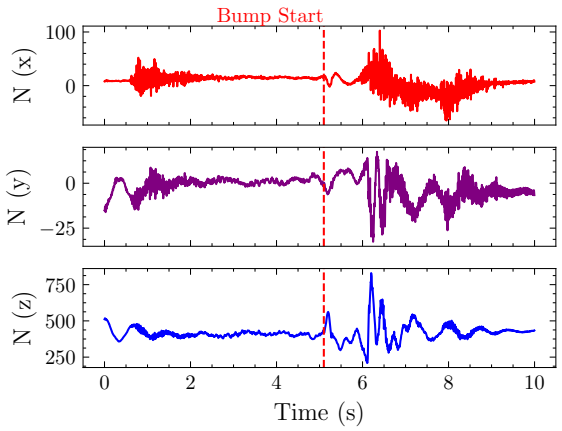
Figure 6.6: REEM-C driving a serpentine path. a) Real hardware snapshots of the REEM-C humanoid driving a serpentine path. b) The path travelled by the REEM-C in the x-y plane. The c) forces and d) torques measured at the right ankle. The spikes in the graphs indicate points that the robot turned the LeanSteer; they settled at the end of the trajectory.



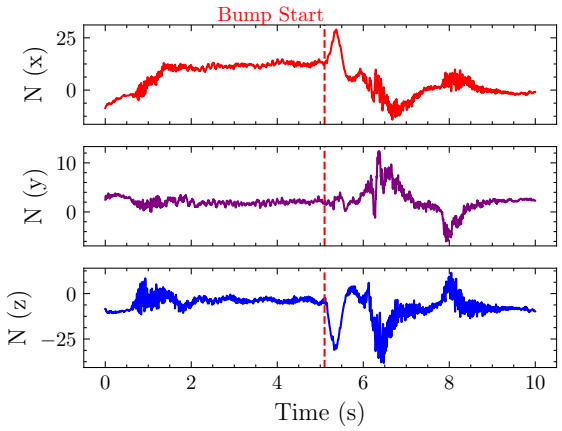
(a)



(b)



(c)



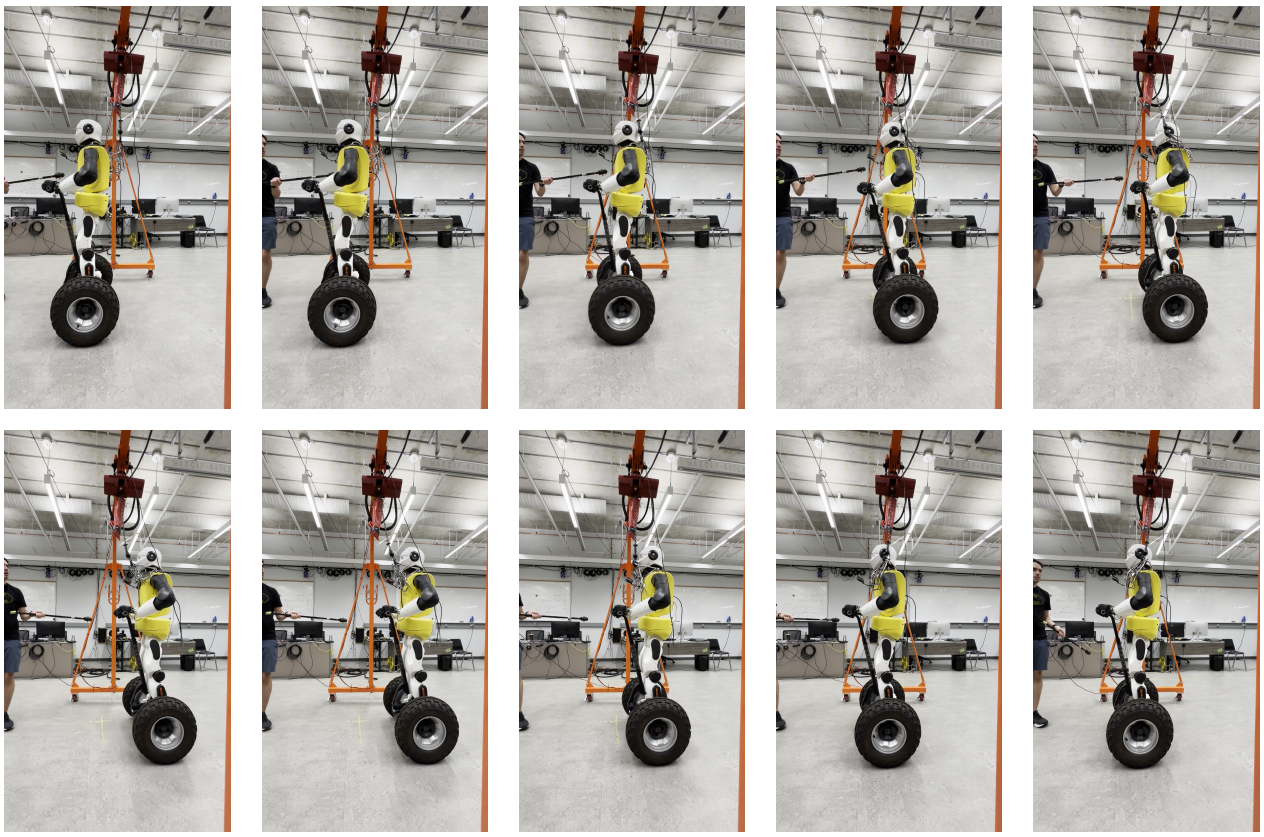
(d)

Figure 6.7: REEM-C driving over a bump. a) Snapshots of the experiment where REEM-C drove over a bump (exercise mats). b) The path taken by the REEM-C plotted in 3D. The bump is denoted by vertical dashed lines. c) The left ankle forces during the trial. Climbing the bump, the forces are contained but descending the bump causes large spikes that settles after a few seconds. d) The forces measured at the left wrist. We see that similar to the ankle forces, the bump is clearly noted in the plots.

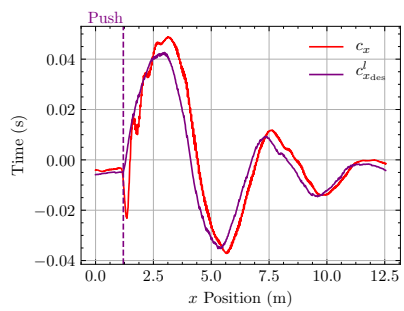
the local frame to counteract the push. After a few seconds, the push is absorbed and the REEM-C travels back to the original location and the system velocity reaches steady-state around 0 m/s (as shown in Figure 6.8 d)). This trial shows the controller robustness and system adaptability to unknown disturbances. The controller can sustain even larger pushes, with the largest one recorded at 176 N.

Driving Over Uneven Terrain

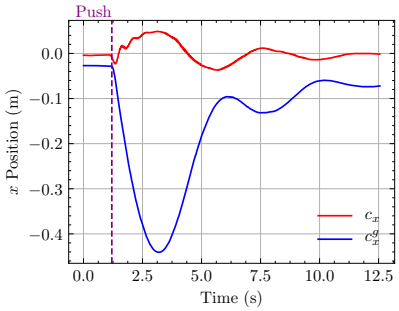
In this trial, the REEM-C drove over a set of bike tires that acted as disturbances or rough terrain. The tires were randomly thrown into the air and landed on the ground in the configuration seen in the snapshots of Figure 6.9. The robot started from rest and drove forward with automatic heading correction activated. The heading correction controller monitored the floating base yaw angle and corrected the heading back to the reference, zero, by manipulating the LeanSteer handlebar. It maintained its forward velocity throughout this trial and successfully drove across the tires. The controller was able to readily reject the tire disturbances, manipulate the LeanSteer and drive to the goal location.



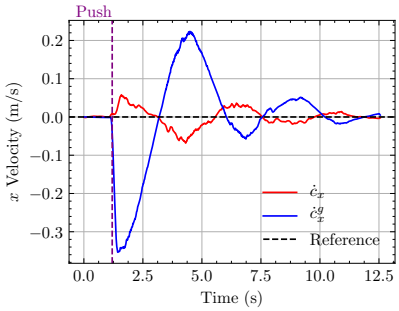
(a)



(b)



(c)



(d)

Figure 6.8: REEM-C self-balancing on a Segway and rejecting a push disturbance. a) The REEM-C humanoid rejecting a disturbance from an instrumented push stick (the push happens on frame ii, t=1.2s). The push was 113N and lasted for 0.7s. b) The desired local CoM in the x-direction ($c^l_{x_{\text{des}}}$) plotted against the measured local CoM in the x-direction (c^l_x). c) The measured local CoM in the x-direction plotted against the measured global CoM in the x-direction (c^g_x). d) The measured local CoM velocity in the x-direction $\dot{c}^l_x$ plotted against the measured global CoM velocity in the x-direction $\dot{c}^g_x$.

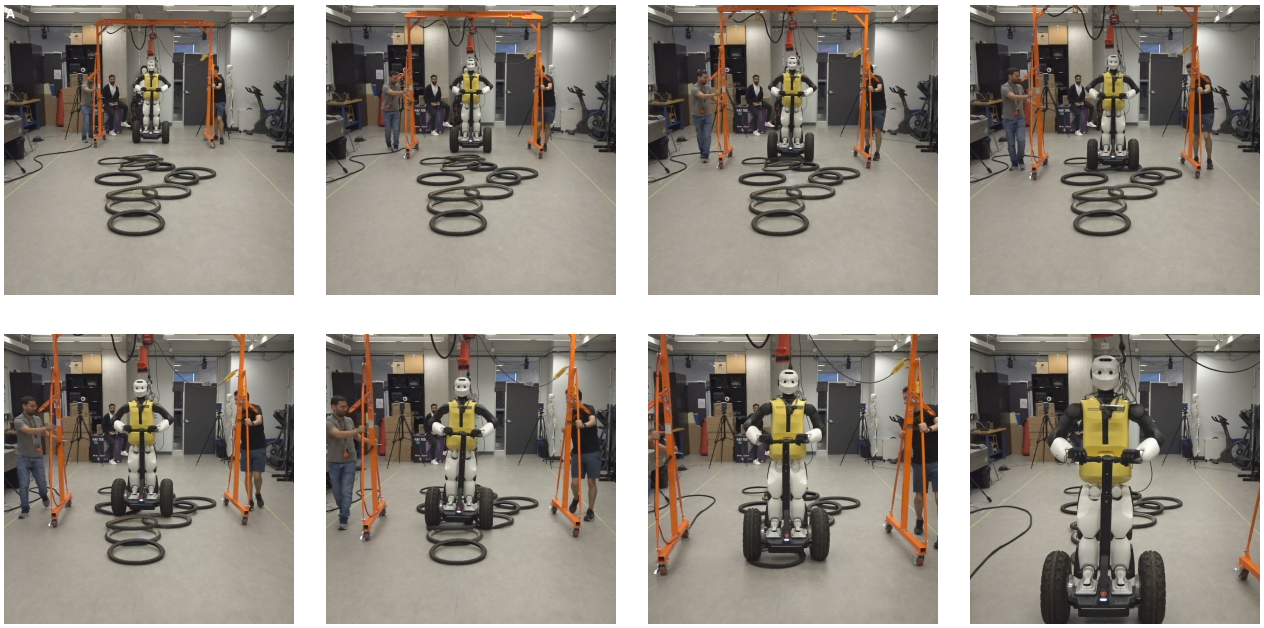


Figure 6.9: The REEM-C humanoid driving over a set of randomly strewn bike tires. In these snapshots, the REEM-C drives over a set of bike tires that act as disturbances or uneven terrain. The robot can easily drive over the tires with the proposed controller.

6.4 Discussion

We introduce a stack-of-tasks-based whole-body controller designed for a full-sized humanoid robot to drive a Segway PT. This controller leverages a hierarchy of tasks balanced through optimization, enabling the humanoid to operate the PT safely and reliably. The formulation incorporates multiple constraints, including a bimanual constraint that allows the humanoid to manipulate the handlebar while simultaneously shifting its center of mass (CoM) to control the Segway. This type of hybrid motion presents significant challenges, as the robot is constrained both at its feet (standing in double support on the Segway) and at its hands (controlling the LeanSteer handlebar).

The controller demonstrates robust and adaptive performance across diverse scenarios, including navigating over bumps, bicycle tires (rough terrain), and unexpected disturbances. Its success underscores the effectiveness of the hybrid strategy, making it resilient even in conditions that would typically disrupt balance and control.

As humanoid robotics progresses, achieving versatile locomotion will be crucial. In human environments, wheeled transport is vital for efficient travel over longer distances, suggesting that humanoids must acquire similar capabilities to become genuinely multi-functional. By developing human-like skills such as operating personal transporters, humanoids can better integrate into workspaces, households, and other human-centric environments, making these capabilities essential for their seamless adaptation to everyday life.

6.4.1 Limitations and Future Work

Our approach demonstrates promising results in controlling a Segway PT, showcasing robustness across diverse terrains, resilience to external disturbances, and the capability to produce mixed motions. However, there are a few limitations that will be addressed in future work. The current implementation uses a position-controlled robot with a relatively low control bandwidth (200 Hz), making it vulnerable to very large disturbances (e.g., a push exceeding 200N). This limitation is mainly due to hardware constraints, and with faster onboard computing, the control bandwidth could be increased to improve performance.

Another limitation is the absence of onboard global state estimation. For this study, VICON motion capture was integrated to validate the controller's robustness across various test scenarios. In future work, we aim to incorporate visual odometry, enabling the system to operate independently in diverse environments without external aids.

Future research will also explore the controller's adaptability to other personal transporters (PTs). While Segways are challenging for humanoids to control due to their underactuated nature and inherent instability, other PTs used in industrial settings, campus security, and similar venues are statically stable with three or four wheels. These systems still require an active controller to maintain balance during motion but are generally easier to control. We also plan to investigate more sophisticated control schemes, such as model predictive control (MPC) or reinforcement learning.

In the context of learning-based control, we believe the proposed controller provides a strong foundation for a wide range of PTs. The current method considers key elements of PT riding, such as standing in double support and grasping handlebars, which could be further enhanced by a learning-based approach that develops a higher-level policy. For instance, learning could generate reference trajectories for the center of mass (CoM) or handlebar orientation. This hybrid approach would combine things like hard constraints and easy objective tuning of optimization-based control with the generalizability of learning-based methods, expanding the controller's versatility across various tasks and environments.

Chapter 7

Discussion and Conclusions

In this chapter we provide a discussion and conclusion for this thesis by summarizing the results presented, and providing an outlook for future work.

7.1 Summary of Work

The research presented in this thesis has shown that it is feasible and more efficient for humanoids to operate personal transporters when travelling long distances. To extend the capabilities of the REEM-C to drive a Segway (used as an exemplar personal transporter), a comprehensive motion generation pipeline was proposed and implemented on real hardware.

First, in Chapter 3 a study involving humans riding Segways was implemented to reverse engineer the Segway's controller to create a high-fidelity simulation model. We synthesized controllers for the the humanoid robot REEM-C to ride a Segway in simulation, motivated by human Segway riding. We performed motion capture experiments of a human riding a Segway and identified human whole-body behaviour as well as the Segway's internal controllers. We then showed that the REEM-C could successfully generate translational, rotational and mixed motions of the Segway in simulation. The Segway is controlled by targeted motions of the REEM-C using an inverted pendulum based LQR controller for pitch control and an admittance controller for the LeanSteer to command a yaw-rate.

In Chapter 4, stabilizer design for humanoid balancing on actuated balance boards was explored. We qualitatively studied how humans balance on balance boards to motivate our humanoid controller design. The developed approach combined various strategies from the literature including ankle admittance, CoM admittance, base pitch admittance, foot force difference admittance, contact wrench distribution and angular momentum regulation in a new integration to achieve stabilization on an actuated balance board. We explored the frontal and sagittal plane balance board problems along with pushes at the upper body and showed that our approach works in simulation and on real hardware. The work brought us one step closer to balancing on unactuated balance boards and other dynamic environments such as moving platforms and transportation devices.

In Chapter 5, a bimanual manipulation workspace analysis for humanoid robots was developed and used for planning manipulable and stable trajectories. This analysis considered manipulability and whole-body stability for a workspace where constraints existed between the hands of the humanoid induced by the object shape and grasp. A combined manipulability-stability metric based on the volume of the manipulability ellipsoid and the distance of the capture point from the edge of the support polygon was proposed. This metric was visualized in a variety of workspace scenarios including those where the humanoid's center of mass is moving at a certain velocity and where it is grasping and carrying objects of different masses and shapes. With a focus on tightly coupled bimanual manipulation of varying symmetry, objects studied included boxes, a broom, and a rolling pin. A general workspace and a box manipulation workspace are visualized for both the REEM-C and TALOS humanoids, showing differences in the generated workspace volumes due to the varying topologies of the humanoids. These visualizations aimed to provide insights into how manipulability and whole-body stability are affected by bimanual manipulation scenarios. Subsequently, the combined manipulability-stability metric is used to plan bimanual manipulation motions. A BiRRT planner planned collision-free trajectories for the robot and object being manipulated using constrained whole-body inverse kinematics with the combined manipulability-stability metric as a cost. A TSID whole-body QP generateed a final motion with dynamics considerations for the object being manipulated using the previously planned path. This pipeline was proven in simulation as well as experimentally on the REEM-C robot for the various objects.

Finally, in Chapter 6, the works from all previous chapters are combined to enable the REEM-C humanoid to operate a Segway. We developed control strategies that enable humanoids to safely and effectively operate PTs while maintaining balance. This control task differs from walking control and presents unique challenges, requiring quick reactions and precise adjustments. Our approach utilized a stack of tasks quadratic program (QP) formulation, which considers contacts, stability, and bimanual manipulation constraints. We demonstrated the effectiveness of our controller through experimental results with a REEM-C humanoid operating a Segway x2 SE. The Segway model developed in Chapter 3 was used to validate the developed controller (which in turn is based on the work in Chapter 4). The bimanual motion generation pipeline presented in Chapter 5 is used to operate the LeanSteer handlebar of the Segway.

7.2 Limitations and Future Work

Several successful real robot experiments were shown in this work but there are still some limitations and scope for improvement. Firstly, this thesis only considered the control of the PT once the robot was already mounted onto it. Mounting and egressing the PT is a challenge on its own given that many PTs are unstable when weight is first placed on them. This will need to be addressed using methods from stair climbing literature [41] combined with balancing strategies on unstable platforms [197, 246]. Secondly, Vicon was used in the loop to estimate the floating base position and orientation of the humanoid-Segway system in free space. This would not be possible in a general outdoor environment, so a method

such as visual odometry or visual SLAM would need to be investigated and integrated.

To continue this line of research, testing this controller on more PTs would be needed. A Segway is arguably harder to operate than a statically stable PT such as a four-wheeled scooter. Nonetheless, the control scheme would need to be adapted based on the humanoid-PT dynamics and kinematics. Using methods like inverse optimal control can help to automatically tune the whole-body controller weights and gains, limiting the need for hand tuning. In terms of sensing, using a solution such as robotic skin [49] could prove useful for more precise maneuvers on PTs. The work in [49] shows examples of applying robotic skin on the whole body of a humanoid. Having extra force sensing capabilities at the hands and feet of the humanoid can allow feedback of more accurate center of pressure for balancing and more accurate forces for upper body admittance control. Investigating the use of MPC or reinforcement learning for the task of operating PTs is also an interesting future work. The lookahead capabilities of MPC and rapid exploration of the action space with RL could lead to a more general purpose control policy that can be ported to many different PTs. Apart from controller improvements, integrating a full navigation stack would allow the robot to autonomously traverse any area. This would involve the integration of high-level planners that take in semantic environment information and synthesize paths for the PT to follow. Obstacle avoidance, and safe navigation in the presence of humans [196] is also a line of work that could propel this research forward.

References

[1] Atlas™. https://www.bostondynamics.com/atlas. Accessed: 2023-01-15.

[2] C++ Reimplementation of eiquadprog. https://github.com/stack-of-tasks/eiquadprog. Accessed: 2023-01-15.

[3] Efficient Task Space Inverse Dynamics (TSID) based on Pinocchio. https://github.com/stack-of-tasks/tsid. Accessed: 2023-01-15.

[4] Isaac Lab. https://isaac-sim.github.io/IsaacLab/main/index.html. Accessed: 2024-10-30.

[5] KickScooter E22 User Manual. https://www.bhphotovideo.com/lit_files/626231.pdf. (Accessed on 10/26/2021).

[6] Ninebot S-Max User Manual. https://files.bbystatic.com/jMP8iMXZghtMJ67lhzAXHA%3D%3D/User%2BManual. (Accessed on 10/26/2021).

[7] Quadratic Programming Solver. https://github.com/quadprog/quadprog. Accessed: 2023-01-15.

[8] Quadratic programming solvers in Python with a unified API. https://github.com/qpsolvers/qpsolvers. Accessed: 2023-01-15.

[9] REEM-C - PAL Robotics : Leading company in service robotics. https://pal-robotics.com/wp-content/uploads/2019/07/Datasheet_REEM-C.pdf.

[10] RoboSavvy-Balance. http://wiki.ros.org/Robots/RoboSavvy-Balance. Accessed: 2021-08-03.

[11] Segway Drift W1 User Manual. https://store.segway.com/pub/media/wysiwyg/warranty/drift-w1-user-manual.pdf. (Accessed on 10/26/2021).

[12] Segway x2 Adventure. https://grabcad.com/library/segway-x2-adventure-1. Accessed: 2021-01-30.

[13] Segway x2 User Manual. https://static4.segway.com/wp-content/uploads/2019/09/24010-00001_ab_pr_se_user_manual.pdf. (Accessed on 10/26/2021).

[14] SEROW Framework for Humanoid Robot Walking Estimation. https://github.com/mrsp/serow. Accessed: 2023-01-15.

[15] Time Parameterization. http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/time_parameterization/time_parameterization_tutorial.html. Accessed: 2020-06-30.

[16] REEM-C. https://robots.ieee.org/robots/reemc/, Aug 2018.

[17] Bruno Vilhena Adorno, Philippe Fraisse, and Sébastien Druon. Dual position control strategies using the cooperative dual task-space framework. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 3955–3960, October 2010.

[18] Navid Aghasadeghi and Timothy Bretl. Inverse optimal control for differentially flat systems with application to locomotion modeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6018–6025, 2014.

[19] Figure AI. Figure AI: Developing General-Purpose Humanoids. https://www.figure.ai, 2024. Accessed: 2024-10-22.

[20] R. McN. Alexander. The Gaits of Bipedal and Quadrupedal Animals. *The International Journal of Robotics Research*, 3(2):49–59, 1984.

[21] F. Allgower, Rolf Findeisen, and Z.K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35:299–315, 05 2004.

[22] Stuart Anderson and Jessica Hodgins. Adaptive torque-based control of a humanoid robot on an unstable platform. pages 511 – 517, 01 2011.

[23] Apptronik. Apollo - Apptronik's General-Purpose Humanoid Robot. https://apptronik.com/apollo, 2024. Accessed: 2024-10-22.

[24] Tamim Asfour, Kazuhito Yokoi, C. Lee, and James Kuffner. Humanoid Robotics. *IEEE Robotics Automation Magazine - IEEE ROBOT AUTOMAT*, 19:108–118, 03 2012.

[25] Donghoon Baek, Youngwoo Sim, Amartya Purushottam, Saurabh Gupta, and Joao Ramos. Toward Control of Wheeled Humanoid Robots with Unknown Payloads: Equilibrium Point Estimation via Real-to-Sim Adaptation, 2024.

[26] P. Bakaráč, M. Klaučo, and M. Fikar. Comparison of inverted pendulum stabilization with PID, LQ, and MPC control. In *2018 Cybernetics Informatics (K I)*, pages 1–6, 2018.

[27] Moses Bangura and Robert Mahony. Real-time Model Predictive Control for Quadrotors. *IFAC Proceedings Volumes*, 47(3):11773–11780, 2014. 19th IFAC World Congress.

[28] Alberto Bemporad. Model Predictive Control, 2021.

[29] Khairidine Benali, Jean-François Brethé, François Guérin, and Marc Gorka. Dual arm robot manipulator for grasping boxes of different dimensions in a logistics warehouse. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 147–152, 2018.

[30] Mehdi Benallegue and Florent Lamiraux. Estimation and Stabilization of Humanoid Flexibility Deformation Using Only Inertial Measurement Units and Contact Information. *International Journal of Humanoid Robotics*, 12(3):15500255, September 2015.

[31] Dmitry Berenson, Siddhartha S. Srinivasa, Dave Ferguson, and James J. Kuffner. Manipulation planning on constraint manifolds. In *2009 IEEE International Conference on Robotics and Automation*, pages 625–632, 2009.

[32] Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. Evidence for Composite Cost Functions in Arm Movement Planning: An Inverse Optimal Control Approach. *PLoS computational biology*, 7:e1002183, 10 2011.

[33] Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten, and Marco Hutter. Keep Rollin'—Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots. *IEEE Robotics and Automation Letters*, 4(2):2116–2123, Apr 2019.

[34] BMW Group Press. Rethinking Micro-Mobility – The BMW Group Personal Mover Concept: Electric, Safe, Agile – For Short Distances Across Sites. https://www.press.bmwgroup.com/global/article/detail/T0280762EN/rethinking-micro-mobility-%E2%80%93-the-bmw-group-personal-mover-concept-electric-safe-agile-%E2%80%93-for-short-distances-across-sites?language=en, 2018. Accessed: 2024-10-22.

[35] BMW Group Press. Successful Test of Humanoid Robots at BMW Group Plant Spartanburg. https://www.press.bmwgroup.com/usa/article/detail/T0444268EN_US/successful-test-of-humanoid-robots-at-bmw-group-plant-spartanburg?language=en_US, 2023. Accessed: 2024-10-22.

[36] Luc Boutin, Antoine Eon, Saïd Zeghloul, and Patrick Lacouture. From human motion capture to humanoid locomotion imitation Application to the robots HRP-2 and HOAP-3. *Robotica*, 29:325–334, 03 2011.

[37] Arthur Bryson, Y.-C Ho, and George Siouris. Applied Optimal Control: Optimization, Estimation, and Control. *Systems, Man and Cybernetics, IEEE Transactions on*, 9:366 – 367, 07 1979.

[38] Fabrizio Caccavale and Masaru Uchiyama. *Cooperative Manipulation*, pages 989–1006. Springer International Publishing, Cham, 2016.

[39] Eduardo Camacho and Carlos Bordons. *Model Predictive Control*, volume 13. 01 2004.

[40] Leobardo Campos, Oscar Carbajal-Espinosa, Alexander Loukianov, and Eduardo Bayro-Corrochano. Stabilization method for dynamic gait in bipedal walking robots. pages 1276–1281, 11 2016.

[41] Stephane Caron. Floating-Base Estimation for Legged Robots. https://scaron.info/robotics/floating-base-estimation.html, 2024. Accessed: 2024-10-28.

[42] Stéphane Caron, Abderrahmane Kheddar, and Olivier Tempier. Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 277–283, 2019.

[43] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, 2019.

[44] A. Castro. Modeling and dynamic analysis of a two-wheeled inverted-pendulum. 2012.

[45] Konstantinos Chatzilygeroudis, Bernardo Fichera, Ilaria Lauzana, Fanjun Bu, Kunpeng Yao, Farshad Khadivar, and Aude Billard. Benchmark for Bimanual Robotic Manipulation of Semi-Deformable Objects. *IEEE Robotics and Automation Letters*, 5(2):2443–2450, 2020.

[46] Lawrence Yunliang Chen, Baiyu Shi, Roy Lin, Daniel Seita, Ayah Ahmad, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. Bagging by Learning to Singulate Layers Using Interactive Perception, 2023.

[47] Lawrence Yunliang Chen, Baiyu Shi, Daniel Seita, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. AutoBag: Learning to Open Plastic Bags and Insert Objects. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3918–3925, 2023.

[48] Shuxiao Chen, Jonathan Rogers, Bike Zhang, and Koushil Sreenath. Feedback Control for Autonomous Riding of Hovershoes by a Cassie Bipedal Robot. *CoRR*, abs/1907.11353, 2019.

[49] Gordon Cheng, Emmanuel Dean-Leon, Florian Bergner, Julio Rogelio Guadarrama Olvera, Quentin Leboutet, and Philipp Mittendorfer. A Comprehensive Realization of Robot Skin: Sensors, Sensing, Control, and Applications. *Proceedings of the IEEE*, 107(10):2034–2051, 2019.

[50] Gordon Cheng, Sang-Ho Hyon, Jun Morimoto, Aleš Ude, Joshua G. Hale, Glenn Colvin, Wayco Scroggin, and Stephen C. Jacobsen. CB: A Humanoid Research Platform for Exploring NeuroScience. *Advanced Robotics*, 21(10):1097–1114, 2007.

[51] Teng-Hu Cheng, Han-Pang Huang, Jiu-Lou Yan, and Yi-Wen Chao. Development of a walking stabilizing controller for humanoid robots. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 40–45, 2010.

[52] E. Chiovetto, B. Berret, and T. Pozzo. Tri-dimensional and triphasic muscle organization of whole-body pointing movements. *Neuroscience*, 170(4):1223–1238, 2010.

[53] Enrico Chiovetto and Martin A. Giese. Kinematics of the Coordination of Pointing during Locomotion. *PLOS ONE*, 8(11):1–16, 11 2013.

[54] William Clark. *Control of a Beam-Hoverboard Self-Transportation System with Model Predictive Control*. PhD thesis, 05 2019.

[55] Debora Clever, Monika Harant, Henning Koch, Katja Mombaur, and Dominik Endres. A novel approach for the generation of complex humanoid walking sequences based on a combination of optimal control and learning of movement primitives. *Robotics and Autonomous Systems*, 83:287–298, 2016.

[56] Debora Clever, Monika Harant, Katja Mombaur, Maximilien Naveau, Olivier Stasse, and Dominik Endres. COCoMoPL: A Novel Approach for Humanoid Walking Generation Combining Optimal Control, Movement Primitives and Learning and its Transfer to the Real Robot HRP-2. *IEEE Robotics and Automation Letters*, 2(2):977–984, 2017.

[57] Debora Clever, Yue Hu, and Katja Mombaur. Humanoid gait generation in complex environments based on template models and optimality principles learned from human beings. *The International Journal of Robotics Research*, 37(10):1184–1204, 2018.

[58] Debora Clever and Katja Mombaur. A new template model for optimization studies of human walking on different terrains. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 500–505, 2014.

[59] Benjamin Cohen, Sachin Chitta, and Maxim Likhachev. Search-based planning for dual-arm manipulation with upright orientation constraints. In *2012 IEEE International Conference on Robotics and Automation*, pages 3784–3790, 2012.

[60] Michael F. Cohen, J. J. Gelfand, and Charles F. Rose. Verbs and adverbs: multidimensional motion interpolation using radial basis functions. 1999.

[61] Thomas Cohn, Seiji Shaw, Max Simchowitz, and Russ Tedrake. Constrained Bimanual Planning with Analytic Inverse Kinematics, 2023.

[62] Silvia Cruciani, Diogo Almeida, Danica Kragic, and Yiannis Karayiannidis. Discrete Bimanual Manipulation for Wrench Balancing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631–2637, 2020.

[63] Airton R. da Silva Jr. and Frank C. Sup IV. A Robotic Walker Based on a Two-Wheeled Inverted Pendulum. *J. Intell. Robotic Syst.*, 86(1):17–34, 2017.

[64] Ewen Louis Dantec, Rohan Budhiraja, Adria Roig, Teguh Lembono, Guilhem Saurel, Olivier Stasse, Pierre Fernbach, Steve Tonneau, Sethu Vijayakumar, Sylvain Calinon, Michel Taïx, and Nicolas Mansard. Whole Body Model Predictive Control with a Memory of Motion: Experiments on a Torque-Controlled Talos. In *International Conference on Robotics and Automation (ICRA 2021)*, Xi'an, China, May 2021. IEEE.

[65] Andrea D'Avella and M.C. Tresch. Modularity in the motor system: Decomposition of muscle patterns as combinations of time-varying synergies. *Advances in Neural Information Processing Systems*, 14:141–148, 01 2001.

[66] P. de Leva. Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics*, 29:1223–1230, 1996.

[67] Andrea Del Prete, Nicolas Mansard, Oscar E. Ramos, Olivier Stasse, and Francesco Nori. Implementing Torque Control with High-Ratio Gear Boxes and Without Joint-Torque Sensors. *International Journal of Humanoid Robotics*, 13(01):1550044, March 2016. Publisher: World Scientific Publishing Co.

[68] Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[69] Holger Diedam, Dimitar Dimitrov, Pierre-Brice Wieber, Katja Mombaur, and Moritz Diehl. Online walking gait generation with adaptive foot positioning through Linear Model Predictive control. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1121–1126, 2008.

[70] C. R. G. Dreher, Mirko Wächter, and Tamim Asfour. Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks. *IEEE Robotics and Automation Letters*, 5(1):187–194, 2020.

[71] George Ellis. Chapter 6 - Four Types of Controllers. In George Ellis, editor, *Control System Design Guide (Fourth Edition)*, pages 97–119. Butterworth-Heinemann, Boston, fourth edition edition, 2012.

[72] Johannes Englsberger, George Mesesan, Alexander Werner, and Christian Ott. Torque-Based Dynamic Walking - A Long Way from Simulation to Experiment. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 440–447, 2018.

[73] Johannes Englsberger, Christian Ott, and Alin Albu-Schäffer. Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion. *IEEE Transactions on Robotics*, 31(2):355–368, 2015.

[74] Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, and Alin Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 916–923, 2014.

[75] Kemalettin Erbatur and Utku Seven. An inverted pendulum based approach to biped trajectory generation with swing leg dynamics. In *IEEE-RAS Conf Human Rob*, pages 216–221, 2007.

[76] Tom Erez, Kendall Lowrey, Yuval Tassa, Vikash Kumar, Svetoslav Kolev, and Emanuel Todorov. An integrated system for real-time model predictive control of humanoid robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 292–299, 2013.

[77] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, New York, 2008. OCLC: ocn190774140.

[78] Martin L. Felis. RBDL: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, pages 1–17, 2016.

[79] Michael W. Foley, Rhonda H. Julien, and Brian R. Copeland. A Comparison of PID Controller Tuning Methods. *The Canadian Journal of Chemical Engineering*, 83(4):712–722, 2005.

[80] Michael G. Forbes, Rohit S. Patwardhan, Hamza Hamadah, and R. Bhushan Gopaluni. Model Predictive Control in Industry: Challenges and Opportunities. *IFAC-PapersOnLine*, 48(8):531–538, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[81] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation, 2021.

[82] R.J. Full and D.E. Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 12 1999.

[83] Jianfeng Gao, You Zhou, and Tamim Asfour. Projected Force-Admittance Control for Compliant Bimanual Tasks. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, 2018.

[84] Manolo Garabini, Danilo Caporale, Vinicio Tincani, Alessandro Palleschi, Chiara Gabellieri, Marco Gugliotta, Alessandro Settimi, Manuel Giuseppe Catalano, Giorgio Grioli, and Lucia Pallottino. WRAPP-up: A Dual-Arm Robot for Intralogistics. *IEEE Robotics & Automation Magazine*, 28(3):50–66, 2021.

[85] Gianluca Garofalo, Christian Ott, and Alin Albu-Schäffer. Walking control of fully actuated robots based on the Bipedal SLIP model. In *2012 IEEE International Conference on Robotics and Automation*, pages 1456–1463, 2012.

[86] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Compliant leg behavior explains basic dynamics of walking and running. *Proceedings. Biological sciences / The Royal Society*, 273:2861–7, 12 2006.

[87] Martin A. Giese, Albert Mukovskiy, Aee ni Park, Lars Omlor, and Jean-Jacques E. Slotine. Real-Time Synthesis of Body Movements Based on Learned Primitives. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2008.

[88] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[89] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, September 1983.

[90] Yukai Gong, Ross Hartley, Xingye Da, Ayonga Hereid, Omar Harib, Jiunn-Kai Huang, and Jessy Grizzle. Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway. In *Am Ctrl Conf*, pages 4559–4566, 2019.

[91] Ruben Grandia, Andrew Taylor, Andrew Singletary, Marco Hutter, and Aaron Ames. Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions. *Robotics: Science and Systems XVI*, Jul 2020.

[92] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer. JOE: a mobile, inverted pendulum. *IEEE Trans Ind Elec*, 49:107–114, 2002.

[93] Robert J. Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. Walking stabilization using step timing and location adjustment on the humanoid robot, Atlas. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017.

[94] Erico Guizzo and Evan Ackerman. The hard lessons of DARPA's robotics challenge. *Spectrum, IEEE*, 52:11–13, 08 2015.

[95] Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and Match: Supercharging Imitation with Regularized Optimal Transport. (arXiv:2206.15469), February 2023. arXiv:2206.15469 [cs].

[96] Siddhant Haldar, Jyothish Pari, Anant Rai, and Lerrel Pinto. Teach a Robot to FISH: Versatile Imitation from One Minute of Demonstrations. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023.

[97] Corey Hart and Simon Giszter. Distinguishing synchronous and time-varying synergies using point process interval statistics: motor primitives in frog and rat. *Frontiers in Computational Neuroscience*, 7:52, 2013.

[98] Yanhao He, Min Wu, and Steven Liu. An Optimisation-Based Distributed Cooperative Control for Multi-Robot Manipulation with Obstacle Avoidance. *IFAC-PapersOnLine*, 53(2):9859–9864, 2020. 21st IFAC World Congress.

[99] Bernd Henze, Máximo A. Roa, and Christian Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *The International Journal of Robotics Research*, 35(12):1522–1543, 2016.

[100] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online Walking Motion Generation with Automatic Foot Step Placement. *Advanced Robotics*, 24(5-6):719–737, 2010.

[101] Hirohisa Hirukawa, Fumio Kanehiro, Kenji Kaneko, Shuuji Kajita, Kiyoshi Fujiwara, Yoshihiro Kawai, Fumiaki Tomita, Shigeoki Hirai, Kazuo Tanie, Takakatsu Isozumi, Kazuhiko Akachi, Toshikazu Kawasaki, Shigehiko Ota, Kazuhiko Yokoyama, Hiroyuki

Handa, Yutaro Fukase, Jun-ichiro Maeda, Yoshihiko Nakamura, Susumu Tachi, and Hirochika Inoue. Humanoid robotics platforms developed in HRP. *Robotics and Autonomous Systems*, 48:165–175, 10 2004.

[102] At Hof. The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human movement science*, 27:112–25, 03 2008.

[103] Michael A. Hopkins, Dennis W. Hong, and Alexander Leonessa. Compliant locomotion using whole-body control and Divergent Component of Motion tracking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5726–5733, 2015.

[104] Yue Hu and Katja Mombaur. Bio-Inspired Optimal Control Framework to Generate Walking Motions for the Humanoid Robot iCub Using Whole Body Models. *Applied Sciences*, 8:278, 02 2018.

[105] Jensen Huang. Nvidia's Jensen Huang Promises a New Computing Age Led by Taiwanese Tech Companies, 2024. Accessed: 2024-10-22.

[106] Weiwei Huang, Chee-Meng Chew, Yu Zheng, and Geok-Soon Hong. Pattern generation for bipedal walking on slopes and stairs. In *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pages 205–210, 2008.

[107] Mikulas Huba. *Selected Topics on Constrained and Nonlinear Control*, pages 87–185. 01 2011.

[108] Marco Hutter, C. David Remy, Mark A. Höpflinger, and Roland Siegwart. SLIP running with an articulated robotic leg. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4934–4939, 2010.

[109] Sang-Ho Hyon, Joshua G. Hale, and Gordon Cheng. Full-Body Compliant Human–Humanoid Interaction: Balancing in the Presence of Unknown External Forces. *IEEE Transactions on Robotics*, 23:884–898, 2007.

[110] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation*, 25(2):328–373, 02 2013.

[111] Vladimir Ivan, Yiming Yang, Wolfgang Merkt, Michael P. Camilleri, and Sethu Vijayakumar. EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control. In Anis Koubaa, editor, *Robot Operating System (ROS)*, volume 778, pages 211–240. Springer International Publishing, Cham, 2019. Series Title: Studies in Computational Intelligence.

[112] Hyobin Jeong, Okkee Sim, Hyoin Bae, KangKyu Lee, Jaesung Oh, and Jun-Ho Oh. Biped walking stabilization based on foot placement control using capture point feedback. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5263–5269, 2017.

[113] Wanxin Jin, Dana Kulić, Jonathan Feng-Shun Lin, Shaoshuai Mou, and Sandra Hirche. Inverse Optimal Control for Multiphase Cost Functions. *IEEE Transactions on Robotics*, 35(6):1387–1398, 2019.

[114] Steven G. Johnson. The NLopt nonlinear-optimization package. https://github.com/stevengj/nlopt, 2007.

[115] Steven Jens Jorgensen, Mihir Vedantam, Ryan Gupta, Henry Cappel, and Luis Sentis. Finding Locomanipulation Plans Quickly in the Locomotion Constrained Manifold. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6611–6617, 2020.

[116] Peter Kaiser, David Gonzalez-Aguirre, Fabian Schültje, Júlia Borràs, Nikolaus Vahrenkamp, and Tamim Asfour. Extracting whole-body affordances from multimodal exploration. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1036–1043, 2014.

[117] Peter Kaiser, Nikolaus Vahrenkamp, Fabian Schültje, Júlia Borràs, and Tamim Asfour. Extraction of Whole-Body Affordances for Loco-Manipulation Tasks. *International Journal of Humanoid Robotics*, 12:1550031, 06 2015.

[118] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Conf Rob Autom*, volume 2, pages 1620–1626, 2003.

[119] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1620–1626 vol.2, 2003.

[120] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. 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)*, volume 1, pages 239–246 vol.1, 2001.

[121] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1405–1411 vol.2, 1991.

[122] Shuuji Kajita, Hirohisa Hirukawa, Kensuke Harada, and Kazuhito Yokoi. ZMP and Dynamics. In *Introduction to Humanoid Robotics*, pages 69–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[123] Shuuji Kajita, Mitsuharu Morisawa, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, Kiyoshi Fujiwara, and Hirohisa Hirukawa. Biped Walking Pattern Generator allowing Auxiliary ZMP Control. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2993–2999, 2006.

[124] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shin'ichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496, 2010.

[125] R. E. Kalman. When Is a Linear Control System Optimal? *Journal of Basic Engineering*, 86(1):51–60, 03 1964.

[126] Tohru Katayama, TAKAHIRA OHKI, TOSHIO INOUE, and TOMOYUKI KATO. Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control - INT J CONTR*, 41:677–699, 03 1985.

[127] Mahmoud Khaled, Ali Mohammed, M.Shaaban Ibraheem, and Reda Ali. *Balancing a Two Wheeled Robot*. PhD thesis, 07 2009.

[128] Kyunam Kim, Patrick Spieler, Elena-Sorina Lupu, Alireza Ramezani, and Soon-Jo Chung. A bipedal walking robot that can fly, slackline, and skateboard. *Science Robotics*, 6(59):eabf8136, 2021.

[129] Kohei KIMURA, Hiroto MIZOHANA, Shunichi Nozawa, Kei Okada, and Masayuki INABA. Speed governing behavior on parallel two-wheeled electric scooter by life-sized humanoid robot based on learning control with online tuning for PID gain during control. *Transactions of the JSME (in Japanese)*, 06 2018.

[130] Kohei Kimura, Shunichi Nozawa, Hiroto Mizohana, Kei Okada, and Masayuki Inaba. Riding and Speed Governing for Parallel Two-Wheeled Scooter Based on Sequential Online Learning Control by Humanoid Robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[131] Victor Klemm, Alessandro Morra, Ciro Salzmann, Florian Tschopp, Karen Bodie, Lionel Gulich, Nicola Küng, Dominik Mannhart, Corentin Pfister, Marcus Vierneisel, Florian Weber, Robin Deuber, and Roland Siegwart. Ascento: A Two-Wheeled Jumping Robot. *CoRR*, abs/2005.11435, 2020.

[132] Kai Koch, Debora Clever, Katja Mombaur, and Dominik Endres. Learning movement primitives from optimal and dynamically feasible trajectories for humanoid walking. pages 866–873, 11 2015.

[133] John Koetsier. Tesla Bot Optimus: Everything We Know So Far, 2022. Accessed: 2024-10-22.

[134] Franziska Krebs and Tamim Asfour. A Bimanual Manipulation Taxonomy. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11031–11038, 2022.

[135] Franziska Krebs, Andre Meixner, Isabel Patzer, and Tamim Asfour. The KIT Bimanual Manipulation Dataset. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 499–506, 2021.

[136] J.J. Kuffner and S.M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001 vol.2, 2000.

[137] Scott Kuindersma, Robin Deits, Maurice Fallon, Andres Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pierre Marion, and Russ Tedrake. Optimization-Based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot. *Autonomous Robots*, 40(3):429–455, 2016.

[138] Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-Training for Robots: Offline RL Enables Learning New Tasks from a Handful of Trials. (arXiv:2210.05178), September 2023. arXiv:2210.05178 [cs].

[139] Marco Laghi, Michele Maimeri, Mathieu Marchand, Clara Leparoux, Manuel Catalano, Arash Ajoudani, and Antonio Bicchi. Shared-Autonomy Control for Intuitive Bimanual Tele-Manipulation. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, 2018.

[140] Riddhiman Laha, Marvin Becker, Jonathan Vorndamme, Juraj Vrabel, Luis F.C. Figueredo, Matthias A. Müller, and Sami Haddadin. Predictive Multi-Agent-Based Planning and Landing Controller for Reactive Dual-Arm Manipulation. *IEEE Transactions on Robotics*, 40:864–885, 2024.

[141] Riddhiman Laha, Luis F.C. Figueredo, Juraj Vrabel, Abdalla Swikir, and Sami Haddadin. Reactive Cooperative Manipulation based on Set Primitives and Circular Fields. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 6577–6584, Xi'an, China, May 2021. IEEE.

[142] Riddhiman Laha, Jonathan Vorndamme, Luis F.C. Figueredo, Zheng Qu, Abdalla Swikir, Christoph Jähne, and Sami Haddadin. Coordinated Motion Generation and Object Placement: A Reactive Planning and Landing Approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9401–9407, 2021.

[143] Alberto Leardini, Zimi Sawacha, Gabriele Paolini, Stefania Ingrosso, Roberto Nativo, and Maria Grazia Benedetti. A new anatomically based protocol for gait analysis in children. *Gait Pos*, 26:560–571, 2007.

[144] Hyungjik Lee, Ho Jin Choi, Jun Hyung Park, Jong Hyun Lee, and Seul Jung. Center of gravity based control of a humanoid balancing robot for boxing games : BalBot V. In *Conf Soc Inst Ctrl Eng*, pages 124–128, 2009.

[145] S. Lee. Dual redundant arm configuration optimization with task-oriented dual arm manipulability. *IEEE Transactions on Robotics and Automation*, 5(1):78–97, 1989.

[146] Konrad Leibrandt, Lyndon da Cruz, and Christos Bergeles. Designing Robots for Reachability and Dexterity: Continuum Surgical Robots as a Pretext Application. *IEEE Transactions on Robotics*, 39(4):2989–3007, August 2023.

[147] Teguh Santoso Lembono, Carlos Mastalli, Pierre Fernbach, Nicolas Mansard, and Sylvain Calinon. Learning How to Walk: Warm-starting Optimal Control Solver with Memory of Motion, 2020.

[148] Puttichai Lertkultanon and Quang-Cuong Pham. A Certified-Complete Bimanual Manipulation Planner. *IEEE Transactions on Automation Science and Engineering*, 15(3):1355–1368, 2018.

[149] Jeongsoo Lim, Hyoin Bae, Jaesung Oh, Inho Lee, Inwook Shim, Hyobin Jung, Hyun-Min Joe, Okkee Sim, Taejin Jung, Seunghak Shin, Kyungdon Joo, Mingeuk Kim, Kangkyu Lee, Yunsu Bok, Dong-Geol Choi, Buyoun Cho, Sungwoo Kim, Jungwoo Heo, Inhyeok Kim, and Jun-Ho Oh. *Robot system of DRC-HUBO+ and control strategy of team KAIST in DARPA robotics challenge finals*, pages 27–69. 04 2018.

[150] Hsiu-Chin Lin, Joshua Smith, Keyhan Kouhkiloui Babarahmati, Niels Dehio, and Michael Mistry. A Projected Inverse Dynamics Approach for Multi-Arm Cartesian Impedance Control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5421–5428, 2018.

[151] Jonathan Feng-Shun Lin, Vincent Bonnet, Adina M. Panchea, Nacim Ramdani, Gentiane Venture, and Dana Kulić. Human motion segmentation using cost weights recovered from inverse optimal control. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1107–1113, 2016.

[152] Jonathan Feng-Shun Lin, Pamela Carreno-Medrano, Mahsa Parsapour, Maram Sakr, and Dana Kulić. Objective learning from human demonstrations. *Annual Reviews in Control*, 51:111–129, 2021.

[153] Yuan Lin, John McPhee, and Nasser L. Azad. Comparison of Deep Reinforcement Learning and Model Predictive Control for Adaptive Cruise Control. *IEEE Transactions on Intelligent Vehicles*, 6(2):221–231, 2021.

[154] Chang Liu, Seungho Lee, Scott Varnhagen, and H. Eric Tseng. Path planning for autonomous vehicles using model predictive control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 174–179, 2017.

[155] Junjia Liu, Yiting Chen, Zhipeng Dong, Shixiong Wang, Sylvain Calinon, Miao Li, and Fei Chen. Robot Cooking With Stir-Fry: Bimanual Non-Prehensile Manipulation of Semi-Fluid Objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166, 2022.

[156] Junjia Liu, Hengyi Sim, Chenzui Li, and Fei Chen. BiRP: Learning Robot Generalized Bimanual Coordination using Relative Parameterization Method on Human Demonstration, 2023.

[157] Luyu Liu, Qianyuan Liu, Yong Song, Bao Pang, Xianfeng Yuan, and Qingyang Xu. A Collaborative Control Method of Dual-Arm Robots Based on Deep Reinforcement Learning. *Applied Sciences*, 11(4), 2021.

[158] Yiping Liu, Patrick M. Wensing, David E. Orin, and Yuan F. Zheng. Dynamic walking in a humanoid robot based on a 3D Actuated Dual-SLIP model. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5710–5717, 2015.

[159] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, planning, and Control*, chapter 5.4 Manipulability, pages 196–199. Cambridge University Press, Cambridge, United Kingdom, 2019.

[160] Ian R. Manchester. AMME4500: Guidance, Navigation, and Control. https://web.aeromech.usyd.edu.au/AMME4500/AMME4500notes.pdf, 2013.

[161] Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderrahmane Kheddar. A versatile Generalized Inverted Kinematics implementation for collaborative working humanoid robots: The Stack Of Tasks. In *2009 International Conference on Advanced Robotics*, pages 1–6, 2009.

[162] Mathworks. Model Predictive Control Toolbox. https://www.mathworks.com/products/model-predictive-control.html, 2021.

[163] Mathworks. Perform Co-Simulation between Simulink and Gazebo. https://www.mathworks.com/help/robotics/ug/perform-co-simulation-between-simulink-and-gazebo.html, 2021.

[164] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[165] Benjamin Michaud and Mickaël Begon. biorbd: A C++, Python and MATLAB library to analyze and simulate the human body biomechanics. *Journal of Open Source Software*, 6(57):2562, 2021.

[166] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion-An inverse optimal control approach. *Auton. Robots*, 28:369–383, 04 2010.

[167] Katja Mombaur, Heike Vallery, Yue Hu, Jonas Buchli, Pranav Bhounsule, Thiago Boaventura, Patrick M. Wensing, Shai Revzen, Aaron D. Ames, Ioannis Poulakakis, and Auke Ijspeert. Chapter 4 - Control of Motion and Compliance. In Maziar A. Sharbafi and André Seyfarth, editors, *Bioinspired Legged Locomotion*, pages 135–346. Butterworth-Heinemann, 2017.

[168] Federico L. Moro, Nikos G. Tsagarakis, and Darwin G. Caldwell. A human-like walking for the COmpliant huMANoid COMAN based on CoM trajectory reconstruction from kinematic Motion Primitives. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 364–370, 2011.

[169] J. B. Morrell and D. Field. Design of a closed loop controller for a two wheeled balancing transporter. In *IEEE-RSJ Conf Intel Robot Syst*, pages 4059–4064, 2007.

[170] Malte Mosbach, Kara Moraw, and Sven Behnke. Accelerating Interactive Human-like Manipulation Learning with GPU-based Simulation and High-quality Demonstrations. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, page 435–441, Ginowan, Japan, November 2022. IEEE.

[171] Albert Mukovskiy, Christian Vassallo, Maximilien Naveau, Olivier Stasse, Philippe Souéres, and Martin A. Giese. Adaptive synthesis of dynamically feasible full-body movements for the humanoid robot HRP-2 by flexible combination of learned dynamic movement primitives. *Robotics Auton. Syst.*, 91:270–283, 2017.

[172] Masaki Murooka, Iori Kumagai, Mitsuharu Morisawa, Fumio Kanehiro, and Abderrahmane Kheddar. Humanoid Loco-Manipulation Planning Based on Graph Search and Reachability Maps. *IEEE Robotics and Automation Letters*, 6(2):1840–1847, 2021.

[173] Richard M. Murray, S. Shankar Sastry, and Li Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., USA, 1st edition, 1994.

[174] F.A. Mussa-Ivaldi and S.A. Solla. Neural primitives for motion control. *IEEE Journal of Oceanic Engineering*, 29(3):640–650, 2004.

[175] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming Exploration in Reinforcement Learning with Demonstrations. (arXiv:1709.10089), February 2018. arXiv:1709.10089 [cs].

[176] Gabriele Nava, Daniele Pucci, Nuno Guedelha, Silvio Traversaro, Francesco Romano, Stefano Dafarra, and Francesco Nori. Modeling and Control of Humanoid Robots in Dynamic Environments: iCub Balancing on a Seesaw. 07 2017.

[177] Lars Omlor and Martin Giese. Anechoic Blind Source Separation Using Wigner Marginals. *Journal of Machine Learning Research*, 12:1111–1148, 03 2011.

[178] David Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35, 10 2013.

[179] Santos M. Orozco-Soto, Juan M. Ibarra-Zannatha, and Abderrahmane Kheddar. Robust Position Regulation of a Seesaw Actuated by a Humanoid. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 1315–1320, 2022.

[180] Christian Ott, Christoph Baumgärtner, Johannes Mayr, Matthias Fuchs, Robert Burger, Dongheui Lee, Oliver Eiberger, Alin Albu-Schäffer, Markus Grebenstein, and Gerd Hirzinger. Development of a biped robot with torque controlled joints. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 167–173, 2010.

[181] Christian Ott and Yoshihiko Nakamura. Admittance Control using a Base Force/Torque Sensor.*. In *SyRoCo*, 2009.

[182] Deepak Paramkusam. URDF to Equations of Motion. https://github.com/DeepakParamkusam/urdf2eom, 2021.

[183] Dmytro Pavlichenko, Diego Rodriguez, Max Schwarz, Christian Lenz, Arul Selvam Periyasamy, and Sven Behnke. Autonomous Dual-Arm Manipulation of Familiar Objects. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, 2018.

[184] Stylianos Piperakis, Maria Koskinopoulou, and Panos Trahanias. Nonlinear State Estimation for Humanoid Robot Walking. *IEEE Robotics and Automation Letters*, 3(4):3347–3354, 2018.

[185] Oliver Porges, Mathilde Connan, Bernd Henze, Andrea Gigli, Claudio Castellini, and Maximo A. Roa. A wearable, ultralight interface for bimanual teleoperation of a compliant, whole-body-controlled humanoid robot. 01 2019.

[186] Pascal Poupart. CS885: Advanced Topics in Artificial Intelligence - Lecture 3A. https://cs.uwaterloo.ca/~ppoupart/teaching/cs885-fall22/slides/cs885-lecture3a.pdf, 2022. Accessed: 2024-10-30.

[187] M. Powell. A View of Algorithms for Optimization Without Derivatives. *Mathematics TODAY*, 43, 01 2007.

[188] M. J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. 2009.

[189] J.E. Pratt and R. Tedrake. *Velocity-Based Stability Margins for Fast Bipedal Walking*, pages 299–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[190] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture Point: A Step toward Humanoid Push Recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, 2006.

[191] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture Point: A Step toward Humanoid Push Recovery. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, 2006.

[192] Jerry Pratt, Twan Koolen, Tomas De Boer, John Rebula, Sebastien Cotton, John Carff, Matthew Johnson, and Peter Neuhaus. Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower body humanoid. *The International Journal of Robotics Research*, 31:1117–1133, 08 2012.

[193] Jerry E. Pratt, Sylvain Bertrand, and Twan Koolen. *Stepping for Balance Maintenance Including Push-Recovery*, pages 1419–1466. Springer Netherlands, Dordrecht, 2019.

[194] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[195] Marc H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.

[196] Vidyasagar Rajendran, Pamela Carreno-Medrano, Wesley Fisher, and Dana Kulić. Human-Aware RRT-Connect: Motion Planning for Safe Human-Robot Collaboration. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 15–22, 2021.

[197] Vidyasagar Rajendran, Francisco Javier Andrade Chavez, and Katja Mombaur. Whole-body Stabilization of the REEM-C Humanoid on Actuated Balance Boards in Sagittal and Frontal Directions. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2023.

[198] Vidyasagar Rajendran, Jonathan Feng-Shun Lin, and Katja Mombaur. Towards Humanoids Using Personal Transporters: Learning to Ride a Segway from Humans. In *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 01–08, 2022.

[199] Vidyasagar Rajendran and Katja Mombaur. Humanoids Operating Mobility Devices Designed for Humans: Experiments on a Segway. In *Walking Robots into Real World*, pages 219–224, Cham, 2024. Springer Nature Switzerland.

[200] Vidyasagar Rajendran and Katja Mombaur. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 2024, Submitted.

[201] Vidyasagar Rajendran, Will Thibault, and Katja Mombaur. Motion Planning for Bimanual Manipulation of Humanoid Robots based on a Novel Manipulability-Stability Metric and Object-Specific Constraints. *IEEE Transactions on Robotics*, 2024, Under Review.

[202] N Ramuzat, G Buondonno, S Boria, and Olivier Stasse. Comparison of Position and Torque Whole Body Control Schemes on the Humanoid Robot TALOS. In *20th International Conference on Advanced Robotics (ICAR)*, Virtual event, Slovenia, December 2021.

[203] N. Ramuzat, O. Stasse, and S. Boria. Benchmarking Whole-Body Controllers on the TALOS Humanoid Robot. *Frontiers in Robotics and AI*, 9, 2022.

[204] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning, 2022.

[205] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: system overview and integration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2478–2483 vol.3, 2002.

[206] Ajay Suresha Sathya, Wilm Decre, Goele Pipeleers, and Jan Swevers. A Simple Formulation for Fast Prioritized Optimal Control of Robots using Weighted Exact Penalty Functions. In *2022 International Conference on Robotics and Automation (ICRA)*, page 5262–5269, May 2022.

[207] Stefan Schaal. Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics. 2006.

[208] Robert Schuller, George Mesesan, Johannes Englsberger, Jinoh Lee, and Christian Ott. Online Centroidal Angular Momentum Reference Generation and Motion Optimization for Humanoid Push Recovery. *IEEE Robotics and Automation Letters*, 6(3):5689–5696, 2021.

[209] Michael Ulbrich Sebastian Albrecht, Marion Leibold. A bilevel optimization approach to obtain optimal cost functions for human arm movements. *Numerical Algebra, Control Optimization*, 2(1):105–127, 2012.

[210] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition, 2021.

[211] João Silvério, Sylvain Calinon, Leonel Rozo, and Darwin G. Caldwell. Bimanual Skill Learning with Pose and Joint Space Constraints. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 153–159, 2018.

[212] Rohan P. singh, Zhaoming Xie, Pierre Gergondet, and Fumio Kanehiro. Learning Bipedal Walking for Humanoids With Current Feedback. *IEEE Access*, 11:82013–82023, 2023.

[213] Rohan Pratap Singh, Mehdi Benallegue, Mitsuharu Morisawa, Rafael Cisneros, and Fumio Kanehiro. Learning Bipedal Walking On Planned Footsteps For Humanoid Robots, 2022.

[214] R. Smits. KDL: Kinematics and Dynamics Library. http://www.orocos.org/kdl.

[215] Matthew Spenko, Stephen Buerger, and Karl Iagnemma. *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*. 01 2018.

[216] Norikazu Sugimoto and Jun Morimoto. Trajectory-model-based reinforcement learning: Application to bimanual humanoid motor learning with a closed-chain constraint. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 429–434, 2013.

[217] Ashok M. Sundaram, Oliver Porges, and Máximo A. Roa. Planning realistic interactions for bimanual grasping and manipulation. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 987–994, 2016.

[218] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 2nd edition, 2018.

[219] Toru Takenaka, Takashi Matsumoto, and Takahide Yoshiike. Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1091, 2009.

[220] JA Tenreiro Machado and Manuel Silva. An overview of legged robots. In *Int Symp Math Meth Eng*, pages 1–40, 2006.

[221] Tesla. Optimus Navigating Around — Tesla. https://www.youtube.com/watch?v=DrNcXgoFv20, 2024. [Online; accessed 28-October-2024].

[222] William Thibault, William Melek, and Katja Mombaur. Learning Velocity-based Humanoid Locomotion: Massively Parallel Learning with Brax and MJX, 2024.

[223] William Thibault, Vidyasagar Rajendran, William Melek, and Katja Mombaur. Learning Skateboarding for Humanoid Robots through Massively Parallel Reinforcement Learning, 2024.

[224] William Thibault, Vidyasagar Rajendran, and Katja Mombaur. Bimanual Manipulation Workspace Analysis of Humanoid Robots with Object Specific Coupling Constraints. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 141–148, 2022.

[225] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, 2017.

[226] D. Toishi and E. Konaka. Point stabilization of two-wheeled vehicle based on machine learning. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 175–180, 2012.

[227] Nikolaus Vahrenkamp and Tamim Asfour. Representing the robot's workspace through constrained manipulability analysis. *Autonomous Robots*, 38:1–14, 01 2014.

[228] Nikolaus Vahrenkamp, Tamim Asfour, and Rudiger Dillmann. Simultaneous Grasp and Motion Planning: Humanoid Robot ARMAR-III. *IEEE Robotics & Automation Magazine*, 19(2):43–57, 2012.

[229] Nikolaus Vahrenkamp, Tamim Asfour, and Rüdiger Dillmann. Robot placement based on reachability inversion. In *2013 IEEE International Conference on Robotics and Automation*, pages 1970–1975, 2013.

[230] Nikolaus Vahrenkamp, Tamim Asfour, Giorgio Metta, Giulio Sandini, and Rüdiger Dillmann. Manipulability analysis. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 568–573, 2012.

[231] Nikolaus Vahrenkamp, Markus Przybylski, Tamim Asfour, and Rüdiger Dillmann. Bimanual grasp planning. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 493–499, 2011.

[232] Jean Chagas Vaz and Paul Oh. Material Handling by Humanoid Robot While Pushing Carts Using a Walking Pattern Based on Capture Point. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9796–9801, 2020.

[233] Dmytro Velychko, Dominik Endres, Nick Taubert, and Martin Giese. Coupling Gaussian Process Dynamical Models with Product-of-Experts Kernels. pages 603–610, 09 2014.

[234] Nahuel A. Villa, Pierre Fernbach, Maximilien Naveau, Guilhem Saurel, Ewen Dantec, Nicolas Mansard, and Olivier Stasse. Torque Controlled Locomotion of a Biped Robot with Link Flexibility. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 9–16, 2022.

[235] L. C. Visser, S. Stramigioli, and R. Carloni. Robust bipedal walking with variable leg stiffness. In *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1626–1631, 2012.

[236] Jiangping Wang, Shirong Liu, Botao Zhang, and Changbin Yu. Inverse kinematics-based motion planning for dual-arm robot with orientation constraints. *International Journal of Advanced Robotic Systems*, 16(2):1729881419836858, 2019.

[237] Ruoshi Wen, Quentin Rouxel, Michael Mistry, Zhibin Li, and Carlo Tiseo. Collaborative Bimanual Manipulation Using Optimal Motion Adaptation and Interaction Control: Retargeting Human Commands to Feasible Robot Control References. *IEEE Robotics & Automation Magazine*, pages 2–14, 2023.

[238] Patrick Wensing and David Orin. *Control of Humanoid Hopping Based on a SLIP Model*, pages 265–274. 01 2013.

[239] Patrick M. Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-Based Control for Dynamic Legged Robots. *IEEE Transactions on Robotics*, 40:43–63, 2024.

[240] Pierre-Brice Wieber. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. In *IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006.

[241] David Winter. Biomechanics and Motor Control of Human Movement, Fourth Edition. 09 2009.

[242] Fan Xie, Alexander Chowdhury, M. Clara De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep Imitation Learning for Bimanual Robotic Manipulation. In *Advances in Neural Information Processing Systems*, volume 33, page 2327–2337. Curran Associates, Inc., 2020.

[243] Songyan Xin, Yangwei You, Chengxu Zhou, Cheng Fang, and Nikos Tsagarakis. A torque-controlled humanoid robot riding on a two-wheeled mobile platform. In *IEEE-RSJ Conf Intel Robot Syst*, pages 1435–1442, 2017.

[244] Katsu Yamane and Akihiko Murai. *A Comparative Study Between Humans and Humanoid Robots*, pages 1–20. Springer Netherlands, Dordrecht, 2016.

[245] Chenguang Yang, Yiming Jiang, Zhijun Li, Wei He, and Chun-Yi Su. Neural Control of Bimanual Robots With Guaranteed Global Stability and Motion Precision. *IEEE Transactions on Industrial Informatics*, 13(3):1162–1171, 2017.

[246] Shunpeng Yang, Hua Chen, Zhen Fu, and Wei Zhang. Force-feedback based Whole-body Stabilizer for Position-Controlled Humanoid Robots. *CoRR*, abs/2108.06652, 2021.

[247] Yiming Yang, Vladimir Ivan, Zhibin Li, Maurice Fallon, and Sethu Vijayakumar. iDRM: Humanoid motion planning with realtime end-pose selection in complex environments. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 271–278, 2016.

[248] Yiming Yang, Vladimir Ivan, Wolfgang Merkt, and Sethu Vijayakumar. Scaling sampling-based motion planning to humanoid robots. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1448–1454, 2016.

[249] Yiming Yang, Wolfgang Merkt, Henrique Ferrolho, Vladimir Ivan, and Sethu Vijayakumar. Efficient Humanoid Motion Planning on Uneven Terrain Using Paired Forward-Inverse Dynamic Reachability Maps. *IEEE Robotics and Automation Letters*, 2(4):2279–2286, 2017.

[250] Santheep Yesudasu, Wafae Sebbata, Jean-François Brethé, and Patrick Bonnin. Depalletisation humanoid torso: Real-time cardboard package detection based on deep learning and pose estimation algorithm. In *2023 27th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 228–233, 2023.

[251] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 1004–1009, 1985.

[252] Ming Yue, Cong An, and Jian-Zhong Sun. An Efficient Model Predictive Control for Trajectory Tracking of Wheeled Inverted Pendulum Vehicles with Various Physical Constraints. *International Journal of Control, Automation and Systems*, 16, 01 2018.

[253] SooYong Yun, WooSang Lee, JongHyeok Park, and Kwan-Woong Gwak. Development of a Virtual Environment Co-Simulation Platform for Evaluating Dynamic Characteristics of Self-Balancing Personal Mobility. *IEEE Trans Veh Tech*, 70:2969–2978, 2021.

[254] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 3229–3236, October 2007.

[255] Stanislaw Zak. An Introduction to Model Predictive Control, 2017.

[256] G. Zambella, G. Lentini, M. Garabini, G. Grioli, M. G. Catalano, A. Palleschi, L. Pallottino, A. Bicchi, A. Settimi, and D. Caporale. Dynamic Whole-Body Control of Unstable Wheeled Humanoid Robots. *IEEE Rob Autom*, 4:3489–3496, 2019.

[257] Haitao Zhou, Xu Li, Haibo Feng, Jiachen Li, Songyuan Zhang, and Yili Fu. Model Decoupling and Control of the Wheeled Humanoid Robot Moving in Sagittal Plane. In *IEEE-RAS Conf Human Rob*, pages 1–6, 2019.

[258] P. Zometa, M. Kögel, T. Faulwasser, and R. Findeisen. Implementation aspects of model predictive control for embedded systems. In *2012 American Control Conference (ACC)*, pages 1205–1210, 2012.

Appendices

Appendix A

Linear MPC for Segway Reference Tracking

Foreword

This appendix summarizes the development of an MPC controller (for the course SYDE 655: Optimal Control) for the Segway model used throughout this work. In the end, this was not used for simulations due to computational costs but the controller performed as well as the classical PID control method used in this thesis.

A.1 Introduction and Literature Survey

Wheeled robots have become very popular in academia, research and industry. They have a wide range of applications from personal transportation, last mile delivery, rescue operations and much more. They are also generally more energy efficient than legged robots and have simpler dynamics.

Within the class of wheeled robots, two-wheeled robots have gained popularity in research due to the unique stabilization challenges they present. They are also popular in general society, despite their complex dynamics, because they are small, light, and highly maneuverable.

Segways are famous robots that take on this two-wheeled configuration. They are usually modeled as two-wheeled inverted pendulums [92], [44], [169].

In this work, we adapt pre-existing two-wheeled inverted pendulum models [92], [63] and apply linear MPC to stabilize them in their upright position and track reference trajectories. Prior work has mainly focused on using PID control, pole placement or LQR (Linear Quadratic Regulator) control techniques to stabilize the system and track reference [92], [127]. This may be due to the paper released by Segway engineers giving a brief overview of their two-wheeled balancing transporter where they explain the use of state-feedback along with a multitude of sensors and processors [169]. There have been some works that investigate

the use of Linear MPC (Model Predictive Control) for Segways and two-wheeled inverted pendulums, for example, [54], [26], [258], [252]. These works generally linearize the Segway dynamics about an operating point to track various reference trajectories. In comparison with this work, most of them use reduced models of the Segway and don't consider turning or control performance in the presence of external disturbances. In this work, stabilization about the upright position is achieved similar to these works via linearization, but no restrictions are made with regards to motion around the vertical axis (turning axis).

In [91], the authors propose a theoretical unification of Nonlinear Model Predictive Control (NMPC) with Control Lyapunov Functions (CLFs). This framework achieves optimal control performance while ensuring stability guarantees. They evaluate the combined NMPC and CLF formulation against baseline CLF and NMPC showing that parameters are easier to tune and greater control performance is achieved as compared with NMPC.

In [226] the authors propose an MPC method that can achieve a long prediction horizon with short computation time. This is achieved by optimizing the input sequence as a graph search problem by restricting the inputs to discrete values. Next a learning approach is used to learn the optimized control result and this is shown to allow longer MPC horizons (leading to better simulated and experimental performance).

In [54], an MPC is developed to control a two-wheeled inverted pendulum with a flexible beam. Several models of varying complexity (including nonlinear versions) are developed and stabilized with MPC. It was found that the MPC worked well when parameters were known completely but performed sub-optimally in the presence of model mismatches.

Despite some of the shortcomings of MPC (particularly the lack of robustness in the presence of model uncertainty), it is still a viable control approach for stabilizing a Segway. In this work, we present the control of a two-wheeled inverted pendulum (herein referred to as "Segway") with Linear MPC and utilize a model adapted from [92] and [63]. We first decouple the Segway into two independent subsystems, one that models the stability around the lateral axis (pitch) and displacement, and the other that acts on the vertical axis (yaw). Then we develop two MPCs that control each subsystem. Each MPC outputs a torque around its respective axis which is fed into a simple linear decoupling matrix that translates these signals into left and right wheel torques. This scheme is similar to the one presented in [92]. After simulation with a nonlinear plant in Simulink, the controllers are tested with a high-fidelity Segway model developed in URDF format for use in Gazebo. The model is based on a previously designed CAD model [12]. Results from the Gazebo simulation do not match exactly with the Simulink simulation, most likely due to a mismatch in the dynamic models used in each scenario. However, when tested in Simulink, the dual-MPC controller scheme performs well at tracking reference trajectories and we also show that the performance is better than an optimal LQR-based solution (as was used in [63]).

The rest of this report is organized as follows: Section A.2 gives some background on linear MPC, Section A.3 details the Segway model used in this work, Section A.4 formulates the optimal control problem for the Segway, Section A.5 describes the various simulations run to test the performance of the MPC and Section A.6 summarizes the work and provides concluding remarks.



Figure A.1: High-fidelity Segway model created in URDF format, simulated in ROS/Gazebo.

A.2 Background

In this section some background is provided on MPC control and it's advantages when coupled with underactuated systems such as the Segway.

Prior to MPC, LQR (and LQG) were (and still are) quite popular due to the simplicity of implementation. However, one major drawback of these control schemes is that they do not consider constraints on system input or outputs. To some extent, constraints can be included in LQR and LQG formulations through the retuning of cost functions, but this is often only effective when the nominal operation of the plant is far from its constraint boundaries [160]. Modern MPC originated in the process industry to satisfy these types of issues prevalent with LQR and LQG control.

A.2.1 General MPC Description

MPC, at its core, is a type of control strategy that uses the system model, an objective function and an optimization algorithm to determine the optimal control action for the system.

The control structure can be summarized as [54], [39]:

1. Future outputs of the model are determined for a future horizon $y(t+k)$, where $k = 1, 2, ..., N_p$, where N_p is the prediction horizon. We use the current model outputs at time t and a set of control inputs $u(t+l)$, where $l = 1, 2, ..., M$, where M is the control horizon.

2. The next control inputs $u(t+1+l)$ are calculated by optimizing a user-determined criterion to keep the model outputs $y(t+k)$ as close to a reference trajectory $v(t+k)$ as possible.

3. Generally, the first control input that was calculated in step 2 is applied to the system, and the rest are discarded.

4. Step 1 is repeated with $y(t+1+k)$ and $u(t+1+k)$ as long as the system continues to run.

A diagram of the above process is below:

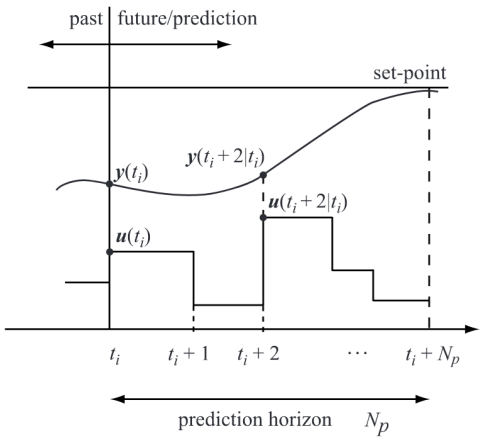


Figure A.2: Controller action using MPC [255]. $u(t+k|t)$ is the predicted control action at $t+k$ given $u(t)$. $y(t+k|t)$ is the predicted output at $t+k$ given $y(t)$.

MPCs have a number of advantages including their versatility to be used with many types of models and objective functions, their ability to handle constraints and their ability to run for long periods of time without intervention. Some disadvantages of MPCs include their lack of robustness, and lack of proofs related to stability guarantees. One way this is combated is by using a terminal cost to approximate the infinite horizon control problem. However this is still not a guarantee and is only really an estimate. In terms of robustness, one qualm with linear MPCs is that they perform poorly in the presence of model uncertainty.

A.2.2 Linear Constrained MPC

While there are different formulations of MPC, be it unconstrained or constrained, or for different objective functions, in this work we use linear constrained MPC and as such describe it in this section. The formulation described herein is based on [28].

First, we use the following linear discrete-time prediction model:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{A.1}$$

where $x \in R^n$, $u \in R^m$ and $y \in R^p$. We would like to enforce the following constraints on the inputs and outputs of the system:

$$\begin{aligned} u_{min} \leq u(t) \leq u_{max},\ u_{min}, u_{max} \in R^m \\ y_{min} \leq y(t) \leq y_{max},\ y_{min}, y_{max} \in R^p \end{aligned} \tag{A.2}$$

The constrained optimal control problem, given as a quadratic performance index, is the following:

$$\begin{aligned} \min_z x_N^\intercal P x_N + \sum_{k=0}^{N-1} x_k^\intercal Q x_k + u_k^\intercal R u_k \\ s.t.\ u_{min} \leq u(t) \leq u_{max}, k = 0, ..., N-1 \\ y_{min} \leq y(t) \leq y_{max}, k = 1, ..., N \end{aligned} \tag{A.3}$$

where $R^\intercal \succ 0$, $Q^\intercal \succeq 0$, $P^\intercal \succeq 0$ and $z = [u_0 u_1 ... u_{N-1}]^\intercal$. P is the solution to the discrete time algebraic Riccati equation.

The convex quadratic optimization problem can be written as:

$$\begin{aligned} V(x_0) = \frac{1}{2} x_0^\intercal Y x_0 + \min_z \frac{1}{2} z^\intercal H z + x_0^\intercal F^\intercal z \\ s.t.\ Gz \leq W + S x_0 \end{aligned} \tag{A.4}$$

In Equation A.4, $H^\intercal \succ 0$ and H, F, Y, G, W, S depend on the weights Q, R, P, upper and lower bounds, U_{min}, u_{max}, y_{min}, y_{max} and model matrices A, B, C.

A.2.3 Linear MPC for Tracking

With the formulation for linear constrained MPC presented in Subsection A.2.2, we can easily extend the objective function for tracking reference signals:

$$\begin{aligned} \min_z e_N^\intercal P e_N + \sum_{k=0}^{N-1} e_k^\intercal Q e_k + \Delta u_k^\intercal R \Delta u_k \\ s.t.\ u_{min} \leq u(t) \leq u_{max}, k = 0, ..., N-1 \\ y_{min} \leq y(t) \leq y_{max}, k = 1, ..., N \end{aligned} \tag{A.5}$$

where $R^\intercal \succ 0$, $Q^\intercal \succeq 0$, $P^\intercal \succeq 0$, $z = [\Delta u_0 \Delta u_1 ... \Delta u_{N-1}]^\intercal$ and $e_k = x_k - x_{r_k}$.

The details on the exact convex quadratic programming formulation are given in [28]. This is the exact objective function that will be used in the MPC formulation for Segway reference tracking.

A.3 Segway Model

The Segway model used for all control is developed in this section. The system is approximated as a two-wheeled inverted pendulum as shown in Figure A.3.

Since this work is not focused on system modelling, we present the equations of motion but refer the reader to two texts from which the model was derived [92], [63]. The various

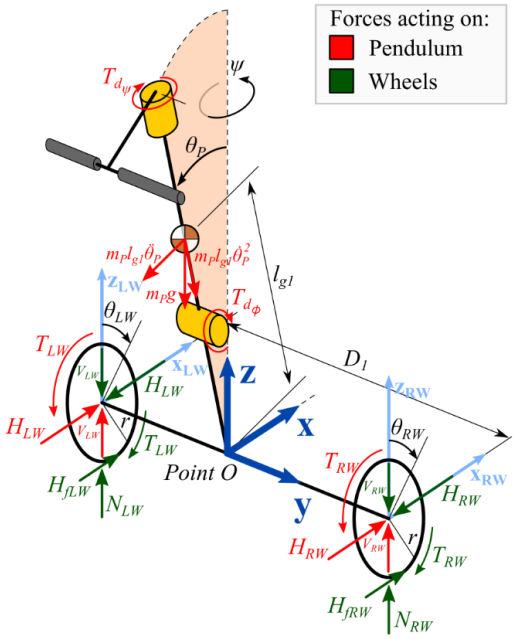


Figure A.3: Segway diagram of forces and moments acting on the body, pendulum, and wheels [63]. As a note, a few parameters and symbols are different than the ones presented in this section.

parameters used in the formulation are summarized in Table A.1. The parameter descriptions and values used in this work are given. The system is linearized about the upright position, that is, $\theta = 0, \dot{\theta} = 0$:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{m_p^2 g l_{gl}^2}{\alpha}\theta + \frac{J_\theta - m_p l_{gl} r}{r\alpha}(T_L + T_W) + \frac{m_p l_{gl}}{\alpha} T_{d_\theta} \\ \dot{\theta} \\ \frac{m_p l_{gl}\beta}{\alpha}\theta + \frac{m_p l_{gl} - r\beta}{r\alpha}(T_L + T_W) + \frac{\beta}{\alpha} T_{d_\theta} \\ \dot{\psi} \\ \frac{D_1}{2rJ\psi}(T_L - T_W) + \frac{1}{J_{psi}} T_{d_\psi} \end{bmatrix}$$

where

$$J_\psi = \frac{D_1^2}{2}(m_W + J_W/r^2) + J_{P_z} \tag{A.6}$$

$$J_\theta = J_{P_y} + m_P l_{gl}^2 \tag{A.7}$$

$$\beta = 2m_w + 2J_w/r^2 + m_p \tag{A.8}$$

$$\alpha = J_\theta \beta - m_p^2 l_{gl}^2 \tag{A.9}$$

From this, we take our state as $[x\ \dot{x}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^\intercal$ and rewrite the equations of motion in state-space form:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & B_{23} & 0 \\ 0 & 0 & 0 & 0 \\ B_{41} & B_{42} & B_{43} & 0 \\ 0 & 0 & 0 & 0 \\ B_{61} & B_{62} & 0 & B_{64} \end{bmatrix} \begin{bmatrix} T_L \\ T_R \\ T_{d_\theta} \\ T_{d_\psi} \end{bmatrix}$$

where

$$A_{23} = \frac{m_p^2 g l_{gl}^2}{\alpha}\ A_{43} = \frac{m_p g l_{gl} \beta}{\alpha} \tag{A.10}$$

$$B_{21} = \frac{J_\theta - m_p l_{gl} r}{r\alpha}\ B_{22} = \frac{J_\theta - m_p l_{gl} r}{r\alpha}\ B_{23} = \frac{m_p l_{gl}}{\alpha} \tag{A.11}$$

$$B_{41} = \frac{m_p l_{gl} - r\beta}{r\alpha}\ B_{42} = \frac{m_p l_{gl} - r\beta}{r\alpha}\ B_{43} = \frac{\beta}{\alpha} \tag{A.12}$$

$$B_{61} = \frac{D_1}{2rJ_\psi}\ B_{62} = -\frac{D_1}{2rJ_\psi}\ B64 = \frac{1}{J_\psi} \tag{A.13}$$

In this work we assume all state variables are measureable, thus the C output matrix in the state space formulation is the identity matrix, I_{6x6}. The feed-forward matrix D is assumed to be the zero matrix, 0_{6x4}.

For the Segway system described in state space form, there are 4 inputs, 2 of which are disturbances that cannot be used to directly control the system. The torques T_L, T_R are available for control. Following the formulation in [92], we decompose the state space into two independent subsystems. One subsystem models the dynamics around the vertical z-axis and the other on the lateral y-axis. This essentially translates into producing an output torque T_ψ about the vertical axis and T_θ about the lateral axis. Each of these torques only influence the dynamics about their respective axes. In order to apply these torques to the Segway system, we need a decoupling unit that transforms T_ψ, T_θ to T_L, T_R. In this work, we take the decoupling controller as:

$$\begin{bmatrix} T_\theta \\ T_\psi \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix} \tag{A.14}$$

With this new decoupling transform, we can write the state space as two separate subsystems, one as the rotational dynamics about the y-axis,

Parameter	*Description*	*Value*	*Unit*
m_p	pendulum mass	37	kg
m_w	wheel mass	4.6	kg
D_1	distance between left and right wheel	0.55	m
l_{gl}	distance from base to CoM of pendulum	0.475	m
r	wheel radius	0.27	m
g	gravitational acceleration	9.81	$\mathrm{m/s^2}$
x	displacement along x-axis	-	m
θ	linearized Segway tilt angle	-	rad
ψ	heading angle of Segway	-	rad
T_{d_θ}	disturbance torque on pitch axis	-	Nm
T_{d_ψ}	disturbance torque on yaw axis	-	Nm
T_L, T_R	Left and right wheel actuator torques	-	Nm
J_w	wheel inertia	0.1643	$\mathrm{kg\,m^2}$
J_{p_y}	pendulum and chassis inertia along y-axis	0.779	$\mathrm{kg\,m^2}$
J_{p_z}	pendulum and chassis inertia along z-axis	3.1757	$\mathrm{kg\,m^2}$

Table A.1: The parameters for the Segway system.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{21} & B_{23} & 0 \\ 0 & 0 & 0 \\ B_{41} & B_{43} & 0 \end{bmatrix} \begin{bmatrix} T_\theta \\ T_{d_\theta} \\ T_{d_\psi} \end{bmatrix} \tag{A.15}$$

and another for the rotational dynamics about the z-axis,

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B_{61} & 0 & B_{64} \end{bmatrix} \begin{bmatrix} T_\psi \\ T_{d_\theta} \\ T_{d_\psi} \end{bmatrix} \tag{A.16}$$

With the state space decoupled and all dynamic parameters defined, we are ready to formulate the control system to stabilize and track reference trajectories with the Segway. In this work we consider reference trajectories for velocity, $\dot{x}$ and angular velocity $\dot{\psi}$.

A.4 Model Predictive Control Formulation for Segway System

For this specific decoupled system, the problem is formulated using two MPCs, one controlling the rotational dynamics about the y-axis, Equation A.15, and another controlling the rotational dynamics about the z-axis, Equation A.16. This scheme is shown in the Simulink block diagram in Figure A.5. MATLAB's MPC Toolbox [162] was utilized for both of the MPCs. The MPC parameters for each subsystem are given in Table A.2 and Table A.3.

For the Segway balance and position MPC, the prediction horizon was tuned to minimize any overshoot to velocity step references. This was done to ensure smooth operation of the Segway, especially when coming to a complete stop. The prediction horizon was tuned first after setting the sampling time of the controller and then held constant afterwards. In the end, this value for the prediction horizon was satisfactory as it stabilized the system and seemed to anticipate constraint violations early enough to mitigate them. The control horizon was kept small so as to try and keep the plant internally stable.

For the Segway yaw MPC, the prediction horizon was also tuned to minimize any overshoot to angular velocity step references. In the end, a larger prediction horizon, than for the Segway balance and position MPC, was used so that the system could react and stabilize quickly.

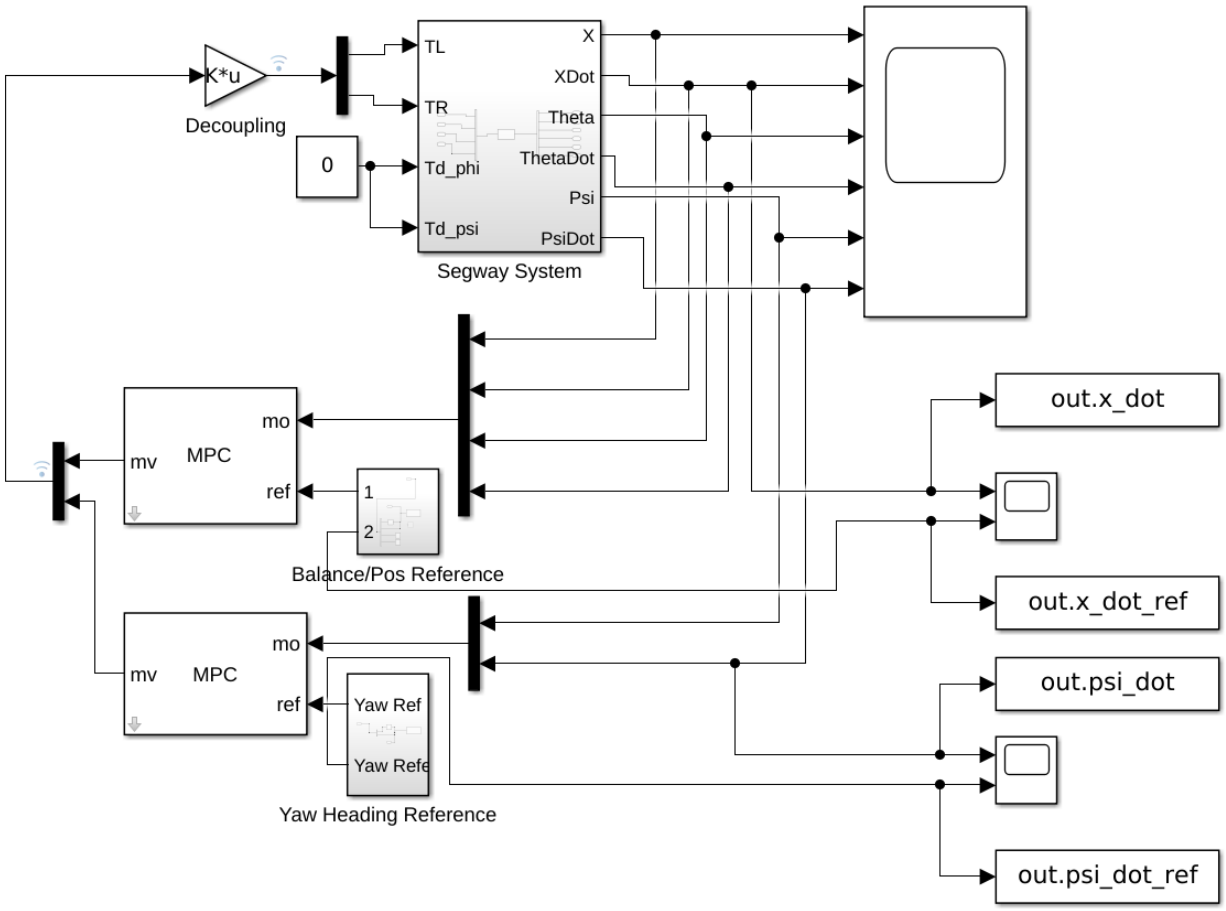


Figure A.4: Decoupled MPC control scheme to track reference trajectories with the Segway system.

The relevant input and output constraints are given in Equation A.17. As a note, the limits on T_L and T_R for a Segway of this size is usually around 100 Nm. As such, the torque limits for T_θ and T_ψ were set to ± 50 Nm so that the maximum torque produced by the decoupling matrix for one actuator of the Segway would be 100 Nm (see Equation A.14 for the T_L and T_R calculations). The maximum speed of this Segway model is 20 km/h, and as such, the constraints on $\dot{x}$ were set to ± 5 m/s.

Parameter	*Value*
Prediction Horizon	10
Control Horizon	2
Sampling Rate	0.1 s

Table A.2: The parameters for the Segway balance and position MPC, controlling the system described in Equation A.15.

Parameter	*Value*
Prediction Horizon	20
Control Horizon	4
Sampling Rate	0.1 s

Table A.3: The parameters for the Segway yaw MPC, controlling the system described in Equation A.16.

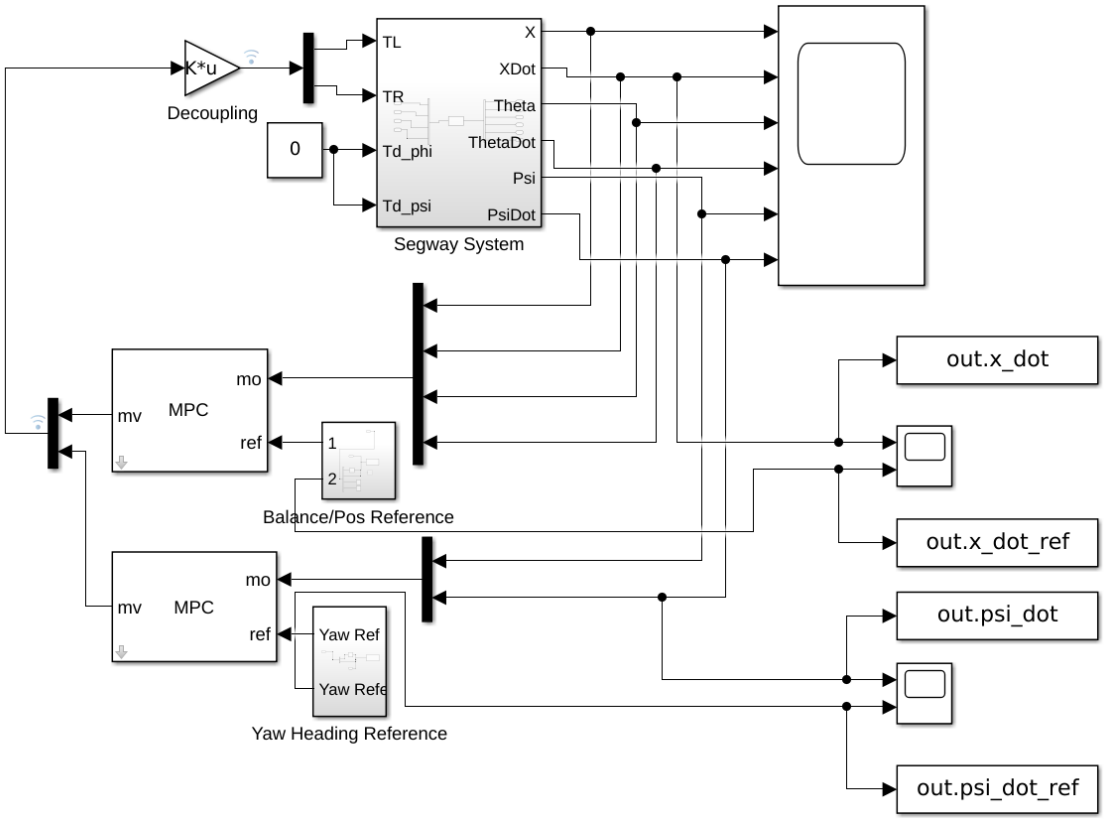


Figure A.5: Decoupled MPC control scheme to track reference trajectories with the Segway system.

$$\begin{aligned} -50 \text{ Nm} \leq T_\theta \leq 50 \text{ Nm}, \\ -50 \text{ Nm} \leq T_\psi \leq 50 \text{ Nm}, \\ -5 \text{ m/s} \leq \dot{x} \leq 5 \text{ m/s}, \\ -\pi/12 \text{ rad} \leq \theta \leq \pi/12 \text{ rad}. \end{aligned} \tag{A.17}$$

Since we have constraints on θ to keep it in a linear region and all system constraints are linear, we can loosely say that the specific problems solved by each MPC of the Segway system are convex.

A.4.1 Choosing the Weights for Q and R

Choosing the weights for the Q and R matrices is important to achieve the desired system performance. The optimality of the solution is relative to the cost function and as such, choosing the weights carefully for the specific problem is important. A starting point for selecting the Q and R matrices is given by Bryson's rule [37]: for each state and control signal, one decides the largest value allowable under normal operation, denoted $\bar{x_1}, \bar{x_2}, ..., \bar{x_n}$, and $\bar{u_1}, \bar{u_2}, ..., \bar{u_m}$. Then the Q and R matrices are given as:

$$Q = \begin{bmatrix} \frac{1}{\bar{x_1}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\bar{x_2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\bar{x_n}^2} \end{bmatrix}, \quad R = \begin{bmatrix} \frac{1}{\bar{u_1}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\bar{u_2}^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\bar{u_n}^2} \end{bmatrix}.$$

This essentially normalizes the individual states and control inputs by the maximum allowable value for each. Bryson's rule is popularly used with LQR and LQG control schemes and in this work is used purely as a starting point for the MPC weight matrices. It turns out, in the end, some manual tuning for the given control problem is usually always required.

For the Segway problem, Bryson's rule was used as a starting point before further tuning the weight matrices. In the end, R was set to 1 and for the Q matrix, x was set to 10000, θ was set to 1000 and ψ was set to 1000. The rest of the states were set to 1. This ensured that the Segway maintained its balance and did not displace too much when perturbed with disturbance forces.

A.4.2 Gazebo/Simulink Co-Simulation

For the high-fidelity Gazebo simulation introduced in this work, Gazebo/Simulink co-simulation was used [163]. This essentially synchronizes the Gazebo and Simulink clocks allowing messages to be passed to and from Gazebo, from Simulink. The Simulink model developed for the co-simulation is shown in Figure A.6 and attached with this submission.

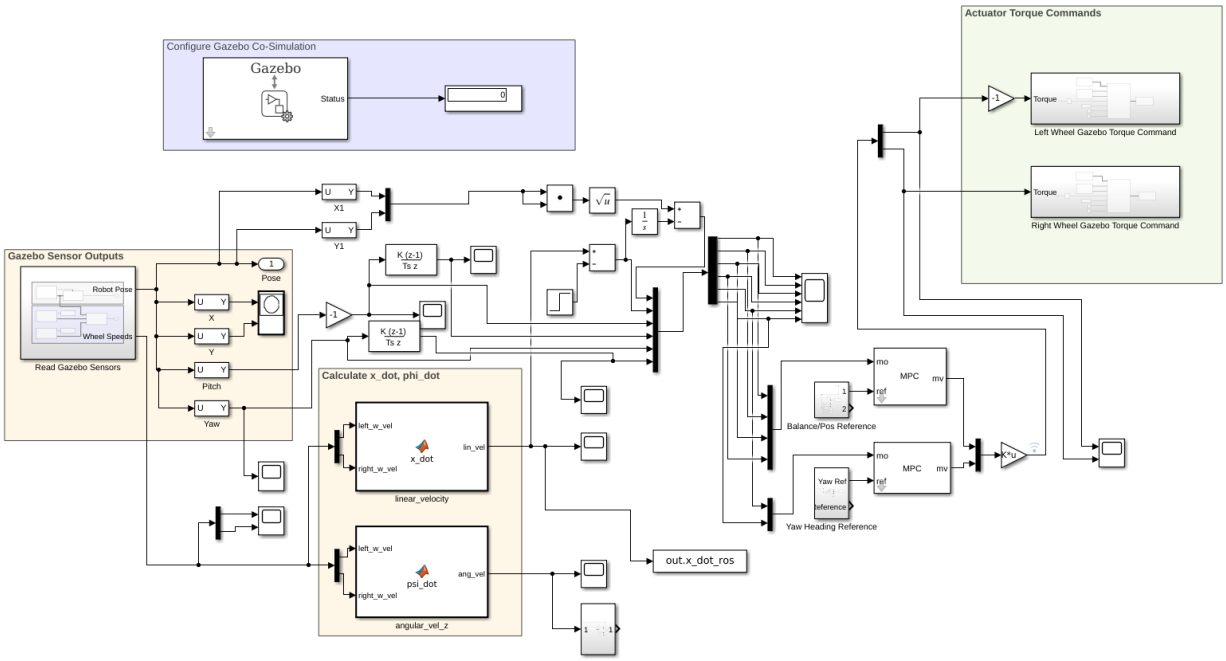


Figure A.6: Decoupled MPC control scheme to track reference trajectories with the Segway system using Gazebo/Simulink co-simulation.

A.5 Simulation Experiments

In this section we outline the various simulation results from the tests done using the decoupled MPC control structure. First, we outline a test done with velocity and angular velocity step references using the linear Segway model in Simulink. Next, we compare the decoupled MPC formulation against a decoupled LQR scheme using disturbance torques applied at the handle of the Segway. Finally, we discuss a test done in the Gazebo simulator for a velocity and angular velocity step reference.

The first test that was performed is shown in Figure A.7, where a step reference of 1 m/s was applied to $\dot{x}$ and another step of 0.2 rad/s was applied to $\dot{\psi}$. From the graph we can see that the Segway system is able to quickly track the reference trajectories. For both $\dot{x}$ and $\dot{\psi}$, the Segway stabilizes at steady state in around 2 s. Figure A.8 shows that the MPC also drives θ to 0 quickly.

Figure A.9 shows the Segway's response to a 5 Nm ramp disturbance at the handlebar. This same test was done in [63] with LQR (linear quadratic regulator) control. Here, the results from that work were reproduced but the Q and R matrices used were the same ones used for the MPC control scheme. As such, this is a fair test to see whether MPC performs better than LQR for the Segway system. In response to the disturbance torque profile, the linear displacement of the Segway is around 0.4 m when using LQR and the system is stabilized quickly. However, when we use MPC, we can see the performance is vastly improved. The Segway only displaces 0.1 m. All the states are driven to 0 much quicker than LQR. The graph makes it look like the Segway's response is a bit chaotic but this is only because the values are very small (see the y-axis of b) and c) for the major difference

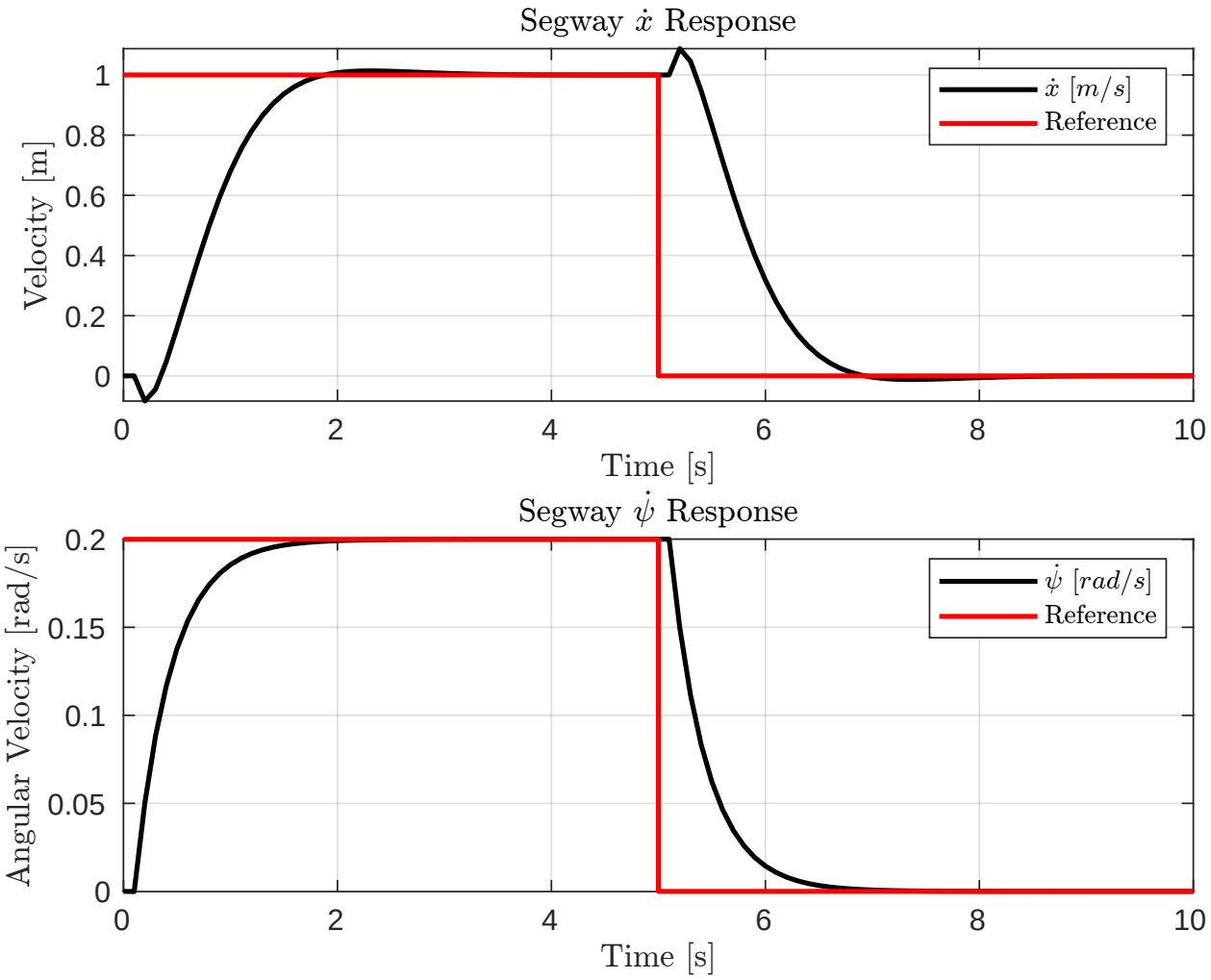


Figure A.7: Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs. The model used here is the linearized equations of motion, and the simulation was done in Simulink as shown in Figure A.5.

between LQR and MPC). This shows that a well-formed MPC as compared to LQR (with the same weighting matrices) is better at rejecting disturbances. This is a major advantage and reason to use MPC. In fact, if even better disturbance rejection was required, the sample time could be reduced at the expense of increased computational complexity.

A final test was run using the Simulink controllers developed in Figure A.5 and using them directly in a Gazebo/Simulink co-simulation. This essentially emulates simulating the controllers on a high-fidelity (and non-linear) plant model. However, the results here were a bit different than the previous tests. The Segway yaw rate response ($\dot{\psi}$) was as designed and was able to track the reference signal easily. However, the $\dot{x}$ response was quite slow taking almost 20s to settle at steady state. This is likely due to a mismatch in parameters between the previous Simulink model and the Gazebo model. The Segway in the Gazebo model, pictured in Figure A.1, was developed by the author from a previously available CAD model. Inertias were calculated using specific material properties in SolidWorks before being

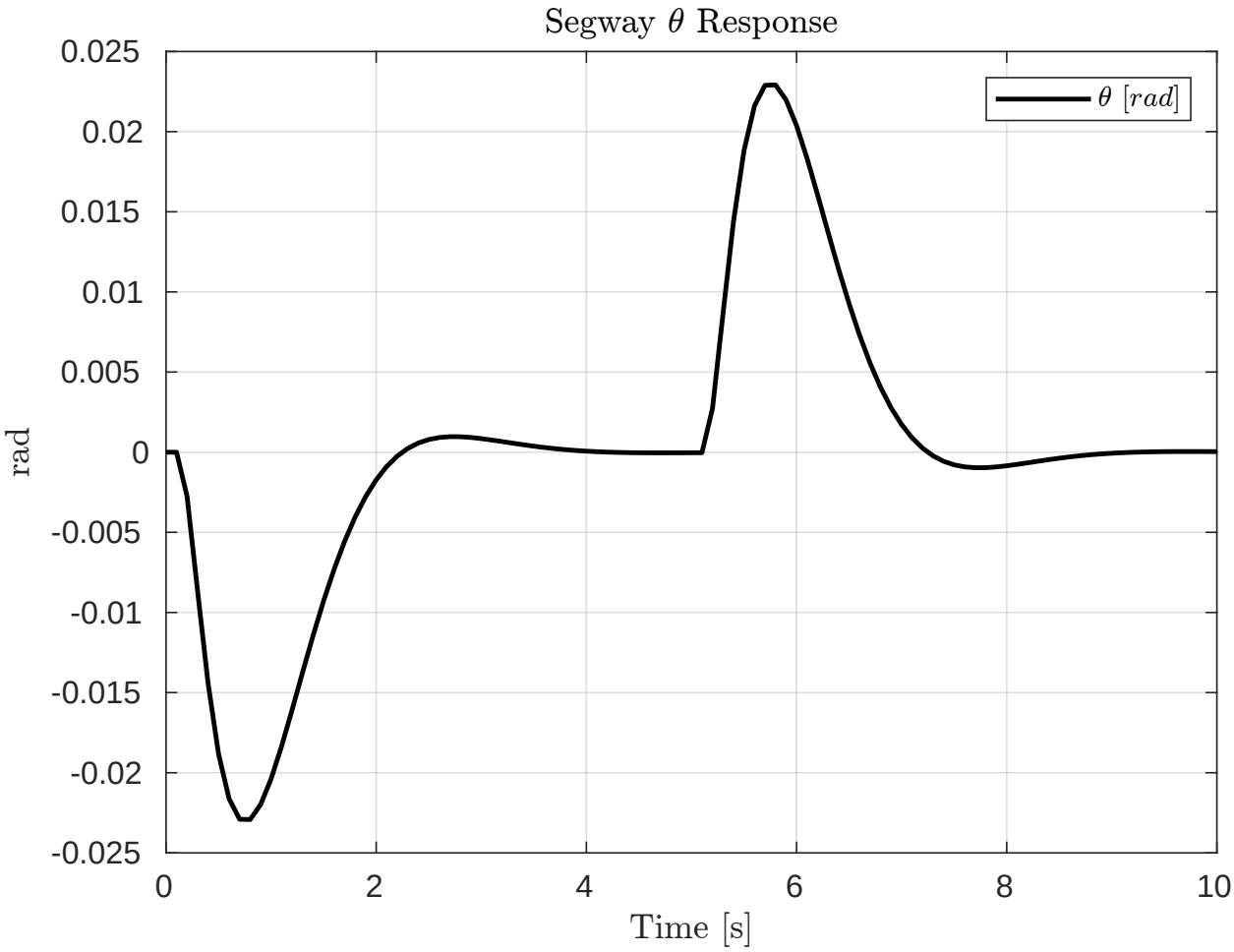


Figure A.8: θ response for Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs.

imported as STL files into a URDF (Unified Robot Description Format). The URDF format specifies the Segway's kinematic and rigid body model. It is likely that these do not translate one to one with the equations of motion presented in Section A.3. However, efforts were made to try and bridge the gap between the two models but further work needs to be done in this regard. A video of the Segway driving around in Gazebo is attached with this submission.

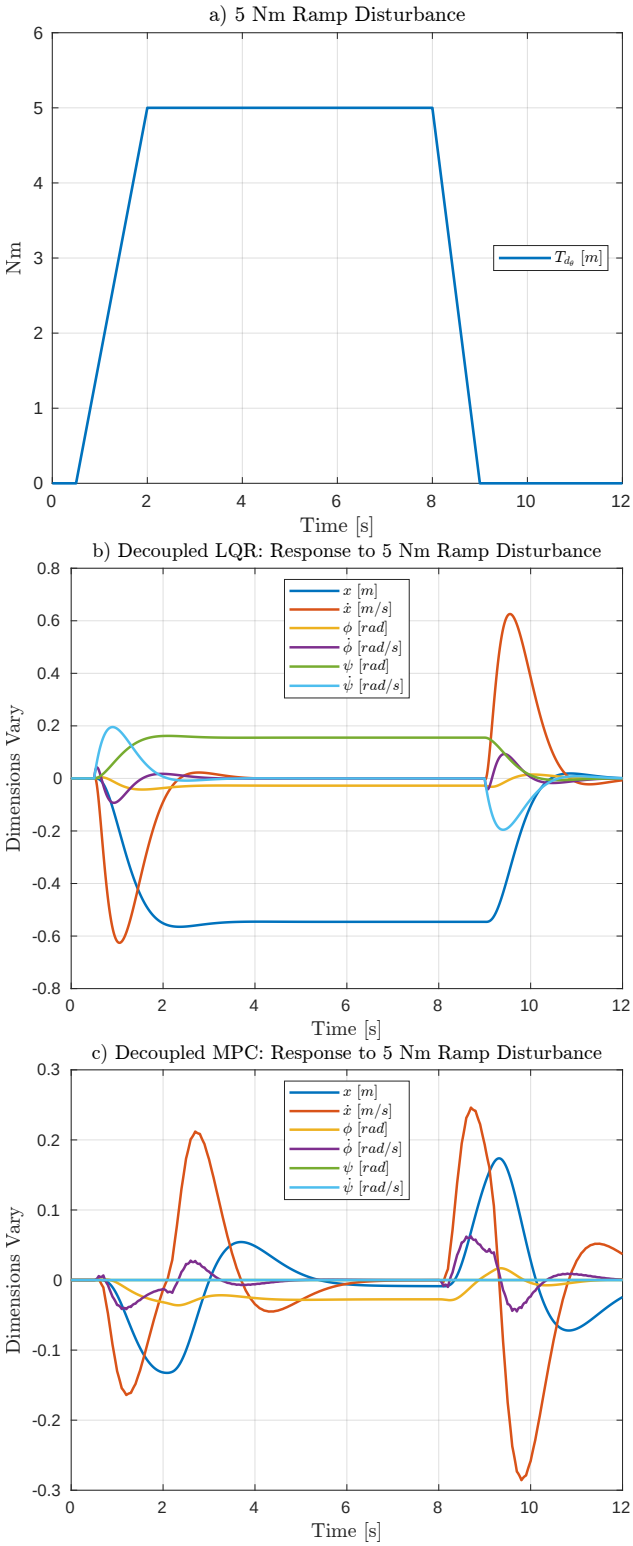


Figure A.9: Simulation of a) 5 Nm ramp disturbance acting on Segway handlebar b) system response using decoupled LQR control c) system response using decoupled MPC.

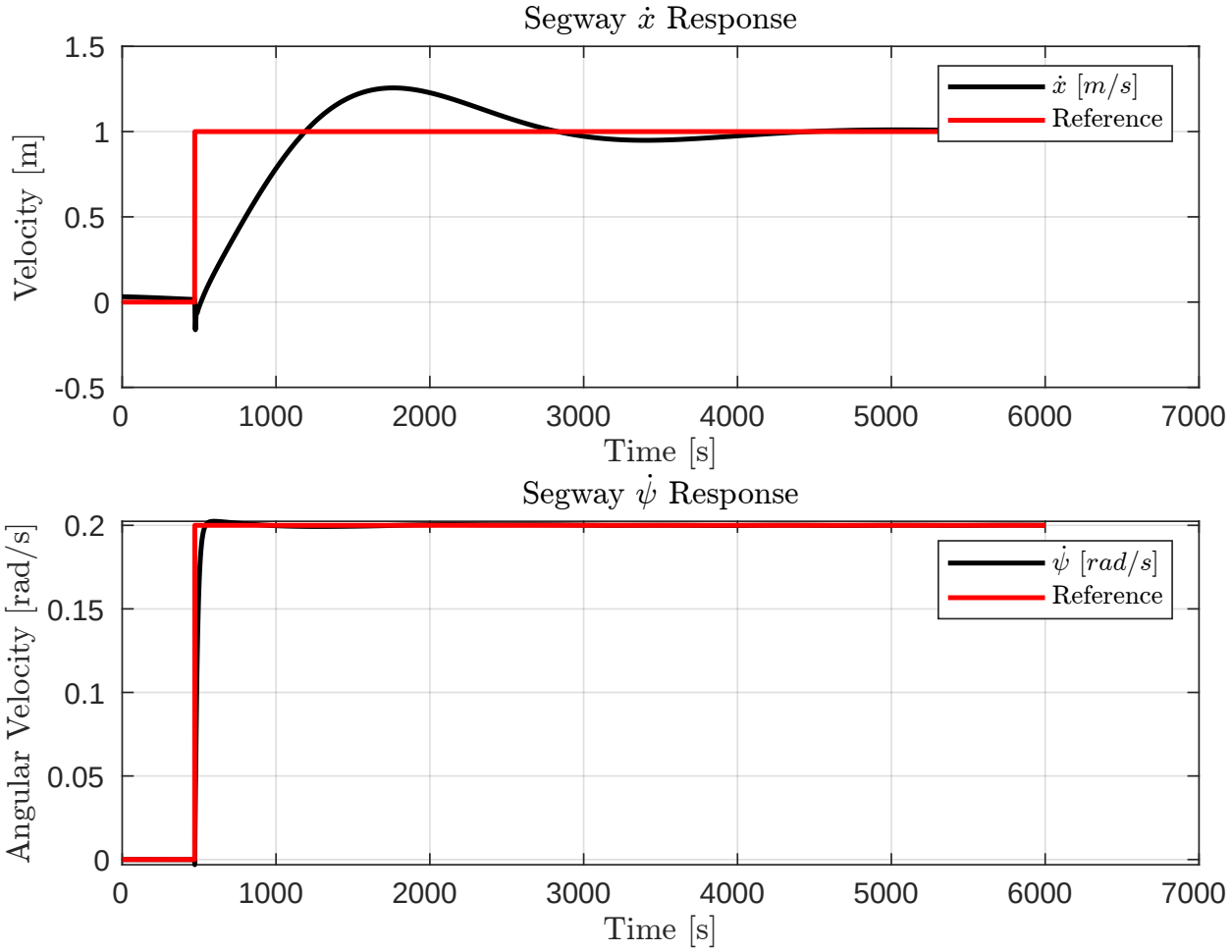


Figure A.10: Segway reference tracking to step changes in the $\dot{x}$ and $\dot{\psi}$ state inputs. The model used here is the high fidelity model in Gazebo, and the simulation was performed using Gazebo/Simulink co-simulation [163]. As a note, the x-axis is labelled incorrectly and should read $Time[steps]$.

A.6 Conclusions

In this work we presented a decoupled MPC formulation to control, stabilize and track reference trajectories with a two-wheeled Segway. Theory and background on MPC was introduced before developing the Segway model. A dual linear MPC formulation with constraints on inputs and outputs was presented to stabilize and control the Segway. It was shown that the resulting controllers were able to stabilize and track reference signals. The linear MPC controller was also compared against LQR and it was found that it performed better in the presence of disturbances. The same MPCs were tested on a high fidelity Segway model in Gazebo but this yielded sub-optimal performance for tracking velocity step inputs. Some explanation on the causes for this discrepancy were given, with the most likely factor being a model mismatch between the equations of motion and the URDF format used to describe the Segway in Gazebo.

Future work on this project could go in many directions. Firstly, the URDF model developed in Gazebo should match more closely with the equations of motion developed in Simulink. For example, there has been recent work to symbolically convert rigid-body models described in URDF format into equations of motion [182]. The author was unaware of this work at the time of development but this could be a viable option to bridge the modelling gap between the high-fidelty Gazebo simulator and Simulink. The work can also be extended to utilize Non-linear MPC to attain better tracking performance beyond the linear region of control used in this work. Segways were developed for human riders, and as such, exploring the developed MPC's capability for adapting to human rider dynamics would be interesting. To start off, the human could be modeled as a cylinder and point mass as seen in Figure A.11.

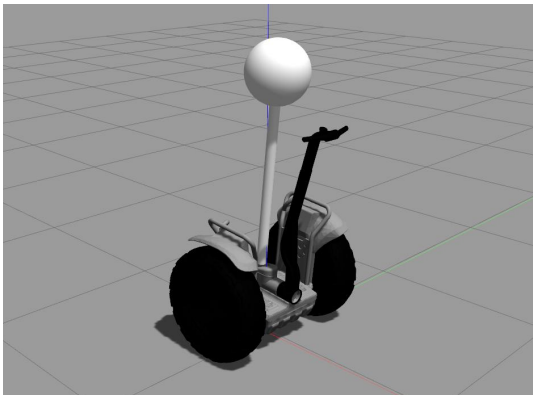


Figure A.11: Segway model with a human point mass rider.

Appendix B

REEM-C Hardware Malfunction and Fix

This appendix section provides a brief account of the hardware malfunction encountered in June 2024. After two weeks of intensive debugging, the issue was traced to the mSATA solid-state drive (SSD) connector on the control PC board. Figure B.1 illustrates the robot disassembled, with cabling coming out from its torso. Figure B.2 captures my colleagues and I troubleshooting the logic box, a compact cube containing numerous intricately nested boards. Despite its complexity, the design proved logical and straightforward to debug. We benefitted greatly from guidance provided by PAL Robotics employees during a visit to their office in May 2022 and through subsequent video calls and emails.

As shown in Figure B.3, the control PC board, with the mSATA SSD and connector highlighted, was identified as the root cause. Removing the SSD from the connector allowed the motherboard to boot. We also confirmed that the SSD itself was functional by testing it via an mSATA enclosure on a working desktop computer. While the precise failure within the control PC board remains unclear and warrants further investigation, an interim solution was implemented. As shown in Figure B.4, an external SSD, loaded with the control computer image, was mounted on the robot, enabling experiments to resume after a two-week delay.

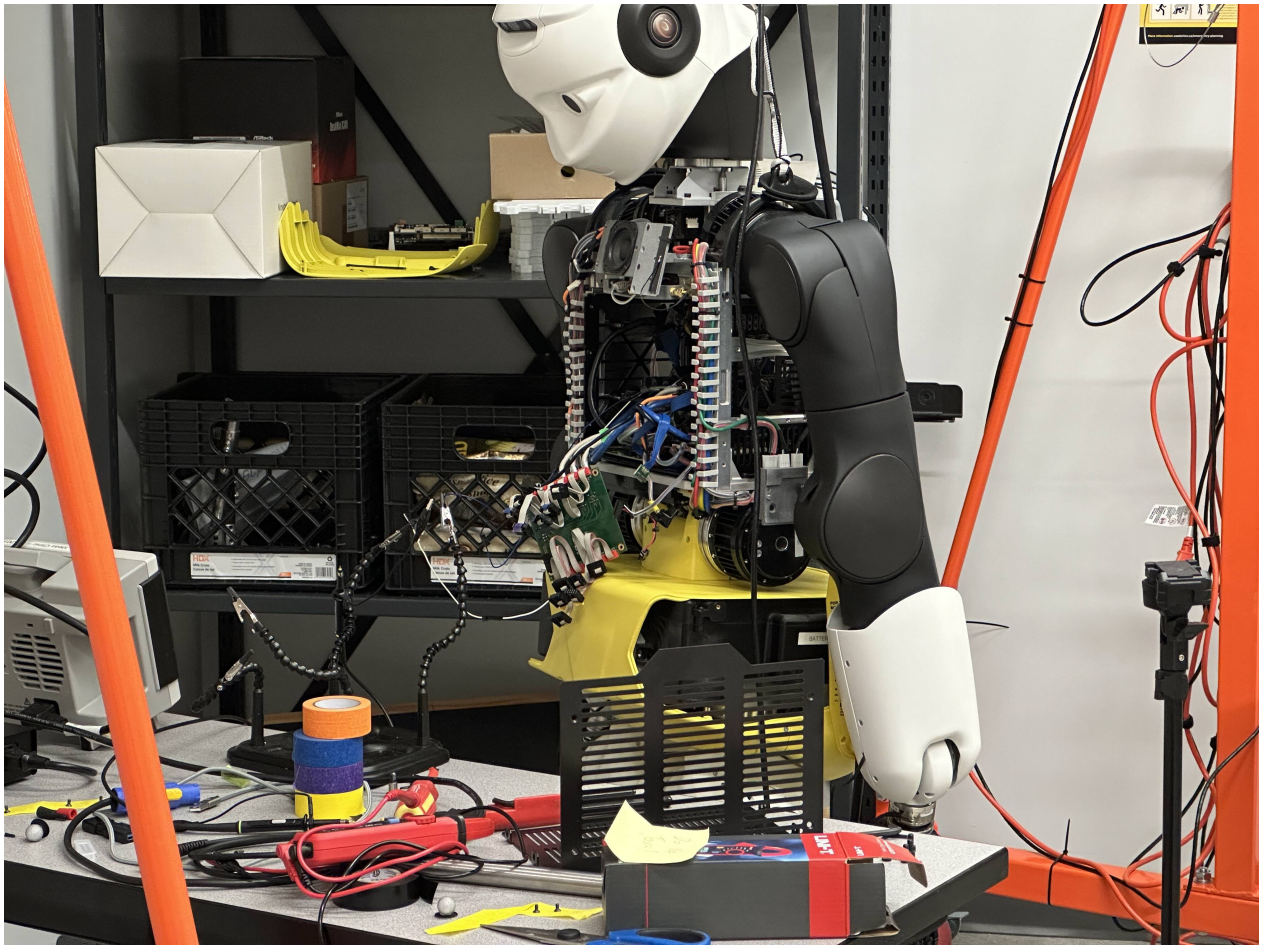


Figure B.1: The REEM-C taken apart for debugging purposes.



Figure B.2: My colleagues and I debugging the robot's logic box and compute module, which had been removed from its torso.

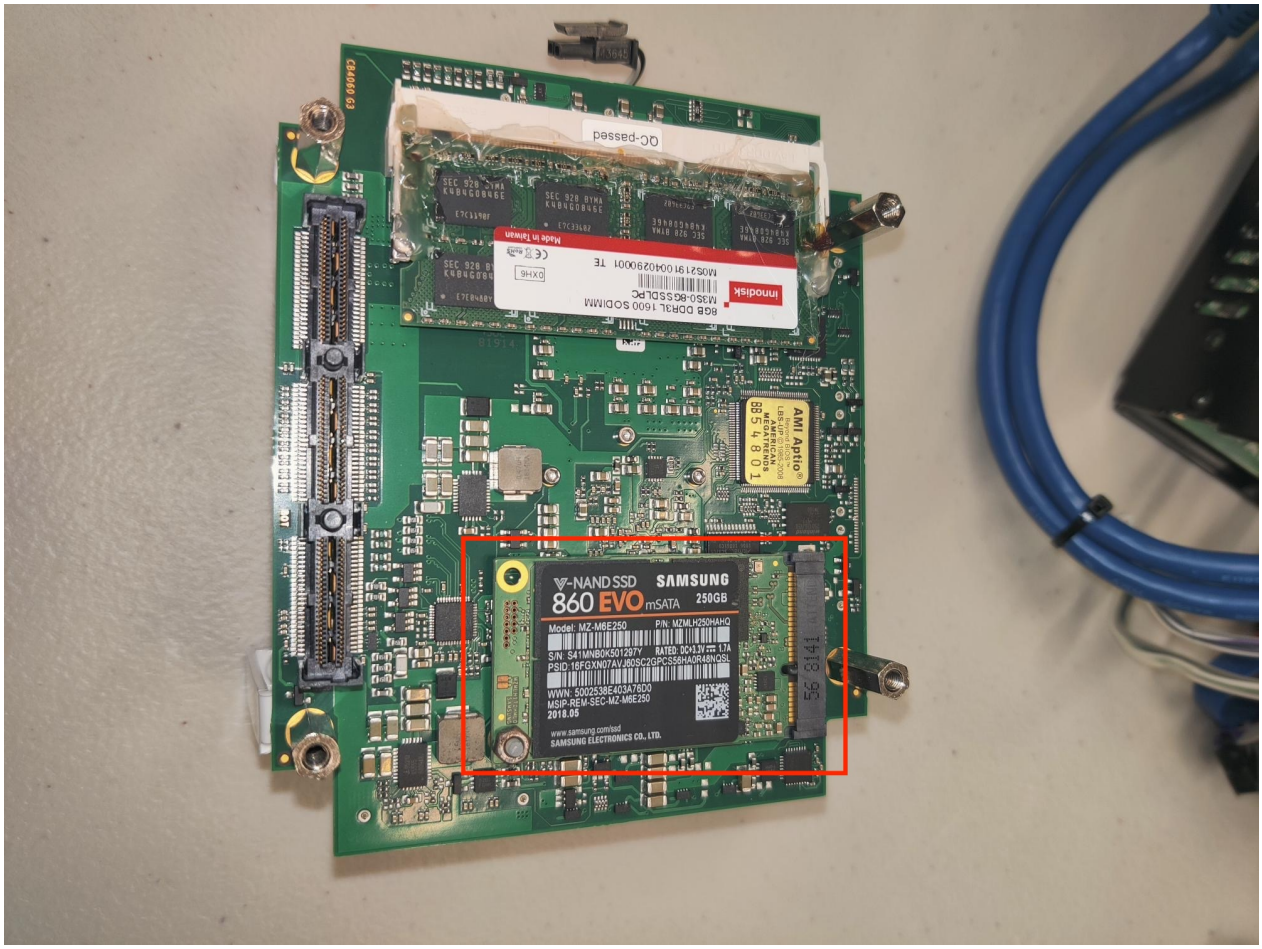


Figure B.3: The REEM-C control computer board is shown here. Highlighted in red is the mSATA SSD connector. Once the SSD was removed from this connector, the motherboard was able to complete its POST (Power-On Self-Test). While this may seem like a simple fix, the board is located deep within the robot's torso, requiring the entire compute logic box to be disassembled (see Figure B.2 and Figure B.1).



Figure B.4: The final solution to operationalize the robot involved mounting an external SSD onto the robot, running the control computer over USB 3.0. This configuration was sufficient to perform all tests described in Chapter 6.

Appendix C

REEM-C Energy Expenditure Comparison Between Walking and Personal Transporter Usage

In this appendix section, we detail the calculations for the energy expenditure comparison between walking and Segway usage for the REEM-C humanoid. Snapshots of the REEM-C walking on flat ground are shown in Figure C.1. Figure C.2 shows a graph of the charge used for riding a Segway compared against walking for 0.75 m. Driving a Segway takes 72% less charge.

To evaluate the runtime of the REEM-C under different operating conditions, we compare the total recorded current draw when the robot is walking and when it is using a personal transporter. The total current draw is measured in Coulombs, which we convert to ampere-hours for runtime estimation. The comparison reveals that REEM-C can operate for significantly longer durations when using a personal transporter compared to walking, given the reduced energy consumption. This analysis assumes consistent current draw and full utilization of the battery capacity during each mode of operation.

The conversion follows (assuming a nominal voltage of 44.4v and a battery capacity of 27.85 Ah, specifications from PAL Robotics):

$$1\,\mathrm{Ah} = 3600\,\mathrm{C}. \tag{C.1}$$

The total current consumption in ampere-hours, I_{total}, is given by:

$$I_{\text{total}} = \frac{Q_{\text{total}}}{3600}, \tag{C.2}$$

where Q_{total} is the total current in coulombs.

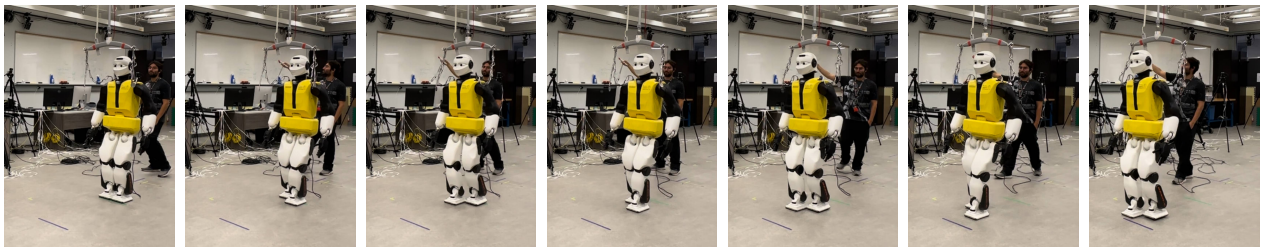


Figure C.1: The REEM-C walking. The REEM-C humanoid walking 0.2 m/s over flat ground for 0.75 m (from left to right).

Example 1: Walking (64667 Coulombs)

For walking, the total recorded current is 64667 coulombs:

$$I_{\text{total}} = \frac{64667\,\text{C}}{3600\,\text{C/Ah}} \approx 17.96\,\text{Ah}. \tag{C.3}$$

The battery's capacity is calculated as:

$$C_{\text{battery}} = \frac{1225\,\text{Wh}}{44.4\,\text{V}} \approx 27.58\,\text{Ah}. \tag{C.4}$$

The estimated run time for walking, T_{walking}, is:

$$T_{\text{walking}} = \frac{C_{\text{battery}}}{I_{\text{total}}} = \frac{27.58\,\text{Ah}}{17.96\,\text{Ah}} \approx 1.54\,\text{hours} \approx 92\,\text{minutes}. \tag{C.5}$$

Example 2: Personal Transporter (18271 Coulombs)

For personal transporter usage, the total recorded current is 18271 coulombs:

$$I_{\text{total}} = \frac{18271\,\text{C}}{3600\,\text{C/Ah}} \approx 5.08\,\text{Ah}. \tag{C.6}$$

The corresponding run time, $T_{\text{transporter}}$, is:

$$T_{\text{transporter}} = \frac{C_{\text{battery}}}{I_{\text{total}}} = \frac{27.58\,\text{Ah}}{5.08\,\text{Ah}} \approx 5.43\,\text{hours} \approx 326\,\text{minutes}. \tag{C.7}$$

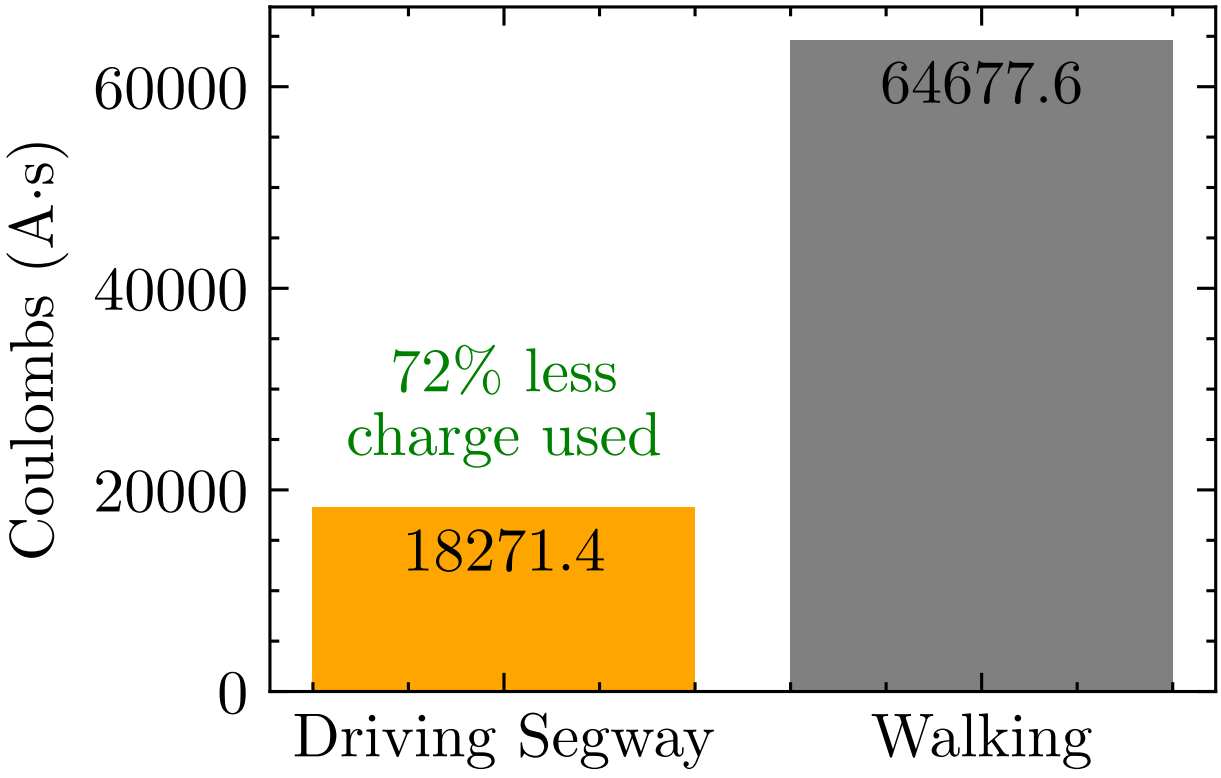


Figure C.2: REEM-C driving a Segway compared against walking. To travel a distance of 0.75 m, the REEM-C used 72% less energy when driving a Segway, as compared to walking. Using the internal 48v 1225 Wh battery, the REEM-C would last 5.4 hours riding a Segway and only 1.535 hours walking.