Towards Robust Bipedal Locomotion:
From Simple Models To Full-Body Compliance

THÈSE N^{O} 8760 (2018)

PRÉSENTÉE LE 24 AOÛT 2018
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE BIOROBOTIQUE
PROGRAMME DOCTORAL EN ROBOTIQUE, CONTRÔLE ET SYSTÈMES INTELLIGENTS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Salman FARAJI

acceptée sur proposition du jury:

Prof. P. Dillenbourg, président du jury
Prof. A. Ijspeert, directeur de thèse
Prof. O. Khatib, rapporteur
Prof. C. Atkeson, rapporteur
Prof. S. Coros, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

If you can't explain it simply,
you don't understand it well enough.
— Albert Einstein

To my parents,
my sweet brother,
and my lovely wife
…

Acknowledgements

I'd like to dedicate a few paragraphs in the beginning to thank all the great people who helped me learn amazing things in my project. I know the space is too limited here to accommodate everybody, and my words are never enough, but without friends and colleagues, these achievements were impossible. First of all, highest and most profound gratitude goes to Auke, my excellent supervisor who helped me through all the up and down moments in my Ph.D. and guided me to reach this point. I can firmly say that the topic of my project and objectives matched 100% to my interests, and this was only thanks to Auke who really supervised everything to follow smoothly. He was always helpful in any aspect of life in academia or outside, and I relied on his help almost in all steps throughout my Ph.D. At this point, I'd also like thank professors Oussama Khatib, Christopher Atkeson and Stellian Coros who accepted to attend my private Ph.D. defense.

Next, I would like to thank the European Union and FP7 program for funding the Walk-Man project which supported my Ph.D. financially. I'm so glad and honored to know such a vast network of people across Europe in different cultures, backgrounds, and scientific interests. I'd especially thank Nikolaos Tsagarakis who coordinated the Walk-Man project, provided help with our robot and gave inspiring ideas. I would also like to thank Antonio Bicci, Darwin Caldwell, Tamim Asfour, Renaud Ronsse and Lucia Pallottino for fruitful collaborations in the project, and lots of memorable nights we spent together after the meetings. They helped me understand various topics related to my project.

In different parts of my thesis, I received very technical comments and advice from multiple people across the world. I had the chance to do my master's thesis with Chris Atkeson from Carnegie Mellon University. I can firmly say he is one of the top three pillars in the field of dynamic walking and I'm desperately thankful for all the helpful comments and tough questions I received from him. In the very early stages of Ph.D., Eric Whitman, a former student of Chris also helped me to understand and set up state of the art controllers in my project. Without his help, I was probably one year behind the point I'm currently standing on.

I like to thank Soha Pouya, a former Ph.D. in our lab who actually pulled me into this scientific community by supervising my semester project and later co-supervising my master project. Soha and her husband Mohammad really helped me understand walking, control, and mechanics almost from zero. In some sense, I followed the path

of model-based control in our lab which was started by Ludovic Righetti and later continued by Soha. I was so lucky to meet Ludovic many times during my Ph.D. in different conferences and also his lab in the Max-Plank Institute. I can roughly say that each time, Ludovic corrected my research path by 30 degrees! All the bits of help, comments, and updates from Ludovic genuinely deserve great acknowledgment. Besides, a very knowledgeable (and busy) control scientist, Philippe Muellhaupt saved weeks and months of my time struggling with meaningless learning optimizations. His deep insight into control theories actually forced me to think entirely out of the box and change my direction almost 90 degrees! It was only thanks to him that I could find a meaningful interpretation for my time-projection controller.

Thanks to my supervisor Auke and the Walk-Man project, I had the chance to travel many times and attend conferences and meetings all around the world. Although I faced visa problems for almost five conferences, I still managed to participate in six/seven others in which I learned an incredible amount of new things which indeed increased my productivity. I very much appreciate Sylvie Fiaux, the administrative assistant of our laboratory for handling the paperwork. Following the unfortunate visa stories, however, I'd like to express my best gratitude for Amy Wu, my representative in US robotics conferences! She kindly agreed to present all the papers and pieces of work I had submitted, but I wasn't able to present. I'm so happy that my Ph.D. overlapped a few years with Amy's stay in BioRob. Although we worked on different topics, her research line and hours and days of discussions really gave a biomechanics spirit to my research and made it more valuable. I'm so proud of our joint publications.

While writing these words, I feel I can never leave BioRob, and it is my second home now. This is only because of a large number of close friends and collaborators I've found here. I was probably the least up-to-date person about movies, which left me almost speechless at the launch meetings. But otherwise, I really enjoyed the fantastic atmosphere which is unforgettable in my life. I would like to thank Nicolas Van der Noot and Luca Colasanto for their kind support and collaboration in various parts of the Walk-Man project. I also enjoyed fruitful discussions with more senior members in BioRob, including Rico Möckel, Massimo Vespignani, and Mostafa Ajallooeian. I would especially thank Jesse van den Kieboom for his kind support. Jesse took my hands as a beginner and walked me through almost all the details about the robot. I also find it really hard to express my gratitude in words for Peter Eckert who was like a saving angel after numerous robot crashes. This is greatly appreciated. Moreover, I'm proud that I had very lovely officemates in BioRob all the time, especially Florin Dzeladini, Kamilo Melo, Alexandre Tuleu and Laura Paez. The kind of joyful moments I had with them would probably never repeat again. I wish good luck for all the past and new BioRob members forever.

Studying in Switzerland for almost seven years and interactions with so many great people were only one side of the coin. On the other side, being in Lausanne among various other Iranian students and their families gave me an infinite amount of energy end enthusiasm. In particular, I would like to thank Mohammad Hashemi for organizing

amazing outdoor hikes which left me with about 50 gigabytes of memorable photographs! During these years, I had the chance to fly back home and visit my family regularly almost twice a year. This gave me a tremendous amount of energy to come back every time and continue my studies. However, each visit was too short to compensate months of staying away from my parents and my little brother. I don't know if any language can ever describe the kind of heartbreaking moments happened when leaving my parents back to Switzerland every time. I'm really thankful for all these supports which can't be quantified in any form. Finally, I would like to express highest appreciations for my sweet wife Mahya who supported me all the way throughout these years and motivated me to achieve higher and higher steps in my carrier. It was truly exciting when she first came to Switzerland and started a whole new chapter in my life, which has been full of sweet moments so far. I wish great success for her future carrier.

Lausanne, July 25, 2018 Salman Faraji

Abstract

Thanks to better actuator technologies and control algorithms, humanoid robots to date can perform a wide range of locomotion activities outside lab environments. Recent dexterous and robust demos with humanoids seemed almost impossible with earlier robots of the 80s and 90s. However, they are still far from what humans do every day. Humanoid robots face various control challenges like high dimensionality, contact switches during locomotion and a floating-base nature which makes them fall all the time. A rich set of sensory inputs and a high-bandwidth actuation are often needed to ensure fast and effective reactions to unforeseen conditions, e.g., terrain variations, external pushes, slippages, unknown payloads, etc. State of the art technologies today seem to provide such valuable hardware components. However, regarding software, there is plenty of room for improvement.

Locomotion planning and control problems are often treated separately in conventional humanoid control algorithms. The control challenges mentioned above are probably the main reason for such separation. Here, planning refers to the process of finding consistent open-loop trajectories, which may take arbitrarily long computations off-line. Control, on the other hand, should be done very fast online to ensure stability. However, there is a fundamental problem. Ankle joints can provide direct control authority to stabilize the system, but the feet in humanoids are too small to allow for capturing even moderate perturbations with this stabilization strategy. Therefore, conventional rigid combinations of off-line planning and online control are too limited. The robot is not limited, however; it can use other stabilization strategies to recover from significant perturbations like humans. One of these strategies is to continuously adjust footstep locations online which requires modification of trajectories. The new trajectories should remain consistent with system dynamics to ensure maximum stability. However, this requires expensive computations with traditional mechanical models.

In this thesis, we want to link planning and control problems again and enable for online trajectory modification in a meaningful way. First, we propose a new way of describing robot geometries like molecules which breaks the complexity of conventional models. We use this technique and derive a planning algorithm that is fast enough to be used online for multi-contact motion planning. Similarly, we derive 3LP, a simplified linear three-mass model for bipedal walking, which offers orders of magnitude faster computations than full mechanical models. Molecule-based simplified geometries and the 3LP model both produce more consistent trajectories, i.e., including essential dynamics of the system in fast locomotions. Next, we focus more on walking and use the 3LP model to formulate online control algorithms based on the foot-stepping strategy. Because of faster computations, our method can combine planning and control to achieve more robustness by modifying trajectories online. The method is based on model predictive control, however, we also propose a faster controller with time-projection that demonstrates a close performance without numerical optimizations. We also deploy an efficient implementation of inverse dynamics together with advanced sensor fusion and actuator control algorithms to ensure a precise and compliant tracking of the simplified 3LP trajectories. Extensive simulations and hardware experiments on COMAN robot demonstrate effectiveness and strengths of our method.

This thesis goes beyond humanoid walking applications. We further use the developed modeling tools to analyze and understand principles of human locomotion. Our 3LP model can describe the exchange of energy between human limbs in walking to some extent. We use this property to propose a metabolic-cost model of human walking which successfully describes trends in various conditions, including unusual frequencies, wide steps, added mass, reduced gravity, and flat walking. In another study, we use the 3LP model to explain geometric asymmetries of human gait in inclined walking and torso-bending conditions. Such asymmetries can be only explained by a mechanical model that encodes internal interactions between the body segments in human. Intrinsic power of the 3LP model to generate walking gaits in all these conditions makes it a handy solution for walking control and gait analysis, despite being yet a simplified model. To fill the reality gap, finally, we propose a kinematic conversion method that takes 3LP trajectories as input and generates more human-like postures. Using this method, the 3LP model, and the time-projecting controller, we introduce a graphical user interface in the end to simulate periodic and transient human-like walking conditions. We hope to use this combination in future to produce faster and more human-like walking gaits, possibly with more capable humanoid robots.

Keywords: Mechanics, Control, Optimization, Bipedal Walking, Planning, Compliance

Zusammenfassung

Dank neuer Antriebstechnologien und Kontrollalgorithmen ist es aktuellen humanoiden Robotern möglich, eine breite Palette von Bewegungsabläufen ausserhalb der Laborumgebung zu demonstrieren. Diese geschickten und robusten Demonstrationen schienen mit früheren Robotern der 80er und 90er Jahren weitestgehend unmöglich. Dennoch sind auch heutige Roboter immer noch weit von den Fähigkeiten eines Menschen entfernt. Humanoide Roboter begegnen diversen Herausforderungen in ihrer Kontrolle. Einige Beispiele sind die hohe Dimension der Kontrollparameter, sich ändernde Kontakteigenschaften während der Bewegung und ihre Eigenart auf einer schwebenden Referenz zu basieren, alles Faktoren, welche oft zum Versagen durch Umfallen führen. Oft werden sowohl ein vielfältiges Set von Sensoren als auch Antriebe mit hoher Bandweite benötigt, um schnelle und effektive Reaktionen zu unvorhergesehenen Ereignissen, wie z. B. sich verändernder Untergrund, externe Stösse, Schlupf, und unbekannte Kontakte bei der Ladung, sicherzustellen. Heutige Technologien können solche mechanische Komponenten zur Verfügung stellen. Im Gegensatz hierzu ist in Bezug auf die Kontrollimplementierung noch viel Raum für Verbesserungen.

Bewegungsplanung und Kontrollprobleme werden in konventionellen Kontrollalgorithmen oft separat behandelt. Die zuvor erwähnten Kontrollherausforderungen sind möglicherweise der Hauptgrund für diese Aufteilung. In dieser Dissertation bezieht sich Planung auf den Prozess zur Erzeugung konsistenter Bewegungstrajektorien im offenem Regelkreis, wobei beliebig lange Berechnungszeiten, offline, d.h. nicht während der eigentlichen Bewegung, in Kauf genommen werden können. Um die Bewegungserzeugung zu garantieren, sollte der Kontrollablauf allerdings online (während der Bewegung) zur Verfügung stehen und sehr schnell agieren, um die notwendige Stabilität zu garantieren. Genau hier aber liegt das Hauptproblem. Die vorhandenen Aktuatoren der Fussgelenke könnten die Stabilisierung übernehmen aber die Füsse in humanoiden Roboter sind für diese Stabilisierungsstrategie sind zu klein. Dies gilt bereits moderate Störungen. Folglich ist die konventionelle Kombination von offline Planung und online Kontrolle zu limitierend. Der Roboter selber ist aber nicht limitiert; er kann wie Menschen auch andere Stabilisierungsstrategien verwenden. Eine dieser Strategien ist, die Platzierung der Füsse anzupassen, wozu eine Änderung der Bewegungstrajektorien nötig ist. In dieser Dissertation möchten wir die Planungs- und Kontrollprobleme miteinander verknüpfen, sodass eine sinnvolle online Änderung der Bewegungstrajektorien realistisch wird.

Die daraus entstehenden, modifizierten Trajektorien sollten mit der Systemdynamik konsistent bleiben, um maximale Stabilität sicherzustellen. In traditionellen mechanischen Modellen benötigt dies aber lange Berechnungszeit. Zur Lösung dieses Problems schlagen wir eine neue Art vor, Geometrien in Robotern zu beschreiben. Die Geometrieerfassung als Moleküle verringert die Komplexität in der Anwendung konventioneller Modelle. Wir zeigen die Verwendung dieser Technik und leiten einen Planungsalgorithmus her, welcher schnell genug ist, um Bewegungsabläufe mit vielen Kontaktpunkten online zu berechnen. Mit einer ähnlichen Vorgehensweise leiten wir ein vereinfachtes, lineares Modell mit drei Massen für das zweibeinige Gehen „3LP“ her, welches die Berechnungszeit um einige Grössenordnungen reduziert. Beide Herleitungen produzieren konsistentere Trajektorien, da sie die auftretende Dynamik des Systems bei schneller Fortbewegung miteinschliessen. Der folgende Focus liegt auf dem Gehen an sich. Wir verwenden das besagte lineare Modell um online Kontrollalgorithmen zur Bewegungserzeugung, basierend auf der Strategie der Fussplatzierungen, herzuleiten. Aufgrund der schnelleren Berechnungszeit kann unsere Methode die Planung und den Regler kombinieren und erzielt durch eine online Änderung der Trajektorien eine höhere Robustheit. Die Methode basiert auf dem Modell-Voraussagungs-Regler, wobei wir jedoch einen schnelleren Regler mit Zeit-Projektion vorschlagen, welcher eine ähnliche Verhaltensweise ohne nummerische Optimierungen zeigt. Des Weiteren setzen wir eine effiziente Implementierung von inverser Dynamisch zusammen mit fortgeschrittener Sensor-Verschmelzung und Antriebs-Kontrollalgorithmen ein, um präzise und anpassungsfähige Bewegungsaufzeichnungen von vereinfachten Trajektorien sicherzustellen. Ausgiebige Simulationen und Experimente mit dem COMAN Roboter demonstrieren die Effektivität und Stärken unserer Methode.

Diese Dissertation geht zudem über die Anwendungsmöglichkeiten von humanoiden Robotern hinaus. Weiterführend benutzen wir die erarbeiteten Modellierungstechniken, um Prinzipien von humanoider Fortbewegung zu analysieren und verstehen. Unser 3LP Model kann den Energieaustausch zwischen Extremitäten zu einem gewissen Grade erklären. Wir verwenden diesen Aspekt um ein Modell für das Gehen, basierend auf den metabolischen Kosten vorzuschlagen. Das besagte Modell beschreibt erfolgreich Trends in diversen Bedingungen, wie zum Beispiel in abnormalen Frequenzen, weiten Schritten, hinzugefügter Masse, reduzierter Gravitation und ebenem Gehen. Obwohl das 3LP ein vereinfachtes Modell ist, zeigt es in der Generierung von Gangarten in all diesen Bedingungen seine Stärke als praktische Lösung für Regler und Gang-Analyse. Um den Realitäts-Unterschied zu überwinden schlagen wir abschließend eine Methode für kinematische Konvertierung vor, welche aus 3LP Trajektorien menschenähnlichere Körperhaltungen generiert. Unter der Verwendung dieser Methode, dem 3LP Modell und dem zeit-projizierenden Regler präsentieren wir eine GUI (graphical user interface), in welcher periodische und transiente menschenähnliche Gangarten simuliert werden können. Wir hoffen, diese Kombination in Zukunft weiterhin nutzen zu können, um schnellere und menschenähnlichere Gangarten generieren zu können und diese auf möglicherweise mit leistungsfähigeren humanoiden Robotern zur Anwendung zu bringen.

Schlüsselwörter: Mechanik, Kontrolle, Optimierung, Zweibeinige Gehen, Planung, Konformität

Résumé

Grâce aux avancées technologiques d'actionneurs et à de meilleurs algorithmes de contrôle, les robots humanoïdes peuvent à ce jour effectuer une large gamme d'activités de locomotion en dehors des environnements de laboratoire. Les récentes démonstrations d'habilités et de robustesses des humanoïdes semblaient presque impossibles avec les robots précédents des années 80 et 90. Cependant, ils sont encore loin de pouvoir reproduire ce que les humains font tous les jours. Les robots humanoïdes font face à divers défis de contrôle tels que la haute dimensionnalité, les changement de contact pendant la locomotion et une nature à base flottante qui les font tomber tout le temps. Un ensemble riche d'entrées sensorielles et un actionnement à bande passante élevée sont souvent nécessaires pour assurer des réactions rapides et efficaces à des conditions imprévues, telles que des variations de terrain, des poussées externes, des glissements, des charges utiles inconnues, etc. Les technologies de l'état de l'art semblent aujourd'hui fournir des composants matériels aussi précieux. Cependant, en ce qui concerne les logiciels, il y a beaucoup de place pour de l'amélioration.

Les problèmes de planification et de contrôle de la locomotion sont souvent traités séparément dans les algorithmes de contrôle pour des humanoïdes conventionnels. Les défis de contrôle mentionnés ci-dessus sont probablement la raison principale d'une telle séparation. Ici, la planification se réfère au processus de recherche de trajectoires cohérentes en boucle ouverte et peut prendre des calculs arbitrairement longs hors ligne. D'un autre côté, le contrôle devrait être fait très rapidement et en ligne pour assurer la stabilité. Cependant, il y a un problème fondamental. Les articulations de la cheville peuvent fournir une autorité de contrôle directe pour stabiliser le système, mais les pieds dans les humanoïdes sont trop petits pour permettre de capturer même des perturbations modérées avec cette stratégie de stabilisation. Par conséquent, les combinaisons rigides classiques de planification hors ligne et de contrôle en ligne sont trop limitées. Le robot n'est pas limité, cependant ; il peut utiliser d'autres stratégies de stabilisation pour se remettre de perturbations significatives comme les humains. Une de ces stratégies consiste à ajuster les emplacements des pieds, ce qui nécessite une modification des trajectoires. Dans cette thèse, nous voulons relier les problèmes de planification et de contrôle et permettre une modification significative de la trajectoire en ligne.

Les trajectoires modifiées doivent rester cohérentes avec la dynamique du système pour assurer une stabilité maximale. Cependant, cela nécessite des calculs coûteux avec des modèles mécaniques traditionnels. Tout d'abord, nous proposons une nouvelle façon de décrire les géométries de robots comme des molécules qui brisent la complexité des modèles conventionnels. Nous utilisons cette technique et en déduisons un algorithme de planification suffisamment rapide pour être utilisé en ligne pour la planification de mouvements multi-contacts. De même, nous dérivons 3LP, un modèle linéaire à trois masses simplifié pour la marche bipède qui offre des ordres de grandeur calculs plus rapides que les modèles mécaniques complets. Ces dérivations produisent toutes deux des trajectoires plus cohérentes, c'est-à-dire comprenant la dynamique essentielle du système dans les locomotions rapides. Ensuite, nous nous concentrons davantage sur la marche et utilisons le modèle linéaire pour dériver des algorithmes de contrôle en ligne basés sur la stratégie du pas à pas. Grâce à des calculs plus rapides, notre méthode peut combiner planification et contrôle pour obtenir plus de robustesse en modifiant les trajectoires en ligne. La méthode est basée sur la commande prédictive, bien que nous proposons aussi un contrôleur plus rapide avec projection temporelle qui démontre une performance proche sans optimisations numériques. Nous déployons également une implémentation efficace de la dynamique inverse ainsi que des algorithmes avancés de fusion de capteurs et de contrôle d'actionneurs pour assurer un suivi précis et conforme des trajectoires simplifiées commandées. Des simulations approfondies et des expériences matérielles sur robot COMAN démontrent l'efficacité et les points forts de notre méthode.

Cette thèse va au-delà des applications de marche humanoïde. Nous utilisons également les outils de modélisation développés pour analyser et comprendre les principes de la locomotion humaine. Notre modèle 3LP peut décrire l'échange d'énergie entre les membres du corps humain dans la marche dans une certaine mesure. Nous utilisons cette propriété pour proposer un modèle métabolique de la marche humaine qui décrit avec succès les tendances dans diverses conditions, y compris avec des fréquences inhabituelles, des longs pas, de la masse ajoutée, une gravité réduite et la marche à plat. Dans une autre étude, nous utilisons le modèle 3LP pour expliquer les asymétries géométriques de la marche dans des conditions de marche inclinée et de flexion du torse. De telles asymétries ne peuvent s'expliquer que par un modèle mécanique qui prend en compte les interactions internes entre les segments corporels chez l'humain. La puissance intrinsèque du modèle 3LP pour générer des allures de marche dans toutes ces conditions en fait une solution pratique pour le contrôle de la marche et l'analyse de la marche, bien qu'elle soit encore un modèle simplifié. Pour combler l'écart de réalité, nous proposons enfin une méthode de conversion cinématique qui prend en entrée les trajectoires 3LP et génère plus de postures humaines. En utilisant cette méthode, le modèle 3LP et le contrôleur de projection temporelle, nous introduisons finalement une interface graphique pour simuler des conditions périodiques et transitoires de marche semblables à celles de être humain. Nous espérons utiliser cette combinaison à l'avenir pour produire des allures de marche plus rapides et plus humaines, éventuellement avec des robots humanoïdes plus performants.

Mots-clés : Mécanique, Contrôle, Optimisation, Bipède, Planification, Conformité

Contents

Acknowledgements **v**

Abstract (English/Français/Deutsch) **ix**

List of figures **xxi**

List of tables **xxv**

1 Introduction **1**
- 1.1 Motivation 1
- 1.2 Approach 2
- 1.3 Contributions 4
- 1.4 Thesis Outline 5
- 1.5 Brief Overview 12

I Starting Point **15**

2 Model Predictive Control **17**
- 2.1 Background 18
- 2.2 Whole body optimization 21
- 2.3 Trajectory pattern generator 23
- 2.4 Foot-step planner 25
- 2.5 Results 28
  - 2.5.1 Range of speeds 29
  - 2.5.2 External pushes 30
  - 2.5.3 Model errors 30
  - 2.5.4 Perturbations 31
  - 2.5.5 Uneven terrain 32
- 2.6 Discussion 32

3 Inverse Dynamics **35**
- 3.1 Background 36
- 3.2 Torque tracking 38

3.2.1 Model identification 38
3.2.2 Control law 39
3.3 Robot state estimation 40
3.3.1 Stage 1: base velocity estimation 42
3.3.2 Stage 2: base position and yaw angle estimation 43
3.4 Controllers 44
3.4.1 Inverse dynamics layer 44
3.4.2 Cartesian control layer 45
3.4.3 Cartesian trajectory generation layer 46
3.5 Results 46
3.5.1 Torque tracking 46
3.5.2 State estimation and control 47
3.6 Discussion 50

II Contributions in Modeling 51

4 Point-Wise Model 53
4.1 Background 54
4.2 Molecule-inspired modeling 55
4.2.1 Vector-based equation of motion 57
4.2.2 Case study: humanoid model 59
4.3 Static posture optimization 60
4.3.1 Optimization setup 61
4.4 Results 63
4.4.1 Global convergence 64
4.4.2 Sensitivity to initial condition 65
4.4.3 Sensitivity to optimization parameters 66
4.4.4 Computation time 66
4.5 Discussion 66

5 Motion Planning 69
5.1 Background 70
5.1.1 The curse of dimensionality 70
5.1.2 Simplified dynamics, full kinematics 70
5.1.3 Trajectory parameterization 71
5.1.4 Fixed contact sequences 72
5.1.5 Emergent contact sequence 72
5.1.6 Environment complexity 73
5.1.7 Rooms for improvement 73
5.1.8 Contributions, novelties 74
5.2 Problem formulation 74
5.2.1 Model formulas 75

5.2.2 Optimization setup 77
5.3 Results 79
5.3.1 Swing dynamics 79
5.3.2 Static vs. dynamic motions 80
5.3.3 Optimization performance 80
5.4 Discussion 80

6 3LP Model 85
6.1 Background 86
6.1.1 Hierarchical controllers 86
6.1.2 Inverted pendulums 86
6.1.3 Multi-link pendulums 88
6.1.4 Spring-loaded pendulums 89
6.1.5 Control difficulty 89
6.1.6 Why 3LP? 89
6.2 3LP dynamics 90
6.2.1 Single support 93
6.2.2 Double support 95
6.2.3 Full step 98
6.3 Periodic gaits 98
6.3.1 Pseudo-passive gaits manifold 101
6.3.2 Actuated gaits manifold 103
6.3.3 Numerical examples 104
6.4 Comparison with human data 108
6.4.1 Sagittal dynamics 109
6.4.2 Vertical GRF 109
6.4.3 Lateral dynamics 110
6.4.4 Transversal rotation torques 111
6.5 Discussion 111

III Contributions in Control 115

7 Time-Projection Control 117
7.1 Background 118
7.1.1 Template models 118
7.1.2 Discrete control 119
7.1.3 Limitations of discrete control 119
7.1.4 Continuous control 120
7.1.5 A better template model? 122
7.1.6 Time-projection controller 123
7.1.7 Novelty 125
7.2 Time-projection 125

7.2.1 Linear system 126
7.2.2 DLQR controller 127
7.2.3 Projection controller 127
7.3 Simulations 129
7.3.1 Intermittent pushes 130
7.3.2 Speed modulation 131
7.4 Analysis 132
7.4.1 Eigen values 133
7.4.2 Push recovery strength 134
7.4.3 Viable regions 135
7.5 Discussion 138
7.5.1 Physical explanation 138
7.5.2 3LP advantage 138
7.5.3 Comparison to MPC 139
7.5.4 Different from intermittent control 139
7.5.5 Measurement requirements 140
7.5.6 Performance 140
7.5.7 Other applications 140

8 Walking and Push Recovery 143
8.1 Background 144
8.1.1 Compliance in walking 144
8.1.2 Swing dynamics 145
8.1.3 Gait generation 145
8.1.4 Control approach 146
8.1.5 Novelty 149
8.2 Hardware control 149
8.2.1 Low-level challenges 151
8.2.2 Open-loop control 155
8.2.3 Closed-loop control 158
8.3 Results 164
8.3.1 In-place walking 164
8.3.2 Intermittent push recovery 167
8.3.3 Continuous pushes 167
8.3.4 Walking 168
8.3.5 Effective ankle stiffness 169
8.4 Discussion 170
8.4.1 Dynamic walking 170
8.4.2 Decoupling assumptions 172
8.4.3 Constant CoM height 172
8.4.4 Hardware limitations 172
8.4.5 Compliance 173

IV Application to Human Walking 175

9 Energetic Model 177
- 9.1 Background . . . 178
- 9.2 Methods . . . 180
  - 9.2.1 Metabolic cost model . . . 180
- 9.3 Results . . . 186
  - 9.3.1 Step width . . . 188
  - 9.3.2 Added mass on the leg . . . 189
  - 9.3.3 Extra swing foot lift . . . 190
  - 9.3.4 Simulated reduced gravity . . . 191
  - 9.3.5 Flat-trajectory walking . . . 191
  - 9.3.6 Walking with obesity . . . 191
- 9.4 Discussion . . . 192

10 Walking Asymmetries 199
- 10.1 Background . . . 200
- 10.2 Methods . . . 203
  - 10.2.1 Experiment . . . 203
  - 10.2.2 Analysis . . . 204
- 10.3 Results . . . 205
- 10.4 Discussion . . . 206

11 Human-Like Animation 211
- 11.1 Background . . . 212
  - 11.1.1 Interpolation of recorded data . . . 212
  - 11.1.2 Artificial trajectory synthesis . . . 213
  - 11.1.3 Physics-based simulation . . . 215
  - 11.1.4 The proposed method . . . 216
  - 11.1.5 Novelties . . . 217
- 11.2 Kinematic conversion . . . 217
  - 11.2.1 Pelvis height . . . 219
  - 11.2.2 Ground clearance . . . 223
  - 11.2.3 Knee target points . . . 223
- 11.3 Results . . . 225
  - 11.3.1 Different speeds . . . 226
  - 11.3.2 Model sizes . . . 227
  - 11.3.3 Inclined walking . . . 227
  - 11.3.4 Walking frequency . . . 227
  - 11.3.5 Backward walking . . . 227
  - 11.3.6 Ground clearance . . . 228
  - 11.3.7 Torso style . . . 229
  - 11.3.8 Dragging forces . . . 229

11.3.9 Push recovery . . . 230
11.4 Discussion . . . 230
11.4.1 Closed-form solutions . . . 231
11.4.2 Limitations . . . 231
11.4.3 Applications and future work . . . 233

12 Conclusion 235
12.1 Low-level hardware complexities . . . 235
12.1.1 Sensory noise . . . 235
12.1.2 Backlashes and mechanical softness . . . 236
12.1.3 Control delays . . . 237
12.2 Hierarchical control challenges . . . 237
12.2.1 Singularities . . . 237
12.2.2 CoM height . . . 238
12.2.3 Swing dynamics . . . 238
12.2.4 Torso balancing dynamics . . . 239
12.2.5 Arm dynamics . . . 240
12.3 Walking gait analysis . . . 240
12.3.1 Missing vertical pelvis motions . . . 240
12.3.2 Footstep adjustment . . . 241
12.3.3 Gait parameters . . . 242
12.3.4 Flat walking consequences . . . 242
12.3.5 Leg lift principles . . . 243
12.3.6 Thigh-shank-foot coordinations . . . 243
12.4 Promising control aspects . . . 245
12.4.1 Capturing systematic perturbation by foot-stepping . . . 245
12.4.2 Compliance . . . 245
12.4.3 Swing dynamics . . . 246
12.4.4 Model-based control . . . 246
12.4.5 Decomposition of planning and control . . . 247
12.5 Future directions . . . 248
12.5.1 Better template models . . . 248
12.5.2 Better planning . . . 249
12.5.3 Better stabilization . . . 249
12.5.4 Better lower-body control . . . 250
12.5.5 Further biomechanic studies . . . 250
12.6 Final words . . . 250

A Force Estimation 253
A.1 Background . . . 254
A.1.1 Inverse dynamics . . . 254
A.1.2 State estimation . . . 254
A.1.3 Torque tracking . . . 255

A.1.4 Dynamics inconsistency 255
A.1.5 External force estimation 256
A.2 Methodology 258
A.2.1 First stage: acceleration estimation 259
A.2.2 Second stage: force strength estimation 260
A.2.3 Third stage: force location estimation 260
A.3 Results 263
A.3.1 First scenario: exploring different body links 263
A.3.2 Second scenario: exploring force directions 264
A.3.3 Third scenario: dynamic motion 264
A.4 Discussion 265

B Inverse Kinematics 269
B.1 Background 270
B.2 Methodology 272
B.2.1 IK_1: Error integration 273
B.2.2 IK_2: Conjugate gradient method 273
B.2.3 IK_3: Integrating errors with inequality constraints 274
B.2.4 IK_4: Direct position optimization 275
B.2.5 IK_5: Two-slack optimization 275
B.3 Results 275
B.3.1 Symmetric squatting without singularity 276
B.3.2 Symmetric squatting with singularity 277
B.3.3 Asymmetric squatting with singularity 278
B.3.4 Simulating other joint limits 278
B.3.5 Effect of boundary design 278
B.3.6 Effect of γ 279
B.3.7 How singularities improve power consumption 281
B.4 Discussion 282

C 3LP Symbolic Equations 287

D DLQR for Constrained Systems 289

E Time-Projection for Constrained Systems 293

F Time-Projection Stability Proof 295

Bibliography 322

Curriculum Vitae 323

List of Figures

1.1 Proposed control architecture . . . 3
1.2 Document structure . . . 6

2.1 Control architecture for walking simulations . . . 20
2.2 Sequence of steps planned by MPC . . . 28
2.3 CoM limit-cycles produced by MPC controller . . . 29
2.4 CoM velocity trajectories produced by MPC controller . . . 30
2.5 Push recovery and other perturbed scenarios with MPC walking controller 31

3.1 Low-level actuator control diagram . . . 39
3.2 Actual friction curves in COMAN robot . . . 40
3.3 CoP detection in each foot based on the ankle torques . . . 41
3.4 Tracking performance of the proposed low-level torque controller and high-level inverse dynamics . . . 48
3.5 State estimation precision when building kinematic chains from CoP points in each foot . . . 49
3.6 Different scenarios of compliant balance behavior with inverse dynamcis . 49

4.1 Point-Wise modeling inspired by computational chemistry methods . . . 56
4.2 Dynamic variables in point-wise modeling . . . 58
4.3 Composition of humanoid postures with point-wise modeling . . . 59
4.4 Representation of joint limits with point-wise models . . . 60
4.5 Comparison of joint-space and point-wise optimizations for posture planning 64
4.6 Sensitivity of joint-space posture planning to initial conditions . . . 65
4.7 Sensitivity of vector-based posture planning to different parameters . . . 67

5.1 Popular motion planning template models . . . 71
5.2 Point-wise model for a limb with inertia . . . 75
5.3 Flat multi-contact motions and study of swing dyanmics . . . 81
5.4 Rough-terrain multi-contact motions and study of frequency . . . 82
5.5 Rough-terrain multi-contact motions and study of accelerations . . . 83

6.1 Different template models in the literature . . . 87
6.2 3LP model variables . . . 91

6.3 3LP transition matrices over a single step phase 99
6.4 Singular values of 3LP gait matrice in passive conditions 102
6.5 Singular values of 3LP gait matrice in active conditions 104
6.6 3LP gait snapshots 106
6.7 Different 3LP gait styles 107
6.8 Sagittal vs. lateral CoM velocity in 3LP 108
6.9 3LP time-trajectories of CoM and feet 109
6.10 Dynamic trajectories of 3LP vs. Human 110

7.1 Overview of a discrete event-based walking controller 119
7.2 The effect of intermittent perturbations 120
7.3 Optimality horizon of DLQR, MPC and time-projeciton controllers 123
7.4 Overview of a continuous online walking controller 124
7.5 Conceptual Time-trajectories of open-loop system, DLQR and time-projecting controllers 129
7.6 Time-projection trajectories for a walking application 130
7.7 Push recovery performance of time-projeciton and DLQR controllers . . . 132
7.8 Speed transition performance of time-projection controller 133
7.9 Eigen-values of open-loop system, DLQR and time-projection controllers . 134
7.10 Sensitivity analysis with respect to the timing of intermittent pushes . . . 135
7.11 Viable regions of DLQR and time-projeciton controllers 137
7.12 The effect of walking frequency and speed on viable regions 137

8.1 Walking control architecture 150
8.2 Typical trajectories of joints during in-place walking gait 152
8.3 Systematic errors due to series elastic elements 153
8.4 Limited leg-lift capabilities of the real robot 154
8.5 Histograms of typical knee spring compressions 155
8.6 Different leg-lift strategies 157
8.7 The quality of filtered state errors over a task of in-place walking 158
8.8 Smooth dead-zone function 159
8.9 Leg-lift challenges during phase-transition moments 160
8.10 The effect of ankle spring compressions in early swing phases 162
8.11 Time-projeciton control look-up-tables 163
8.12 The effect of dead-zones in repeatability of in-place walking 165
8.13 Phase transition distortions without dead-zone funstions 165
8.14 Recovery from moderate pushes on the real robot 166
8.15 Recovery from a strong push on the real robot 166
8.16 Emergent walking gaits 168
8.17 Walking gait of the real robot at 0.2m/s 169
8.18 Quantificaiton of effective ankle stiffness 171

9.1 Formulations of the four cost components 181

9.2 Correlation between predicted and measured metabolic rates 187
9.3 The metabolic cost model at different walking speeds and step frequencies 188
9.4 Predictions of the metabolic cost model in different walking conditions . . 190
9.5 Comparison of experimental step frequencies and model-optimal frequencies 194
9.6 Sensitivity analysis for the energetic cost model 197

10.1 Possible sources of asymmetries in walking 201
10.2 The 3LP model and human walking experiment setup 205
10.3 Measured trunk angles in asymmetry studies 206
10.4 Comparing asymmetries of 3LP and human quantitatively 207
10.5 Asymmetric human walking postures 207
10.6 Inter-leg asymmetry from a broader viewpoint 208

11.1 Human-like walking animation components 214
11.2 3LP model adapted for troso bending and inclined walking conditions . . 218
11.3 Calculation of pelvis arcs in human-like walking animation 220
11.4 Time-trajectories of different variables in human-like walking animations . 224
11.5 Kinematic configurations of the leg segments in human walking 225
11.6 Snapshots of human walking and synthesized walking trajectories 226
11.7 Gait snapshots produced at different walking conditions 228
11.8 Different scenarios of perturbed human-like walking 229
11.9 Human-like walking animation GUI 232

A.1 Disturbance estimation control architecture 255
A.2 Principles of external force estimation 257
A.3 Calculation of force application point on different body segments 261
A.4 External force estimation in static conditions 263
A.5 Detection of forces in different directions 264
A.6 Dynamic motion scenarios for external force detection 265
A.7 External force detection during dynamic tasks 266

B.1 Symmetric non-singular squatting scenarios 277
B.2 Symmetric singular squatting scenarios 279
B.3 Asymmetric singular squatting scenarios 280
B.4 Joint limits and singular postures happening together 281
B.5 The role of joint boundaries in singular positions 282
B.6 Symmetric singular squatting scenarios on the real robot 283
B.7 Energy efficiency in singular and non-singular balancing tasks 284

F.1 Time projection for a one degree-of-freedom system. 297

List of Tables

4.1 Posture optimization dimensions 63

5.1 Motion Planning optimization performance 84

6.1 3LP variable notations 92
6.2 3LP phase varibale definitions 93
6.3 Numeric parameters for adult-size and kid-size 3LP models 102

9.1 Trends and offsets for predictions of the energetic model 189
9.2 Energetic model prediciton accuracies 198

10.1 Fitting coefficients for the analysis of asymmetry 208

B.1 Numeric values of the parameters used in inverse kinematic algorithms . . 276

1 Introduction

Different locomotion types have advantages and drawbacks in robotic applications. Wheeled robots can possibly carry heavier payloads while flying robots are limited in this sense, but they can go to very unstructured environments. Legged robots are different, however; they can carry more payloads than flying robots, and perform locomotion in more complex terrains compared to wheeled robots. Legged robots are probably less efficient, but the stepping capability enriches their locomotion skills and makes them very interesting for search and rescue applications. However, legged robots effectively possess many control complexities, e.g. the large number of degrees of freedom, sensory inputs, actuation mechanisms, under-actuation dimensions, noises, nonlinearities, geometrical and dynamical limits, mechanical softness, and more importantly, continuity or discontinuity of the motion. The floating nature of these robots imposes a great balance control challenge, though enables them to move around. Bipedal walking and stability issues make the control problem even more challenging in humanoids. Because of such control complexities, humanoids do not probably demonstrate their full locomotion capabilities yet compared to humans. Besides, the existing limited hardware technologies further reduce their efficiency and ability to carry heavy payloads.

1.1 Motivation

This project was funded by the European Walk-Man project [1] which aimed at developing loco-manipulation algorithms for search and rescue tasks. Following these objectives, the present doctoral thesis proposes better ways of controlling humanoid locomotion. We look for efficient and **robust algorithms** that enable the robot to balance and walk given many hardware limitations and task complexities. We want our algorithms to be **generic across different platforms**, fast, energy efficient and compliant. These objectives are sometimes conflicting in existing humanoid control literature, though the human locomotor system may demonstrate all these features at the same time. Healthy

---

[1] http://walk-man.eu/, last accessed in April 2018.
In addition to Biorobotics laboratory at EPFL, the consortium was composed of:
- Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Italy
- Research Centre “Enrico Piaggio”, Università di Pisa (UNIPI), Italy
- High-Performance Humanoid Technologies Lab, Karlsruhe Institute of Technology (KIT), Germany
- Center for Research in Mechatronics, Université Catholique de Louvain, Belgium

humans can easily stabilize in different walking conditions, e.g., when an extra mass is added (Browning and Kram, 2007), or when some assistive device helps (Zhang et al., 2017). They can also walk very fast and yet comply with perturbations (Vlutters et al., 2016). In this regard, the ideal control algorithm we are looking for should be human-like, though we acknowledge that in terms of mechanics, perception, and actuation, our humanoid platforms are way more limited than humans.

Many existing control algorithms in the literature are only optimized for a single robot or character (Heess et al., 2017; Mordatch et al., 2012; Lengagne et al., 2013; Faloutsos et al., 2001; Geijtenbeek et al., 2013; Geyer and Herr, 2010). When transferring the algorithm to a different platform, the entire optimization routine often needs to be repeated. Learning and optimization algorithms are very good at finding local behaviors (Geijtenbeek et al., 2013; Van der Noot et al., 2015), though hard to generalize. Besides, outside the optimized region, such controllers or pattern generators might produce highly infeasible actions due to a potential over-fitting. Instead, a vast part of the literature uses human-data for pattern generation (Tsai et al., 2010; Choi et al., 2003; Yin et al., 2007; Lee et al., 2002; Okada and Miyazaki, 2013). However, a library of different motions together with a proper trajectory adjustment algorithm might be needed.

For complex systems like bipeds, due to high dimensionality and contact switches which make the system discontinuous, it is often popular to use simplified models in control (Herdt et al., 2010b; Feng et al., 2016). These models, which are called template models, approximate overall system dynamics in certain conditions, but enable us to do much faster computations. Template models can simplify the task of motion generation, and more importantly, they enable for online computations which can stabilize the robot in case of significant perturbations. The ability to re-plan dynamically consistent trajectories online plays a key role to stabilize unstable floating-based systems like humanoids.

1.2 Approach

Among policy learning, motion-capture imitation and model-based control approaches discussed, the contribution of this work mainly falls in the third category. We might take inspiration from human trajectories to better design our controllers in some parts, but we do not use motion-capture data directly. Also, the entire thesis is based on **controlling the robot online**. We may collect data to identify models for different parts of the robot, specifically the actuators, but we avoid learning behaviors with offline optimizations. While full models provide a precise interface with the real hardware in our framework, template models enable us to predict the behavior and correct it in the form of online model predictive control. An extensive review of the existing literature has helped us better understand the advantages and limitations of such hierarchical decompositions. Among various advantages, such decomposition helps us isolate different hardware challenges and tackle them separately from the main control algorithm. Based on this philosophy, we propose a **hierarchical architecture** (shown in Figure 1.1) which allows us to define different tasks and tackle them in separate blocks:

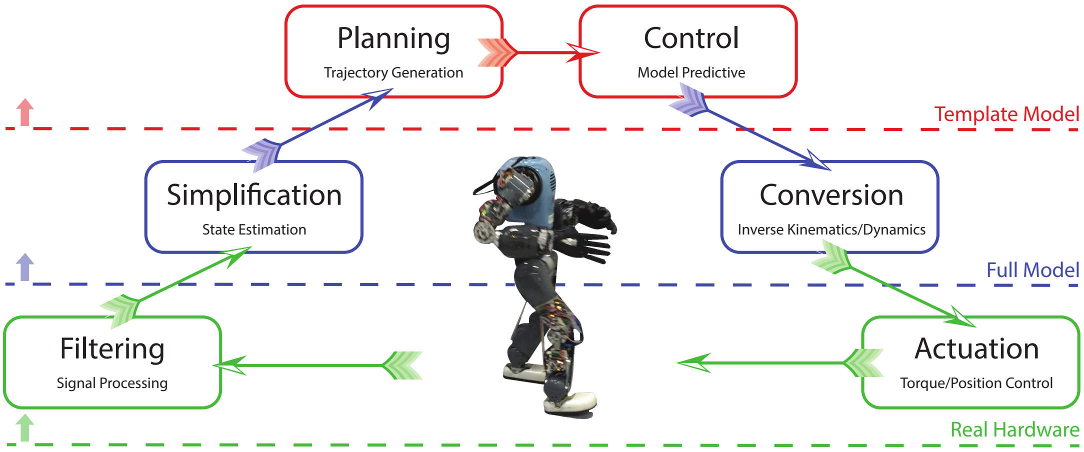


Figure 1.1 – The proposed control architecture in different model abstraction levels. We have specific blocks to deal with real hardware details, forward and inverse kinematics/dynamics algorithms to convert between the template and full-space models and eventually, template motion planning and control algorithms which can run very fast.

- State estimation: using a full forward model to fuse sensory data while handling noises, delays, and backlashes. This model allows us to extract important state variables, e.g., CoM and feet positions as well as torso orientation.
- Motion planning: we can simply think about a few important state variables and plan trajectories with template models. We do not need to consider all details, e.g., actuator dynamics, individual joints, etc.
- Stabilization: we simply optimize the future evolution of important states and find reasonable adjustments of inputs.
- Tracking: hardware details such as yaw joint angles, series-elastic elements, actuator properties, etc. are packed together and handled using full-body inverse kinematics or dynamics as well as a precise actuator control.

In other learning-based control approaches, hardware details should be included in the simulation environment to fill the reality gap and to find robust controllers (Van der Noot et al., 2015). Although many rigid-body modeling tools are commercially available (Docquier et al., 2013; Hollars et al., 1991), one probably needs to identify and include actuator dynamics, sensory noises, and communication delays as well to account for uncertainties. These effects are often hard to quantify and model. Other issues like mechanical backlashes, springs, and soft contacts are also very hard to simulate precisely. Besides, system states are easily accessible in simulations whereas obtaining such clean signals from the robot may require extensive and complicated filtering which often comes with a significant delay penalty.

Using a hierarchical approach, we can define certain control blocks and develop each separately without mixing everything in a single offline optimization. By knowing details of each block, we can build a consistent control hierarchy with minimal dependencies by concentrating on the important states of the system. We encode dynamics by full or template mechanical models, whereas other control approaches encode this information in human trajectories (Yin et al., 2007), non-linear learning kernels (Heess et al., 2017) or neural circuits (Geyer and Herr, 2010). In our approach, control principles and trajectories are easily scalable with the size of robot and the walking gait parameters. We do not need to redo offline optimizations or recollect human data for new conditions. Also, we avoid the potential over-fitting of task-specific offline optimizations by using **mechanical models and classical control** rules in our method. We can always make sure that our full models produce valid motions, and the operating regions of our template models remain easy to characterize. These models facilitate calculations and decrease sensitivity to hardware details, both in control and state estimation.

Locomotion planning and control problems are often treated separately in conventional humanoid control algorithms. Here, planning refers to the process of finding consistent open-loop trajectories, which may take arbitrarily long computations off-line. Control, on the other hand, should be done very fast online to ensure stability. There are probably three main approaches that can stabilize a walking gait; 1) Center of Pressure (CoP) modulation by ankle torques (Asano et al., 2004), 2) adjustment of next footstep locations (Koolen et al., 2012), 3) rotation of the upper-body in reaction to perturbations (Kuo, 1999). These three strategies simply stabilize the Center of Mass (CoM) by the ankle, swing hip or stance hip torques respectively. In this regard, the CoM state is probably the "principal" state to be controlled while footstep locations and torso angles follow afterward. Ankle joints can provide direct control authority to stabilize the system, but the feet in humanoids are too small to allow for capturing even moderate perturbations with this stabilization strategy. Therefore, conventional rigid combinations of off-line planning and online control are too limited. The robot is not limited, however; it can use other stabilization strategies to recover from significant perturbations like humans. One of these strategies is to continuously adjust footstep locations online which requires modification of trajectories. The new trajectories should remain consistent with system dynamics to ensure maximum stability. However, this requires expensive computations with traditional mechanical models. Focusing on this problem in this thesis, we want to link planning and control problems again and enable for **online trajectory modification** in a meaningful way. We exclusively use the **foot-stepping strategy** for stabilization and set up our control problem on this basis.

1.3 Contributions

Driven by properties of our robot COMAN, especially considerable swing dynamics, we develop better modeling formulations which achieve considerably faster computation speeds. These models are later used for posture optimization, multi-contact motion planning and walking gait generation. Our new models provide a better dynamical match with the actual system which makes them very useful for **robot control**, study of **human walking energetics**, **gait asymmetries** and **human-like animations**.

Our proposed walking model is in a sweet spot between the very simple linear inverted pendulum model and the very complex and nonlinear full-body dynamical model. With our model, we can encode much more physical effects of walking while keeping the same computational properties of simple models. In brief, the original contributions of this work are listed as:

- Combination of state of the art model predictive control, inverse dynamics, state estimation and torque tracking algorithms to achieve walking and push recovery in simulations (Faraji et al., 2014b,a) and compliant balancing on the real COMAN robot (Faraji et al., 2015).

- Development of an original way of modeling robots like molecules, used for very fast multi-contact posture (Faraji and Ijspeert, 2018a) and motion (Arreguit et al., 2018) planning. Derivation of the 3LP walking model with the same philosophy which takes into account leg swing and torso dynamics (Faraji and Ijspeert, 2017a).

- Application of the 3LP model to robot control: development of time-projection control theories to recover inter-sample disturbances (Faraji et al., 2018a), and application to the 3LP model for walking and push recovery on COMAN hardware (Faraji et al., 2018b).

- Application of the 3LP model to human walking: to estimate the metabolic cost landscape (Faraji et al., 2018c), to study gait asymmetries in inclined walking and torso bending conditions (Wu et al., 2018), and to develop a fast (micro-second) method for synthesis of human-like gaits with knee and ankle joints, suitable for walking animations (Faraji and Ijspeert, 2018b).

The next section provides detailed information about these contributions.

1.4 Thesis Outline

This document presents our contributions in four main parts: implementation of state of the art control algorithms, contributions in modeling, walking control on COMAN robot, and analysis of human walking. In the first part, we discuss state of the art control ingredients of the proposed hierarchy to achieve balancing and walking tasks. In the second part, we discuss new modeling techniques developed to overcome limitations of the state of the art algorithms with our specific robot hardware. In the third part, based on our new models, we develop new control theories and apply them to walking control on the real hardware. Finally in the last part, we present other applications of our modeling and control tools in biomechanics and animation communities. The complete structure of this thesis is depicted in Figure 1.2. In each part, we present two or three articles which describe our fundamental contributions. Extra publications which less related to walking are moved to appendix. The following list summarizes our contributions with their specific applications.

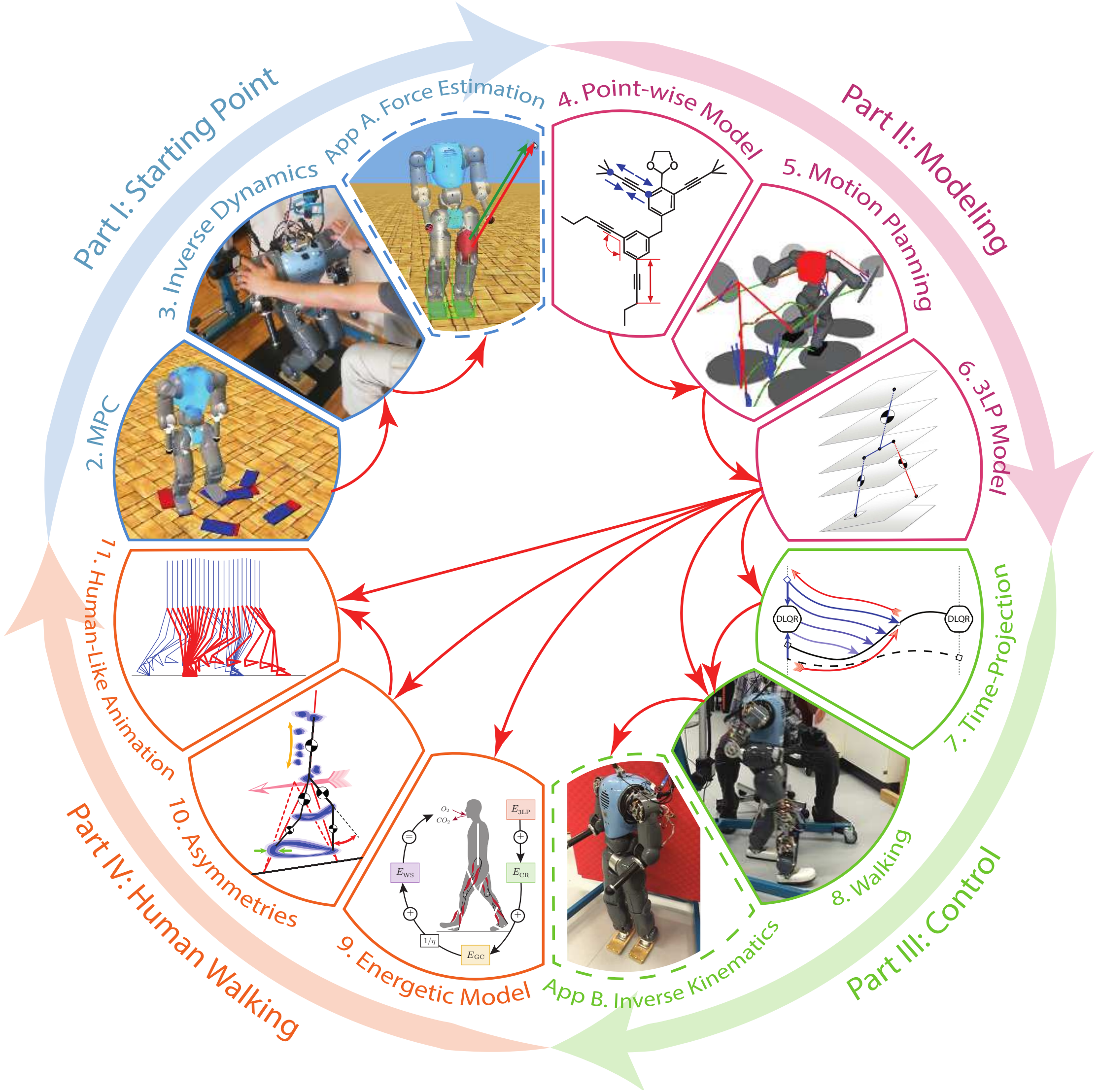


Figure 1.2 – Structure of this document. We have four different parts to present contributions in modeling, control, hardware experiments and extended application domains. In each part, we have three main publications which present methodologies of different components in our control architecture. The red arrows in the middle show dependencies of these contributions while dashed blocks are moved to appendix as less related to walking directly.

Publication Note: The material presented in the summary list is adopted from abstracts of the publications mentioned in each item.

Part I: Starting Point

- **Chapter 2: Model Predictive Control** for walking and push recovery (Faraji et al., 2014a)
  In this chapter, we formulate a novel hierarchical controller for walking applications on torque-controlled humanoid robots. Our method is based on task-space inverse dynamics which generates joint torques, given Cartesian accelerations of end-effectors (feet and CoM). Our implementation is in the form of a very efficient quadratic optimization that can be solved in less than a millisecond. Using the Linear Inverted Pendulum (LIP) model (Kajita and Tani, 1991), we derive a formulation where footstep locations serve as discrete inputs. We then use an online Model Predictive Control (MPC) to plan optimal footsteps, given desired sagittal and steering velocities determined by the user. Over various simulations in this chapter, this method shows robustness against external pushes, sensory noises, model errors and delayed communications in the control loop. Besides, it can perform blind walking over slopes and uneven terrains, and turn rapidly at the same time.

- **Chapter 3: Inverse Dynamics** for performing task-space motions (Faraji et al., 2015)
  Although task-space inverse dynamics can improve tracking and compliance in dynamic tasks, it requires the internal and global states of the system, a precise torque control, and a proper model. In this chapter, we discuss practical issues to implement inverse dynamics on a torque-controlled robot. By identifying models for the electrical actuators, inverting such models and estimating the friction online, a high bandwidth torque controller is achieved. Also, a cascade of optimization problems fuses all the sensory data coming from IMU, joint encoders and contact force sensors, and estimates the global states robustly. Our state estimation builds the kinematic chain of each leg from the center of pressure which is more robust in case of rotational slippages in the foot. Thanks to a precise and fast torque control, robust state estimation and optimization-based whole-body inverse dynamics, the real robot can keep balance with a very small stiffness and damping in the Cartesian space. It can also recover from strong pushes and perform dexterous tasks. Such compliant and stable performance is based on a pure torque control, without any joint damping or position/velocity tracking.

- **Appendix A: Force Estimation** (Faraji and Ijspeert, 2016)
  We propose a novel method to estimate external forces applied to a compliantly balancing robot in simulations. Using a cascade of different optimizations, we estimate global states of the robot by fusing the sensory data. This includes estimation of accelerations in addition to positions and velocities. We then use similar motion equations used in the inverse dynamics controller to find mismatches in the available sensory data and associate them with an unknown source of external force. Then, by decomposing Jacobians, we search over all body segments to find

the force application point. By approximating segment volumes with ellipsoids, we can derive analytic solutions to solve the search problem very fast in real time. The proposed approach is tested on a complex humanoid robot in simulations and outperforms static estimators over fast dynamic motions. We foresee many applications for this method especially in human-robot interactions where it can serve as a whole body virtual suit of tactile sensors. It can also be very useful in identifying inertial properties of the objects being manipulated or mounted on the robot like a backpack.

Part II: Contributions in Modeling

- **Chapter 4: Point-Wise Model** for posture planning applications (Faraji and Ijspeert, 2018a)
  Traditional joint-space models used to describe equations of motion for humanoid robots offer nice properties linked directly to the way these robots are built. However, from a computational point of view and convergence properties, these models are not the fastest when used in planning optimizations. In this chapter, inspired by Cartesian coordinates used to model molecular structures, we propose a new modeling technique for humanoid robots. We represent robot segments by vectors and derive equations of motion for the full body. Using this methodology in a complex task of multi-contact posture planning with minimal joint torques, we set up optimization problems and analyze the performance. We demonstrate that compared to joint-space models that get trapped in local minima, the proposed point-wise model offers a much faster computational speed and a possibly suboptimal but unique final solution. The underlying principle lies in reducing the nonlinearity and exploiting the sparsity. Apart from the specific case study of posture optimization presented in this chapter, these principles can make the proposed technique a promising candidate for many other optimization-based complex tasks in robotics.

- **Chapter 5: Motion Planning** in multi-contact scenarios (Arreguit et al., 2018)
  We present a new method for multi-contact motion planning which efficiently encodes internal dynamics of the robot without needing to use full models. Our approach is based on a five-mass model which is formulated by Cartesian points instead of joint angles. We solve direct optimization problems which include distance constraints between these points, Newtonian equations and integration constraints. We consider a given rhythm of contact switches but leave the phase-timings and contact positions free inside the optimization to provide more flexibility. Due to simpler equations and sparser problem structures, we can achieve very short optimization times in the order of few hundred milliseconds, which make the method suitable for application of online model predictive control. Aside from contact position and time adjustment properties, we can include precise foothold regions and synthesize dynamic motions by taking internal dynamics and momentums into account.

- **Chapter 6: 3LP Model** to study and model walking dynamics (Faraji and Ijspeert, 2017a)
  In this chapter, we present a new mechanical model for biped locomotion, composed of three linear pendulums (one per leg and one for the whole upper body) to describe stance, swing, and torso balancing dynamics. The methodology used to derive 3LP equations is similar to the previous point-wise model, but with linearizing assumptions. In addition to a double support phase, this model has different actuation possibilities in the swing hip and stance ankle which produce a broad range of walking gaits. Without numerical time-integration, closed-form solutions of the 3LP model help to find periodic gaits which could simply scale in certain dimensions to modulate the motion. Thanks to the linear properties, this model can provide a computationally fast platform for model predictive controllers to predict the future and consider meaningful inequality constraints to ensure feasibility of the motion. Such property comes from a description of limb dynamics with joint torques directly and therefore, reflecting hardware limitations more precisely, even in the very abstract template space. The proposed model produces human-like torque and ground reaction force profiles, and thus, compared to point-mass models, it is more promising for generation of dynamic walking trajectories. Despite being linear and lacking many features of human walking like CoM vertical excursions, knee flexions, and ground clearance, we show that the proposed model can explain one of the main optimality trends in human walking, i.e., the nonlinear speed-frequency relationship. In this chapter, we mainly focus on describing the model and its capabilities, comparing it with human data and calculating optimal human gait variables.

Part III: Contributions in Control

- **Chapter 7: Time-Projection Control** for walking and push recovery (Faraji et al., 2018a)
  We present a new control paradigm in which inter-sample errors are projected to sampling events where a discrete LQR (DLQR) controller suggests corrections. These input corrections are then projected back to the inter-sample time and applied to the system. The projecting controller reacts to disturbances immediately and compared to the DLQR controller, it provides superior performance in recovering intermittent perturbations. Further analysis of closed-loop eigenvalues and disturbance rejection strength show strong stabilization properties for this architecture. An analysis of viable regions also shows that the proposed controller covers most of the maximal viable set of states. Although it does not consider inequality constraints, it can perform similarly to MPC most of the time. It is computationally much faster than MPC and yet optimal over an infinite horizon.

- **Chapter 8: Walking and Push Recovery** which demonstrates a compliant behavior (Faraji et al., 2018b)
  In this chapter, we present a simple control framework for on-line push recovery

with dynamic foot-stepping strategy. Due to relatively heavy legs in our robot, we need to take swing dynamics into account and thus use the 3LP model. We formulate discrete LQR controllers and use the time-projection method to adjust the next footstep location on-line during the motion. This adjustment, which is found based on both pelvis and swing foot tracking errors, naturally takes the swing dynamics into account. Suggested adjustments are added to the Cartesian 3LP gaits and converted to joint-space via inverse kinematics. Fixed and adaptive foot lift strategies are also used to ensure enough ground clearance in perturbed walking conditions. The proposed architecture is robust, yet uses very simple state estimation and basic position tracking. We rely on physical springs in the robot to absorb impacts while introducing simple laws to compensate their tracking bias. Extensive hardware experiments demonstrate the functionality of different control blocks in this chapter and prove the effectiveness of time-projection in extreme push recovery scenarios. We also show self-produced and emergent walking gaits when the robot is subject to continuous dragging forces. These gaits feature dynamic walking robustness due to relatively soft springs in the ankles, and we avoid any active Zero Moment Point (ZMP) control in our proposed architecture.

- **Appendix B: Inverse Kinematics** algorithm to handle joint limits and singularities (Faraji and Ijspeert, 2017b)
  We propose a nonlinear inverse kinematics formulation which solves for joint positions directly. Compared to various other popular methods that integrate velocities, this formulation can better handle asymmetric and singular-postured balancing tasks for humanoid robots. We introduce joint position and velocity boundaries as inequality constraints in the optimization to ensure feasibility. Such boundaries provide safety when approaching or getting away from joint limits or singularities. Besides, mixing positions and velocities in our proposed algorithm facilitates the recovery from singularities, which is delayed in conventional inverse kinematics methods. Extensive demonstrations on the real hardware prove effectiveness and energy efficiency of the proposed algorithm. Our formulation automatically handles different numerical and behavioral difficulties rising from singularities, which makes it a reliable conversion block for different Cartesian planners.

Part IV: Application to Human Walking

- **Chapter 9: Energetic Model** which provides biomechanical predictions (Faraji et al., 2018c)
  Since the advent of energy measurement devices, gait experiments have shown that energetic economy has a significant influence on human walking behavior. However, few cost models have attempted to capture the major energy components under different walking conditions. Here we present a simple but unified model that uses walking mechanics to estimate the metabolic cost at different speeds and step lengths, and for six other biomechanically relevant gait experiments in literature. This includes at various gait postures (e.g., extra foot lift), anthropometric dimensions

(e.g., added mass), and reduced gravity conditions, without the need for parameter tuning to design new gait trajectories. Our results suggest that the metabolic cost of walking can be explained by a linear combination of four costs: swing and torso dynamics, CoM velocity redirection, ground clearance, and body weight support. The overall energetic cost is a trade-off among these separable components, shaped by how they manifest under different walking conditions.

- **Chapter 10: Walking Asymmetries** and study of posture consequences (Wu et al., 2018)
  The trunk can serve many purposes in walking, from carrying objects to avoiding overhead obstacles. As the heaviest body segment, changes in trunk kinematics can have an important mechanical impact on whole-body control and configuration. To understand the role of trunk flexion, we investigated the changes in inter-leg asymmetry by asking five subjects to walk at three different speeds, three inclination angles, and four trunk angles (three imposed, one self-selected). Asymmetry was defined as the difference between the angle of the trailing leg and the angle of the leading leg relative to a vector orthogonal to the walking surface during double support. We determined the relative contribution of speed, inclination angle, and trunk angle on asymmetry from empirical data and compared these results with 3LP, a linear model composed of three pendulums which represent trunk and the two legs. We found that trunk lean has the opposite effect on asymmetry, compared with the other two factors. Thus trunk lean may be used to compensate for asymmetries that may be unpleasant for humans or induce mechanically negative effects.

- **Chapter 11: Human-Like Animation** of different gaits (Faraji and Ijspeert, 2018b)
  We present a new framework to generate human-like lower-limb trajectories in periodic and non-periodic walking conditions. In our method, dynamics is encoded in the 3LP model, and we use the time-projecting controller to stabilize the motion. On top of gait generation and stabilization in the template space, we introduce a kinematic conversion method that synthesizes more human-like trajectories by combining geometric variables of the 3LP model adaptively. Without any tuning, numerical optimization or offline data, our human-like walking gaits are scalable with respect to body properties and gait parameters. We can change various parameters such as body mass and height, walking direction, speed, frequency, double support time, torso style, ground clearance and terrain inclination. We can also simulate the effect of constant external dragging forces or momentary perturbations. The proposed framework offers closed-form solutions in all the three stages which enable simulation speeds orders of magnitude faster than real time. This can be used for video games and animations on portable electronic devices with limited power. It also gives insights for generation of more human-like walking gaits with humanoid robots.

1.5 Brief Overview

In different parts of the thesis, although we might switch between different controllers, e.g., between MPC and time-projection or between inverse kinematics and inverse dynamics, we keep the same control hierarchy demonstrated in Figure 1.1. The order of chapters is roughly chronological although publication dates might sometimes be delayed. The following list describes chronological relations of different chapters together.

- We started with inverse dynamics and MPC control, and both were developed earlier in previous works (Faraji et al., 2013, 2014b), but adapted to the robot available in our laboratory and presented in chapter 2.
- Later, we developed hardware interfaces (state estimation and torque control) which enabled us to do compliant balance, described in chapter 3.
- In parallel, we developed the point-wise model targeting multi-contact posture optimization and motion planning tasks described in chapters 4 and 5.
- Since our inverse dynamics was not precise enough to compensate swing dynamics on the real hardware, we took the point-wise modeling technique and developed the 3LP model, hoping to produce better-matching Cartesian trajectories for the end-effectors (CoM and feet). The 3LP model encodes falling, swing and torso balancing dynamics as well as lateral bounces described in chapter 6. The well-known LIP model does not produce all these features together by default.
- In the range of motions supported by the robot, the inequality constraints initially encoded in our MPC footstep planner were not activated most of the time. Therefore, we developed the time-projection control (described in chapter 7) which somehow serves as closed-form solutions for MPC without constraints, but with similar online reaction properties.
- The 3LP model and time projection control were used to perform walking with the real hardware, described in chapter 8.
- In parallel, the 3LP model was used in biomechanics studies of walking energetics and geometric asymmetries, described in chapters 9 and 10. This part of the thesis was done in collaboration with a biomechanist colleague in the lab, Dr. Amy Wu. She had an in-depth knowledge of human locomotion and helped us better identify different aspects of our modeling and control tools.
- Finally, we used the 3LP model and time-projection control to produce human-like motions with a novel inverse kinematics method, described in chapter 11. This part is inspired by previous biomechanics studies, and hopefully forms the basis for faster and more human-like walking gaits on the humanoid robot.

Our research has various novel aspects in terms of modeling, control, integration and inter-disciplinary research. We give an equal weighting to different parts, though the most important chapters are probably the 3LP model (chapter 6), compliant balance control (chapter 3), time-projection control (chapter 7), the energetic model (chapter 9), and human-like motion generation (chapter 11). These chapters probably represent the most novel contributions of this thesis.

Starting Point Part I

2 Model Predictive Control

We start this thesis by introducing our first walking controller. The material presented in this chapter is composed of two parts: inverse dynamics controller and Model Predictive Control (MPC) for footstep adjustment. This method is, in fact, an extension of author's master thesis published in (Faraji et al., 2014b) by adding steering features and adapting the method to COMAN robot. Referring back to Figure 1.1, we use the linear inverted pendulum model in template space and formulate a MPC controller which can produce footstep locations according to reference sagittal and steering velocities given. This controller is also able to recover from pushes by adjusting the step locations online, i.e., producing a dynamic walking behavior. Appropriate turning trajectories are also generated in the Cartesian space and given to the inverse dynamics controller. This chapter also characterizes stability in various perturbed walking conditions, especially on uneven terrain and slopes. Our former publication (Faraji et al., 2014b) further extends the analysis to the case of modeling errors, sensory noises and control delays in all which the method shows convincing robustness to some extent. This suggests that a hierarchical controller would be a good solution for walking as long as a robust low-level controller like inverse dynamics can fill the reality gap between the template model and the complex robot. The primary technical challenge we overcome in this chapter is to implement numerical optimizations of MPC and inverse dynamics efficiently to be solved in 1-2ms [1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji, Soha Pouya, and Auke Jan Ijspeert. *"Robust and agile 3D biped walking with steering capability using a footstep predictive approach."* In Proceedings of Robotics: Science and Systems (RSS), Berkeley, USA, 2014.

The role of first author was to develop the method and to write the manuscript. The second author supervised this work and the previous version (Faraji et al., 2014b) which was done on Atlas robot during a master project.

---

[1] All the videos of this chapter could be found at https://youtu.be/zbSqH1csvsQ

2.1 Background

Legged robots and especially bipeds are difficult to control as one needs to maintain stability while performing desired tasks. Although quadruped robots have some intrinsic stability, the case is more restricted for bipeds or humanoids since the support region for Center of Mass (CoM) is smaller. A traditional way to control the robot is to keep the CoM inside the support region all the time while taking steps (statically-stable walking). However, this produces un-natural lateral bounces and non-smooth sagittal motions compared to humans. One can control the Zero Moment Point (ZMP) (Vukobratović and Borovac, 2004) instead to allow the CoM move more freely while maintaining a dynamic stability. Keeping ZMP inside the support region will prevent the feet from tilting or rolling when the COM is outside.

Various methods are introduced in literature to perform locomotion by modulating ZMP. Based on Inverted Pendulum Model (IPM) (Kajita and Tani, 1991), one can produce desired CoM motions to be tracked by the ankle torques ((Choi et al., 2004; Englsberger et al., 2011)). More complicated forms of the IPM which assume inertia for the base are also used in literature to improve walking stability. In (Pratt et al., 2012), the inertia is used to rest after taking a step while in (Whitman, 2013), it compensates swing dynamics. Inverted pendulum models simplify dynamics, aiming at predicting the motion with a lower computational cost compared to using the full model of the system. An example of exploiting such simplification is (Koolen et al., 2012) where the concept of capture point is introduced. This point indicates an immediate footstep location which brings the robot to rest condition.

Planning locomotion and performing low-level joint control of the robot are two related topics. Using Jacobians, one can translate Cartesian variables into joint variables and vice-versa. Virtual Model Control method ((Pratt et al., 2001)) is an example of translating forces while various methods translate velocities as well (Choi et al., 2004) or integrate them to obtain joint positions. However, when we target agile and versatile locomotion, incorporating the knowledge of dynamics in control helps a lot. While Jacobians merely provide information about the geometry of the robot, dynamic models can predict required joint torques to realize desired motions. Without this knowledge, one should rely on the performance of individual joint controllers and their tracking to determine the overall behavior. Using high gains for better tracking typically leads to a stiff behavior which could be harmful to the environment and to the robot itself, especially in case of impacts at each step. Besides, stability, compliance, and accuracy depend on the tuning of various parameters, the posture of the robot and the desired speed. Thus, one prefers to use softer feedback gains to reject perturbations compliantly and rely more on feed-forward dynamics to avoid high stiffness.

Inspired from operational space formulations of Khatib (Khatib, 1987) and unified formulations of Aghili (Aghili, 2005), inverse-dynamics methods have been widely used on humanoid robots recently either using joint-space trajectories like (Righetti et al., 2013) or Cartesian trajectories like (Whitman and Atkeson, 2010; Stephens and Atkeson, 2010a). In such formulations, we can optimize joint torques and constraint forces, given desired joint-space or Cartesian accelerations. Although equality constraints ensure tracking of contact accelerations, in the closed-form inverse-dynamics method (Righetti et al., 2013), we cannot include inequality constraints for the Center of Pressure (CoP), friction cones or joint torque limits easily. This motivates solving a quadratic constrained problem using fast QP solvers where we can consider all constraints at the same time ((Herzog et al., 2013), (Whitman and Atkeson, 2010), (Whitman, 2013) and (Kuindersma et al., 2014)). With such formulations, we track the desired trajectories of end-effectors with simple PD controllers while being compliant, maintaining balance and satisfying all constraints.

In this chapter, we propose a hierarchical control architecture for locomotion of torque-controlled humanoids similar to our previous work (Faraji et al., 2014b), but with steering capabilities. The method is composed of three layers:

1. Whole body optimization: generates joint torques given Cartesian accelerations of the feet and the CoM.
2. Trajectory pattern generator: produces and tracks task-space trajectories for feet and CoM, given footstep positions and orientations. The outputs of this layer are Cartesian accelerations, to be given to the previous layer.
3. Footstep planner: produces footstep patterns, given desired sagittal, lateral and turning speeds. The first footstep position is then given to the previous layer.

The first layer is described in section 2.2 where we formulate a quadratic problem to generate joint torques. The features of our formulation and comparison with other implementations will be presented, however, the main contribution of this work is not at this level. In section 2.3, we describe our smooth Cartesian trajectory generation and tracking policies of the second layer. These trajectories are defined between the initial and final footstep positions in the current step phase. The latter is determined by the third layer introduced in section 2.4. This layer optimizes future footsteps according to the LIP model and a reference footstep plan, formed based on Cartesian velocities that the user determines with a joystick. The optimization in this layer is written as a Model Predictive Control (MPC) problem (Kothare et al., 1996) over future footsteps. Figure 2.1 visualizes our control layers and the flow of information between them.

The novelty of this work is mainly in the third layer where we plan future footsteps in a discrete space. In (Kuindersma et al., 2014), optimal trajectories are found offline based on (Posa et al., 2014). Using these trajectories, the method proposed in (Kuindersma et al., 2014) finds optimal CoP trajectories that satisfy frictional and torque limitations. A linear Time Variant LQR then finds CoM accelerations over an arbitrary horizon, which are then given to a QP solver to generate joint torques. The major difference of (Kuindersma et al., 2014) with our method is the fact that we do not have any offline optimization. We rather optimize next footstep positions online, considering the constraints mentioned and a reference footstep pattern defined in closed-form. Moreover, although the time-variant LQR in (Kuindersma et al., 2014) allows a change in CoM

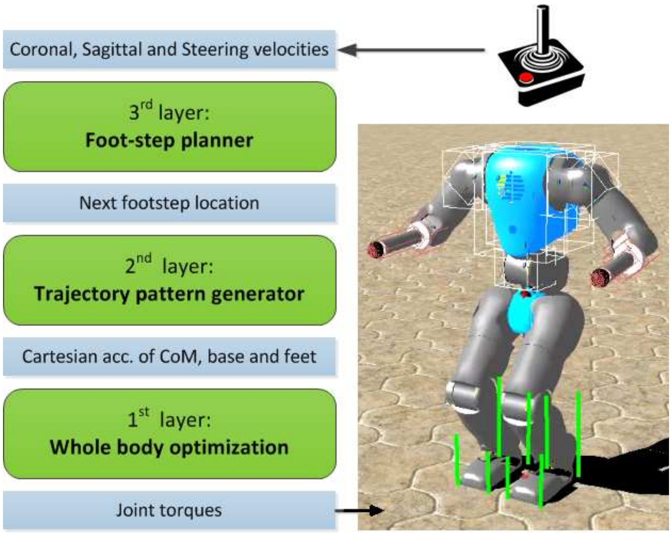

Coronal, Sagittal and Steering velocities
3rd layer:
Foot-step planner
Next footstep location
2nd layer:
Trajectory pattern generator
Cartesian acc. of CoM, base and feet
1st layer:
Whole body optimization
Joint torques

Figure 2.1 – In this figure, the proposed hierarchical control architecture is shown together with a picture of the real robot. We have three different layers which receive simple commands from the user and generate joint torques for the robot.

height, keeping the CoM height constant has a minimal influence on balance performance in their robot. Comparatively, we fix the height and obtain a simple closed-form discrete linear system describing future step phases. In (Kuindersma et al., 2014), authors incorporate a simple terrain estimation algorithm to determine the desired CoM height profile. However, we will show that our blind method with constant CoM height is notably robust against moderate terrain variations. We try to reject such disturbances by taking proper footsteps, rather than modulating the CoP and relying on ankle torques (Kuindersma et al., 2014; Englsberger et al., 2011; Stephens and Atkeson, 2010b).

While push recovery is done in the literature over a single step (Stephens and Atkeson, 2010b) or multiple steps (Koolen et al., 2012), we aim at a more generic scenario, i.e., bringing the system to a desired non-zero velocity (given by joystick). The idea of capture points in (Koolen et al., 2012) is similar regarding the formulation future footsteps in closed-form by simplifying the system with LIP. However, we consider the CoM position and speed separately rather than combining them in a single quantity. Our approach can also be applied on Spring Loaded Inverted Pendulum (SLIP) as a simplified model of running. Mordatch et al. in (Mordatch et al., 2010) predict up to four future half cycles combining single support, double support and ballistic motion phases with nonlinear formulations of the SLIP model. Although it can perform walking and running on various terrains, the planning itself takes considerable time, making the robot not responsive to strong perturbations such as pushes. Our reactive footstep planner takes less than 0.2ms instead to solve the problem.

After introducing the three control layers, we will shortly discuss the simulation platform and the COMAN robot (Spyrakos-Papastavridis et al., 2013) used to validate our controller. Simulation results over a wide range of forward and steering speeds is then presented in the same section. We will also test our method against various perturbations, either internally like noise or model errors or externally like pushes or terrain variations.

2.2 Whole body optimization

In this section, we present our QP problem formulation used in low-level joint control. Similar to several previous works (Whitman and Atkeson, 2010; Stephens and Atkeson, 2010b,a; Herzog et al., 2013), the objective function has a quadratic form while constraints include the Equation of Motion (describing dynamics of the robot), end-effector tasks, CoP regions and friction polyhedra. The difference of our implementation, however, is that we solve the problem by including the joint torque limits in the constraints. This is in contrast to Dynamic Balance Force Control (DBFC) methods (Stephens and Atkeson, 2010a,b) which divide the problem into two levels: (i) finding optimal contact force distributions and then (II) finding joint torques via pseudo-inversion and treating the torque limits by saturation.

To formulate our problem, we first define the state vector as $q = \left[p_b, o_b, q_j\right]^{\mathrm{T}}$, where $p_b \in \mathbb{R}^3$ and $o_b \in \mathbb{R}^4$ represent position and quaternion orientation of the pelvis frame and $q_j \in \mathbb{R}^n$ represents $n = 23$ joint angles. Throughout this chapter, we use a convention to show positions by p, quaternions by o and both of them together with x. We update the global variables p_b and o_b by an internal odometry method based on IMU and joint sensors. We write the Equation of Motion (EoM) for this rigid body as:

$$\begin{aligned} M(q)\ddot{q} + h(q, \dot{q}) &= \tau + J_C^{\mathrm{T}}(q)F \\ J'_C(q)\ddot{q} + \dot{J}'_C(q)\dot{q} &= \ddot{x}_C \end{aligned} \tag{2.1}$$

Where $M(q) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the inertia matrix, $h(q, \dot{q}) \in \mathbb{R}^{n+6}$ represents the floating base centripetal, Coriolis and gravity forces, $\tau = \left[0 \in \mathbb{R}^6, \tau_j \in \mathbb{R}^{\mathrm{n}}\right]^T$ is the vector of actuated joint torques, $J_C(q) \in \mathbb{R}^{k\times(n+6)}$ is the Jacobian of k linearly independent constraints and $F \in \mathbb{R}^k$ is the vector of k constraint forces. Note that there is no need to consider the 4th element of the quaternion vector o_b in derivations. The variable $\ddot{x}_C \in \mathbb{R}^{\mathrm{k}'}$ denotes the Cartesian translational and rotational accelerations of the controlled points on the robot's body (namely end-effectors and the COM). Here k' is the total number of constraint equations introduced by these points. Note that $J'_C(q) \in \mathbb{R}^{\mathrm{k}'\times(\mathrm{n}+6)}$ corresponds to all controlled Cartesian points being in contact or moving freely and thus, $J_C(q)$ is a sub-matrix of $J'_C(q)$. For the points being in contact and fixed, the corresponding $\ddot{x}_C$ is zero while for the floating points, this reference acceleration is determined by the second layer of our controller. Thus for walking, k' is always 12 (as we have two feet, each introducing 6 constraint equations). The parameter k (number of contacting points) is either 6 in single support or 12 in double support phase.

Given the Cartesian accelerations ($\ddot{p}_{\mathrm{CoM}}$, $\ddot{o}_b$ and $\ddot{x}_C$), we use a quadratic program as formulated in (2.2) to minimize $\ddot{q}$, τ and F under various physical constraints.

$$
\begin{aligned}
\min_{\ddot{q},\tau_j,F,\sigma} V_{Q_q}(\ddot{q}-\ddot{q}_d) + V_{Q_\tau}(\tau) + V_{Q_F}(F) + V_{Q_\sigma}(\sigma)& \\
M\ddot{q} + h = \tau + J_C^{\mathrm{T}} F& \\
\ddot{x}_C = J'_c\ddot{q} + \dot{J}'_c\dot{q}& \\
\sum F_i = m\ddot{p}_{\mathrm{CoM}} + \sigma& \\
A\left[\tau_j^{\mathrm{T}}, F^{\mathrm{T}}\right]^{\mathrm{T}} \leq B&
\end{aligned} \tag{2.2}
$$

Where $V_Q(v) = v^T Q v$, the variable m is robot's mass, the matrices Q_i are diagonal quadratic costs and σ induces a soft constraint on CoM dynamics. We encapsulate coefficients introduced by physical inequality constraints into A and B for conciseness. The joint torques are bounded to certain values according to actuator specifications. CoP regions limit the moment available at each contact (refer to (Whitman and Atkeson, 2010) and (Whitman, 2013)). We also include friction polyhedra at each contact similar to (Kuindersma et al., 2014). In (2.2), $\ddot{q}_d$ is set to zero except for the base rotational acceleration which is $\ddot{o}_b$. In fact, we control the base orientation directly, while the method in (Herzog et al., 2013) controls it by regulation of the total angular momentum rate. Note that the equality constraint for $\ddot{p}_{\mathrm{CoM}}$ in (2.2) is a simpler alternative of using CoM Jacobian for relating $\ddot{p}_{\mathrm{CoM}}$ to joint accelerations which makes the optimization slower.

With such problem definition, robot's dynamics, feet accelerations and inequality constraints are defined as hard constraints. However, Cartesian accelerations determined by the second layer of our controller are followed by soft constraints with large quadratic costs. If the second layer gives infeasible reference accelerations regarding available frictions or torque limits, our QP finds the closest solution without violating the hard constraints. A similar approach is taken in (Kuindersma et al., 2014) where the second equality constraint (contact accelerations) is soft. In their formulation, this soft constraint has the same effects on infeasible $\ddot{x}_C$. However in our formulation, the softness on CoM and base orientation accelerations deal with infeasible task accelerations $\ddot{x}_C$ as well. We consider larger quadratic costs for the CoM compared to the base orientation. As a result, the Cartesian tracking of the CoM and feet is more precise with the cost of small variations mainly on base orientation, which happen in case of large perturbations and swing dynamics. Note that this tracking is more important for us compared to the robot's posture since we want our full robot to match with the LIP used for planning.

It is worth mentioning that for walking, we keep the upper body joints fixed, i.e., assuming zero joint accelerations. Stiff PD position controllers realize this assumption. The quadratic costs in (2.2) determine our weightings for torques vs. contact forces and accelerations. Our robot is fully actuated in the single support phase, considering 6 degrees of freedom in each leg and the number of constraints. There exists a unique solution in the QP problem according to the analysis provided in (Aghili, 2005) and (Mistry and Righetti, 2012). Note however that the robot is under-actuated in the double support phase and our force distribution weighting between the two feet in this phase follows a similar policy described in (Whitman and Atkeson, 2010). We give larger weights to the foot closer to the CoM. The double support phase only happens shortly before starting the rhythmic stepping, described in the next section.

Using the well known QP solver CVXGEN (Mattingley and Boyd, 2012), we can solve the whole problem including all joints in less than 1.2ms (6 iterations on average) on a Core i5 1.7GHz machine, coded in C++. With fixed upper body assumption, however, there is no point in considering the corresponding elements in the mass matrix and Jacobians, since accelerations are zero. Therefore, we avoid defining such sparse matrices and instead break them into several blocks in CVXGEN which reduce the problem size drastically. These fixed joints are around 50% of all joints in most of the humanoid robots including COMAN. So, we can decrease the optimization time down to around 0.7ms in total, still being able to calculate gravity compensation torques for the fixed joints. For closed-form solutions, however, one needs to reformulate the alternative pseudo-inversion formula in (Mistry and Righetti, 2012) to make it efficient. Note that the active-set based QP solver in (Kuindersma et al., 2014) uses the solution found previously to speed up convergence and has a superior average performance, compared to the QP solver of the present work.

In this section, we presented how we calculate the joint torques, given Cartesian accelerations of the controlled points, i.e., CoM, base orientation and feet. We also described strengths of the proposed problem formulation, tailored to our objectives in the upper layers of the controller. In next section, we explain how these Cartesian accelerations are generated in the second layer.

2.3 Trajectory pattern generator

In this section, we explain the policy we use to generate gait trajectories as well as the stepping rhythm. In this layer, a state machine with fixed timing switches between left and right support phases. There is only a short double support phase in the beginning when the robot starts from the normal posture and shifts left or right to start a rhythmic stepping properly. The input to this layer of our architecture is the next footstep position at the end of each swing phase. This layer is responsible for generating Cartesian accelerations for the CoM, base orientation and the feet.

As will be described later in the third layer, we assume that the robot follows a simple foot-less LIP model in the left or right support phases which induces no ankle torques. In our method, ankles mainly control the posture and compensate swing dynamics rather than inducing motions on the CoM. This assumption is the key factor in making our blind robot robust against terrain variations. However, to enhance accuracy, we take into account the actual non-flat orientations of the feet in calculating friction polyhedra or CoP constraints of (2.2). In the LIP model, the x and y components of the variable $\ddot{p}_{\text{CoM}}$ are determined by:

$$\ddot{p}_{\text{CoM}} = \frac{\mathrm{g}}{\mathrm{z}_0}(p_{\text{CoM}} - p_{\text{base}}) \tag{2.3}$$

Where z_0 is the reference constant CoM height and p_{base} is the center location of the stance foot. Note that the z component of $\ddot{p}_{\text{CoM}}$ is determined by a PD feedback, tracking the constant CoM height reference z_0.

In order to investigate the coupling of x, y and yaw motions, from (Koolen et al., 2012) one can write equations of motion for a full inverted pendulum with ankles and inertia by:

$$\begin{aligned} m\ddot{p}_{\text{CoM}} &= f + mg \\ I\dot{\omega} &= \tau_{\text{hip}} - \omega \times (I\omega) \end{aligned} \tag{2.4}$$

Where I is the moment of inertia and ω is the angular velocity of the body. Moment-balance equations for this model are:

$$-(p_{\text{CoM}} - p_{\text{base}}) \times f - \tau_{\text{hip}} + \tau_{\text{ankle}} = 0 \tag{2.5}$$

Substituting τ_{hip} and f from (2.4) to (2.5), we obtain:

$$(p_{\text{CoM}} - p_{\text{base}}) \times (\text{m}g - \text{m}\ddot{p}_{\text{CoM}}) + \tau_{\text{ankle}} = I\dot{\omega} + \omega \times (I\omega) \tag{2.6}$$

Note that x and y components of $\ddot{p}_{\text{CoM}}$ are in proportion to those components in $(p_{\text{CoM}} - p_{\text{base}})$, described in (2.3). So the cross product on the left has no z component. One can easily prove that if ω and $\dot{\omega}$ are zero, τ_{ankle} will also be zero which is equivalent to the basic ankle-less and inertia-less LIP used in (2.3).

For the complex model of (2.6), however, we should make sure that τ_{ankle} will compensate all the right-hand-side moments so that our prediction which is based on a simple LIP stays valid. It is difficult to find bounds for $\dot{\omega}$ since I is time variant and depends on the posture of the robot. Assuming a diagonal I however, one can write the approximate validity condition as:

$$| \begin{bmatrix} I_{xx}\dot{\omega_x} + (I_{zz} - I_{yy})\omega_z\omega_y \\ I_{yy}\dot{\omega_y} + (I_{xx} - I_{zz})\omega_x\omega_z \\ I_{zz}\dot{\omega_z} + (I_{yy} - I_{xx})\omega_y\omega_x \end{bmatrix} | \leq \begin{bmatrix} w_{\text{foot},y}/2 \\ w_{\text{foot},x}/2 \\ \mu_{\text{rot}} \end{bmatrix} |mg| \tag{2.7}$$

Where $w_{\text{foot},x}$ and $w_{\text{foot},y}$ are foot sizes in x and y directions and μ_{rot} is the rotational friction coefficient. If x and y components of ω are negligible as well as the non-diagonal elements of I, then large yaw acceleration values will only appear in the z component of τ_{ankle} which has larger bounds. Since μ_{rot} is much larger than the foot width of our robot, (in SI units, 0.5 vs. 0.08 respectively) and I_{zz} is smaller than I_{xx} and I_{yy}, the method can tolerate relatively swift yaw motions. By increasing the steering speed up to a certain point, the whole body optimization in the first layer generates torques that completely track the reference orientation. Above that, tracking will not be precise and the induced motion will be on the margins of constraints.

So far we discussed how we generate CoM accelerations. At the end of each swing phase, we assume that the base and swing foot are rotated by $\omega_{\text{joy}}T$ around z axis where

T is the duration of swing phase and ω_{joy} is the reference steering velocity determined by the joystick. The third layer also determines the final location of the swing foot (i.e., the next footstep location). Thus, the second layer generates trajectories between initial and final orientations of the base and also positions and orientations of the swing foot. We use Spherical linear interpolation (Slerp) transition functions (Leeney, 2009) to generate smooth Cartesian trajectories and use quaternions to avoid singularities of Euler rotations. To lift the swing foot and having enough clearance, we use sinusoids of the form:

$$p_{C,z}(t) = \frac{z_{cl}}{\frac{1}{2\omega} - \frac{1}{6\omega}}\left(\frac{\sin(\omega t)}{2\omega} - \frac{\sin(3\omega t)}{6\omega}\right), \quad \omega = \frac{\pi}{T} \tag{2.8}$$

Where $p_{C,z}(t)$ is the reference height for the swing foot, t is the time counted from the beginning of each phase and z_{cl} is the clearance height when the foot is in the. Such function ensures zero positions, velocities, and accelerations at boundary times. The generated reference trajectories are all tracked by PD controllers in this layer which produce $\ddot{o}_b$ and $\ddot{x}_C$, given to the first layer together with $\ddot{p}_{\text{CoM}}$ described earlier. The alteration of stance and swing roles is also based on the pattern of phases produced in this layer. In section 2.5, we evaluate the performance of such pattern generation policy over uneven terrains where the assumption of flat ground is violated. This assumption is in fact used when keeping the CoM height constant and generating arc trajectories of (2.8).

So far, we have defined the Cartesian trajectories and tracking principles. In the next section, we will describe how the next footstep location is determined by the third layer.

2.4 Foot-step planner

In this section, we will formulate a Model Predictive Control (MPC) problem (Kothare et al., 1996) to find a stabilizing plan of future footsteps. We use the word future because this plan is updated at every control time-step. The inputs to this layer of our controller are sagittal ($v_{\text{joy},x}$), lateral ($v_{\text{joy},y}$) and steering velocities (ω_{joy}) determined by a joystick (assumed to be in robot's coordinate frame).

At this level of control, the robot is simplified with a foot-less LIP. The prismatic actuator in the leg keeps the CoM height always constant. This model allows us to predict the future motion assuming no ankle torques. Our formulation forms a discrete-time model of the robot where CoM positions and velocities are the states and footstep locations are inputs. Recall from (2.3) where the CoM acceleration depends on its distance from the stance foot. One can solve this differential equation and obtain the solution in time for the x and y components of p_{CoM}, expressed in (2.9).

$$\begin{aligned} \ddot{p}_{\text{CoM}} &= \frac{g}{z_0}(p_{\text{CoM}} - p_{\text{base}}) \\ p_{\text{CoM}}(t) &= ae^{-t/\tau} + be^{t/\tau} + p_{\text{base}} \\ \tau &= \sqrt{z_0/g} \\ a &= 0.5(-\tau\dot{p}_{\text{CoM}}(0) + p_{\text{CoM}}(0) - p_{\text{base}}) \\ b &= 0.5(\tau\dot{p}_{\text{CoM}}(0) + p_{\text{CoM}}(0) - p_{\text{base}}) \end{aligned} \tag{2.9}$$

We can then predict the x and y components of CoM position at time T using their current values:

$$p_{\text{CoM}}(T) = p_{\text{base}}(\frac{-1}{2h} + \frac{-h}{2} + 1) + p_{\text{CoM}}(0)(\frac{1}{2h} + \frac{h}{2}) + \dot{p}_{\text{CoM}}(0)(\frac{-1}{2h} + \frac{\tau h}{2}) \tag{2.10}$$

Where $h = e^{\frac{T}{\tau}}$. One can follow the same procedure and obtain the x and y components of $\dot{p}_{\text{CoM}}(T)$. We define the simplified robot's state $\hat{q}$ as:

$$\hat{q} = \begin{bmatrix} p^T_{\text{CoM},x} & p^T_{\text{CoM},y} & \dot{p}^T_{\text{CoM},x} & \dot{p}^T_{\text{CoM},y} \end{bmatrix}^T \tag{2.11}$$

Using current state of the CoM $\hat{q}(t)$ and the remaining time of the current step phase $T - t$, we can express the state of CoM at the end of the current step phase by:

$$\hat{q}(T - t) = A(T - t)\hat{q}(t) + B(T - t)p_{\text{base}} \tag{2.12}$$

Where $p_{\text{base}} \in \mathbb{R}^2$ and $A \in \mathbb{R}^{4\times4}$ and $B \in \mathbb{R}^{4\times2}$ are matrices containing all the coefficients of (2.10). Since p_{base} and $\hat{q}(t)$ are already available at each moment, $\hat{q}(T - t)$ could be calculated easily, called $\hat{q}[1]$ hereafter. Following the same procedure, we can predict the future motion by:

$$\hat{q}[i + 1] = A\hat{q}[i] + Bp_{\text{base}}[i], \quad i \geq 1 \tag{2.13}$$

Assuming fixed phase durations, A and B are functions of T and future footsteps are expressed by $p_{\text{base}}[i]$. It is straightforward to show that matrices A and B form a controllable system by checking the rank of $\begin{bmatrix} B & AB \end{bmatrix}$ (Ogata, 1995). Equation (2.13) serves as the basis model used in our MPC to plan future footsteps. The next footstep position given to the second layer of our controller is therefore $p_{\text{base}}[1]$ determined in this layer.

We aim at guiding the future footsteps such that they induce the desired velocity vector given by the joystick. To this end, we first define a reference footstep plan. One may expect that lateral CoM motions are minimal in natural walking conditions. Define the variable $s = 1$ for right support and $s = -1$ for left support phases. Assume also that the footsteps normally have a distance of $2d$ in the lateral plane. We define delta-motion for each step by $\Delta x = 2v_{\text{joy},x}T$, $\Delta y = 2v_{\text{joy},y}T$ and $\Delta\theta = \omega_{\text{joy}}T$. If we define the rotation vector $R(\eta) = \Big[cos(\eta), sin(\eta)\Big]^T$, we can plan a reference discretized path by initializing:

$$\begin{aligned}\theta[0] &= o_{b,z} \\ p_{\text{des}}[0] &= p_{\text{base}} \\ m[0] &= p_{\text{des}}[0] + d\ R(\theta[0] + s\pi/2)\end{aligned} \tag{2.14}$$

And defining the steps for $1 \leq i \leq N$ as:

$$\begin{aligned}\theta[i] &= \theta[i-1] + \Delta\theta \\ m[i] &= m[i-1] + \Delta x\ R(\theta[i-1]) + \Delta y\ R(\theta[i-1] + \pi/2) \\ p_{\text{des}}[i] &= m[i] + d\ R(\theta[0] - s(-1)^i\pi/2) \\ v[i] &= v_{\text{joy},x}\ R(\theta[i]) + v_{\text{joy},y}\ R(\theta[i] + \pi/2)\end{aligned} \tag{2.15}$$

In these notations, θ represents the steering angle, m denotes the midpoint of the two feet, p_{des} represents the ideal footstep plan and v represents the ideal CoM speed. The goal of MPC controller in the third layer is to track such sequence of ideal footsteps and desired CoM velocities determined by the joystick. Therefore, we define a quadratic optimization problem of the form:

$$\begin{gathered}\min_{\hat{q}[i], p_{\text{base}}[i]} \sum V_{Q_q}(\hat{q}_p[i+1] + \hat{q}_p[i] - 2\ m[i]) + V_{Q_{dq}}(\Delta\hat{q}_p[i+1, i] - T\ v[i]) + \\ V_{Q_p}(p_{\text{base}}[i] - p_{\text{des}}[i]) + V_{Q_{dp}}(\Delta p_{\text{base}}[i+1, i] - \Delta p_{\text{des}}[i+1, i]) \\ s.t. \\ \hat{q}[i+1] = A\hat{q}[i] + Bp_{\text{base}}[i], \quad i \geq 1\end{gathered} \tag{2.16}$$

Where $V_Q(v) = v^T Q v$, the matrices Q_i are quadratic costs, $\hat{q}_p[i]$ denotes position components (x and y) of the state vector $\hat{q}[i]$ and the operator Δ denotes the difference between consecutive $i+1$ and i indexes. Such objective function implicitly minimizes both states and input control variables, together with their derivatives in discrete space. The first two quadratic costs are written over two consecutive CoM positions so that they preserve symmetry of the limit-cycle. In normal conditions when the robot is moving solely in the forward direction, the value of objective function becomes zero. This formulation translates the desired lateral and sagittal local speeds to the global coordinate frame. In practice, however, we always give a zero reference lateral speed to the planner. Sample planned and reference paths are shown in Figure 2.2.

In practice, we use higher weights for the matrices Q_{dp} and Q_{dq} so that the robot does not react aggressively or take very large steps when it is perturbed with strong pushes. Such weighting policy is more robust for maintaining average speeds, determined by the joystick. One can easily introduce additional constraints on the individual footsteps too. For example, CoM position should not go further than $|\mu z_0|$ of the stance foot position where μ is the friction coefficient. In this case, the ground reaction force vector will fall out of the friction polyhedron. We have added this constraint to our optimization,

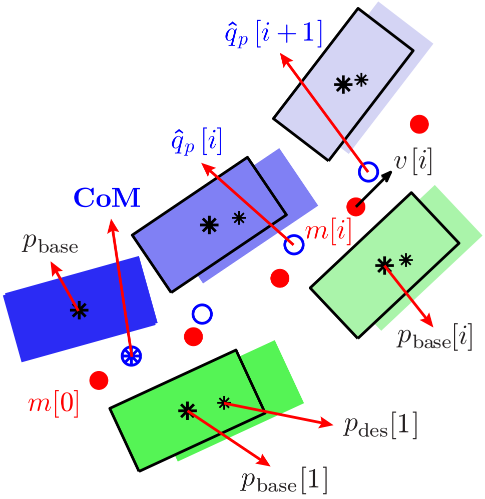


Figure 2.2 – Sequence of steps planned by MPC during a left support phase. The ideal sequence of (2.15) is shown in background while the optimal plan extracted from (2.16) is plotted on top with solid bounds.

though it is rarely triggered in the range of speeds that our method can produce. Note also that the inter-foot distance ($2d$) is chosen to be large enough to avoid self-collisions.

Our MPC controller of this layer is implemented via CVXGEN (Mattingley and Boyd, 2012) up to a horizon of $N = 5$ footsteps and takes around 0.2ms on average to solve. Once the next footstep $p_{\text{base}}[1]$ is optimized, it is transferred to the second layer of the controller and treated as a target point for the swing leg. Note that if the CoM is perturbed, the corresponding $p_{\text{base}}[1]$ will adapt in a single time-step while in many methods such as (Mordatch et al., 2010), the corrective response is delayed. In the next section, we briefly introduce our simulation platform and analyze the control performance in perturbed and periodic walking conditions.

2.5 Results

In this section, we evaluate the performance of our control approach on COMAN robot (Spyrakos-Papastavridis et al., 2013) in simulations. COMAN is a child-sized torque-controlled robot with electric motors and series-elastic elements. It weights around 30 kg and has a total of 23 degrees of freedom, 6 per leg, 4 per arm and 3 between the pelvis and the torso. For this work, we keep the upper body fixed with the policies described earlier. The robot carries usual joint position and velocity sensors as well as an IMU unit on the pelvis and contact force sensors in the ankles. In the proposed controller, however, we do not use the contact sensors, accelerometers or any perception. Our method is robust against the series-elastic elements in simulations, where the torque tracking performance is ideal. Although the control loop of the real robot is relatively fast (around 500Hz), the torque controllers are not responsive and fast enough yet to be suitable for this controller. They are typically slow, introducing about 20ms of delay

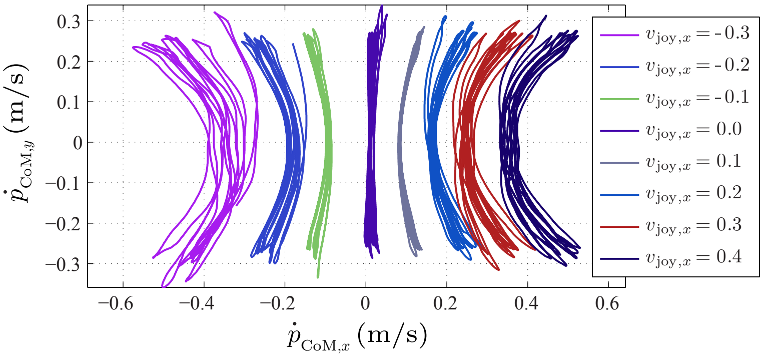


Figure 2.3 – Resulting limit-cycles of CoM speeds plotted for a wide range of desired forward velocities. The controller tracks the desired velocity, however, the repeatability of limit-cycles is affected by foot trajectory tracking performance and contact simulation in high speeds.

with respect to the given torque profiles. As improving the real robot is still ongoing, we show simulation results in this section using the ODE-based simulator Webots (Michel, 2004).

2.5.1 Range of speeds

With the same set of parameters, we can cover a wide range of backward and forward speeds between -0.2m/s to 0.4m/s, shown in Figure 2.3. However, the tracking depends on various parameters. It is acceptable in forward-walking motions, but slightly biased when walking backward due to ankle asymmetries.

We additionally test the performance of our controller when combining steering and forward motions. To this end, we perform four tests shown in Figure 2.4 to find the maximum reference velocity combinations possible. Achieving fast forward and steering velocities at the same time is highly challenging. One can see in Figure 2.2 that the outer foot must take larger steps compared to the inner foot and thus, the steering motion limits the maximum forward speed.

The walking performance of our controller on COMAN is comparable in relative terms to real-sized humanoids which are taller. For example, the method presented in (Feng et al., 2013) uses on-line trajectory optimization and achieves 1.14m/s on flat ground with Atlas robot, twice as tall as COMAN. A restricting assumption for our method is keeping the feet flat while Feng in (Feng et al., 2013) uses toes at lift-off moments. Including a push-off phase is a possible improvement for our method. So far we characterized stable periodic walking conditions without any perturbation. In the following parts, we will examine its performance against perturbations, starting from external pushes next.

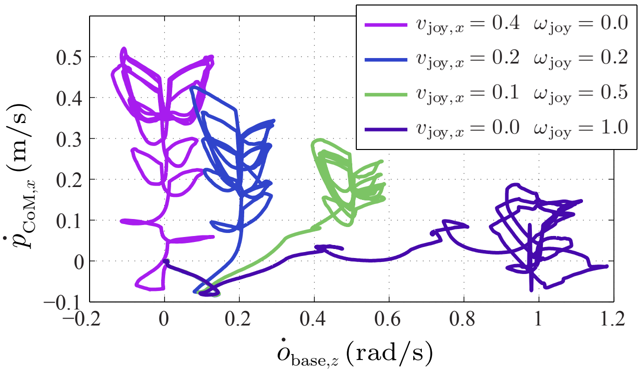


Figure 2.4 – Evolutions of the forward CoM ($\dot{p}_{\text{CoM},x}$) and base yaw velocities ($\dot{o}_{\text{base},z}$). In each test, the reference joystick velocity linearly increases up to the indicated value and remains constant after about 3s. The robot starts from rest condition and goes to the final limit-cycle. These four tests show maximum reference speeds that could be given to the robot for stepping stably. We can see that steering limits the forward velocity as the outer foot must take larger steps.

2.5.2 External pushes

One important and challenging task for bipeds is to stabilizes after being pushed from different directions while performing walking. Here, we consider a scenario where the robot is going forward with a moderate speed of 0.2m/s. During locomotion, we apply impulsive pushes to the torso each having 3Ns of strength. The robot takes corrective steps to capture the accumulated energy rather than relying on ankle torques and modulating the CoP. Note that external pushes perturb the CoM state, and the method simultaneously changes the future footstep plan, starting from (2.12). Since our planning is online, the robot can react to perturbations as fast as possible. The resulting behavior is shown in Figure 2.5A.

2.5.3 Model errors

In model-based methods, one needs to make sure that the internal model of the robot matches the real robot as much as possible. The aim is to reduce feedback gains which correct these errors and thus, making the robot more compliant. We test our method in some scenarios where the robot carries additional weight in simulations, and the controller is blind to it. The aim is to know how robust the method is against unknown errors. The resulting limit-cycles at the nominal speed of 0.2m/s and maximum tolerable errors are shown in Figure 2.5B as well as walking on a slope of 15 deg degrees. The latter test evaluates the performance where the assumption of flat-ground is violated. This slope is the maximum that the method can tolerate while being blind. We observe that the walking is still stable, even though the limit cycles become skewed, asymmetric or enlarged.

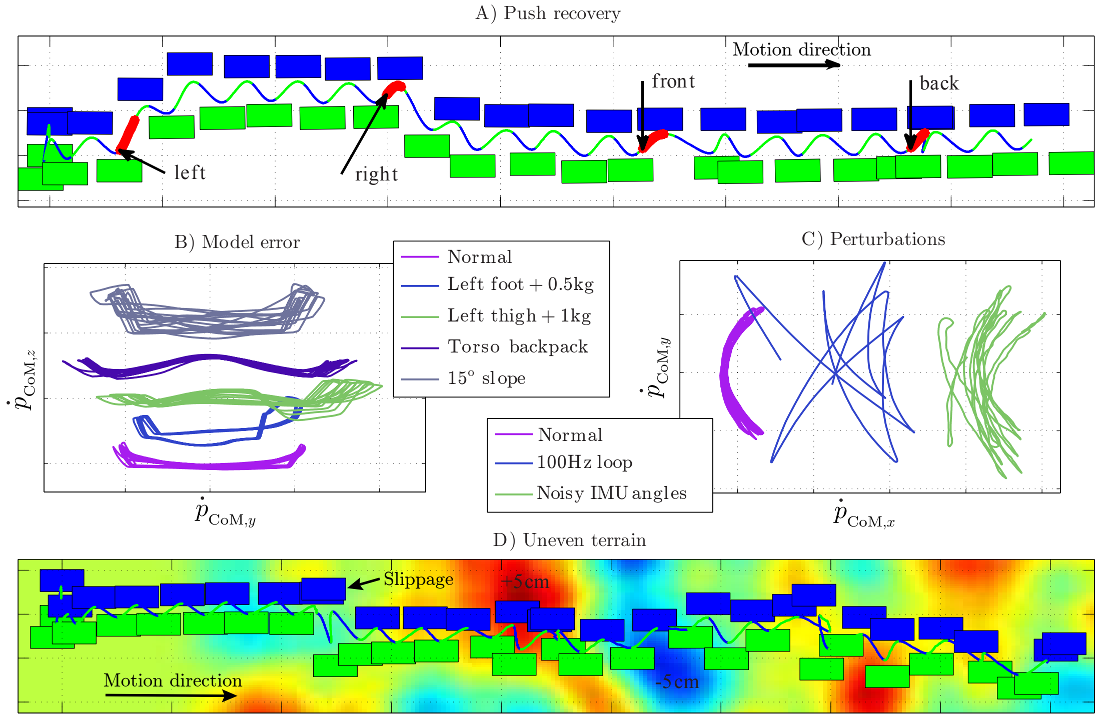


Figure 2.5 – A) Push recovery: The robot is subject to 10N external pushes applied to the torso from various directions while walking at 0.2m/s. Each push lasts 0.3s and then, the robot recovers by taking proper steps. B) Model Errors: Extra weight is added to various parts of the robot. In the "torso backpack" scenario, we add 1kg to the torso and shift its CoM -1cm back. In the slope scenario, the robot gradually takes larger steps on steeper slopes until it falls at 15 deg. C) Perturbations: In the noise scenario, a white Gaussian noise of std=3 deg is added to the IMU orientation angles and a noise of std=1 deg is added to the joint position sensors. The robot also tolerates 10ms of delay in the communication link, while still being stable. D) Uneven terrain: Here, terrain variations are ±10% of the leg length and the robot still walks robustly. Larger variations could also be tolerated, but our simulator has problems simulating contacts on complex meshes.

2.5.4 Perturbations

Apart from systematic model errors, we test the robustness against delayed communication and sensory noises as well. These scenarios are shown in Figure 2.5C together with the nominal limit-cycles. We add Gaussian noises to the IMU orientation and joint position sensors. The standard deviation (std) of IMU noises could reach 3 deg at maximum while the std of joint sensors could reach 1 deg. Note that the joint errors accumulate in the kinematic chain of the leg and affect the end-effector tracking performance severely. Also, since IMU angles determine the whole orientation of the robot, a small wrong rotation can induce a large motion on the feet. Stronger perturbations lead to self-collision of the feet or result in taking very large steps which are physically impossible. In practice, a Kalman filter could be used to filter out these noises which we avoid in these tests. The robot also tolerates delayed communications up to 10ms. Although it takes unpredicted steps in different directions, it is still able to maintain the dynamic stability.

2.5.5 Uneven terrain

Our final test validates the performance over uneven terrain, i.e., random external geometric errors. We assume flat-ground when planning footsteps and forming arc trajectories for the swing foot. However, they are not often valid since the swing foot might touch the ground earlier or later than expected. The intrinsic compliance of the first layer improves stability in the sense that minimal bouncing happens at these moments. However, we are interested to know the effect of such perturbations in the footstep planning and the cyclic behavior of the robot. The result of our test is shown in Figure 2.5D where terrain variations are around ±10% leg length. Although the robot slips at some point due to weaknesses of contact modeling in our simulator, it can recover by taking proper steps again. Our responsive footstep planner enables the robot to recover from fast perturbations like slippages and external pushes which can happen in the real world as well.

2.6 Discussion

In this chapter, we presented a hierarchical controller able to perform walking over a wide range of forward and steering speeds. Our controller is based on a dynamics model without needing any offline optimization. Different levels translate the problem first from the joint torques to Cartesian accelerations, then to footstep positions and then only to forward and steering desired speeds, given by the user. The specific formulation of the first layer makes the robot compliant and considers most of the physical constraints on the robot including torque limits, frictions, and CoP regions. Such flexible low-level controller plays a crucial role in behaving compliant and robust when the robot is exposed to various kinds of perturbations that make its interaction with the environment unpredictable. The second layer produces smooth Cartesian trajectories for the feet and CoM and tracks them with Cartesian PD controllers. In the third layer, we simplify the robot by a LIP model which allows us to predict the motion in closed-form over multiple steps. We can then find a desired footstep position which captures the extra energy while realizing a desired locomotion speed.

A distinguishing feature in our method is the focus on footstep adjustment instead of CoP modulation. The ankle torques can in fact help only up to the limited size of the feet. However, one can take larger steps and use the available friction to guide the CoM. This requires planning over hybrid states, i.e., switching the supporting leg frequently. Possible improvement for our method could be combining the footstep planning with CoP modulation strategies in our second layer where we feed the open-loop LIP accelerations to the CoM. Another future work is adding constraints to the footstep planner to avoid self-collisions which make the problem non-convex. Furthermore, we can consider exact footholds (determined by perception) which may need a mixed-integer programming. Our formulation provides basis for efficiently adding various constraints that might not be realizable with other non-linear planning methods. Evident by various kinds of robustness tests and minimal parameter tunings, we can suggest the method for walking on a wide range of torque-controlled bipedal robots. The accompanying movie shows more scenarios where the robot performs walking and steering as well as recovering from perturbations.

3 Inverse Dynamics

This chapter presents our implementation of inverse dynamics on the real hardware. Remember that this control block was first presented in chapter 2 and combined with our MPC controller for walking applications. However, due to hardware imperfections, we needed to develop additional control layers that facilitate interfacing inverse dynamics with the real hardware. This chapter presents these layers, namely state estimation and torque control. The goal of state estimation is to fuse encoder and IMU signals and provide a clean estimation of floating-based states. Our torque controller, on the other hand, inverts the model of actuators identified offline to improve the torque-tracking speed. It also uses torque sensors to estimate frictions and compensate them without vibrations. These two layers enable for smooth and very compliant whole-body behaviors without using any joint position and velocity reference, or joint-level damping. With the proposed state estimation and torque tracking, our robot can keep balance compliantly and perform a wide range of motions, including multi-contact scenarios. Despite hardware imperfections which would be discussed in chapter 8 as well, our hierarchical controller can thoroughly demonstrate all the advantages of task-space inverse dynamics for humanoids. Specifically, it can demonstrate a whole-body compliance and stability[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji, Luca Colasanto, and Auke Jan Ijspeert. *"Practical considerations in using inverse dynamics on a humanoid robot: Torque tracking, sensor fusion and Cartesian control laws."* In Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, pages 1619–1626, 2015.

The first author developed the method, performed experiments and wrote the manuscript. The second and third authors gave helpful comments and corrections.

---

[1] All the videos of this chapter could be found at https://youtu.be/BBFWHUW-S6c

3.1 Background

In legged robotics, kinematics-only algorithms and position controllers were traditionally popular. Even though dynamic models were available in different formats, for long it was computationally expensive to use them in the control of humanoid robots. Recently, thanks to powerful computational tools, performing such calculations is becoming affordable on-line. Furthermore, humanoids gradually start to interact with humans and getting out of laboratories which require a great compliance and robustness. Therefore, torque-controlled robots are becoming popular while control algorithms try to provide robustness, compliance, and preciseness together. The wish to perform more complex and dynamical tasks has motivated researchers to think of task-space motion planning to reduce the complexity of high-dimensional problems. However, a precise controller that converts Cartesian tasks to actuator inputs is crucial in the loop to ensure precision and feasibility regarding physical constraints.

Although in many position-controlled robots, contact force sensors provide a partial observation of the dynamics, in torque-controlled robots we additionally have sensing and control over the torque in all individual joints. Regarding the control aspects, respecting inequality constraints such as torque and friction limits as well as Center of Pressure (CoP) regions could not be easily addressed in classical formulations like (Mistry and Righetti, 2012). Stephens and Atkeson (Stephens and Atkeson, 2010a) formulated the problem into a two-staged optimization. At first, they optimized contact forces and moments, given desired momentum change rates. Friction and CoP limitations were also considered at this stage. Then they found joint torques using more conventional pseudo-inverse methods like (Mistry and Righetti, 2012). Herzog (Herzog et al., 2014) followed a similar approach with a hierarchical formulation that executed the desired tasks with different priorities. In chapter 2 we combined all these stages in a single optimization process. Given desired Cartesian accelerations for feet, hands, CoM and the torso (tasks), we find joint torques directly by optimizing all the unknown variables in a single quadratic programming problem subject to various equality and inequality constraints. We have successfully tested this layer combined with a footstep planner to generate walking on Atlas (Faraji et al., 2014b) and COMAN (chapter 2) robots in simulation. Inverse dynamics helps to make the robot compliant, precise and robust against various sources of noise and errors.

In this chapter, we discuss how inverse dynamics is interfaced with our real COMAN robot (Tsagarakis et al., 2013) to perform balancing and other Cartesian tasks. While this article solves practical problems of a specific robot, COMAN, we think that the methods presented here could help improving whole-body torque control on other robots such as Sarcos (Herzog et al., 2014) and Atlas (Xinjilefu et al., 2014a) as well. Our control chain consists of robot's state estimation, Cartesian task planner, inverse dynamics and actuator torque control. In our previous simulations of chapter 2, position encoding, and torque tracking performance were ideal. The robot's global state was computed with simple kinematics, and the output torques were directly realized in the joints. However, on the real robot, joint encoders have low resolution, the control delay is considerable, and torque tracking is slow, implemented by a PI loop only. The IMU is broadcasting data with slower rates, and due to magnetic interferences, the yaw angle we get from IMU is unreliable. Our goal in the present work is to propose estimation and actuation methods that improve the interface of inverse dynamics with the real hardware.

The contribution of the present work is twofold. In torque tracking, we propose a friction observer that acts as a feedback term together with the inverse of motor's model as a feed-forward term. Such architecture improves bandwidth, latency and also transparency of the joint when commanding zero torque references. In other relevant works, fast PID loops over the current (Hutter et al., 2011) or torque (Boaventura et al., 2012; Hutter et al., 2011; Le Tien et al., 2008; Mosadeghzad et al., 2013) compensate disturbances, sometimes only considering internal actuator dynamics (Boaventura et al., 2012) or only the friction observation (Le Tien et al., 2008). The proposed architecture, however, combines the two on a voltage-controlled motor with a novel observer design. By modeling the motor resistance and back electromagnetic effects (Pillay and Krishnan, 1989), we find a feed-forward term that essentially replaces the additional current loop (Hutter et al., 2011).

In state estimation, a two-stage quadratic programming method is proposed. We fuse IMU data, leg kinematics, and contact forces together to estimate the world-frame position and velocity of the robot along with the yaw angle. In case of slight rolling/tilting of the feet which frequently happens in very dynamic tasks, the CoP moves to the borders. We assume that the foot's surface at the CoP has no relative translational motion with respect to the ground. Therefore, at each instance of time, the inverse chain of the leg can start from the CoP point in the world-frame instead of the ankle position. This strategy improves the precision and stability while we do not make any restricting assumption on foot/terrain orientations like (Rotella et al., 2014). Our second contribution in this chapter is to estimate the global states with this strategy. Our formulation is similar to Kalman filtering with fixed covariances. However, other methods proposed in (Rotella et al., 2014; Bloesch et al., 2013; Xinjilefu et al., 2014b,a) linearize the model to update the covariance matrices for statistical optimality. Our method is computationally faster but requires a minimal offline tuning. Moreover, we assume that leg kinematics (positions and velocities) are perfect like (Rotella et al., 2014; Bloesch et al., 2013) while Xinjilefu in (Xinjilefu et al., 2014a,b) puts such assumption on positions only and uses the full model of the robot to filter velocities. We consider such improvement for future work as the case is different due to existence of series springs in COMAN.

The chapter starts with describing our torque controllers. We continue in the third section by formulating a hierarchy of optimization problems that fuse the sensory data to estimate the global state of the robot. In the fourth section, we introduce our inverse dynamics formulation and Cartesian controllers suited for dexterous demonstrations discussed in the results section.

3.2 Torque tracking

The baseline PI torque controller in COMAN does not have the bandwidth required for our tasks. Therefore we identify motor parameters offline and invert the resulting model to find a feed-forward voltage. We also need to estimate the friction which is considerable in COMAN due to high gear ratios. In literature, friction identification and compensation is studied broadly in position/velocity/torque-controlled robots (Johnson and Lorenz, 1992; Kelly et al., 2000; Traversaro et al., 2014). There are various statical or dynamical friction models used in different robots or experimental setups. In a position/velocity control paradigm, since the desired direction of motion and its velocity are known, it is easy to compensate the Coulomb friction. Such compensation improves the position tracking and reduces the effort made by feedbacks (Johnson and Lorenz, 1992). More complex models such as LuGre (Olsson et al., 1998) are also identified and studied in (Tan and Kanellakopoulos, 1999) by exploiting a set of observers to estimate the parameters on-line.

In torque-controlled motors, however, less complicated models are considered such as hyperbolic tangent (Hur et al., 2012) or Coulomb-viscous (Traversaro et al., 2014). In general, since the desired direction of motion is unknown in a torque controller, it is not easy to compensate the Coulomb friction. Therefore other works in torque-controlled joints either observe the friction with some dynamics and latency (Le Tien et al., 2008) or make a fast PID loop to compensate the torque error (Hutter et al., 2011; Boaventura et al., 2012).

In COMAN, there are two encoders before and after the spring and a torque sensor before the spring. Due to the low resolution of post-spring encoders and therefore modeling difficulty, we assume negligible transient drive in springs. In this section we describe our procedure to identify the motor's electromagnetic properties like (Kelly et al., 2000; Johnson and Lorenz, 1992; Traversaro et al., 2014), and then we formulate an inverse control law which generates proper actuator input (voltage) to realize the desired torque in the output (Figure 3.1).

3.2.1 Model identification

To identify motor properties, we track two sets of reference trajectories (sinusoids and trapezoids to explore motor dynamics and friction properties respectively) with a simple PD controller of low gains that produces the motor voltage. After recording velocities (by finite differentiation) and output torques, we set up a least square optimization problem which fits the following model to the signals:

$$\begin{aligned} \frac{v - K_e N\dot{\theta}}{R_m} k_t N &= \tau_{\text{fric}} + J_m N^2 \ddot{\theta} + \tau = \tau_{\text{mot}} \\ \tau_{\text{fric}} &= \tau_{\text{co}}\ \text{sign}(\dot{\theta}) + \tau_v \dot{\theta} \end{aligned} \tag{3.1}$$

where v is the voltage of the motor, θ is the joint angle before the spring, τ is the load torque and τ_{fric} corresponds to all friction torques. The constant k_e is back EMF

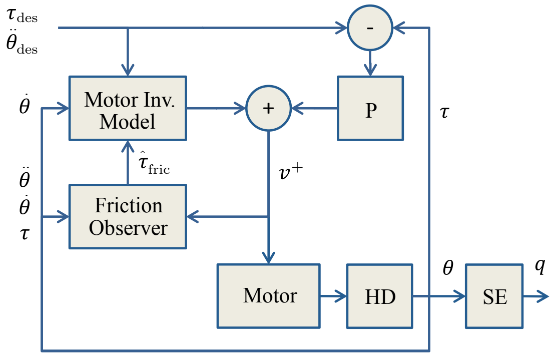


Figure 3.1 – A typical joint in COMAN consists of an electrical motor, a Harmonic Drive (HD) and an in-series spring (SE). After identification, we model the relation between motor voltage v and the torque τ after the harmonic drive. Such model is used to estimate the friction and produce feed-forward voltage added to a simple proportional feedback (P).

coefficient, k_t is torque constant, N is harmonic drive ratio, R_m is motor resistance and J_m is total inertia of the motor and the harmonic drive. We use a simple combination of Coulomb (τ_{co}) and viscous (τ_v) frictions inspired from the type of data we observe. Adding stiction or Stribeck effects or using more advanced dynamic models like Dahl or LuGre (Olsson et al., 1998) turned out having no considerable improvement in our setup.

The goal of our identification is to adjust motor parameters k_t, k_e, R_m, J_m we already have from the data-sheet and identify friction coefficients τ_{co} and τ_v. Since COMAN has joint torque sensors, we can perform identification for each joint individually unlike (Traversaro et al., 2014). We assume equal torque and back EMF constants and also N is known from the data-sheet. Figure 3.2 shows a typical trained model, plotted as a friction-velocity profile. The red curve shows our friction model while the black curve corresponds to the characteristic of friction estimated from the measured torque and the motor model. In practice, to explore the whole range of output torques and high currents, we apply different external loads to the joint during the experiment. According to the data-sheet, the motors in COMAN have very low magnetic reluctance and therefore, we observed negligible improvement by adding reluctance term during the identification process.

3.2.2 Control law

After optimizing the model (3.1) for each joint individually, we invert it to find the feed-forward voltage terms v^+ to be applied at the next time-step:

$$v^+ = k_e N \dot{\theta} + \frac{R_m}{k_t N}(\hat{\tau}_{\text{fric}} + J_m \ddot{\theta}_{\text{des}} + \tau_{\text{des}}) \tag{3.2}$$

where $\hat{\tau}_{\text{fric}}$ stands for the estimated friction. Although the optimized set of parameters better describe the data, the difference is still considerable in Figure 3.2. Unlike using

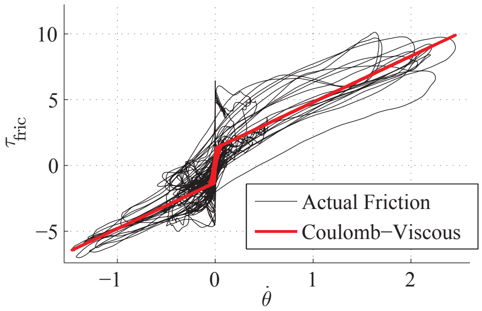


Figure 3.2 – The friction in shoulder roll joint and the approximate coulomb-viscous curve. Note that the actual curve is obtained via estimating motor parameters.

the trained parameters (Hur et al., 2012), we try to estimate the friction based on the torque measurements. knowing the voltage applied in the previous time-step v^- and the resulting motor velocity $\dot{\theta}$, motor acceleration $\ddot{\theta}$ and the output torque τ_{out}, we can estimate the friction by:

$$\begin{aligned} v^- &= k_e N\dot{\theta} + \frac{R_m}{k_t N}(\hat{\tau}_{\text{fric}} + J_m\ddot{\theta} + \tau) \\ \hat{\tau}_{\text{fric}} &= \hat{\tau}_{\text{co}} + \tau_v\dot{\theta} \end{aligned} \tag{3.3}$$

where the viscous friction is assumed to follow the model. Subtracting (3.3) from (3.2) yields:

$$v^+ - v^- = \frac{R_m}{k_t N}(J_m(\ddot{\theta}_{\text{des}} - \ddot{\theta}) + (\tau_{\text{des}} - \tau)) \tag{3.4}$$

which basically means the voltage of the motor is integrating the torque and acceleration errors, ensuring convergence to zero. Finally we add a simple proportional gain to correct the frequency response of the system (Figure 3.1). In practice we multiply the estimated Coulomb friction $\hat{\tau}_{\text{co}}$ by a factor close to one to make the system more stable against un-modeled effects. The controller can show remarkable zero-torque transparency and track very fast torque profiles precisely. It can also be easily applied on other robots which have torque sensing capabilities, unless a model is fitted to the actuators. The performance will be further demonstrated in section 3.5 and compared to the momentum based estimation proposed in (Le Tien et al., 2008). In next section, we describe another crucial component required for agile motions which is the global state estimation.

3.3 Robot state estimation

Using inverse dynamics requires precise and robust estimation of the robot's state in the global coordinates frame (i.e., 6 DoF of the pelvis and their derivatives). These values should be indeed consistent with contact constraints. One can easily assume that the geometric center of the foot is fixed and use this constraint to determine the global

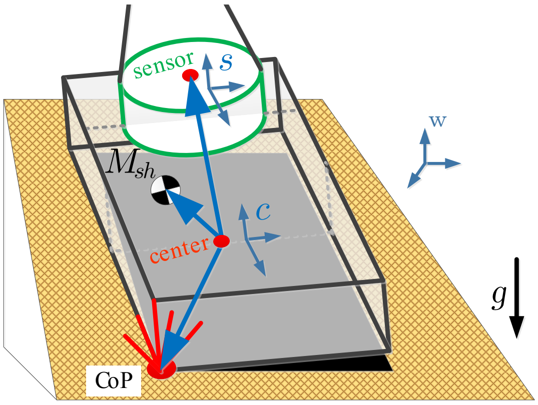


Figure 3.3 – An arbitrary posture of the foot over an inclined surface. It is very usual on the real robot that the inverse dynamics algorithm uses boundaries of the support region to provide balance. This perturbs the state estimation in case of tilting or rolling of the foot. In this figure, the frame c refers to the geometric center of the bottom surface, s refers to the frame attached to the 6D force sensor and w refers to the world-frame. Note that M_{sh} corresponds to the total mass under the sensor.

base (pelvis) position. The IMU mounted on the pelvis can also be used to determine orientations. However, this simple method on the real robot results in slippage when the CoP goes to the borders like Figure 3.3. Small motions of the foot quickly result in considerable perturbations on the whole inverse dynamics. A standard method to cancel such vibrations out is to add a term on the reference torques to damp joint velocities. However, these unwanted forces sometimes cause more slippage and imprecise tracking. Therefore we are interested in estimating the robot's global state robustly and precisely to avoid joint damping and consequently reach faster motions.

We assume that the feet have no relative translational motion at the CoP point, but they might have relative rotational motion. Such assumption makes velocity estimation more precise. However other methods either consider fixed foot position (Xinjilefu et al., 2014b,a; Rotella et al., 2014; Bloesch et al., 2013) or at most, add fixed foot orientation constraint (Rotella et al., 2014). Possible slippages, tilting or rolling, are considered as a disturbance in these methods. However, the fact that the reference foot position/orientation can slowly change and adapt through Kalman filtering of (Bloesch et al., 2013; Rotella et al., 2014) is not yet implemented in the present work.

COMAN is equipped with a precise Microstrain IMU on the pelvis which computes orientations, angular velocities, and linear accelerations. COMAN also carries 6-axis force-torque sensors on each foot. We do not have a considerable backlash in the joints and rely on pre-spring encoders to obtain joint velocities, assuming stiff springs. In fact, post-spring encoders are used to build kinematics, but the signal does not have enough resolution for differentiation. Similar methods in the literature perform steady state Kalman filtering considering full-body dynamics to estimate joint velocities (Xinjilefu et al., 2014b; Fallon et al., 2014). In this work, however, we only consider estimation of the global states and leave the joint states and springs for future work.

We set up a staged problem to estimate base position x_b^{w}, velocity $\dot{x}_b^{\mathrm{w}}$ and yaw angle ϕ_z. The superscript w indicates that the variable is expressed in the world-frame. Inputs are joint positions q_j, joint velocities $\dot{\theta}_j$, base frame angular rate ω^b, base linear acceleration $\ddot{x}_{\mathrm{IMU}}$, base pitch ϕ_y and roll ϕ_x angles, contact forces f_s and moments m_s which are expressed in the sensor frames $\{s\}$. We rely on the internal filtering of the IMU to cancel biases and temperature effects. The first stage of our method estimates the floating-base velocity and CoP in each contact while the second stage estimates the base position and yaw angle.

3.3.1 Stage 1: base velocity estimation

In this stage we use the base linear acceleration and contact forces to determine the CoP in the feet as well as the base velocity in the local frame. For hand tips we actually assume point contact with fixed CoP, though the robot has small spheres that can slightly roll. Assuming small rotational motions, we get the static contact moment equilibrium from (Sentis et al., 2010) for the feet and transfer it from the world-frame w to the contact's frame $\{c\}$ (Figure 3.3):

$$\begin{aligned} m_{\mathrm{CoP}} &= S(f_s + f_{\mathrm{sh}})x_{\mathrm{CoP}} + S(f_s)x_s + m_s + S(f_{\mathrm{sh}})x_{\mathrm{sh}} \\ f_{\mathrm{sh}} &= \mathrm{M}_{\mathrm{sh}}(R_c^{\mathrm{w}})^{-1}g \end{aligned} \tag{3.5}$$

where M_{sh} refers to the weight of the parts under the force sensor, x_{sh} refers to their center of mass, g is gravity, x_s denotes sensor frame location, S is skew symmetric matrix and f and m correspond to forces and moments respectively. The variable x_{CoP} is the CoP relative position inside the contact polygon. Note that all x, m and f variables are expressed in c frame. In our formulations, R_a^b is generally the rotation matrix of the frame a expressed in b. We use superscripts in our notations to show the reference frame. In (3.5), we want to be free of the global rotation whereas it appears in f_{sh}. However since the gravity g is along the z axis, any yaw rotation does not change g. Therefore we only need IMU roll and pitch angles in each time-step.

Now assuming that the two surfaces have no relative translational motion at the CoP point, we can relate the translational velocity of the contact point (which is zero) to the base velocity using kinematic relations:

$$-\dot{y} = (R_c^b S(\omega^c) + S(\omega^b)R_c^b)x_{\mathrm{CoP}} + \dot{x}_c^b + S(\omega^b)x_c^b \tag{3.6}$$

where we have defined $\dot{y} = (R_b^{\mathrm{w}})^{-1}\dot{x}_b$, the superscript b refers to the base (pelvis) frame, ω^b and ω^c are local angular velocities of frames b and c, x_c^b refers to foot's center-point position (Figure 3.3) and x_{CoP} is the same variable in (3.5), expressed in c frame. One can also estimate the new velocity of the base by:

$$\dot{y}_{\mathrm{des}} = \dot{y}(t - \triangle t) + \ddot{y}_{\mathrm{IMU}}\triangle t \tag{3.7}$$

where $\triangle t$ is the duration of a time-step. We can now combine (3.5), (3.6) and (3.7) and form a constrained quadratic optimization as:

$$
\begin{aligned}
\min_{\dot{y}, x_{\text{CoP}}} \sum V_{Q_\delta}(\delta) + \sum V_{Q_{m_{\text{CoP}}}}(m_{\text{CoP}}) + V_{Q_y}(\dot{y} - \dot{y}_{\text{des}}) \\
m_{\text{CoP}} = A x_{\text{CoP}} + B \\
-\dot{y} + \delta = C x_{\text{CoP}} + D \\
x_{\text{CoP}} \in \text{support polygon surface}
\end{aligned}
\tag{3.8}
$$

Here we define $V_Q(\psi) = \psi^T Q \psi$ and each active contact (either foot or hand) appears in the optimization with its own constraints. We have compacted (3.5) and (3.6) and avoided to index different contacts for simplicity.

If we consider Q matrices to be the inverse of error's covariance matrix for each equation, the optimization in (3.8) becomes equivalent to Kalman filtering, but with inequality constraints. Here we assume fixed covariances, and in fact, we tune them to get the desired performance. We possibly lose optimality regarding statistical properties, but avoid large calculations of optimal Kalman gains and still get the desired performance. Regarding CoP calculations, the costs for tangential elements of m^c_{CoP} and $m^{c_l}_{\text{CoP}}$ are set to large values while the cost for z components is set to zero. Adding slack variables δ to (3.6) means filtering kinematic data which is crucially needed due to encoder limitations in COMAN. The optimization (3.8) in fact filters all the sensory data together with a minimal setup where the internal process model is assumed to be a floating IMU. In future work, we would consider whole body dynamics and integrate the full model like (Xinjilefu et al., 2014a) to estimate the joint velocities as well.

3.3.2 Stage 2: base position and yaw angle estimation

So far we have determined the base linear velocity and we already know the angular rate from the IMU, both expressed in the base frame. In this stage we are going to solve another optimization problem to find the yaw angle ϕ_z and the position x^{w}_b of the base, considering the CoP found in the previous stage and available roll and pitch angles from IMU. We express the orientation of the base by:

$$
R^{\text{w}}_b = R(\triangle\phi_z) R(\phi_z(t - \triangle t)) R_{\phi_x\phi_y} = R(\triangle\phi_z) R' \tag{3.9}
$$

where $R_{\phi_x\phi_y}$ denotes the rotation matrix of the pitch and roll angles we get from the IMU and $R(\triangle\phi_z)$ is the delta rotation matrix around z axis. Now we can write the CoP in the world-frame as:

$$
x^{\text{w}}_{\text{CoP, des}} = x^{\text{w}}_c(0) + R^{\text{w}}_c(0) x^c_{\text{CoP}} = x^{\text{w}}_b + R(\triangle\phi_z) R' x^b_{\text{CoP}} \tag{3.10}
$$

where $x^{\text{w}}_c(0)$ and $R^{\text{w}}_c(0)$ represent the initial foot center frame. The variable $x^{\text{w}}_{\text{CoP, des}}$ is therefore the reference center position plus the relative displacement of the CoP at the current time-step. We can also use the velocities and accelerations to approximate the

x_b^{w} as:

$$x_b^{\mathrm{w}}(\mathrm{t}) = x_b^{\mathrm{w}}(t - \triangle t) + R(\triangle\phi_z)R'(\dot{y}\triangle t + \ddot{y}_{\mathrm{IMU}}\frac{\triangle t^2}{2}) \tag{3.11}$$

Defining $\triangle\phi_{z,\ \mathrm{des}} = \omega_z^b \triangle t$ where ω_z^b is the IMU angular rate, we can combine (3.11) and (3.10) in an optimization problem:

$$\begin{aligned} \min_{x_b^{\mathrm{w}}, \triangle\phi_z} \sum Q_\delta \delta^2 + Q_\phi(1 - \cos(\triangle\phi_{z,\ \mathrm{des}} - \triangle\phi_z)) \\ x_b^{\mathrm{w}} + \delta = R(\triangle\phi_z)A + B \end{aligned} \tag{3.12}$$

Equation (3.11) is always present in the optimization (in compact form) while different active contacts can participate with their own constraints and costs, again in a similar constraint form. We avoid indexing for the sake of simplicity. The optimization in this stage is in fact nonlinear and we solve it via Matlab's symbolic engine by setting the derivatives of the objective function to zero. It turns out that if we choose equal costs in each diagonal cost matrix, we can find x_b^{w} linearly depending on sine and cosine of $\triangle\phi_z$. Therefore one can easily replace it in the equations and find multiple solutions for $\triangle\phi_z$ where we choose the closest one to zero and thus calculate x_b^{w} as well. Therefore the full state of the pelvis can be calculated with these two stages which take 0.2ms on a modern computer.

This method can easily disable different contacts by setting their costs to zero. Using inequality constraints, we limit the CoP and ensure stability in case of slight tilting or rolling of the foot. Introduction of slack variables enables us to filter kinematics and contact forces together in a minimal setup and improve the robustness against noise. In future work we will find a policy to re-estimate the reference CoP position in (3.10) like (Bloesch et al., 2013) and update it during locomotion/slippage. Note that the proposed method can be easily applied on most of the humanoid robots with IMU, joint encoders and contact force sensors. In addition to torque tracking and state estimation, there are few remarks on task controllers and parameter tuning of inverse dynamics layer that we explain in next section.

3.4 Controllers

In this section we briefly introduce the control algorithm we use to perform the tasks described in section 3.5. This controller has three sub-layers itself: inverse dynamics, Cartesian controllers and trajectory generation.

3.4.1 Inverse dynamics layer

We use the general inverse dynamics framework of chapter 2. In brief, given the Cartesian accelerations $\ddot{x}_e$ (translational and rotational), we use a quadratic program as formulated in (3.13) to minimize joint accelerations $\ddot{q}$, joint torques τ and contact forces λ under various physical constraints.

$$
\begin{aligned}
\min_{\ddot{q},\lambda,\tau,\sigma} \; & V_{Q_q}(\ddot{q}) + V_{Q_\lambda}(\lambda) + V_{Q_\tau}(\tau) + V_{Q_\sigma}(\sigma) \\
& M\ddot{q} + h = \tau + J_c^{\mathrm{T}}\boldsymbol{\lambda} \\
& \sigma + \ddot{x}_e = J_e\ddot{q} + \dot{J}_e\dot{q} \\
& A\begin{bmatrix}\tau^{\mathrm{T}} & \lambda^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \leq B
\end{aligned}
\tag{3.13}
$$

where the matrices Q_i are diagonal quadratic costs and the variables σ induce a soft constraint on Cartesian tasks. M is the mass matrix, h represents all gravitational, centripetal and Coriolis forces, J_c is the Jacobian of contact points and J_e is the Jacobian of end-effectors (contacts, CoM and torso orientation). The matrices A, B represent physical inequality constraints such as torque limits, friction polyhedra and CoP limitations. In practice, we perform stiff position control on the joints that are not included in the motion. The implementation of (3.13) in CVXGEN (Mattingley and Boyd, 2012) enables us to remove sparsities from the mass matrix and Jacobians which enhance the performance up to 4 times. We reach 0.7ms to 1.1ms on a modern computer depending on the number of contacts involved (2 and 4 respectively).

Our choice of cost gains are uniform diagonal matrices $Q_q = 10^{-2}$, $Q_\lambda = 10^{-2}$, $Q_\tau = 10$, $Q_\sigma = 10^4$ for floating and $Q_\sigma = 10^7$ for contacting points. These costs provide stable performance on the real robot and ensure preciseness of the desired tasks. Fixed contacts have more importance than floating tasks. We penalize joint torques more than contact forces and accelerations to provide smoother torque profiles, but indeed floating tasks have higher priority than torques. Adding soft constraints on the Cartesian tasks is beneficial especially when CoM falls outside the support polygon. If for any reason, the upper layer provides infeasible accelerations, the soft constraints ensure satisfaction of physical inequality constraints. For instance, in case of keeping balance and being pushed extremely so that the CoM falls outside the support polygon, the robot still keeps full contact at the feet without tilting or rolling. Therefore we reject invalid Cartesian accelerations by sacrificing the precision in normal conditions unlike the prioritized hierarchy of (Herzog et al., 2014). Note that the proposed method can be easily applied on torque-controlled robots unless other control layers are present as well as a full model of the robot.

3.4.2 Cartesian control layer

Our task space formulation requires world-frame accelerations for the hands, feet, CoM and torso orientation. For Cartesian translational tasks $\ddot{x}^{\mathrm{w}}_{\mathrm{des}}$, we use a simple PD law as:

$$
\ddot{x}^{\mathrm{w}}_e = \ddot{x}^{\mathrm{w}}_{\mathrm{des}} + K(x^{\mathrm{w}}_{\mathrm{des}} - x^{\mathrm{w}}_e) + D(\dot{x}^{\mathrm{w}}_{\mathrm{des}} - \dot{x}^{\mathrm{w}}_e) \tag{3.14}
$$

For angular tasks, the formulation we use is inspired from the work in (Bacon, 2012), but including accelerations. Imagine the frame $\{e\}$ is attached to an end-effector with orientation R^{w}_e, angular velocity ω^{w}_e and angular acceleration α^{w}_e expressed in the world-frame. The desired target orientation, angular velocity and accelerations are $R^{\mathrm{w}}_{\mathrm{des}}$, $\omega^{\mathrm{w}}_{\mathrm{des}}$ and $\alpha^{\mathrm{w}}_{\mathrm{des}}$. Viewed from the desired reference frame, we want to minimize the error observed in this frame. Therefore we define the PD law as:

$$\alpha_e^{\mathrm{des}} = -K\theta_e^{\mathrm{des}} - D\omega_e^{\mathrm{des}} \tag{3.15}$$

where θ_e^{des} is the angles coming from R_e^{des}. Using standard kinematics, the angular acceleration of the frame $\{e\}$ in the world-frame is then calculated by:

$$\alpha_e^{\mathrm{w}} = R^{\mathrm{w}}_{\mathrm{des}}[S(\omega^{\mathrm{des}} + 2\omega_e^{\mathrm{des}})\omega_e^{\mathrm{des}} + \alpha_e^{\mathrm{des}}] + \alpha^{\mathrm{w}}_{\mathrm{des}} \tag{3.16}$$

Here in fact the Coriolis motion is considered when converting accelerations between different frames. Replacing Cartesian PD controllers by more advanced policies like MPC in future works can improve the tracking performance.

3.4.3 Cartesian trajectory generation layer

In experiments discussed in next section, we produce various agile motions that require proper trajectory generation from an initial to a final configuration. We use smooth exponential functions to generate 3-times differentiable trajectories as well as Quaternion SLERP functions used in (Bacon, 2012), derived once more to obtain the desired accelerations. We skip the details for conciseness.

3.5 Results

In previous sections, we discussed the three-layer controller. A robust and fast method in each of them is crucial for performing agile tasks discussed in this section.

3.5.1 Torque tracking

First of all, we characterize the performance of torque tracking block shown in Figure 3.4A, B, C. A high-level PID controller together with inverse dynamics layer that produces feed-forward torques is used to track sinusoidal trajectories on single joints. We do not choose large-amplitude signals to avoid reaching current limits since a considerable portion of the generated torque is used to accelerate the rotor. The double derivative of the desired trajectory $\ddot{\theta}_{\mathrm{des}}$ is also given to the torque tracker to compensate the torque required for the rotor.

Figure 3.4A shows the Bode diagram of force tracking transfer functions. One can see the poor performance of the basic initial PI controller while using feed-forward and feedback terms improve the transfer function. The remaining phase lag (which is about 15ms) is due to the implementation of torque tracking on an external PC with a delayed communication. In future, we will transfer it to individual motor controller boards to reduce the delay considerably. Figure 3.4B corresponds to the momentum based friction observer proposed in (Le Tien et al., 2008). Even though its tracking is satisfactory for small torque magnitudes, when the user holds the joint, the controller shows a considerable delay in tracking larger torque magnitudes which comes from the filtering nature of this observer. However in Figure 3.4C, our proposed estimator shows a smaller tracking latency.

The advantage of our method over PID controllers (Hutter et al., 2011) or disturbance observers (Le Tien et al., 2008) is that our controller compensates internal dynamics of the actuator while those methods try to resolve these issues with faster loop frequencies. Also with such estimation of the friction from the previous time-step, there is no need to close the torque loop with high PID gains.

3.5.2 State estimation and control

Next, we will demonstrate the performance of our state estimation and inverse dynamics formulation over some multi-joint tasks. The performance is quantified for two tasks of rotation around vertical (z) and lateral (y) axes while other challenging scenarios are demonstrated in the accompanying video. We frequently use the simple kinematics based filter in chapter 2 which starts the chain from the center of the feet and does not use the IMU data.

In scenario 1, the robot genuflects quickly around the hip. Figure 3.4D shows the desired and actual trajectories of the torso pitch angle, showing a fast and stable motion. In fact, this demonstration is challenging due to limited support polygons. The remaining steady state error is due to small PD gains in the Cartesian space, and our CAD model does not match the real robot perfectly. However, the robot is compliant which can be observed in the accompanying video. One can easily integrate the pitch error to converge in steady state. However, we want to ensure that most of the control input is generated by feed-forward terms which indicate more precise modeling of the system.

In scenario 2, we have depicted the performance of our estimator in Figure 3.4E where the robot rotates around the vertical axis at 1Hz. The base yaw angle is plotted using the previous simple kinematics approach versus our new staged optimizer. In simulations, the basic kinematic approach is stable enough to perform all the tasks. However, on the real robot, we have considerable noise that is rejected successfully by the proposed staged filtering.

In scenario 3, we have repeated the same motion at 0.5Hz (Figure 3.4F), but this time replacing the new state estimator in the loop by the simple one to observe the resulting performance. In scenario 4, using back the new state estimator, we have added damping to all the joints in torque tracking level to see if the vibration could be canceled. Such unwanted damping improves stability, but similar to scenario 3, it affects the tracking performance. Comparing Figure 3.4E and Figure 3.4F, one can conclude that tracking is worse in fast motions, but we have fewer vibrations because the joints change direction more rapidly and there is less time to build up static frictions (refer to (Olsson et al., 1998)). Note that all the 12 degrees of freedom in the lower body are active with small start/stop motions. Overcoming the Coulomb friction is still not perfect in torque tracking layer which results in such vibrations. Further improvement could be possibly reducing the previously mentioned delay in the torque tracking loop.

To characterize the effectiveness of starting the chain from the CoP, we have plotted

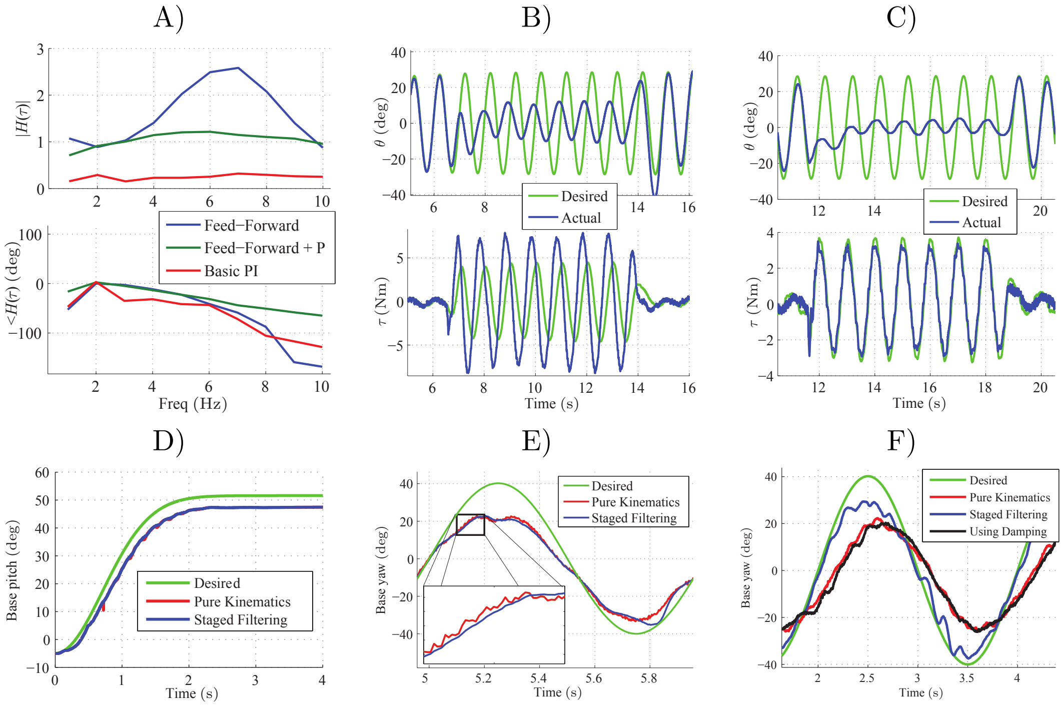


Figure 3.4 – A) The Bode diagram of force tracking transfer function for basic PI, feed-forward only and looped controllers. Note that the feed-forward term already closes an integrating loop in its friction observer. We close a second loop by a simple P gain to flatten the bode diagram. B) Tracking performance of the estimator proposed in (Le Tien et al., 2008) at 1Hz in the presence of an external holding force. The controller shows a considerable delay in higher torques and overshoots when the external force is removed. C) Tracking performance of our estimator at 1Hz in the presence of an external holding force. The controller successfully follows the desired torque (before the spring). Note that the PID gains of the high-level loop (over position) are small so that the user holding the joint can almost stop it. In (B) and (C) graphs, reference position trajectories are the same, though desired torque profiles are different due to the high-level PID controller. D) Scenario 1: rapid genuflection of Figure 3.6F. The full demonstration is shown in the accompanying video where the robot complies with external pushes. E) Scenario 2: rotation around the vertical axis at 1Hz. The main source of noise is, in fact, the precision of post-spring encoders. Our new filtering method successfully rejects noise and spikes and provides a smooth estimation compared to the previous kinematic method used in chapter 2. F) Scenario 3,4: repeating the same vertical rotation scenario at 0.5Hz, but in control, we use once the new state estimation and once the basic kinematics method. We also repeat the same task with the novel state estimator but adding some joint damping in torque tracking level.

few more signals in Figure 3.5 corresponding to scenario 2. As shown in the accompanying video, the feet slightly tilt and slip on the ground. This indicates that the fast motion of this scenario is on the margin of physical constraints. Such whole body rotation at

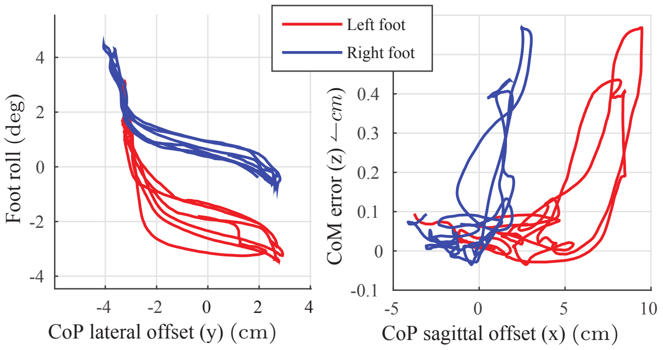


Figure 3.5 – Left: When the CoP goes to the borders, there is a slight tilting or rolling where the geometric center point of the foot is not fixed in the world-frame anymore. Right: demonstrates the difference between estimated CoM vertical position by our staged filtering and the basic kinematics based filter.

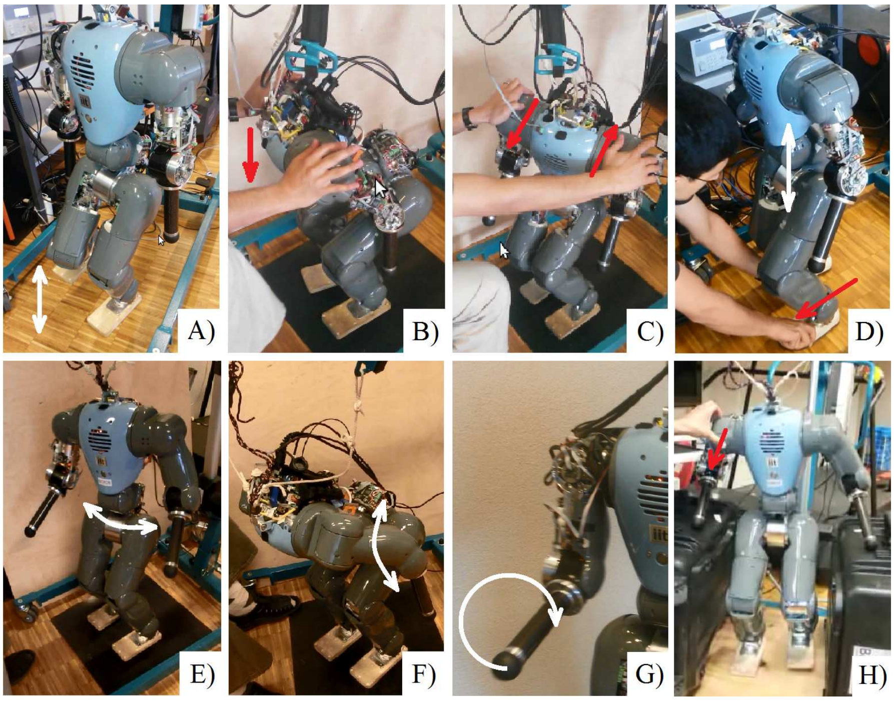


Figure 3.6 – Different scenarios using improved torque control and whole body inverse dynamics. White arrows show robot's motion while red arrows show external push. A) balance on single foot, B, C) withstanding extreme pushes downward and rotational, D) squatting motion while the user slightly perturbs the foot location, E) rotation around vertical axis, F) genuflection around y-axis, G) circular motion with hands, H) full body balance on three contacts while withstanding an external push. Please refer to the accompanying video to see each demonstration.

1Hz is challenging for a human as well. Since there is a thin flexible silicon layer at the bottom of each foot in COMAN, one can expect slight tilt or roll if the CoP goes to the borders as shown in Figure 3.5 left. Now the assumption of fixed center-foot position which is used by the basic kinematics estimator of chapter 2 is not true anymore. Therefore, the two new and old estimation algorithms can systematically differ during such dynamical tasks (Figure 3.5 right). Our new filtering, however, starts the chain from the CoP and efficiently uses the kinematic constraint while in similar works (Rotella et al., 2014; Xinjilefu et al., 2014a), this information might be ignored by considering large covariances.

Finally, we have demonstrated the full body compliance of the robot over different tasks. In fact, the primary goal of modeling the motor, performing torque control and using inverse dynamics is to find proper feed-forward terms and reduce the effect of feedbacks. This approach decreases the stiffness of the robot and makes it compliant. Figure 3.6 shows few snapshots of extreme compliance capabilities without violating physical constraints like contact frictions. Full demonstrations are shown in the accompanying video.

3.6 Discussion

In the present chapter, we have shown the gap between our previous work in simulations and challenges we faced on the real robot. We have decomposed the controller into three levels: state estimation, control, and actuation layers and discussed our strategies to improve the performance. The first novelty of this work lies in the combination of a new friction observer and motor inverse dynamics to improve the bandwidth and precision of the torque-tracking. The second novelty of this work lies in the staged optimization problems that act like a Kalman filter to find base position and orientation from IMU, kinematics and contact force sensor data. We build the kinematic chain starting from the CoP and efficiently handle very dynamic motions where the feet might slightly tilt or roll. The third novelty finally refers to the fast demonstrations and extreme compliance of the robot which is superior compared to similar approaches (Rotella et al., 2014; Herzog et al., 2014; Stephens and Atkeson, 2010a). Future work would be including the modeling of springs to improve torque tracking and state estimation, improving communication delay and adapting reference foot location on-line during state estimation.

Contributions in Modeling Part II

4 Point-Wise Model

We started developing our locomotion and balancing algorithms in the previous part with the linear inverted pendulum model and full-body inverse dynamics (refer to chapters 2 and 3 respectively). These two models were, however, probably at the two opposite extremes regarding computational cost and mechanical complexity. Our robot COMAN has relatively heavy legs and cannot be easily approximated with a single mass. Therefore, the walking controller proposed in the previous part had a limited performance. Given this limitation and also higher-level objectives of the project, particularly multi-contact motion planning and better future predictions, we needed template models with more details but similar fast computations. Therefore, inspired by methods popular in computational chemistry, we developed our point-wise modeling technique. The aim is to represent the robot posture directly with Cartesian positions instead of joint angles to reduce the complexity of full-body dynamics models. This idea is different from task-space inverse dynamics in the sense that we want to do extreme motions, e.g., going to joint limits or maximally-stretched singular postures. Therefore, our Cartesian points include end-effectors, elbow, knee, hip, and shoulder positions (not joint-angles) altogether. This representation reduces the complexity by making the equations sparser. However, we need to bind these points via length constraints in numerical optimizations. Apart from model representation, the simplification part of this methodology also comes from assuming ball joints in the chain and approximating limb dynamics with rods of no inertia around the rod axis. This relieves the need to reconstruct local frames on the body segments of the robot. This chapter presents an application of posture planning with this method, the next chapter 5 presents an application of motion planning, and chapter 6 introduces a linearized version used for walking[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji and Auke Jan Ijspeert. *"Modeling Robot Geometries Like Molecules, Application to Fast Multi-contact Posture Planning for Humanoids."* IEEE Robotics and Automation Letters, 3(1):289–296, 2018.

---

[1] All the videos of this chapter could be found at https://youtu.be/kzCcAoSkBfc

4.1 Background

Humanoid robots normally have many degrees of freedom, enabling them to perform various complex tasks ranging from locomotion to manipulation. For cyclic motions like walking, model-based approaches dominantly rely on simplifications and periodicity analysis techniques to stabilize the motion (Feng et al., 2015). In a larger scale, however, especially for locomotion in complex environments, one would require a plan ahead of time. Dynamics of the robot, contact surface configurations, inherent geometric and actuation limitations and finally, the presence of gravity make motion synthesis and control complicated. In the model predictive control paradigm of chapter 2, handling large perturbations in an online fashion depends directly on a high planning speed, i.e. the ability to synthesize feasible motions in short time spans. A simple model that captures main effects is favorable since it offers fast computational capabilities. On the other hand, optimality and feasibility analysis require dynamic information and indeed more geometric details. Therefore, a computationally fast yet inclusive kino-dynamic model is paramount in multi-contact motion planning problems.

Apart from motion planning, finding feasible and optimal postures for manipulation and balancing tasks requires a fast geometric solver. One popular approach is to find a time evolution of internal coordinates that converges to the optimal posture. This can be done via inverse kinematics (IK) (Escande et al., 2014) or inverse-dynamics (ID) formulations (Faraji et al., 2015; Feng et al., 2015) which offer compliance as well. In these time-integration controllers, an optimal posture is found naturally, but with certain dynamics. This is because only first or second order equations are solved to find velocities or accelerations, favoring their linear properties. Although optimization problems are solved as fast as 1-2ms in each control tick, it takes time for the robot to follow trajectories, i.e. to integrate velocities and accelerations. It typically takes 1-2s to reach the final goal, depending on robot capabilities (Escande et al., 2014).

An alternative to time-integration approaches is to solve for positions directly in a single control tick. This is more suitable for planning postures ahead of time, e.g. to check feasibility or optimality of a multi-contact posture in new environmental conditions. This is, in fact, the same as solving a nonlinear inverse-kinematics optimization problem (refer to appendix B). The nature of this single-shot optimization is obviously different from time-integration approaches. However, the role of Hessians and Jacobians in finding descending search directions seems equivalent to finding accelerations and velocities in time-integration approaches.

Planning "optimal" postures with minimal joint torques is, however, far more complex than simple IK problems due to inclusion of contact forces and joint torques (which are complex functions of the geometric configuration and contact forces) as optimization variables. In these tasks, the inherent redundancy in the system is solved by optimizing the overall joint torques while in normal IK methods, a desired position and orientation is considered for the pelvis, torso or the Center of Mass (CoM). The task of optimizing postures is well addressed in the literature, for example in matching a model with recorded human data in OpenSim (Delp et al., 2007) or analyzing multi-contact human postures in car assembly lines (Howard et al., 2014). In static optimal postures, the equations of motion ensure stability of the humanoid while in normal IK methods, the Zero Moment Point (ZMP) or CoM is forced to lie within the support polygon (appendix B).

Reaching real-time performance is important, but yet a secondary objective. More fundamental problems are convergence properties and nonlinearity of the equations which often affect planning problems. Handling the inherent redundancy, joint limits and of course minimization of joint torques are major critical factors for optimal posture planning. Apart from increased computation time, these issues often lead to local minima which make the final solution very sensitive to the starting point in optimizations. These issues limit the application of optimal posture planning in online control. Besides, sensitivity analysis and convergence properties are rarely discussed or addressed in motion planning and modeling papers.

Our goal is to approach geometric optimization problems differently. Traditional joint space models encapsulate all geometric complexity inside the robot model and lead to very nonlinear equations. In this work, we propose writing mechanical equations in a different way, inspired by molecules. We unroll the compacted inherent complexity of the body and distribute it over the external tasks as well. We demonstrate that formulating kinematic optimization problems becomes much easier in this way while better speed and convergence properties are achieved. We limit our focus on a case study of optimal posture optimization and provide an insightful comparison with joint-space models. A wider range of applications, however, including forwards and inverse dynamics as well as motion planning can be foreseen with this modeling technique.

4.2 Molecule-inspired modeling

The proposed modeling technique is based on methodologies used in molecular mechanics where atoms are represented by points linked together with certain bonds. A very fundamental task in computational chemistry is to determine geometric shapes or conformations of the molecules. Well-known techniques like Nuclear Magnetic Resonance (NMR) spectroscopy or X-ray crystallography reveal information about the mean position and size of atoms as well as length and type of bonds in different materials. Then, various optimization techniques are used to find exact positions of atoms in 3D space, given certain energy functions that determine the optimal length for the bonds and the angles between them. These optimization problems can have multiple local or global solutions depending on the level of energy in the molecule. Figure 4.1.A shows an example conformation, a member of NanoPutian series called NanoBalletDancer (Chanteau and Tour, 2003) which is an artificially synthesized anthropomorphic molecule that resembles human. This particular shape is indeed engineered by experts knowing principles of bond geometries. However, a popular molecular mechanics software (Spartan (Hehre, 2003) for example) can also find this conformation by minimizing energy levels.

In molecular mechanics, there are two dominant coordinates used to model configurations: natural internal vs. Cartesian coordinates (Baker, 1993). In the former, keeping atomic distances fixed, one would use bond angles to express 3D coordinates of atoms.

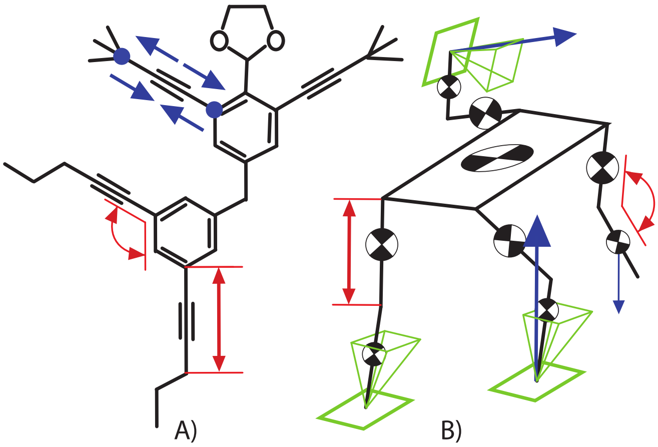


Figure 4.1 – A) NanoBalletDancer, a member of NanoPutian artificial molecules (Chanteau and Tour, 2003) designed with an anthropomorphic shape. B) a typical humanoid robot in multi-contact interaction with the environment, assigned to pick up something with the right hand. Geometric similarities between chain structures (shown in red) make 3D formulations interchangeable. Dynamic interactions are, however, very different (shown in blue, attractive and repulsive forces in molecules versus gravitational and contact forces in robots), indicating that equations of motion might have different properties.

This is very similar to traditional joint-space modeling in robotics. In the latter approach, however, each atom is assigned a 3D position constrained to have certain distances from neighboring atoms. These holonomic constraints can also describe certain angular limitations between the bonds, expressed by vector operations (Baker, 1993). A constrained optimization would then treat these constraints via undetermined Lagrangian multipliers in Verlet time integration (Allen et al., 2004). More formally, imagine a set of atoms represented by 3D coordinates r_i, a potential function $U(r^N)$ where $r^n = (r_1, r_2, ..., r_N)$ and a set of constraints:

$$\chi(r_i, r_j) = |r_i - r_j|^2 - b_{ij}^2 = 0 \tag{4.1}$$

where b_{ij} represents the distance between atoms i and j. Cartesian equations of motion take the form:

$$m_i \ddot{r}_i = -\frac{\partial U(r^N)}{\partial r_i} - \sum_{k \in K_i} \lambda_k \frac{\partial \chi_k}{\partial r_i} \tag{4.2}$$

where m_i is the mass of atom i and the set K_i represents its neighboring atoms. The Verlet time integration updates positions iteratively by:

$$r_i^{(t+\Delta t)} = \hat{r}_i^{(t+\Delta t)} + \sum_{k \in K_i} \lambda_k \frac{\partial \chi_k}{\partial r_i} \Delta t^2 m_i^{-1} \tag{4.3}$$

where Δt is the integration time-step and $\hat{r_i}^{(t+\Delta t)}$ is the unconstrained position of atom i. Putting these update rules into (4.1) yields:

$$\chi^{t+\Delta t}(r_i, r_j) = |r_i^{(t+\Delta t)} - r_j^{(t+\Delta t)}|^2 - b_{ij}^2 = 0 \tag{4.4}$$

which should be satisfied in the next time step. By solving this nonlinear system of equations, one would obtain λ_k, necessary to correct the unconstrained positions during integration. Verlet integration is very similar to nonlinear optimization in the sense that Lagrange multipliers are helping to satisfy constraints. However, in molecular dynamics, one would assume weak coupling between constraints and use linear approximations (Gauss-Seidel method) of the non-linear system to solve for λ_k iteratively (SHAKE algorithm (Allen et al., 2004)). Interior point methods are also popular (Schlegel, 2011) and provide faster convergence rates if all couplings are considered (Doyle, 2003). However, solving full-dimensional system of equations might still be time-consuming, despite their sparse structure.

Even though the weak coupling assumption makes the convergence slow, the intrinsic properties of Cartesian coordinates together with the power of Lagrange multipliers let the atoms move freely to find better solutions, especially in the case of having constraints in the system. In unconstrained conditions, however, natural internal coordinates are more suitable for finding molecular conformations (Baker, 1993). We adapt this idea and use Cartesian coordinates to solve multi-contact posture optimization problems for humanoid robots where a large number of constraints are involved. We only take inspiration from geometric similarities of molecules and humanoids. Interaction forces and "tasks" have entirely different natures, depicted in Figure 4.1. Indeed, the nature of atomic and molecular forces is very different from gravity. Therefore, we limit our focus on static conditions in this chapter. Exploring dynamic similarities and motions are left for future work.

4.2.1 Vector-based equation of motion

Dynamic equations and the notion of potential fields for robots are not similar to molecules due to the different nature of dominant forces. Besides, we are not going to use Verlet integration to solve the posture optimization problem, although it might be possible in an inverse-dynamics paradigm. Our goal is to 1: find an alternative for joint-space coordinates 2: formulate optimization problems and 3: solve them via interior-point methods, aiming at improving the computation time and convergence properties.

Similar to Cartesian coordinate systems used for molecules, we model each segment of the robot body by an approximative vector expressed in the world-frame (Figure 4.2), linking the parent and child joints together. This vector freely rotates in 3D space while its length is constrained. The world-frame Center of Mass (CoM) position for this body segment is:

$$r = b + (uL)\ x \tag{4.5}$$

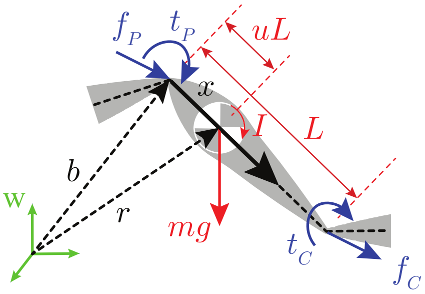


Figure 4.2 – Notations used to model a segment of the robot by vectors in the world-frame. The unitary vector x represents the segment alignment expressed in the world-frame as well as the interaction forces and torques (shown in blue). Model-specific parameters are also shown in red.

where the vector b denotes parent joint position, L represents the segment length, u quantifies the relative CoM position across the segment and the unitary vector x is the free variable shown in Figure 4.2. Writing equations of motion for this system in Cartesian coordinates is straightforward:

$$\begin{aligned} f_p - f_c &= m\ (\ddot{b} + uL\ \ddot{x} - g) \\ t_p - t_c &= uL\ (x \times f_p) + (1-u)L\ (x \times f_c) + I\ (x \times \ddot{x}) \end{aligned} \tag{4.6}$$

where × stands for cross product, f_p and t_p are incoming Cartesian forces and torques applied by the parent link, f_c and t_c are outgoing forces and torques applied to the child link and I is the inertia around any axis orthogonal to x (refer to Figure 4.2). We call (4.6) constrained equation of motion, because of holonomic constraints on x:

$$x^T x = 1, \quad x^T \dot{x} = 0, \quad x^T \ddot{x} + \dot{x}^T \dot{x} = 0 \tag{4.7}$$

There are two underlying assumptions: the CoM of the segment is aligned with joint positions, and the rotational inertia around x is negligible. These assumptions are realistic for most humanoid robots. Otherwise, an auxiliary vector y subject to $y^T y = 1$ and $x^T y = 0$ should be added, to reconstruct the local frame to account for the asymmetry incurred. The advantage of modeling a robot with vectors lies in the ability to express the tasks (end-effector positions) with a linear combination of segment vectors. Besides, joint angles and torques are all expressed with a quadratic combination of variables. Equations (4.6) and (4.7) indeed provide a geometric transformation of the traditional joint-space equations. Therefore, all translational and rotational dynamics are still included. Next, we are going to use these properties and establish a full model for our humanoid robot, consisting of limbs and a torso.

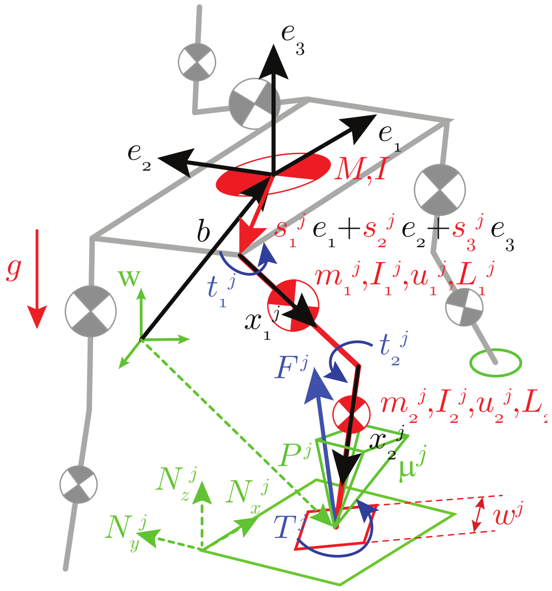


Figure 4.3 – A humanoid robot can be represented by a set of vectors in Cartesian space; all expressed in the world-frame. Fixed model parameters are shown in red, given task parameters including desired contact positions, and orientations are shown in green and free dynamic variables (forces and torques) drawn in blue. Black arrows are free position variables (Refer to the text for further information). Green areas denote surfaces that can provide a supporting force and green circles denote Cartesian points to be reached, without establishing a contact.

4.2.2 Case study: humanoid model

The spirit of vector-based equations is based on breaking the geometry of the robot into individual vectors with certain quadratic relations. Here, we consider a generic humanoid robot with a torso and four limbs (indexed by j), described by the following variables:

- b: world-frame position of the base (root).
- e_1, e_2, e_3: orthonormal basis for orientation of the torso.
- x_1^j, x_2^j: unitary vectors for limb segments.
- t_1^j, t_2^j: Cartesian torques in the knee, elbow, hip and shoulder joints.
- F^j: contact forces constrained in a friction polyhedron with coefficient μ^j.
- T^j: contact torques. The Center of Pressure (CoP) is limited to a square of size w^j.

All these quantities are shown in Figure 4.3. A dynamic model indeed includes derivatives of position vectors as well. Note that parameters s_1^j, s_2^j and s_3^j are used to express the hip and shoulder positions, M and $I = \left[I_x, I_y, I_z\right]$ represent mass and inertial properties of the torso and finally, each link segment has its own geometric and inertial properties

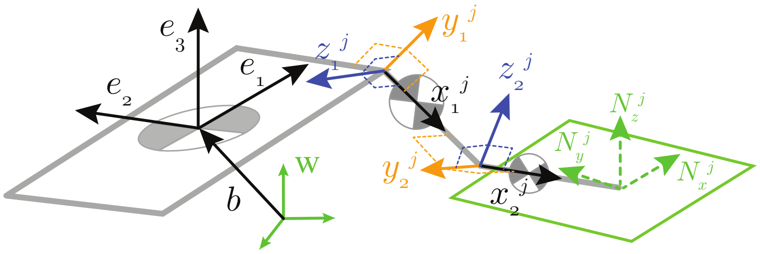


Figure 4.4 – Reconstructing local frames on each body segment to encode joint-space boundaries of articulated joints. These auxiliary orthonormal frames can be either added as decision variables during the optimization to impose joint limits, or calculated offline to convert from vector-space to joint-space.

shown in Figure 4.2. The task parameters are also given by desired contact positions P^j and contact surface coordinates N^j. We skip writing full equations, but principles of (4.6) are simply applied here to describe the relation between variables. The inter-segment interaction forces (formerly f_p and f_c) are indeed resolved by combining equations together. An important property of such modeling technique is adding vectors together to reach the desired end-effector point P^j. This linear constraint potentially simplifies geometry optimization tasks, discussed in the next section.

Note that since most of the humanoid robots are constructed with articulated joints, we need to encode joint limits into the framework as well. To this end, we can introduce auxiliary vectors y^j and z^j for each segment and reconstruct the orthonormal local frame together with x^j. Joint-limits can then be directly added by constraining the dot product of these vectors together or with the given contact frame vectors N^j. A simple demonstration of these local frames is found in Figure 4.4. We do not necessarily need to add derivatives of these vectors unless explicit bounds on joint velocities are desired. Note that Cartesian joint torques t^j can also be projected onto these vectors to find real actuator torques if needed.

4.3 Static posture optimization

In this section, we use models developed earlier to formulate a task of posture optimization. We consider point contacts and drop contact torques T^j to simplify the system, although adding them is straightforward due to linearity. In this case, CoP and rotational friction constraints are linear functions of contact torques and vertical forces (refer to chapter 3). Besides, to give the robot more freedom, we let it rotate the feet (and the wrists) around the normal axis of the contact surface (N_z^j vector in Figure 4.3). This relaxation makes the system redundant but helps rotate the body and avoid restricting joint limits, aiming at finding more optimal postures. The redundancy can be removed by introducing orientation constraints or other simple policies explained later. We also limit our focus on static postures in this chapter but consider six tasks that require very different interactions with the environment. The purpose is to assess the baseline convergence behavior of a minimal setup in symmetric and asymmetric postures as well as in starting from random initial conditions. Joint torque limits can also be added as described earlier, but practically, we found these constraints never activated. We use the model of our robot COMAN with an approximate mass of 30kg, a height of 90cm and joint torque limits of 40Nm.

4.3.1 Optimization setup

Optimization variables

The algorithm has to find position vectors b, e_1, e_2, e_3, x_1^j and x_2^j, contact forces F^j and joint torques t_1^j and t_2^j for each joint j together with auxiliary vectors y_1^j, y_2^j, z_1^j and z_2^j to reconstruct local frames when joint angle bounds are considered. For the joint-space model in a similar optimization setup, we consider the generalized state vector $q \in \mathbb{R}^{22}$, joint torques $\tau \in \mathbb{R}^{16}$ and contact forces F^j.

Objective function

Consists of two dominant terms: the magnitude of contact forces and joint torques. However, we observed that both modeling approaches (joint-space and vector-based) could hardly converge to a unique solution, starting from random initial conditions. This is mainly due to the nonlinearity of the cross products and redundancy in the system. Therefore, we added an auxiliary linear term to lift the base (root) up as much as possible and to orient the limbs along the sagittal vector of the contact surface (N_x^j, shown in Figure 4.3). These terms implicitly reduce joint torques and remove the redundancy mentioned earlier. The objective function for vector-based model is therefore described as:

$$f = Q_\tau(\sum_j |t_1^j|^2 + |t_2^j|^2) + Q_F(\sum_j |F^j|^2) + Q_{lin}(-\alpha^T b + \beta \sum_j (x_1^j - x_2^j)^T N_x^j) \tag{4.8}$$

where the vector $\alpha = \begin{bmatrix}0, 0, 1\end{bmatrix}^T$ denotes upward direction, $\beta = 0.01$ and Q_τ, Q_F and Q_{lin} are 0 or 1, simply enabling or disabling the terms. The parameter β is used to remove the redundancy, but it can indeed affect the overall posture as well, pushing the knees or elbows away from straight postures. Therefore, we choose a small value to minimize this effect yet removing the redundancy. Later, we provide a further analysis and show that the choice of this parameter is not critical. Besides, adding contact orientations to the tasks relieve the need to have any β in the objective function. Note that the same objective function is also used for the joint-space model:

$$f = Q_\tau|\tau|^2 + Q_F(\sum_j |F^j|^2) + Q_{lin}(-\alpha^T b(q) + \beta \sum_j (x_1^j(q) - x_2^j(q))^T N_x^j) \tag{4.9}$$

where the base positions and limb vectors are functions of the generalized state vector q.

Constraints

A set of equality constraints describe the equations of motion, linking all variables together. In the vector-based model additionally, linear task constraints are added for each limb j:

$$b + s_1^j e_1 + s_2^j e_2 + s_3^j e_3 + L_1^j x_1^j + L_2^j x_2^j - P^j = 0 \tag{4.10}$$

while end-effector positions are complex functions of the generalized state vector ($EF^j(q)$) in the joint-space model:

$$EF^j(q) - P^j = 0 \tag{4.11}$$

Motion equations in the vector-based model take the form:

$$\begin{aligned} \sum_j F^j &= M_{tot} g \\ \sum_j P^j \times F^j &= M_{tot}(x_{CoM} \times g) \end{aligned} \tag{4.12}$$

where M_{tot} is the overall body mass and x_{CoM} represents the center of mass position as a linear function of decision variables. Note that for static cases, given that desired task positions P^j are fixed, equations in (4.12) become linear in terms of optimization variables. These equations in the joint-space model appear as:

$$h(q) = \tau + \textstyle\sum_j J^j(q)^T F^j \tag{4.13}$$

where $h(q)$ denotes gravitational forces and $J^j(q)$ represents the Jacobian of each end-effector position. We also have holonomic constraints of the form (4.7) in the vector-based model while these constraints are inherent in the joint-space model. Regarding inequality constraints, we mainly have joint position bounds and friction polyhedra in both models. The former constraint takes the form of simple bounds on optimization variables in the joint-space model while it requires reconstruction of local frames in the vector-based model. In both models, we define friction polyhedra as:

$$\begin{aligned} (F^j)^T N_z^j &\geq 0 \\ \mu^j (F^j)^T N_z^j &\geq |(F^j)^T N_x^j| \\ \mu^j (F^j)^T N_z^j &\geq |(F^j)^T N_y^j| \end{aligned} \tag{4.14}$$

where parameters μ^j are friction coefficients of contact surfaces. We also introduce constraints of the form:

$$x_2^{jT} N_z^j \leq 0 \tag{4.15}$$

Setup	Joint-Space	Vector-Based
variant I	$N = 50, M = 58$	$N = 93, M = 117$
variant II	$N = 50, M = 58$	$N = 69, M = 77$
variant I w.o. torques	$N = 34, M = 42$	$N = 69, M = 93$
variant II w.o. torques	$N = 34, M = 42$	$N = 45, M = 53$

Table 4.1 – Size of optimization problems in different configurations. Here N stands for the number of variables and M stands for the total number of equality and inequality constraints. The variant III optimizer runs variant II and then variant I optimizers in the first and second stages respectively. Note also that joint limits are implemented as simple bounds on variables in the joint-space model, not counted here. Disabling linear terms or contact forces in the objective does not influence the problem size.

in both models to make sure that the limbs stay in front of contact surfaces. We also disable contact forces of floating links simply by multiplying them with constant binary mask parameters, determined by task requirements.

Multi-stage optimization

Despite some linear constraints and quadratic terms in the objective, our optimization setup remains non-convex due to the holonomic quadratic equality constraints and joint angle boundaries in the vector-based model. To further simplify the analysis, we consider three variants of optimization: (I) full formulation as described, (II) a version without joint boundaries and (III) a version in which we solve the problem in two stages: once without joint boundaries and then using the solution as a warm starter in a second stage where boundaries are added. We further consider enabling and disabling the torques and will demonstrate that even the simple linear objective can do the job most of the time. These strategies will be applied on both joint-space (equations (4.9), (4.11), (4.13), (4.14) and (4.15)) and vector-based (equations (4.8), (4.10), (4.12), (4.14) and (4.15)) optimizations respectively.

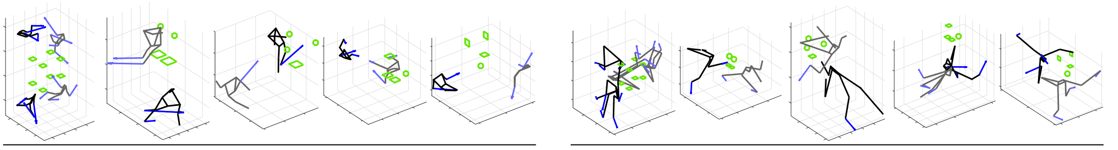
Implementation

All optimizations for both models are implemented using SNOPT (Gill et al., 2005), with a maximum of 250 iterations, feasibility tolerance of 10^{-6} and accuracy level of 10^{-6}. Singularities are handled by a constant damping of the Hessian matrix implemented in this package. We use a dedicated Matlab code to generate vector-based model equations, implemented in c++. Forces and torques are all normalized by the body weight ($M_{tot}g$) and lengths by the body length in the objective and constraints. We avoided including these terms in the equations for the sake of simplicity. Problem sizes are reported in Table 4.1

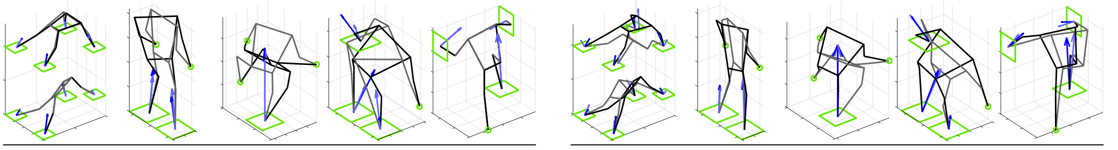
4.4 Results

Using the tools and methods explained, we aim at comparing the two modeling techniques over the complex task of posture optimization where computational aspects play an

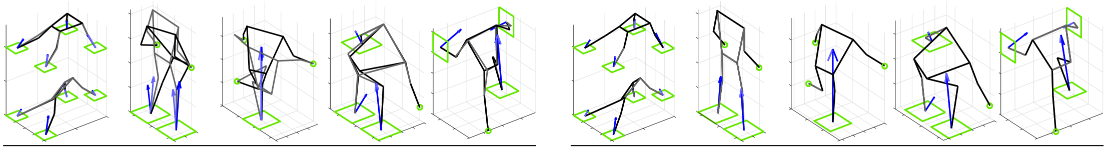
A) Random starting points



(avg. 209 ± 76ms) B) Variant I: full problem (avg. 107 ± 44ms)



(avg. 342 ± 110ms) C) Variant II: removing joint limits (avg. 21 ± 11ms)



(avg. 536 ± 161ms) D) Variant III: resolving joint limits in two stages (avg. 46 ± 13ms)

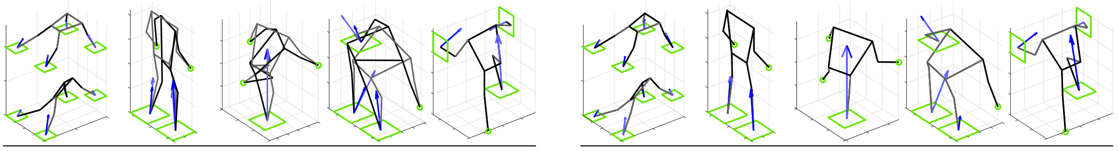


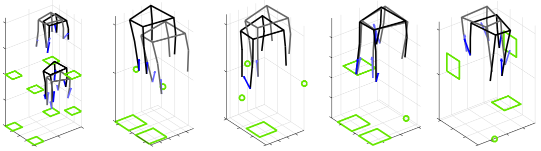
Figure 4.5 – Global convergence analysis for the joint-space (left) and vector-based (right) models with $Q_\tau = 1$, $Q_{lin} = 1$, $Q_F = 0$ and $\beta = 0.01$ in the objective function. Each row consists of two batches of six typical tasks for humanoid robots: (from left) symmetric and asymmetric quadruped postures, normal standing, single support, picking an object on the ground and stair climbing with supports from the walls. A) a typical random starting point. B) Variant I: considering joint limits in the first run leads to local minima. C) Variant II: removing joint limits still does not solve the problem. D) Variant III: adding joint limits back in the second stage of optimization can not lead to a global optimum too. In these figures, we plot two trials of each optimization to distinguish cases which are globally convergent. This is separately verified over 1000 trials, and the average time spent in the optimizer over all tasks is mentioned on top of batches. The vector-based model performs much faster and is more convergent.

important role. Our analysis covers convergence behavior, i.e. finding local or global solutions, perturbation analysis, and computation time performance.

4.4.1 Global convergence

Starting from random initial points, we applied the three optimization variants to various interesting tasks: symmetric and asymmetric quadruped postures, normal standing, single support, picking an object on the ground and stair climbing with supports from the walls. The results are shown in Figure 4.5 for both modeling techniques. Although the starting points are far from desired, both models can lead to meaningful postures. Considering

A) Perturbed starting point



B) Variant III: two-stage optimizer (421 ± 129ms)

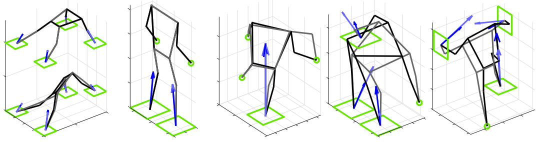


Figure 4.6 – Sensitivity of the joint-space model to initial conditions. (A) The starting points are only slightly different, (B) but the optimization seems very sensitive, converging to different local minima (two trials per task). Here we used variant III optimization with same parameters as Figure 4.5. The vector-based model converges to the same solution every time, not interesting to show.

joint boundaries (variant I) leads to local minima in both models. Removing these constraints in variant II, however, results in a different behavior. The vector-based model converges to the same solution every time while the joint-space model still gets trapped in local minima. We took advantage of this property in variant III and reintroduced joint-boundaries in a second optimization stage which starts from the solution of the boundary-free optimization. As a result, the vector-based model can now find a unique solution while the joint-space model fails. The optimization problems are not convex, and a formal uniqueness proof for the solution is missing. But over a large number of random starting points (10^3 trials), we found the vector-based model convincingly convergent for the popular humanoid tasks considered. We defined uniqueness by a threshold (10^{-3}) on the maximum standard deviation of the final solutions obtained (over different dimensions). More accuracy levels can be achieved by tightening the stopping criteria in the optimization.

4.4.2 Sensitivity to initial condition

This test could be performed in different ways. Here, we considered a single normal starting point and perturbed it randomly by 10%. Thanks to global convergence, the vector-based formulation behaves in the same way as before while the joint-space formulation again converges to different local minima. Figure 4.6 demonstrates few examples. Perturbation test can quantify how continuous the model is if used as a basis in a higher level optimization frameworks. Jumping between different local minima would be confusing for higher level planners, and it makes the joint-space formulation less suitable for planning.

4.4.3 Sensitivity to optimization parameters

So far, all optimizations were using both the linear and joint torque terms in the objective function. It becomes interesting to test them separately, since many similar posture optimization methods in literature rely on minimizing joint torques (Howard et al., 2014; Marler et al., 2011). Enabling contact force terms produce less optimal solutions (Figure 4.7), since they are in conflict with joint torques. When disabling joint torques however, these terms are necessary to produce optimal contact forces. In this case, the results are less optimal, but still meaningful. Without linear terms however, the vector-based model is not globally convergent anymore (Figure 4.7). A variant II optimizer without joint torques and bounds is the fastest and produces convincing postures, useful for robots with large range of joint motions (Figure 4.7). Remember we chose a small β value to reduce the effect of preferred limb orientation on other tasks which stretch the knees and elbows. In Figure 4.7, we show how increasing β can flex the knees and elbows. Our choice of $\beta = 0.01$ seems less conservative and thus reasonable. Disabling β however leads to ambiguities (Figure 4.7) that could be resolved by introducing feet/wrist orientation constraints alternatively.

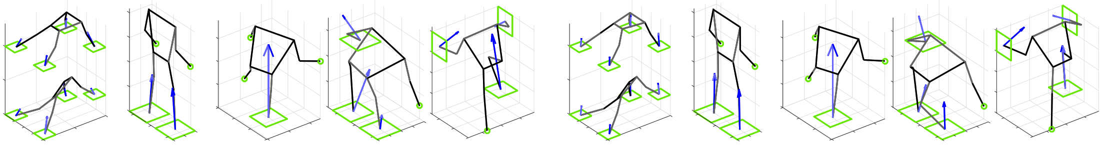
4.4.4 Computation time

In both Figure 4.5 and Figure 4.7, we have reported the average timing performance over 10^3 random trials. Adding joint bounds slows down the process in the vector-based model, regardless of joint-torque terms actually. Producing crouched postures seem less expensive while stretched-limb postures seem more costly (Figure 4.7). It is preferred to keep torque terms for maximum optimality and linear terms to ensure global convergence, even from very different initial conditions. Although the first stage in variant III is globally convergent, we can not still say whether the solution of the second stage is the global optimum of the full problem (variant I). The uniqueness of results is very interesting, however, considering the nonlinear and non-convex properties of multi-contact posture optimization. Our minimal setup (variant II) can reach up to 21ms on average, starting from totally random initial conditions. This could be further improved to 12ms by disabling torque terms and still generating meaningful postures for some tasks. To the best of our knowledge, for such randomly initialized non-linear optimizations, this computation speed was not achieved before. This is due to unrolling complex equations and the convergence properties that our formulation offers.

4.5 Discussion

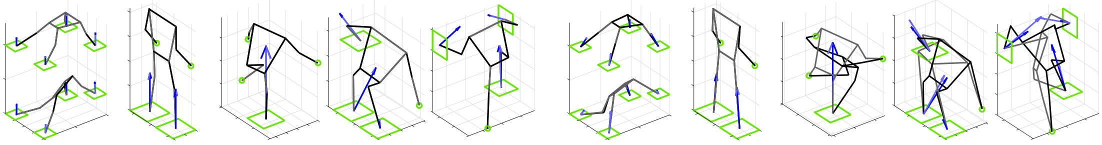
We proposed a fast algorithm that finds unique, feasible and sub-optimal solutions in two stages. The idea of using stages was inspired from the well-known simulated annealing technique (Van Laarhoven and Aarts, 1987) where constraints are tightened gradually to better guide the solution towards a feasible point. The result might not be the global minimum of the full problem, but in our case yet meaningful. The vector-based model offers faster speeds than joint-space formulations in our case study, though the core novelty of this chapter lies in our molecule-inspired modeling technique that distributes

A) Normal: (48 ± 17ms) B) Including forces: (49 ± 18ms)



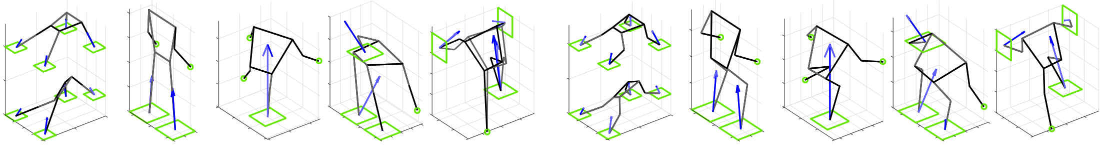
$Q_\tau = 1,\ Q_{lin} = 1,\ Q_F = 0,\ \beta = 0.01$ $Q_\tau = 1,\ Q_{lin} = 1,\ Q_F = 1,\ \beta = 0.01$

C) w/o torques: (45 ± 18ms) D) w/o linear terms: (146 ± 86ms)



$Q_\tau = 0,\ Q_{lin} = 1,\ Q_F = 1,\ \beta = 0.01$ $Q_\tau = 1,\ Q_{lin} = 0,\ Q_F = 0,\ \beta = 0.01$

E) w/o limb orientation: (66 ± 20ms) F) Strong limb orientation: (32 ± 12ms)



$Q_\tau = 1,\ Q_{lin} = 1,\ Q_F = 0,\ \beta = 0.00$ $Q_\tau = 1,\ Q_{lin} = 1,\ Q_F = 0,\ \beta = 0.05$

Figure 4.7 – Sensitivity of vector-based optimization to the choices of objective function parameters. A) Normal choice of parameters similar to Figure 4.5. B) Including contact forces results in minimal and more uniform contact forces, often increasing joint torques (e.g. object picking task). C) Excluding torque terms leads to slightly crouched postures. Note that contact force terms are needed to avoid arbitrary large forces. Formerly, the torque terms could indirectly optimize contact forces as well. D) Excluding linear terms can still generate optimal postures, but without global convergence properties. E) Disabling β will remove desired limb orientation preference and lead to optimal and singular postures, but it can cause ambiguities in leg/arm orientations (e.g. stair climbing task). F) Too large β values result in crouched postures, but faster convergence.

the complexity of the robot to improve the convergence behavior. We break the chain of the robot into multiple segments and link them together with linear constraints. This makes the Jacobian and Hessian matrices simpler in the optimization and speeds up the convergence. The vector-based model has only first and second order polynomial terms while the joint space model has many sine and cosine terms multiplied together. Extensive analysis and comparison demonstrate superior speed and convergence properties in the proposed vector-based modeling technique, despite typically larger problem sizes.

The proposed modeling technique is fundamentally different from task-space formulations of Khatib (Khatib, 1987), although both deal with Cartesian variables. We reconstruct the model via individual vectors that represent body segments while task-space formulations use Jacobians to convert delta-motions from joint-space to task-space and vice versa. The goal of this chapter is not to speed up traditional time-integration IK or ID transformations. Those problems are linear with respect to velocities or accelerations where warm-starting or decomposition techniques (Escande et al., 2014) can reduce optimization times even below 1ms. Linearizing our vector-based model can indeed lead to such performance as well for online control. In this chapter, instead, we directly find final "optimal" postures which involve finding positions, joint torques, and contact forces altogether. This problem is much larger in size and very nonlinear with respect to optimization variables. We also do not have pre-calculated solutions to speed the process by warm-starting. Despite these limitations and starting from totally random initial conditions, the proposed vector-based approach can find unique final postures (46ms, Figure 4.5) much faster than time-integration (1-2s) (Escande et al., 2014) or direct nonlinear optimizations in the joint-space (200-500ms, Figure 4.5). Despite recent advances in fast motion planning which typically use abstract variables (CoM, momentum, contacts etc.) (Ponton et al., 2016), we can not provide any performance comparison yet. In this chapter, we have not set up any motion planning problem. It remains interesting for future work to use vector-based modeling techniques in motion planning too. Due to better convergence properties, we expect to be able to add more dynamics details without compromising optimization times.

To prioritize different tasks and deal with over-constrained cases, we found soft weighting matrices (Sugihara, 2011) robust against sensory noises in our torque-based controller using ID formulations (chapter 3). In this work, all end-effector tasks are coded as equality constraints (assuming feasible tasks). It would be interesting to incorporate weighting matrices or hierarchical priorities of time-integration approaches (Escande et al., 2014; Herzog et al., 2014) in our nonlinear optimizations. These policies would better handle unsolvable or over-constrained situations. In these cases, it becomes interesting to further analyze optimization times and final errors to make sure the algorithm offers a constant timing like (Sugihara, 2011) in case of infeasible tasks. Compared to joint-space models, representing the robot by vectors leads to simpler inequality constraints for collision avoidance. Due to a non-convex nature (Alrifaee et al., 2014), however, these constraints further complicate convergence properties in both joint-space and vector-based optimizations. Since the goal is limited to investigate model properties in this chapter, we consider including them in future work.

In future, we consider exploiting our particular problem structure (which consists of few quadratic and many linear terms) to make efficient use of sparsity and further reduce optimization times. Apart from this, we aim at deriving full-body forward and inverse dynamic equations, inspiring from popular techniques in computational chemistry. The proposed vector-based technique can also facilitate modeling of simpler robots like manipulators, bipeds or monopods.

5 Motion Planning

This chapter extends the idea of point-wise modeling to motion planning. Remember we mentioned in chapter 4 that expressing mechanical equations with Cartesian points and binding them together with length constraints can potentially unravel the complexities of joint-space models and facilitate optimizations. In this chapter, we explore possibilities of using our simple modeling technique in multi-contact motion planning scenarios to generate faster locomotion speeds. We do not notably propose any better contact sequence search algorithm here. We only use a given motion sequence but formulate an optimization problem that can adjust footstep locations, phase timings, and all other trajectories in a physically feasible and meaningful way. We use a five-mass model to better describe internal dynamics and momentum changes in very dynamic motions. Our setup allows for deciding the overall motion duration which together with the number of steps can produce very static or very dynamic motions. Other state of the art methods use a combination of centroidal dynamics and inverse kinematics. The proposed model can find similar dynamic motion plans only by including swing dynamics via the five-mass model. We encode fewer kinematic details and introduce a trade-off in terms of model complexity, but make optimizations faster. This planning method together with a perception pipeline and inverse dynamics controller can realize dexterous locomotion behaviors on the real robot. We consider such integration effort for future work[1].

Publication Note: The material presented in this chapter is adopted from:

- Jonathan Arreguit, Salman Faraji and Auke Jan Ijspeert. *"Fast multi-contact whole-body motion planning with limb dynamics."* in preparation, 2018.

The optimization setup in this work was implemented by Jonathan Arreguit in a master project, continuing our previous point-wise modeling technique (refer to chapter 4). The role of second author in this paper was to perform a literature review and to write the manuscript.

---

[1] All the videos of this chapter could be found at https://youtu.be/0YjVNQkt_kc

5.1 Background

Planning multi-contact locomotion in unstructured environments involves various challenging tasks, ranging from perception, environment modeling, motion planning and compliant control. Despite well-established techniques for different blocks in this complex pipeline (Kaiser et al., 2016; Rebula et al., 2007), the planning part is yet time-consuming and computationally slow compared to human locomotion. The underlying complexity has different folds: limited foothold locations, robot's floating-based dynamics, unilateral frictions, reachability constraints, collisions and planning the sequence of contacts. These complexities produce a very complex mathematical problem in case all mechanical details are considered. Solving such problem for a simple task of sitting down on a chair may, in fact, take up to a few hours (Lengagne et al., 2013; Al Borno et al., 2013). However, it is possible to speed-up this problem with different simplifications and achieve faster planning speeds while compromising for certain features. Taking inspiration from state of the art algorithms, we propose a simple formulation based on the molecule-inspired modeling of chapter 4 that can improve dynamics of the motion in an efficient and meaningful way. Before introducing the proposed method, we review the literature by identifying essential challenges involved and discussing possible improvements.

5.1.1 The curse of dimensionality

Humanoids or multi-legged robots have floating-based dynamics described by $(n+6)$ states where n refers to the number of joints and 6 denotes the dimension of global states which describe the position and orientation of the robot in world-frame. Finding time-trajectories for all these degrees of freedom produces a high-dimensional planning problem. One of the simplifications widely popular in the literature is to use single-mass models which could approximate the overall behavior in slow motions (Hauser et al., 2005). In these models, one can optimize all contact forces to produce a physically meaningful Center of Mass (CoM) trajectory. However, these forces produce a moment on the body according to Centroidal dynamics (Orin et al., 2013). Enforcing the resulting angular momentum rate to zero is possible, but it is more natural to associate it with whole-body inertia like (Winkler et al., 2018) (shown in Figure 5.1A). This simplification is close to reality if the legs are light-weight compared to the main body, e.g., in quadruped robots like ANYmal (Winkler et al., 2018). However, in many humanoid robots including Coman (Colasanto et al., 2012), the legs and arms are comparably heavy.

5.1.2 Simplified dynamics, full kinematics

Simplified models can speed-up calculations by reducing problem dimensions. However, they possibly over-simplify fast motions and especially dynamic effects induced by heavy limbs. To mitigate this problem, it is possible to plan end-effector and CoM trajectories in Cartesian space for a simplified model, and match it kinematically with the full model (shown in Figure 5.1B). Mordatch et al. (Mordatch et al., 2012) converted such abstract trajectories to joint angles via inverse kinematics. By feeding the resulting joint angles into a full dynamics model, they were able to find the whole-body momentum rate which

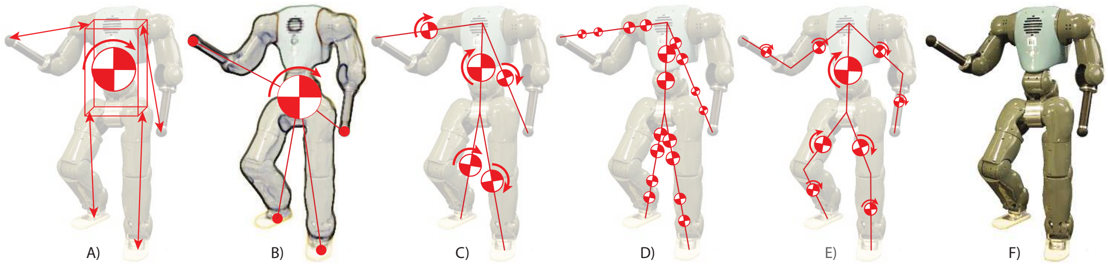


Figure 5.1 – Different template models used in the literature: A) A single mass and inertia with kinematic length constraints (Winkler et al., 2018), B) Centroidal dynamics with full kinematics (Dai et al., 2014), C) the proposed five-mass model with stretchable limbs, D) same as previous, but representing inertias with two masses, E) a nine-mass model with elbows and knees and F) the full robot COMAN (Colasanto et al., 2012). The borders in B) indicate that the simplified Centroidal dynamics match with the full system by using full kinematics.

was then constrained to be equal to the abstract momentum rate. In their approach, there is no need to explicitly include individual joint angles and contact forces in the optimization. These variables are directly calculated from the Cartesian trajectories through inverse kinematics and inverse dynamics respectively. Mordatch et al. used soft constraints to enforce dynamic equations originally (Mordatch et al., 2012). However, Dai et al. included the contact forces explicitly and used hard constraints (Dai et al., 2014). These methods assume no limitation for the joint torques which significantly reduces the number of inequality constraints needed. They also need very efficient kinematic solvers, possibly in closed-form, which could be iteratively used inside bigger optimizations. Herzog et al. also used a similarly decoupled dynamic-kinematic approach and achieved faster optimizations with more efficient solvers (Herzog et al., 2016).

5.1.3 Trajectory parameterization

Due to nonlinearity, it is very common to use splines to encode trajectories over time. This can be done for joint positions (Al Borno et al., 2013), Cartesian positions (Mordatch et al., 2012) and even contact forces (Winkler et al., 2018). However, it requires binding trajectories together at regular intervals via kinematic and dynamic constraints (Winkler et al., 2018). An alternative is to define all variables at every time-sample and link them via integration constraints (Dai et al., 2014; Herzog et al., 2016; Posa et al., 2014). This gives more flexibility in the design of trajectory shapes but increases the dimensionality of the problem. A vast part of literature also aims at optimizing transition postures only, which further simplifies the spline method (Chung and Khatib, 2015; Brossette et al., 2013; Bouyarmane and Kheddar, 2011; Werner et al., 2016). Although these methods may potentially be faster from a computational point of view, they are prone to producing quasi-static motions.

5.1.4 Fixed contact sequences

Given reachability constraints and environmental complexities, finding a kinematically and dynamically feasible set of contacts is not always trivial. The Linear Inverted Pendulum (LIP) model (Hemami and Golliday, 1977) can provide closed-form equations which enable for online Model Predictive Control (MPC) in flat-ground walking conditions (Faraji et al., 2014a; Herdt et al., 2010b), but it is too limited for multi-contact locomotion. Without considering dynamical consistency, it is possible to perform a faster kinematic search for footstep locations and orientations in a first stage. Rapidly-exploring random trees can achieve reasonable plans with simple distance constraints (Werner et al., 2016; Chung and Khatib, 2015) and energy-based cost functions (Huang et al., 2013), but these approaches are also limited to bipedal walking. A library of recorded human motions can give a good initial guess (Borràs et al., 2017). However, the generalization to different environments remain a challenge. Hauser et al. proposed a graph search method in the first stage to find a path of (kinematically) adjacent nodes spanning between the start and end points (Hauser et al., 2005). The second stage in their algorithm then fixed dynamic consistencies. It is possible to link them back after computing them separately (Bouyarmane and Kheddar, 2011), but at the cost of increasing computations in mixed-integer optimizations. However, extraction of convex hulls in the environment to represent supporting surfaces would allow for contact adjustment and result in faster optimization problems (Brossette et al., 2013). Given a simple environment model represented by such convex hulls, one can use mixed-integer convex optimization to find bipedal walking footholds (Deits and Tedrake, 2014) or more complex multi-contact motions (Ponton et al., 2016). The approach presented in (Deits and Tedrake, 2014) tackles kinematics and dynamics separately in different optimization stages while (Ponton et al., 2016) combines both in a single mixed-integer optimization. Depending on the number of existing support surfaces, the discrete search problem may grow in size exponentially.

5.1.5 Emergent contact sequence

To find more elegant contact sequences with dynamic consistency and flexible timing, Mordatch et al. introduced the Contact-Invariant Optimization (CIO) approach (Mordatch et al., 2012). To explain it briefly, consider a single contact point of the robot. Without any assumption on the phase sequence or timing for this point, it is possible to minimize the multiplication between the present contact force and the distance (to the adjacent environment surface) at each instance of time. When the distance is zero, the algorithm is allowed to increase the contact force while it is forced to zero when the distance is non-zero (i.e., a swing motion). Both contact positions and forces are indeed explicit optimization variables. While this method was initially formulated with soft constraints in (Mordatch et al., 2012), Posa et al. used complementarity constraints and formulated the problem with hard constraints which allowed for sliding contacts as well (Posa et al., 2014). This formulation was later used in (Dai et al., 2014) to produce arbitrary contact sequences for different tasks on the Atlas robot. A more restrictive alternative to this flexible approach is to look at each contact individually. The sequence of each contact is fixed, *swing* and then *stance* and vice versa. Winkler et al. assumed the same number of switches for each contact and hard-coded the trajectories in each phase with splines (Winkler et al., 2018). This approach considers variable phase durations per contact point which sum up to the same total motion duration determined beforehand. This approach allows for arbitrary contact sequences like the CIO method, but with a predetermined number of contact switches.

5.1.6 Environment complexity

We mentioned that representing the environment with simple convex-hulls is useful in discrete search methods (Werner et al., 2016; Deits and Tedrake, 2014). The approach presented by Winkler et al. (Winkler et al., 2018) combines all these surfaces and approximates them with a smooth height-map which allows for continuous optimizations. However, in case of large gaps or restricted footholds, a good initial guess is needed to avoid converging to local minima. Mordatch et al. calculated a soft distance-to-surface measure in the CIO method to consider environment complexities, but their simulations were restricted to flat-ground or stair-climbing cases and not traversing huge gaps. A similar limitation also appears in the approach of (Dai et al., 2014) and (Posa et al., 2014) where the contact sequence in restricted foothold cases or monkey-bar scenarios might be pre-determined. Therefore, the CIO approach and the alternative proposed in (Winkler et al., 2018) are probably powerful in finding emergent behaviors in simple environments, but more prone to local minima in complex terrains.

5.1.7 Rooms for improvement

As discussed so far, popular approaches proposed in the literature might share certain positive aspects, but they might be limited from other perspectives. An ideal planner should be able to re-plan the motion online within a few milliseconds and ideally adjust the timing, contact locations and even contact surfaces which involve discrete decisions. Besides, it is expected to generate dynamic and fast motions by considering robot dynamics. Most of the methods proposed in the literature cannot reach such time performance due to many reasons. The mixed-integer nature of the problem is probably the most restricting factor. Additionally, although simplified models speed up the problem, they cannot produce dynamic motions for robots with heavy limbs unless the full dynamics model (Lengagne et al., 2013) or the kinematic model is used (Mordatch et al., 2012; Dai et al., 2014; Herzog et al., 2016). Consider Figure 5.1 which summarizes the important modeling methods used in literature. The single mass-inertia model of Winkler et al. (Winkler et al., 2018) is shown in Figure 5.1A together with the decoupled dynamic-kinematic approach of (Mordatch et al., 2012; Dai et al., 2014; Herzog et al., 2016) shown in Figure 5.1B. These approaches speed-up simulations considerably compared to using the full model (Lengagne et al., 2013) shown in Figure 5.1F. However, it is possible to elaborate the simplified model slightly to better account for internal body dynamics in fast motions.

5.1.8 Contributions, novelties

In this chapter, by splitting the modeling and discrete search challenges apart, we only focus on the simplified model and propose a method to improve the performance from this perspective. We restrict the study to finding dynamic trajectories and use a predetermined contact sequence. We also use simple collision avoidance techniques and do not consider possible knee/elbow contacts with the environment. We propose a simple five-mass model with a torso and four limbs to represent different body parts shown in Figure 5.1C. Using the previously developed optimization technique of chapter 4, we set up a motion planning problem with contact forces, phase times, end-effector, pelvis and mid-shoulder positions as main variables. These trajectories are bound together with dynamic equations, geometric length constraints, reachability constraints and contact friction cones. We show that our method can plan dynamic multi-contact motions in the order of few hundred milliseconds with adjustable contact location and phase timing properties. Although we use a predefined contact sequence which could be the outcome of human motion libraries (Borràs et al., 2017), random search (Huang et al., 2013; Chung and Khatib, 2015; Werner et al., 2016) or mixed-integer problems (Deits and Tedrake, 2014), the novelty of our method lies in the new simplified model which could be used in many other optimization problems as well. It can efficiently describe internal robot dynamics while allowing for implementation of reachability constraints. Despite using time-integration to parametrize trajectories instead of splines which are probably more efficient, we show that we can achieve fast optimization times comparable to single-mass simplified models. The next section describes the model and optimization formulation in details. Next, we demonstrate results over different planning scenarios and analyze the performance from various perspectives. Finally, we conclude with a discussion of advantages and limiting factors as well as future directions for potential improvements.

5.2 Problem formulation

We start this section by introducing our simplified mechanical model. This model is directly formulated with Cartesian position variables instead of joint angles. In chapter 4, we discussed that such modeling technique is popular in computational chemistry to calculate complex molecular conformations. Despite different scales of mechanical forces in humanoids versus molecules, the overall geometries have important similarities, especially the tree-like topology and the fixed-length constraints between points. We formulated a posture optimization problem in which world-frame positions of skeletal joints were the main variables instead of actual joint angles. We considered a mass and inertia for each limb which spans between two adjacent Cartesian points, constrained to have a fixed length inside the optimization. Compared to joint-space models, this new formulation is sparse and mathematically less complicated. Therefore, optimizations of arbitrary multi-contact postures could be done with better convergence properties and up to ten times faster with the new formulation (refer to chapter 4).

In the present work, however, we use a simpler five-mass model (shown in Figure 5.1C) for motion planning. This model is derived from a nine-mass model similar to the

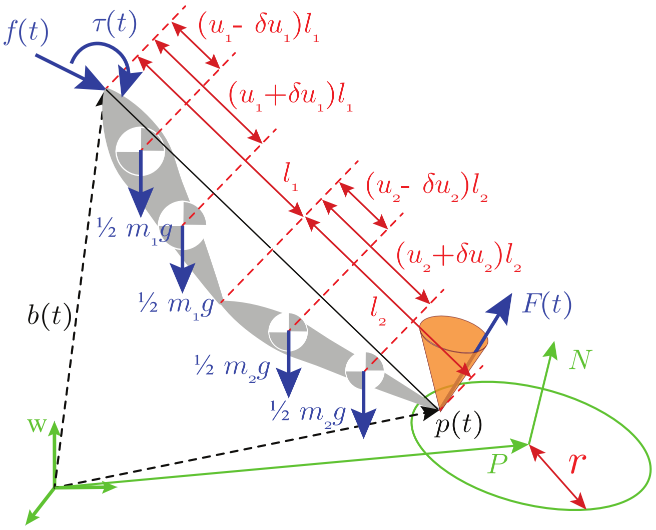


Figure 5.2 – Demonstration of a single limb in the robot with internal forces $f(t)$ and $\tau(t)$, and external forces $F(t)$. This limb could be one arm for example without hands, where the vector $b(t)$ is shoulder position and the vector $p(t)$ is contact position both expressed in the world-frame w. The surface can be approximated by a circle at position P, normal vector N and radius r. The friction cone of this contact is also described by a coefficient μ.

humanoid anatomy (Figure 5.1E). Also, the inertias we assume for each body segment are equivalent to splitting the mass into two half-masses shown in Figure 5.1D. We consider stretchable limbs in the robot where masses also change location along the limb proportionally. More formally, we assume there are two ideal prismatic (massless) actuators on both sides regulating the entire limb length proportionally. We use a fixed length for the torso segment and restrict limbs to a certain range to approximate the effective workspace. The upper limit of this range encodes reachability constraint while the lower limit prevents self-collision.

5.2.1 Model formulas

To derive the equations, consider Figure 5.1D. As mentioned earlier, we consider mass and inertial for both limb segments (upper arm and forearm as well as thigh and shank segments). Since in human (de Leva, 1996) and our robot COMAN, transversal moments of inertia in each limb segment are much smaller than sagittal and longitudinal, and since the last two components are almost the same, we approximate each limb segment by two half-masses. This is equivalent to considering only two nonzero diagonal elements in the inertia matrix. Consider one limb of the robot shown in Figure 5.2 which attaches to the main body at point $b(t)$ (representing the pelvis or the mid-shoulder point), and creates contact with the environment at point $p(t)$. To improve readability of the chapter, we avoid using indices for individual limbs. In our simplified model, we only consider contact forces $F(t)$ and not wrenches. We leave this actuation possibility (realized by ankle and wrist joints) to a lower-level inverse-dynamics algorithm for a better track of the simplified trajectories. We also assume point-contacts which ideally prevent transversal moments. However, the proposed optimization framework is ideally able to consider all these moments and include constraints to ensure their feasibility with respect to the foot/hand size. Coming back to Figure 5.2, further assume that internal forces $f(t)$ and torques $\tau(t)$ are applied to the limb at point $b(t)$. We denote segment masses by m, lengths by l, relative mass positions by $0 \leq u \leq 1$ (refer to Figure 5.2), and sagittal and longitudinal inertias by i. Splitting the half-masses by δul around the actual mass location, one can obtain:

$$\delta ul = \sqrt{\frac{i}{ml^2}} \tag{5.1}$$

Individual limb segments are identified by subscripts 1 and 2. It is obvious that the overall limb length L and mass M are:

$$L = l_1 + l_2, \quad M = m_1 + m_2 \tag{5.2}$$

The overall relative limb mass location U is:

$$U = \frac{m_1 u_1 l_1 + m_2(l_1 + u_2 l_2)}{(l_1 + l_2)(m_1 + m_2)} \tag{5.3}$$

and the overall inertia around this mass is calculated by:

$$\begin{aligned} I \;&=\; \frac{m_1}{2}(UL - (u_1 - \delta u_1)l_1)^2 + \frac{m_1}{2}(UL - (u_1 + \delta u_1)l_1)^2 \\ &+\; \frac{m_2}{2}(UL - l_1 - (u_2 - \delta u_2)l_2)^2 + \frac{m_2}{2}(UL - l_1 - (u_2 + \delta u_2)l_2)^2 \end{aligned} \tag{5.4}$$

which could be described in the simplest form as:

$$I = i_1 + i_2 + \frac{m_1 m_2}{m_1 + m_2}((1 - u_1)l_1 + u_2 l_2)^2 \tag{5.5}$$

Now, having L, M, U and I as a function of individual segment properties, we can define $x(t) = p(t) - b(t)$ and write Newton equations:

$$\begin{aligned} F(t) + f(t) + M(g - U\ddot{x}(t) - \ddot{b}(t)) &= 0 \\ \tau(t) + x(t) \times ((1 - U)F(t) - Uf(t) - \frac{I}{|x(t)|^2}\ddot{x}(t)) &= 0 \end{aligned} \tag{5.6}$$

These equations describe normal rigid body dynamics. We use $|x(t)| = L$ for the torso, since such constraint is explicitly included in the optimization. For other stretchable limbs, although $|x(t)|$ is variable in the range $L_{min} \leq |x(t)| \leq L$, we still use $|x(t)| \approx L$ which is an approximation, but simplifies the symbolic equations considerably.

The entire robot model is composed of five body segments (torso + four limbs) whose properties are calculated as described before. For each segment, we have to include one equality or inequality in the optimization to control the length. However, endpoint positions are all decision variables in the optimization to be found. Note that in the pelvis and mid-shoulder points, the sum of internal forces and torques are zero. Therefore, we can combine all dynamic equations of the limbs (i.e., the set of equations (5.6) for each limb) and obtain the six usual Centroidal dynamic equations. In the single mass models used in (Mordatch et al., 2012; Dai et al., 2014; Herzog et al., 2016), CoM and end-effector positions, as well as contact forces, are optimization variables. The extra computational cost of our model is only replacing the single CoM position with two pelvis and mid-shoulder positions. The rest of six Centroidal dynamic equations remain present. The advantage of having individual limb dynamics is that end-effector movements can now directly influence the six equations, whereas the approaches in (Mordatch et al., 2012; Dai et al., 2014; Herzog et al., 2016) implicitly create this linkage between the inverse kinematics and the full dynamical model. In these approached, more precision could be obtained at the cost of more calculations. We avoid including all the model details in this section for the sake of readability and only provide our optimization problem in abstract form.

5.2.2 Optimization setup

As mentioned, we use a predefined contact sequence in our planning scenarios. The sequence naturally starts from a given state and ends in a terminal posture. The environment is also modeled by contact surfaces given in circular shapes with certain positions P, radius r and normal vectors N similar to (Chung and Khatib, 2015) (refer to Figure 5.2). The sequence also involves the rhythm of contact changes while directly assigning contact points in the robot to those contact surfaces. The optimization can then find dynamically consistent end-effector position and force trajectories, proper contact locations inside the surfaces and variable phase durations. We divide the motion into M phases in which both the upper and lower limbs either perform a swing motion or establish a contact and receive supporting forces from the environment. Each phase is divided into K sub-phases which aim at providing a good resolution for dynamic and kinematic constraints. Sub-phases (indexed by i where $1 \leq i \leq MK$) have variable durations h_i encoded as optimization decision variables which all sum to MT. Here, the parameter T represents an average phase duration and determines the frequency of motion. Exact phase durations h_i are not forced to be exactly equal to T however. End-effector, pelvis and mid-shoulder positions p_i^j and velocities $\dot{p}_i^j$ (where $1 \leq j \leq 6$) are also decision variables at the beginning of each sub-phase. Therefore, trajectories are formed by time-integration constraints over individual sub-phases which form a direct optimization setup (Posa et al., 2014). The contact points are free to move inside the contact circles, and contact forces F_i^j should lie inside friction cones. We also define accelerations by differentiation of velocities $h_i \ddot{p}_i^j = \dot{p}_i^j - \dot{p}_{i-1}^j$. Altogether, the optimization setup is as follows:

$$
\begin{array}{cl}
\min\limits_{h_i, p_i^j, \dot{p}_i^j, F_i^j} \sum\limits_{i,j} h_i \ddot{p}_i^{j\,2} + h_i F_i^{j\,2} + H(p_i^j) & \text{(objective)} \\
s.t. & \\
\sum h_i = MT & \text{(motion duration)} \\
\Gamma(p_i, \ddot{p}_i, F_i) = 0 & \text{(dynamics)} \\
L_{min} \leq L(p_i) \leq L_{max} & \text{(kinematics)} \\
G(p_0, \dot{p}_0) = S_0 & \text{(initial condition)} \\
\text{Contact phase:} & \\
p_i^j \in S(P_k, N_k, r_k) & \text{(circular surfaces)} \\
F_i^j \in F(P_k, N_k, \mu_k) & \text{(friction cones)} \\
\dot{p}_i^j = 0 & \text{(no sliding)} \\
\text{Swing phase:} & \\
F_i^j = 0 & \text{(free swing)} \\
h_i \dot{p}_i = p_i - p_{i-1} & \text{(time integration)} \\
C(p_{i+[\frac{K}{2}]}) \geq 0, \quad i \bmod K = 0 & \text{(collision)}
\end{array}
\tag{5.7}
$$

Here, S_0 is the initial state. The vectors p_i and F_i contain individual quantities p_i^j and F_i^j. The function $G(.)$ maps Cartesian points to a given state and $L(.)$ encodes length constraints. The vectors L_{min} and L_{max} simply contain limb length constraints. For the torso length, the corresponding elements are set to be equal to implement an equality constraint. The function $\Gamma(.)$ encodes dynamic equations (5.6) for each sub-phase i, $F(.)$ denotes friction cones and $S(.)$ denotes contact circles. In practice, we define accelerations by a direct differentiation of velocities and thus, exclude them from the list of optimization decision variables. Zero velocities, accelerations and forces are also defined parametrically to make the optimization faster.

We normalize forces and accelerations by total mass and gravity in the objective function. The function $H(p) = -\hat{z}^T p$ is only applied to the mid-shoulder point to lift the robot up and make the motion more energy optimal. The vector $\hat{z}$ denotes world-frame vertical direction. Since we do not have knee joints in our five-mass model, we cannot minimize knee torques to naturally lift the robot up. In chapter 4 where we optimized postures with a more complete model that had knee joints, we showed that such lifting term could produce similar postures compared to a setup in which only a torque-minimizing objective was used. Alternative approaches to avoid collapsing are constraining end-effector positions in certain limited regions with respect to the body (Winkler et al., 2018), relying on initial trajectories (Herzog et al., 2016) or using nominal knee angles (Dai et al., 2014).

The collision function $C(.)$ simply makes sure that the mid-trajectory end-effector positions always lie on one side of both starting and ending contact surfaces. More precisely:

$$C(p) = \begin{bmatrix} (p - P^-)^T N^- \\ (p - P^+)^T N^+ \end{bmatrix} - c \tag{5.8}$$

where superscripts $-$ and $+$ represent starting and ending contact surfaces for the corresponding swing phase. The scaler $c = 5\%$ of leg length also indicates a desired amount of ground clearance or leg lift for the swing trajectories. Finally, we start optimizations by an intuitive (possibly infeasible) initial solution where $h_i = T$, swing trajectories connect the centers of contact surfaces together, velocities are differentiations of these positions and contact forces are set to the total weight divided by the number of active contacts in each phase. The optimization therefore adjusts all these variables optimally. We use the package CasADi (Andersson et al., 2012) to formulate our problem with automatic symbolic differentiation and solve it via IPOPT (Wächter and Biegler, 2006).

5.3 Results

In this section, we study the optimization results for different motion planning tasks. We consider changing terrain complexity, planning horizon and trajectory precisions. In all experiments, we keep the motion sequence fixed, i.e. we use the same order of limb motions all the time. This sequence is set to move one limb at a time, although it could be easily generalized to moving arbitrary number of limbs and even going to flight phases.

5.3.1 Swing dynamics

As motivated previously, the aim of using a five-mass model in our method is to study the exchange of energy between the base (torso) and the limbs. In chapter 6 we showed that in bipedal walking motions, swing and torso balancing dynamics can influence torso motions mainly by reducing the speed. Mechanically, a transfer of kinetic energy from the torso to the limbs enables them to move, but slows down the torso itself. Single-mass models cannot include this phenomena in optimizations (Winkler et al., 2018) while Centroidal dynamic approaches require inverse kinematics and full-body momentum equations to produce dynamically consistent motions (Dai et al., 2014; Herzog et al., 2016; Mordatch et al., 2012). Our method can easily model this phenomena while reducing the complexity and considerably speeding up optimizations. Depending on the frequency of motion and the robot properties, this exchange of energy might become important. In other words, Cartesian motions that ignore swing dynamics might only be realizable on robots with lightweight limbs (Winkler et al., 2018). Otherwise, the choice of frequency should be slow enough to allow for a feasible tracking of Cartesian trajectories with inverse kinematics or dynamics. Figure 5.3 demonstrates an example of motion where limb-swing dynamics can influence torso motions to some extent. One can observe in Figure 5.3B that torso velocities are generally decreased when the limbs start moving.

The scenario in Figure 5.3 is of particular interest, because the torso almost has to stop in order to move hands. In such flat terrain conditions, ideally, the robot does not need to use hands at all. However, since the rhythm of motion is fixed and given, the planner cannot skip hand-support phases and let the robot naturally walk. It is also not able to shorten these phases, because by design, the hands cannot just move constantly with the body and instead, they have to stop at corresponding contact surfaces for some time. Consequently, because of the lifting terms which bring the legs to their maximum length, the robot cannot keep both feet in contact and move the torso with a high velocity forward. One of the limbs may geometrically violate its reachability limits during double-feet support. As a result, the torso almost stops completely when the arms move. This is indeed the limitation of our predefined motion sequence which requires stationary hand positions and could be solved by deciding a different sequence.

5.3.2 Static vs. dynamic motions

In the second test, we change the frequency of motion and investigate the resulting trajectories. The frequency of normal walking in a child with the same height as our COMAN robot (approximately 90cm) is about $T = 0.45$s (Hausdorff et al., 1999). Here, we consider two gaits of $M = 12$ steps with different average phase durations of $T = 0.3$s and $T = 0.7$s. As demonstrated in Figures 5.4 and 5.5, the choice of $T = 0.7$s produces a slow motion with conservative contact forces and non-smooth trajectories. The choice of $T = 0.3$s produces forces that reach friction cone boundaries, but resulting in smoother trajectories and similar limb acceleration patterns.

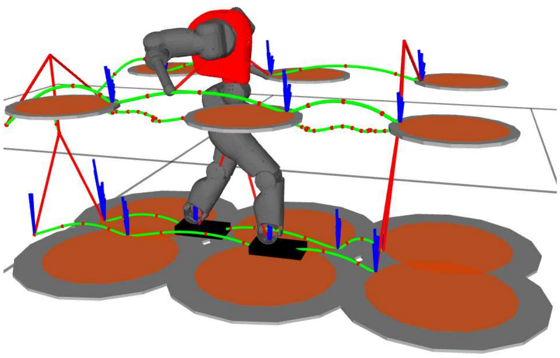
5.3.3 Optimization performance

In the final set of experiments, we change different motion parameters and report the trends in optimization performance. In particular, the horizon length M, trajectory precisions determined by K and motion dynamics determined by T have considerable influences on the optimization performance. Table 5.1 lists different optimization setups to investigate these parameters. It is observed that the number of future steps almost linearly increases the optimization time which is due to the sparsity of our technique similar to (Ponton et al., 2016). Increasing the number of sub-phases has a similar effect according to Table 5.1. Also, generating more static motions is generally faster, since velocities and accelerations are small in magnitude.

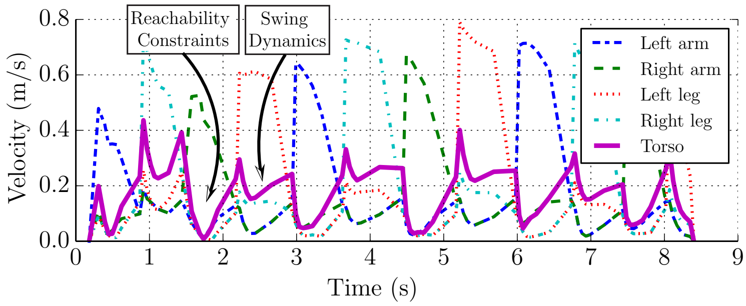
5.4 Discussion

Similar to our posture planning work of chapter 4, we observe that representing geometries by Cartesian points instead of joint angles can speed up motion planning optimizations as well. In the most complicated case, we have $M = 12$ steps and $K = 7$ sub-phases which gives a total of 84 sub-phases and optimization time of 1.39s. In comparison, the combination of Centroidal dynamics and inverse kinematics used in (Herzog et al., 2016) optimizes 100 sub-phases in 30s. While this combination can account for swing dynamics in fast motions, a reduced version without inverse kinematics (Ponton et al., 2016) can run much faster and optimize a static plan (of 4 steps in 8s duration and

A) Terrain profile



B) Absolute velocities



C) Limb lengths

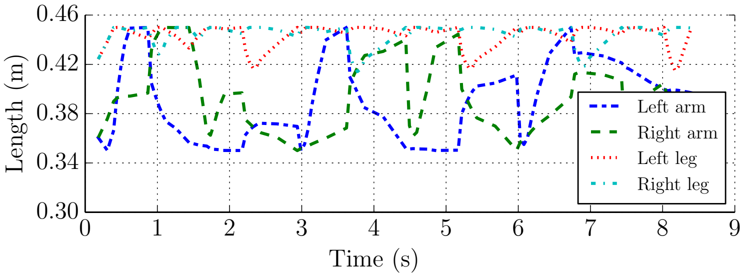
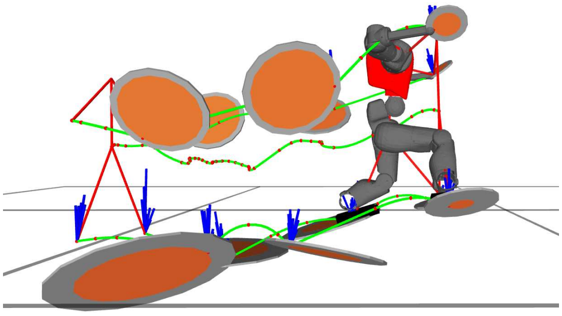


Figure 5.3 – Planning $M = 12$ steps with a fine resolution of $K = 7$ and a nominal phase duration of $T = 0.7$s. In this simple environment shown in A), trajectories are rather simple, but limb swing dynamics influences the torso velocity considerably. In B), we have shown absolute velocities for all the five masses in our model. In this scenario, since hand-support surfaces are placed slightly high, the elbows are normally bent which means the arms hardly reach the boundaries of their workspace. On the other hand, because of the lifting term, the legs are normally stretched and limited. Therefore, when both feet are in contact, the pelvis and torso cannot progress much because of such reachability constraints and thus slow down. This can be seen in the accompanied video. The double-support duration is relatively small in human walking as well (Cappellini et al., 2006). When the two hands are fixed, the pelvis and torso have more freedom to move, though leg-swing dynamics influences their speed slightly.

with 80 sub-phases) in 0.421s. Our method, however, can find more dynamic motions in similar optimization setups thanks to inclusion of internal dynamics in the five-mass

A) Quasi-static motion $T = 0.7$s



B) Fast motion $T = 0.3$s

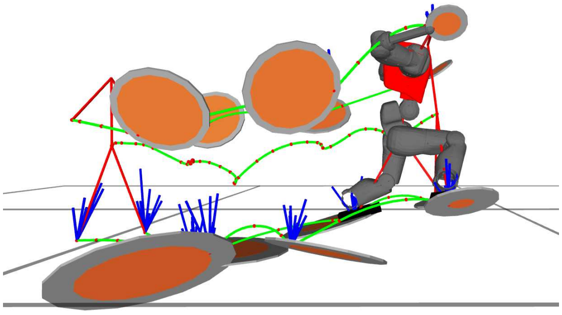
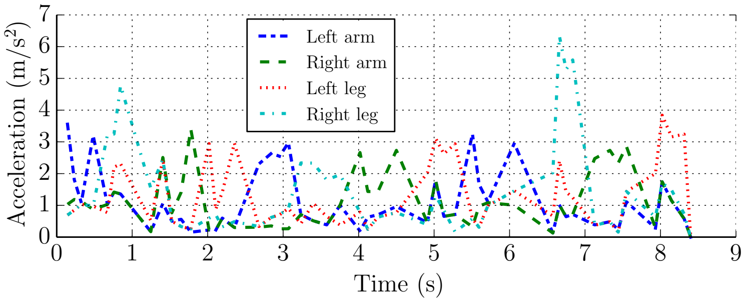


Figure 5.4 – Rough-terrain locomotion with different average phase durations. With a choice of $T = 0.7$s, the robot spends more time in each phase while with $T = 0.3$s, the motion becomes much faster. In this case, contact forces are also less conservative and may reach friction cone boundaries.

model. It also leaves the choice of sub-phase durations free in optimizations to find more optimal solutions compared to (Herzog et al., 2016; Ponton et al., 2016).

We only used an intuitive start point without any prior knowledge. Running these optimizations with a warm start in receding-horizon optimizations can be much faster and allow for a reactive online control of up to 50Hz update rate, depending on the horizon length and the desired resolution. In the smallest setup, we can optimize $M = 4$ steps in 120ms while spending 4.8ms per iteration. For a typical MPC control application with receding horizon, we might only need up to 5 iterations which may take 20-25ms. This convincingly extends our previous MPC controller of chapter 2 for bipedal walking which used the linear inverted pendulum model and could adjust footsteps in 0.2ms. Comparatively, with a cost of longer computations which is still much faster than the normal frequency of motion ($T = 0.5$s), we can optimize a much more complex multi-contact gait. We can adjust contact positions, swing trajectories and phase timings altogether. More importantly, we are not limited to linearity assumptions anymore.

Our method is able to produce motions with different properties despite using a given fixed rhythm. It can also adjust the timing of different phases in order to smoothen trajectories. However, we consider using Cost of Transport (CoT) terms in the objective function and remove the motion duration constraint to naturally find an optimal frequency. For online push-recovery applications which may require activation or deactivation of


A) Quasi-static motion $T = 0.7$s

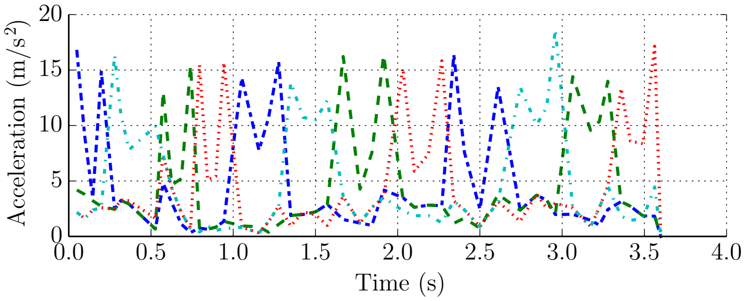

B) Fast motion $T = 0.3$s

Figure 5.5 – Rough-terrain locomotion with different average phase durations similar to Figure 5.4. With a choice of $T = 0.7$s, absolute limb accelerations follow a jerky pattern whereas with $T = 0.3$s, the motion is fast and smooth. In this case, different limb accelerations follow similar patterns.

certain contacts on the fly, assuming that only few integer combinations are decided, our method is potentially fast enough to be used in a mixed-integer optimization solver. A possible scenario would be receiving external pushes during locomotion while available foothold regions are not enough to recover. In this case, the robot may decide to use hands against the wall which is a new discrete decision. We do not have a mixed-integer optimization setup yet, but we plan to add this feature in future work. Ultimately, we would like to integrate this planner with a perception pipeline that extracts environment geometries (Kaiser et al., 2016), and a compliant inverse dynamics controller that provides precise and fast tracking (refer to chapter 3).

M	T	K	Terrain	Var.	Cons.	Optim.	Iter.	Description
4	0.5s	4	Rough	431	606	0.12s	4.8ms	Short horizon
8	0.5s	4	Rough	879	1254	0.29s	12.6ms	Medium horizon
12	0.5s	4	Rough	1327	1902	0.52s	19.3ms	Long horizon
12	0.3s	4	Rough	1327	1902	0.69s	19.7ms	High frequency
12	0.5s	4	Rough	1327	1902	0.52s	19.3ms	Natural frequency
12	0.7s	4	Rough	1327	1902	0.36s	18.0ms	Slow frequency
12	0.5s	4	Rough	1327	1902	0.52s	19.3ms	Normal gait
12	0.5s	7	Rough	2317	3258	1.39s	39.7ms	High resolution
12	0.7s	7	Flat	2317	3258	1.21s	37.8ms	Flat gait

Table 5.1 – Dimensions and performance metrics of different optimization setups. Columns show the number of steps M, average phase durations T, number of sub-phases K, terrain type, number of variables, number of constraints, overall optimization time, time per iteration and scenario description respectively. The first group of three experiments investigates the effect of horizon length. The second group explores different gait frequencies. The third group also investigates the effect of motion resolution. Increasing the horizon almost linearly increases the optimization time. Producing quasi-static gaits can be done faster. Increasing the resolution also linearly increases optimization time which is slightly better for simpler environments.

6 3LP Model

We could achieve a relatively fast walking gait with the linear inverted pendulum model and MPC controller of chapter 2. However, such tracking performance was due to using inverse dynamics and perfect torque tracking in ideal simulation environments. On the real COMAN robot which has relatively heavy legs and delayed torque control, this tracking was hard to achieve. In other words, the inverse dynamics layer was not able to compensate the missing swing dynamics in inverted pendulum trajectories effectively. Improving the torque controllers further was impossible due to control communication delays. However, to produce more consistent Cartesian trajectories, especially to include swing and torso-balancing dynamics, we developed the 3LP model. 3LP, which stands for three linear pendulums, is derived using our point-wise modeling technique of chapter 4. 3LP features limb masses and inertias as well as hip and ankle torques which allow for modulation of the walking gait. In this chapter, we introduce 3LP equations and our simple method to find walking gaits. We compare 3LP with human to better demonstrate the higher precision of 3LP dynamics versus the linear inverted pendulum model. 3LP is, in fact, the focal point of this thesis in the sense that it is used for both robot control and biomechanical studies (refer to Figure 1.2). In chapter 7, we will develop a time-projection controller based on the 3LP model which finally replaces our MPC walking controller of chapter 2. A combination of the 3LP model and time-projection control is later used in chapter 8 for walking experiments on the real hardware. For biomechanics studies, we add missing energy components to the 3LP model in chapter 9 which together predict human walking energetics in a wide range of walking conditions. We also use the 3LP model to study human walking asymmetries in case of inclined walking or torso bending conditions in chapter 10[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji and Auke Jan Ijspeert. *"3LP: A linear 3d-walking model including torso and swing dynamics."* The International Journal of Robotics Research (IJRR), 36(4):436–455, April 2017.

---

[1] All the videos of this chapter could be found at https://youtu.be/jrisgGxsL4s

6.1 Background

Humanoid robots are challenging to control mainly due to their complex structures and a floating base. During locomotion tasks, these systems introduce another complexity compared to wheeled or flying robots, which is the hybrid nature of stepping where the continuous model changes in each phase. It has always been challenging to balance these robots with only unilateral supporting forces from the environment. Also, creating a sequence of motion, the associated timing and the required control architecture are other important topics in controlling humanoid robots. The main objectives are therefore being human-like, energy efficient, versatile and of course agile like humans. In this chapter, we are proposing a new template model that describes main aspects of walking while being computationally very efficient. Such model can be very useful in modern control architectures from the computational perspective. It can also go beyond conventional template models such as Linear Inverted Pendulum (LIP) by producing more natural motions and faster walking speeds, resembling human locomotion.

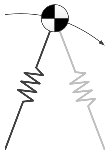
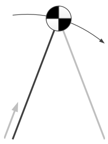
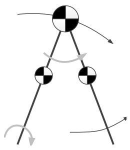
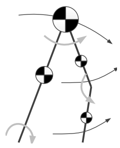
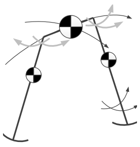
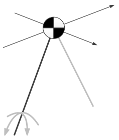
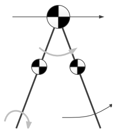
The design of controllers should address many concerns like fast implementation, stability, robustness to unknown model parameters and the degree of dependency on sensory data. Besides, it is desired to handle speed transitions and large disturbance rejection in the same control framework. Candidate methods are typically coming with proper identification of the basin of attraction regarding system states, actuator limitations and violation of model assumptions. This identification is not always straightforward due to its nonlinear nature, though it has been postulated that two steps are enough to stabilize in almost all conditions (Zaytsev et al., 2015). In this regard, Model Predictive Control (MPC) is a powerful framework as it can find optimal policies constrained to certain actuation and state limitations. It can also predict if there is no feasible solution, to let the algorithm take a different decision.

6.1.1 Hierarchical controllers

Recently, hierarchical control approaches are becoming popular, where a simple template model determines the overall dynamics in an abstract way and then, a detailed full-body inverse dynamics controller converts this behavior to individual actuator inputs (Faraji et al., 2014a; Feng et al., 2013; Kuindersma et al., 2014). In dynamical systems, prediction of future evolution is mainly sensitive to the model and sensory data precision (Bhounsule et al., 2015). In hierarchical approaches similarly, dynamical matching between the template and the full model is crucial to ensure precise execution of the abstract plan. In this regard, we briefly review relevant template models proposed for walking, identify missing properties and motivate the new model proposed in this chapter.

6.1.2 Inverted pendulums

One of the earliest template models that roughly explains bipedal mechanics is Inverted Pendulum (IP) (McGeer, 1990). In this model, a single mass rolls over a massless stick with a fixed length. IP is widely used to analyze passive walkers (McGeer, 1990) and energetics of human walking (Kuo et al., 2005). Inspired by IP, many simple robots

	 (Iida et al., 2009)	 (Hemami et al., 1973)	 (Asano et al., 2004)	 (Asano et al., 2004)	 (Kuo, 1999)	 (Kajita and Tani, 1991)	 (Takenaka et al., 2009)
name	SLIP	IP	-	-	-	LIP	-
knee	-	-	-	x	-	-	-
steering	-	-	-	-	-	x	-
3D	-	-	-	-	x	x	-
swing	-	-	x	x	x	-	x
linear	-	-	-	-	-	x	-

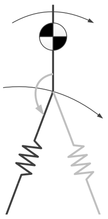
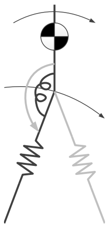
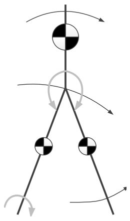
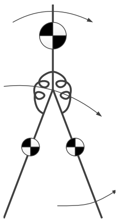
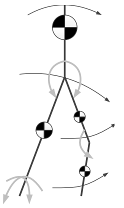
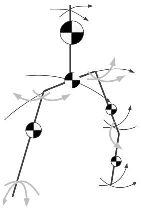
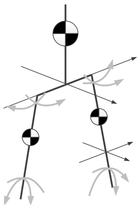
	 (Maufroy et al., 2011)	 (Sharbafi and Seyfarth, 2015)	 (Hasaneini et al., 2013)	 (Gomes and Ruina, 2011)	 (Manchester and Umenberger, 2014)	 (Gregg et al., 2012)	 proposed
name	BSLIP	FMCH	-	-	-	-	3LP
knee	-	-	-	-	x	x	-
steering	-	-	-	-	-	x	(x)
3D	-	-	-	-	-	x	x
swing	-	-	x	x	x	x	x
linear	-	-	-	-	-	-	x

Figure 6.1 – Different key models introduced in the literature for walking. In this table, the model is standing on the left leg, and the right leg is in the swing motion. Solid arrows show the direction of motion and degrees of freedom while gray arrows show actuation torques or push-off forces. For models without swing dynamics, we show the swing leg in gray color to implicitly indicate that attack angle is a control authority. Note that some of these models are only in 2D while more advanced models are in 3D. Some models simulate pelvis width, torso dynamics or ground clearance as well. Most of these models produce compass gait. However, some have ankle actuation or arc foot. In the comparison table, we mention important features such as knee flexion (for ground clearance), steering capabilities, 3D formulations, smooth profiles, the inclusion of torso and swing dynamics and linearity. By smoothness, we mean no collision and push-off impulse, but possibly describing the double support phase. Many of these models allow for torso pitch while we keep it fixed in 3LP for simplicity. Note that 3LP can describe steering like the LIP model use in chapter 2 only if pelvis width is set to zero. Despite being linear, 3LP offers many features not existing in other template models.

are built to walk naturally with minimal energy, injected in push-off, swing hip or both (Collins et al., 2005). Later, this model was simplified to Linear Inverted Pendulum (LIP) (Kajita et al., 2003), favoring analytical solutions instead of numerical integrations. With a proper modulation of Zero Moment Point (ZMP) (Vukobratović and Borovac, 2004), many position controlled robots like ASIMO (Sakagami et al., 2002) perform walking via inverse kinematics methods. These algorithms are usually able to produce slow to moderate walking speeds. However, robots using the LIP method usually walk with crouched knees to keep the Center of Mass (CoM) at a constant height. In addition to increasing energy consumption, it is harmful to the robot in the long term and less human-like, though providing full controllability.

More recent extensions of LIP contain one (Park and Kim, 1998) or two additional masses (Takenaka et al., 2009; Buschmann et al., 2007) in the legs to address swing dynamics while keeping the CoM height constant. For instance, in these extensions, the proposed model is formulated in a 2D space, and parameterized swing trajectories (in the sagittal and vertical planes) are used to generate desired gaits. In (Park and Kim, 1998) specifically, sinusoidal profiles lead to closed-form solutions with few realistic approximations introduced. Despite the advantage of describing ground clearance effects with a minimal coupling to the sagittal dynamics, the swing trajectory is yet imposed to the system, which is not desirable. We look for a more generic model to work at different speeds and frequencies, not relying on parametric trajectories to tune.

6.1.3 Multi-link pendulums

Apart from these single-mass models, there are other nonlinear extensions solved numerically. In (Byl and Tedrake, 2008; Asano et al., 2004), the IP model was extended to have two separate masses for each leg as well as a single mass at the hip level. Using similar actuation schemes, this model could produce compass gaits on 2D-constrained robots. In (Asano et al., 2004), the same model was modified to have another Degree of Freedom (DoF) in the swing leg to provide ground clearance. The stance leg, however, always remains straight in this version. In (Westervelt et al., 2007), this model was augmented with a torso and later, it was also used by (Manchester and Umenberger, 2014) to perform natural walking on uneven terrain, using a library of motion primitives. Another model with four masses in the legs, hip, and torso was proposed in (Gregg and Spong, 2009) without any DoF in the stance knee. This model was used to generate walking trajectories with steering properties.

Aiming at removing impacts, a simpler model with two passive springs in the hips was proposed by Gomes (Gomes and Ruina, 2011). These springs are mainly motivated by elastic properties of human muscles. By exploiting torso motions, Gomes could find zero energy gaits. Another interesting complex model was proposed in (Gregg et al., 2012) with a pelvis of a certain width in 3D and a mass in the center. The swing leg also had a DoF in the knee. This 3D model takes advantage of a limited transversal wrench in the contact point to facilitate steering. However, finding a periodic gait for such complicated model is difficult and computationally expensive.

6.1.4 Spring-loaded pendulums

It is always questionable which template model produces more realistic motion from the viewpoint of geometry, torques or energy. The models above mainly address energetic and geometric similarities. However, specific ground reaction force profiles and the elastic behavior observed in human legs are better produced in another category of models based on Spring Loaded Inverted Pendulum (SLIP) (Blickhan, 1989). The simplest model in this category is composed of two massless springs (legs) connected to a point mass. Observations indicate a better description of energy exchange in this model over faster walking speeds and running, mimicking compliant properties of human tendons (Blickhan, 1989). Based on this model, Iida (Iida et al., 2009) built a hip-actuated robot walking in 2D with various springs, similar to the human muscles. Properties of the passive SLIP itself -without hip actuation- were widely explored later in (Rummel et al., 2010). Using the concept of Virtual Pivot Point (VPP) to stabilize the torso, the model was also extended to have an upper body (Sharbafi and Seyfarth, 2015) which made the motion more human-like.

In Figure 6.1, we have briefly shown key models proposed for walking in the literature. Note that in some of them, passive springs are added to the hip actuators for energy storage, similar to human. In this chapter, however, we do not investigate elastic behaviors and energy-saving mechanisms.

6.1.5 Control difficulty

Except for LIP, all other models presented earlier require numerical integration to obtain time trajectories. Therefore, in a periodic walking paradigm, the Jacobian around a nominal solution linearizes the model and provides the framework for Floquet analysis or discrete controller designs (Rummel et al., 2010). This approach can be used to create an optimal library of primitives (Kelly and Ruina, 2015; Manchester and Umenberger, 2014; Gregg et al., 2012). However, online reaction to disturbances as well as the inclusion of other inequality constraints that are often ignored in calculating a stable basin of attraction limit the generality of this framework. MPC, on the other hand, is powerful in this regard. However, it requires simple and possibly linear models to facilitate online calculations. The LIP model, therefore, fits best in the MPC framework (Faraji et al., 2014a; Herdt et al., 2010b). MPC, its simpler version LQR, and sometimes Discretized LQR (DLQR) (Ogata, 1995) controllers are popular in stabilizing walking gaits and recovering large pushes (Kelly and Ruina, 2015; Byl and Tedrake, 2008). With nonlinear models, however, a library of optimal policies is generated offline, or a discrete transition model is considered at specific events like CoM apex or heel-strike. Such controllers can not react to perturbations very quickly, since adjusting the plan is not computationally affordable online.

6.1.6 Why 3LP?

In this chapter, we propose a more general version of LIP (Kajita et al., 2003) with three linear pendulums (called 3LP) that capture torso and swing dynamics in 3D. This model allows prediction of future at any time in closed-form which is favorable by limited computational resources and MPC. Compared to LIP, 3LP-based trajectories are easier to track by inverse dynamics block in a hierarchical controller. In other words, CoM motion becomes more natural for the humanoid robot, since swing and torso dynamics are taken into account. Swing-leg trajectories are also more natural compared to the template models which track an imposed angle of attack with a stiff controller (Kelly and Ruina, 2015; Byl and Tedrake, 2008; Collins et al., 2005). Besides, using 3LP, one can define meaningful torque limits in MPC frameworks instead of putting vague timing or step-size limits which do not precisely reflect physical facts about the real hardware. It should be noted, however, that the CoM height in 3LP is constant similar to LIP.

The 3LP model provides direct access to the hip and ankle torques. These input dimensions let us find various types of gaits with simple closed-form expressions. 3LP as a template model is therefore very useful for motion planning. Along with falling dynamics like IP (Kuo et al., 2005) and LIP (Kajita et al., 2001a), torso-balancing hip torques are also part of 3LP like (Maufroy et al., 2011). The most outstanding feature of 3LP is in considering swing dynamics in a linear fashion that allow us to calculate natural cycles. With LIP, however, a reference footstep plan is needed (refer to chapter 2). Considering Figure 6.1 again, few models in the literature consider this integral part of walking. In these models, due to nonlinearity, numerical integration is always needed to search for periodic gaits. In 3LP, however, we do not need to integrate the system or to perform numerical optimizations.

This chapter merely focuses on introducing the model, formulating equations, finding gaits and comparing them to human gaits. Setting up control problems remain for future works. In the next section, we will explain model details and the assumptions behind. Next, a method based on geometrical symmetry is introduced to find different periodic gaits. We will demonstrate that for a human-like gait, actuation profiles in 3LP are similar to those in human. Finally, we show that 3LP, despite being linear, quantitatively explains the main optimality trend in human gaits, i.e. speed-frequency relation.

6.2 3LP dynamics

To capture the coupling of swing and torso dynamics, we have added two other pendulums to the standard LIP model, connected with a pelvis of a certain width. In this model, as shown in Figure 6.2, there are 2-DoF actuators in each hip and ankle. We visualize feet of limited size in Figure 6.2, only to mention the availability of ankle torques. The upper body (referred to as torso) and the legs are each represented by a single mass. By construction (assuming ideal controllers), masses stay in horizontal planes of constant height and the torso is always upright without sagittal, frontal or transversal rotations. These assumptions are used in (Gregg et al., 2012) as well to decouple sagittal and lateral dynamics.

Since the torso is connected to an accelerated frame (i.e. pelvis), a balancing torque is always needed to keep the torso upright. This torque is often calculated using the virtual pendulum concept (Sharbafi and Seyfarth, 2015), where the torso can pitch or

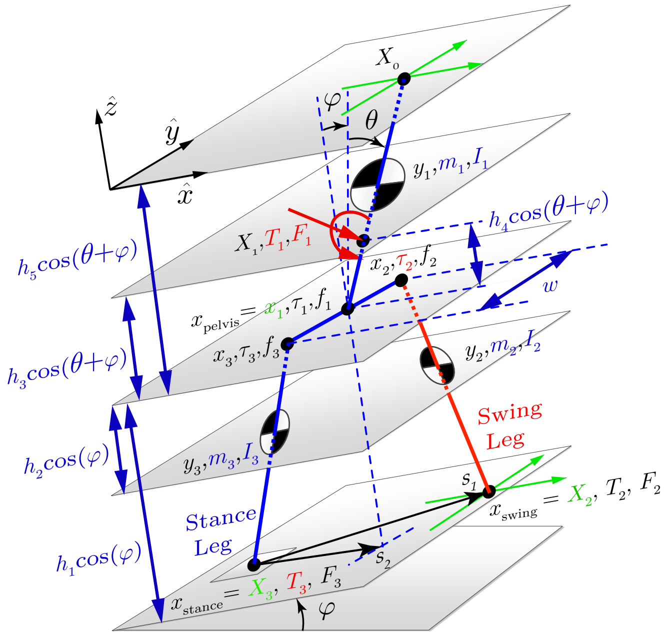


Figure 6.2 – A schematic of 3LP model with all variables and model parameters. The bottom planes show the level ground and all upper planes show where the three masses and the pelvis are constrained to move. The torso is always upright, and the pelvis is along the y-axis by model construction. The swing foot remains inside the bottom plane, i.e. sliding on the ground with no force during the swing phase. Note that in single support all contact forces for the swing leg (T_2, F_2) are zero. The inputs of the model are τ_2 and T_3 shown in red together with arbitrary external perturbations T_1 and F_1 applied to the torso. State variables are feet and pelvis positions, shown in green. Fixed model parameters are also shown in blue, including masses and geometrical dimensions.

roll freely. Inspired by rolling contact constraints that produce more human-like gaits (Hamner et al., 2013), we allow for a transversal wrench at stance foot to help to keep the pelvis orientation fixed. In 3LP, we do not consider steering properties as they make the model nonlinear. It is practically easy however to steer the robot using an inverse dynamics layer, as we showed in chapter 2 where a simple LIP model was used. There is no need to add heel-strike and push-off impulses in 3LP since our double support phase smoothly takes care of the contact transition. In SLIP-based models (Sharbafi and Seyfarth, 2015), the compliant springs automatically produce a double support phase and perform weight transition without impulses. In 3LP, however, switching to double support is triggered when both horizontal components of the swing foot velocity become zero. This assumption is typically used in models with swing dynamics like (Gomes and Ruina, 2011), though unlike 3LP, Gomes removes double support for more simplicity (Gomes and Ruina, 2011).

In Figure 6.2, all external/internal forces and torques, as well as positions, are shown for each interesting point of the model, i.e. contacts, hips and pelvis center. These vector variables in our model are expressed in a rotated Cartesian frame attached to the slope.

Table 6.1 – 3LP variable notations.

X_i	Location of external forces	l	Leg length
F_i	External forces	h_1	Pelvis height = leg length
T_i	External moments	h_2	Relative leg mass height
y_i	Pendulum mass locations	h_3	Relative torso mass height
m_i	Pendulum masses	h_4	Relative force application point height
I_i	Pendulum inertias	h_5	Torso height
x_i	Hip and mid-pelvis locations	h	Foot length
f_i	Pelvis forces	w	Pelvis width
τ_i	Pelvis torques	g	Gravity $= -9.81 m/s^2$
ϕ	Slope angle	θ	Torso angle (with respect to vertical)

Using the notations of Table 6.1 One can easily write geometric relations as:

$$\begin{array}{ccc} x_1 = [x_{1,x} \;\; x_{1,y} \;\; 0]^T + h_1 \cos(\phi)\ \hat{z} & x_2 = x_1 + (\frac{ws}{2})\ \hat{y} & x_3 = x_1 - (\frac{ws}{2})\ \hat{y} \\ X_1 = x_1 + h_4 \cos(\theta + \phi)\ \hat{z} & X_2 = [X_{2,x} \;\; X_{2,y} \;\; 0]^T & X_3 = [X_{3,x} \;\; X_{3,y} \;\; 0]^T \\ y_1 = x_1 + h_3 \cos(\theta + \phi)\ \hat{z} & y_2 = x_2 + \frac{h_2}{h_1}(X_2 - x_2) & y_3 = x_3 + \frac{h_2}{h_1}(X_3 - x_3) \end{array} \tag{6.1}$$

where $s = \pm 1$ determines left or right support phases. These parameters as well as other position variables are shown in Figure 6.2. Defining the gravity vector as:

$$\hat{g} = g\ (\cos(\phi)\hat{z} + \sin(\phi)\hat{x}) \tag{6.2}$$

where $\hat{x}$, $\hat{y}$ and $\hat{z}$ denote the orthonormal rotated frame attached to the slope, we can write total force equations for each mass $i = 1, 2, 3$:

$$m_i(\ddot{y}_i + \hat{g}) = f_i + F_i \tag{6.3}$$

The total moment equations for each mass $i = 1, 2, 3$ are:

$$I_i\ \dot{\omega}_i = (X_i - y_i) \times F_i + (x_i - y_i) \times f_i + T_i + \tau_i \tag{6.4}$$

where all variables are defined in the rotated Cartesian frame (refer to Figure 6.2) and:

$$\omega_1 = 0, \quad \omega_2 = \frac{\dot{X}_2 - \dot{x}_2}{h_1}, \quad \omega_3 = \frac{\dot{X}_3 - \dot{x}_3}{h_1} \tag{6.5}$$

Finally, we can write total force and moment equations for the mass-less pelvis around the center point:

$$\begin{array}{rcl} f_1 + f_2 + f_3 & = & 0 \\ \tau_1 + \tau_2 + \tau_3 + (x_2 - x_1) \times f_2 + (x_3 - x_1) \times f_3 & = & 0 \end{array} \tag{6.6}$$

which link all variables together. In these equations, we consider $x_1 = x_{\text{pelvis}}, X_2 = x_{\text{swing}}$ and $X_3 = x_{\text{stance}}$ as independent variables and solve for others as dependent variables.

Table 6.2 – Sequencing information about the two consecutive phases that form a full step phase.

Full step =	double support +	single support
duration	T_{ds}	T_{ss}
timing order	1	2
stance leg	subscripts 3	subscripts 3
swing leg	-	subscripts 2
control input	ankle	hip/ankle
controllability	over-actuated	fully-actuated

Actuation possibilities for 3LP are selected as stance foot T_3 and swing hip torques τ_2 in the sagittal and lateral planes. Note that T_1 and F_1 represent disturbing external forces (at point X_1 on the torso) which are zero in normal conditions.

Now, differential equations governing the bipedal motion could be obtained for different phases. We consider a full step, consisting of a double support followed by a single support phase, defined in Table 6.2. In double support, the weight is transferred from F_2 to F_3, and in single support, the leg with variables of subscripts 2 will swing forward. Over the next full step phase, the supporting leg can be simply changed by altering the sign of variable d.

The equations (6.3), (6.4) and (6.6) are not enough to solve the system completely. We require 12 new equations to find all accelerations in terms of control inputs and vice versa. This number comes from the fact that in total, we have 12 actuation variables in the system (torques in the hips and ankles). In single support, since the system is fully actuated, the additional constraints can simply be added by considering zero contact force in the swing foot and zero acceleration in the stance foot. In double support, however, because of the inherent redundancy, we need to make assumptions to simplify the system.

6.2.1 Single support

In this phase, the swing foot does not have any external forces and the stance foot is fixed on the ground:

$$F_2 = 0,\ T_2 = 0,\ \ddot{X}_3 = 0 \tag{6.7}$$

To further simplify the system, we consider two modes of input torques: constant (U) and ramp (V) profiles. Despite being linear, these profiles can still lead to a convincing match with human torque profiles, discussed in the next section. More complex terms might improve the matching precision, but make the equations more complicated. Besides, since there are fundamental geometric differences between the limbs in 3LP and human (e.g. the number of articulated joints), higher order terms might lead to an over-fitting which is not desired. Overall, we have eight input parameters U and V in the system:

$$\begin{bmatrix} \tau_{2,y} \\ \tau_{2,x} \\ T_{3,y} \\ T_{3,x} \end{bmatrix} = \begin{bmatrix} U_{h,y} \\ U_{h,x} \\ U_{a,y} \\ U_{a,x} \end{bmatrix} + \frac{t}{T_{\text{ss}}} \begin{bmatrix} V_{h,y} \\ V_{h,x} \\ V_{a,y} \\ V_{a,x} \end{bmatrix} = U + \frac{t}{T_{\text{ss}}} V \tag{6.8}$$

where T_{ss} stands for single support duration, t denotes the time from the beginning of the single support phase, and subscripts a and h denote ankle and hip respectively. Equations of (6.7) and (6.8) together give 12 additional constraints that enable us to solve the system completely. To this end, we define the state vector $X^{\text{ss}}(t)$ and the disturbance vector W as:

$$\begin{aligned} X^{\text{ss}}(t) &= \begin{bmatrix} X_{2,x}(t) & X_{2,y}(t) & x_{1,x}(t) & x_{1,y}(t) & X_{3,x} & X_{3,y} \end{bmatrix}^T \\ W &= \begin{bmatrix} F_{1,x} & F_{1,y} & T_{1,y} & T_{1,x} \end{bmatrix}^T \end{aligned} \tag{6.9}$$

where X_3 is the fixed contact point location. Using Maple (Monagan et al., 2005), we combine all equations and obtain a linear DAE system symbolically:

$$\frac{d^2}{dt^2} X^{\text{ss}}(t) = C_X^{\text{ss}}\ X^{\text{ss}}(t) + C_U^{\text{ss}}\ U + C_V^{\text{ss}}\ V + C_W^{\text{ss}}\ W + C_D^{\text{ss}}\ D \tag{6.10}$$

where constant matrices C^{ss} merely depend on system parameters (refer to Appendix C for further details). The constant vector D is also a function of slope and torso angles ϕ and θ as well as the phase variable s. In this equation, $X^{\text{ss}}(t)$ is the state vector, U and V represent constant and ramp inputs and W represents perturbations. Next, we define the full state vector $Q^{\text{ss}}(t)$ and the constant vector R as:

$$\begin{aligned} Q^{\text{ss}}(t) &= \begin{bmatrix} X^{\text{ss}}(t)^T & \dot{X}^{\text{ss}}(t)^T \end{bmatrix}^T \\ R &= \begin{bmatrix} U^T & V^T & W^T & D^T \end{bmatrix}^T \end{aligned} \tag{6.11}$$

Such abstract formulation facilitates the process of finding periodic gaits, discussed in the next section. Assuming constant external forces W, we can solve this system analytically:

$$Q^{\text{ss}}(t) = A^{\text{ss}}(t)\ Q^{\text{ss}}(0) + B^{\text{ss}}(t)\ R \tag{6.12}$$

where the time-dependent transition matrices $A^{\text{ss}}(t)$ and $B^{\text{ss}}(t)$ describe state evolution over time. These matrices are in fact very easy to calculate, because of their simple structure:

$$\begin{aligned} A^{\mathrm{ss}}(t) &= \sum_{i=1}^{6} A_i^{\mathrm{ss}} \gamma_i^{\mathrm{ss}}(t) \\ B^{\mathrm{ss}}(t) &= \sum_{i=1}^{6} B_i^{\mathrm{ss}} \gamma_i^{\mathrm{ss}}(t) \end{aligned} \tag{6.13}$$

where $\gamma_i^{\mathrm{ss}}(t) = e^{w_i^{\mathrm{ss}} t}$ for $i = 1..4$, $\gamma_5^{\mathrm{ss}}(t) = 1$ and $\gamma_6^{\mathrm{ss}}(t) = t$. The four scalars w_i^{ss} are square roots of the eigenvalues associated to the non-zero part of C_X^{ss} matrix (which has 4 dimensions). These values describe falling and swing dynamics, quantifying instability and divergence properties. A shorter pelvis height, for example, leads to w_i^{ss} values with larger magnitudes which mean that falling and swing dynamics become faster. A_i^{ss} and B_i^{ss} merely depend on constant system parameters, encoding evolution of the initial state and the effect of inputs respectively. Once these individual matrices are calculated offline, $A^{\mathrm{ss}}(t)$ and $B^{\mathrm{ss}}(t)$ can be easily calculated online by few arithmetic operations. Note that in 3LP, lateral and sagittal dynamics are decoupled, but their similarity leads to repeated eigenvalues.

6.2.2 Double support

In this phase, the two feet are fixed:

$$\ddot{X}_3 = 0, \ \ddot{X}_2 = 0 \tag{6.14}$$

and contact forces are being transfered from (F_2, M_2) to (F_3, M_3). Once (F_2 and M_2) become zero, the next single support phase starts, where the leg with subscript 2 performs swing motion. So far, equations (6.14) give us only 4 constraints, while 8 are yet missing.

In double support, we decided a linear transfer of weight from one leg to another. This means the vertical component of the Ground Reaction Force (GRF) in the previous stance leg (which has been stationary during single support) will go linearly to zero during double support. Such straightforward policy provides simple analytic solutions (compared to quadratic or other forms). Assume that:

$$\begin{bmatrix} F_{2,z} \\ T_{2,z} \end{bmatrix} = (1 - \frac{t}{T_{\mathrm{ds}}}) \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix}, \begin{bmatrix} F_{3,z} \\ T_{3,z} \end{bmatrix} = \frac{t}{T_{\mathrm{ds}}} \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix} \tag{6.15}$$

where T_{ds} denotes double support duration and t denote the time from the beginning of the double support phase. The variables $\alpha(t)$ and $\beta(t)$ are (possibly complex) functions of other variables in the system, but we are not going to find them explicitly. The linear policy can be encoded with the following equations by removing $\alpha(t)$ and $\beta(t)$:

$$\frac{t}{T_{\mathrm{ds}}} \begin{bmatrix} F_{2,z} \\ T_{2,z} \end{bmatrix} = (1 - \frac{t}{T_{\mathrm{ds}}}) \begin{bmatrix} F_{3,z} \\ T_{3,z} \end{bmatrix} \tag{6.16}$$

Remember that in single support, the ankle torques in the stance foot are determined by

(6.8) and the vertical GRF is constant (because of the fixed height assumptions in the model). Given that the vertical GRF decreases linearly with time, the Center of Pressure (CoP) position can be simply preserved in double support by linearly decreasing contact reaction moment T_2 in stance leg (and increasing T_3 accordingly):

$$\begin{aligned}
\begin{bmatrix} T_{2,y} \\ T_{2,x} \end{bmatrix} &= (1-\frac{t}{T_{\text{ds}}}) \begin{bmatrix} U_{a,y}+V_{a,y} \\ -U_{a,x}-V_{a,x} \end{bmatrix} \\
\begin{bmatrix} T_{3,y} \\ T_{3,x} \end{bmatrix} &= \frac{t}{T_{\text{ds}}} \begin{bmatrix} U_{a,y} \\ -U_{a,x} \end{bmatrix}
\end{aligned} \tag{6.17}$$

The equations (6.17) together with (6.8) will indeed result in piecewise linear ankle torque profiles. Note that the minus signs behind the lateral-plane ankle torques $U_{a,x}$ and $V_{a,x}$ in (6.17) come from the symmetry concept in the lateral plane. In other words, we assume that these constants move the CoP in opposite directions in the two feet. A positive $U_{a,x}$ for example moves the CoP to the left on the left foot, while moving it to the right on the right foot.

Similarly to (6.15), we implement a transition policy for the hip torques as well. Assume that:

$$\begin{aligned}
\tau_{2,y} &= (\frac{t}{T_{\text{ds}}})U_{h,y} + (1-\frac{t}{T_{\text{ds}}})\gamma(t) \qquad (6.18) \\
\tau_{3,y} &= (1-\frac{t}{T_{\text{ds}}})(U_{h,y}+V_{h,y}) + (\frac{t}{T_{\text{ds}}})\gamma(t)
\end{aligned}$$

where $\tau_{2,y}$ starts from a function $\gamma(t)|_{t=0}$ in the beginning and converges to $U_{h,y}$ at the end of double support phase. Likewise, $\tau_{3,y}$ starts from $(U_{h,y}+V_{h,y})$ and converges to the same $\gamma(t)|_{t=T_{\text{ds}}}$ at the end of the phase. The function $\gamma(t)$ in fact represents the torque in the stance hip which is a complex function of other variables in the system. However, we know linear trajectories of swing hip torques from (6.8). The equation (6.18) therefore encodes a linear transition policy. Similar rules can be written for the lateral plane.

$$\begin{aligned}
\tau_{2,x} &= (\frac{t}{T_{\text{ds}}})U_{h,x} + (1-\frac{t}{T_{\text{ds}}})\zeta(t) \qquad (6.19) \\
-\tau_{3,x} &= (1-\frac{t}{T_{\text{ds}}})(U_{h,x}+V_{h,x}) + (\frac{t}{T_{\text{ds}}})\zeta(t)
\end{aligned}$$

where the minus sign (behind $\tau_{3,x}$) is for symmetry, like before. The function $\zeta(t)$ plays the same role as $\gamma(t)$, but in the lateral plane. Now, we can remove $\gamma(t)$ and $\zeta(t)$ to make more explicit equations:

$$\begin{aligned}
\frac{\tau_{2,y}-\frac{t}{T_{\text{ds}}}U_{h,y}}{1-\frac{t}{T_{\text{ds}}}} &= \frac{\tau_{3,y}-(1-\frac{t}{T_{\text{ds}}})(U_{h,y}+V_{h,y})}{\frac{t}{T_{\text{ds}}}} \\
\frac{\tau_{2,x}-\frac{t}{T_{\text{ds}}}U_{h,x}}{1-\frac{t}{T_{\text{ds}}}} &= \frac{-\tau_{3,x}-(1-\frac{t}{T_{\text{ds}}})(U_{h,x}+V_{h,x})}{\frac{t}{T_{\text{ds}}}}
\end{aligned} \tag{6.20}$$

Now, equations (6.14), (6.16), (6.17) and (6.20) provide the 12 constraints needed to solve the system in the double support phase. Note that except (6.14), other equations are arbitrary choices to remove the redundancy. One might replace them with other policies for the same purpose. However, our simple assumptions preserve linearity and lead to a convincing match with human torque profiles, discussed in the next sections.

As mentioned earlier, 3LP switches to double support when the swing foot velocity becomes zero. This assumption, however, does not guarantee zero acceleration during the touchdown event. Therefore, a negligible discontinuity of torque and force profiles is unavoidable. In the simplest case, even preserving the continuity of horizontal GRF components lead to terms like $tx(t)$ which are linear, but difficult to solve analytically. With our specific linear transition rules, however, the profiles are almost continuous. Some discontinuities happen only in the transversal torques which are small in magnitude, demonstrated later in Figure 6.10.

In double support phase, the state variable consists of base positions only:

$$X^{\text{ds}}(t) = \begin{bmatrix} X_{2,x} & X_{2,y} & x_{1,x}(t) & x_{1,y}(t) & X_{3,x} & X_{3,y} \end{bmatrix}^T \tag{6.21}$$

while other vectors are the same as before. Again, we combine all equations and obtain a symbolic linear DAE system for the double support phase:

$$\frac{d^2}{dt^2} X^{\text{ds}}(t) = C_X^{\text{ds}}\ X^{\text{ds}}(t) + C_U^{\text{ds}}\ U + C_V^{\text{ds}}\ V + C_W^{\text{ds}}\ W + C_D^{\text{ds}}\ D \tag{6.22}$$

where constant matrices C^{ds} merely depend on system parameters (refer to Appendix C for further details). Defining the full state vector $Q(t)$ and the constant vector R as before, we can obtain a similar system of closed-form solutions as:

$$Q^{\text{ds}}(t) = A^{\text{ds}}(t)\ Q^{\text{ds}}(0) + B^{\text{ds}}(t)\ R \tag{6.23}$$

where the time-dependent transition matrices $A^{\text{ds}}(t)$ and $B^{\text{ds}}(t)$ describe state evolution over time. Note that since X_2 and X_3 are constant here, the matrix $A^{\text{ds}}(t)$ is partially diagonal with unit elements, keeping feet positions constant. The transition matrices have the following simple structure:

$$\begin{aligned} A^{\text{ds}}(t) &= \sum_{i=1}^{4} A_i^{\text{ds}} \gamma_i^{\text{ds}}(t) \\ B^{\text{ds}}(t) &= \sum_{i=1}^{4} B_i^{\text{ds}} \gamma_i^{\text{ds}}(t) \end{aligned} \tag{6.24}$$

where $\gamma_i^{\text{ds}}(t) = e^{w_i^{\text{ds}} t}$ for $i = 1..2$, $\gamma_3^{\text{ds}}(t) = 1$ and $\gamma_4^{\text{ds}}(t) = t$. The two scalars w_i^{ds} are replicated square roots of the eigenvalues associated with the non-zero part of C_X^{ds} matrix.

6.2.3 Full step

Having state transition matrices for both phases, we can now find closed-form equations for the full step phase:

$$Q(t) = A(t)\ Q(0) + B(t)\ R \tag{6.25}$$

where t denotes the time from the beginning of the full step phase (= beginning of the double support) and $Q(t) \in \mathbb{R}^{12}$ is defined as:

$$Q(t) = \begin{bmatrix} X(t)^T & \dot{X}(t)^T \end{bmatrix}^T \tag{6.26}$$

and $X(t) \in \mathbb{R}^6$ is:

$$X(t) = \begin{bmatrix} X_{2,x}(t) & X_{2,y}(t) & x_{1,x}(t) & x_{1,y}(t) & X_{3,x} & X_{3,y} \end{bmatrix}^T$$

The transition matrix $A(t)$ is defined as:

$$A(t) = \begin{cases} A^{\text{ds}}(t) & t \le T_{\text{ds}} \\ A^{\text{ss}}(t - T_{\text{ds}}) A^{\text{ds}}(T_{\text{ds}}) & 0 < t - T_{\text{ds}} \le T_{\text{ss}} \end{cases} \tag{6.27}$$

and $B(t)$ is:

$$B(t) = \begin{cases} B^{\text{ds}}(t) & t \le T_{\text{ds}} \\ A^{\text{ss}}(t - T_{\text{ds}}) B^{\text{ds}}(T_{\text{ds}}) +\ B^{\text{ss}}(t - T_{\text{ds}}) & 0 < t - T_{\text{ds}} \le T_{\text{ss}} \end{cases} \tag{6.28}$$

Note that we have used parameters T_{ss} and T_{ds} to calculate transfer matrices. The variable T_{ds} is crucial for double support calculations, though T_{ss} only determines the rate of time-increasing input components (V) in the single support. Therefore, one can easily scale T_{ss} and V such that the ratio remains constant in (6.8). The duration of a full step phase is defined as $T_{\text{step}} = T_{\text{ds}} + T_{\text{ss}}$. Figure 6.3 provides a demonstrates of all transition matrices and timing variables in a full step phase.

So far in this section, we have found transition matrices for the system and defined a full phase, consisting of a double support followed by a single support. We have also formulated matrices such that they are very fast for online calculation. The matrix $A(t)$ is used to find the time evolution of our system, especially until the end of a full step phase where $t = T_{\text{step}}$. In the next section, we are going to find different open-loop periodic gaits based on the type of actuation desired.

6.3 Periodic gaits

Although various types of symmetric and asymmetric gaits can be generated (Rummel et al., 2010), in this chapter, we only focus on symmetric gaits observed in normal human walking. In other words, given timing specifications, we find a space of different vectors that produce symmetric gaits. Each of these vectors contains initial conditions and constant inputs. The concept of symmetry could be encoded in a single matrix along

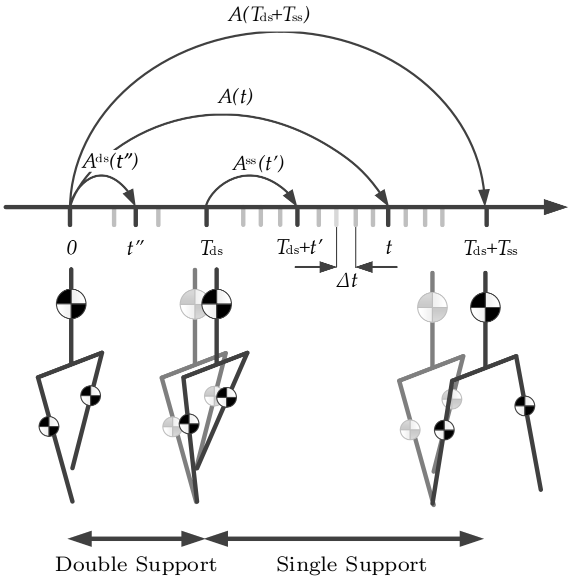


Figure 6.3 – Demonstration of transition matrices on a time axis. Δt represents arbitrary time duration that could be as small as a control tick. Thanks to linearity, one can easily find exact transition matrices for Δt as well which could be used for visualization and control purposes. Otherwise, the matrix $A(t)$ is enough for gait generation.

with the constraint of zero foot velocity at the end of the step.

Consider the vectors $Q(t)$ and $X(t)$ in (6.26) and (6.27) respectively. Indeed, zero-velocity constraints imply:

$$\begin{aligned}
\begin{bmatrix} X_{3,x}(t) & X_{3,y}(t) \end{bmatrix}^T &= \text{const}, \quad 0 \leq t \leq T_{\text{step}} \\
\begin{bmatrix} \dot{X}_{3,x}(0) & \dot{X}_{3,y}(0) \end{bmatrix}^T &= 0 \\
\begin{bmatrix} \dot{X}_{2,x}(0) & \dot{X}_{2,y}(0) \end{bmatrix}^T &= 0 \\
\begin{bmatrix} \dot{X}_{2,x}(T_{\text{step}}) & \dot{X}_{2,y}(T_{\text{step}}) \end{bmatrix}^T &= 0
\end{aligned} \tag{6.29}$$

The first constraint is automatically satisfied by construction of matrices $A^{\text{ss}}(t)$ and A^{ds} where we assumed stationary stance foot (and thus zero acceleration). The other three constraints shall be satisfied in finding periodic gaits.

After a full step phase, contact positions are exchanged by applying the matrix $T \in \mathbb{R}^{12\times 12}$ to the vector $Q(T_{\text{step}})$ where velocities are yet unchanged:

$$T = \begin{bmatrix} T_X & 0 \\ 0 & I_{6\times 6} \end{bmatrix} \tag{6.30}$$

where $T_X \in \mathbb{R}^{6\times 6}$ is defined as:

$$T_X = \begin{bmatrix} . & . & . & . & 1 & . \\ . & . & . & . & . & 1 \\ . & . & 1 & . & . & . \\ . & . & . & 1 & . & . \\ 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \end{bmatrix} \tag{6.31}$$

To implement the symmetry concept, we define relative difference vectors between the base, swing foot, and stance foot positions. These vectors could be extracted from $Q(t)$ by the following matrix $M \in \mathbb{R}^{12\times 12}$:

$$M = \begin{bmatrix} M_X & 0 \\ 0 & M_{\dot{X}} \end{bmatrix} \tag{6.32}$$

where 0 and I are zero and identity matrices. Sub-block are also defined as:

$$M_X = \begin{bmatrix} -1 & . & 1 & . & . & . \\ . & -1 & . & 1 & . & . \\ . & . & 1 & . & -1 & . \\ . & . & . & 1 & . & -1 \end{bmatrix}, \quad M_{\dot{X}} = \begin{bmatrix} \multicolumn{6}{c}{I_{6\times 6}} \\ 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \end{bmatrix} \tag{6.33}$$

where M_X extracts difference vectors from the state vector and $M_{\dot{X}}$ is used to constrain swing foot velocities to zero. Comparing the difference vectors before and after a symmetric full step, in the sagittal plane, components are equal, and in the lateral plane, they are opposite. Besides, we have to encode velocity constraints of (6.29) as well. So, we define a matrix $S \in \mathbb{R}^{12\times 12}$:

$$O = \mathrm{diag}(\ [1, -1, 1, -1, 0, 0, 1, -1, 0, 0, 1, 1]\ ) \tag{6.34}$$

where symmetry conditions are applied to the difference vectors and the base velocity while feet velocities are forced to zero. Now, consider in initial condition vector $Q(0)$ and inputs R. The state at the end of a full step is:

$$Q(T_{\text{step}}) = A(T_{\text{step}})Q(0) + B(T_{\text{step}})R \tag{6.35}$$

Applying contact exchange matrix, the initial state $Q'(0)$ for the next phase is:

$$Q'(0) = TQ(T_{\text{step}}) \tag{6.36}$$

The difference vectors extracted from $Q(0)$ and $Q(T_{\text{step}})$ should satisfy symmetry conditions while initial and final foot velocities in $Q(0)$ and $Q(T_{\text{step}})$ should be zero. One can write these conditions as:

$$MQ(0) = OMQ'(0) \tag{6.37}$$

Now, for a general vector of initial conditions and actuation parameters packed together:

$$Y = \begin{bmatrix} Q^T(0) & R^T \end{bmatrix}^T \tag{6.38}$$

the following equation should hold if Y represents a symmetric periodic gait:

$$DY = 0 \tag{6.39}$$

where:

$$D = \begin{bmatrix} -M + OMTA(T_{\text{step}}) & OMTB(T_{\text{step}}) \end{bmatrix} \tag{6.40}$$

The matrix $D \in \mathbb{R}^{12\times 25}$, in fact, compares the difference vectors before and after a full step phase while forcing feet velocities to zero. Note that apart from state vectors, the hip and contact torques U and V, the disturbance vector W and the phase variable s are considered here. However, it is meaningless in practice to consider a periodic gait with constant external disturbance. Initial contact positions X_3 would be set to zero to avoid redundant null space dimensions.

Remember that in the previous section, there were two variables to decide: T_{ss} and T_{ds}. In this section, we find various types of gaits by selecting different combinations of actuation and timing variables. We do so by considering the matrix D which is, in fact, a function of timing variables. Any periodic solution (a vector containing initial states and actuation inputs) should lie in the null space of the D matrix. In other words, solution manifolds are found by combining different null vectors of the matrix D. A solution contains initial states and actuation parameters where available actuators are swing hip and stance ankle torques (in sagittal and lateral directions), in constant U and time-increasing modes V. Note that only columns attributed to non-zero values in the solution vector are selected. In other words, we normally exclude columns related to contact positions X_3 and disturbances W.

In the rest of this chapter, we use human-like body parameters for numerical simulations, where mass distributions and geometries are taken from (de Leva, 1996). Table 6.3 lists these parameters for two adult-size and kid-size models, used further in this chapter.

6.3.1 Pseudo-passive gaits manifold

First, we would like to know whether the system has any pseudo-passive walking pattern or not. By pseudo-passive, we mean a gait in which swing hip and stance contact torques are zero. The term pseudo indicates that in the stance hip, the actuators might produce or dissipate power and the system is not strictly passive. This term also reflects the fact

Model	adult-size	kid-size	unit
Total mass	70	30	kg
Body length	1.7	1.0	m
h_1	0.89	0.52	m
h_2	0.32	0.19	m
h_3	0.36	0.22	m
m_1	45.7	19.6	kg
$m_2 = m_3$	12.15	5.2	kg
w	0.2	0.12	m

Table 6.3 – Parameters for adult-size and kid-size models used in simulations.

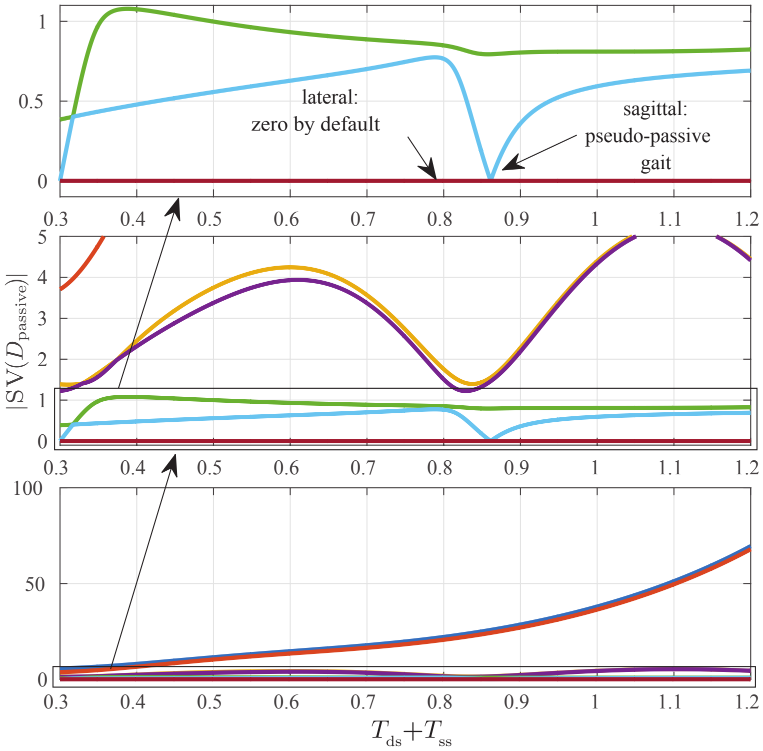


Figure 6.4 – Absolute singular values of D_{passive} versus T_{step}, plotted for an adult-size model. In these plots, we fix the double support time $T_{\text{ds}} = 0.3$s and only change T_{ss}. It is notable that around $T_{\text{step}} = 0.86$s, the system shows zero singular values which correspond to a null space containing an infinite number of periodic solutions. These solutions are all without swing hip or stance contact actuation referred to as pseudo-passive gaits.

that in our linear model, the legs are stretched or shortened by prismatic actuators, as part of the model construction. For pseudo-passive gaits, we remove columns associated to X_3, U, V and W in D and obtain a reduced matrix $D_{\text{passive}} \in \mathbb{R}^{12\times 11}$. Figure 6.4 shows absolute singular values of D_{passive} over time.

One can clearly see that there is a time $T_{\text{step}} = T_{\text{relax}} = 0.86\text{s}$ where the system shows an additional zero singular value. T_{relax} can be found by a simple root-finding algorithm. This singular value refers to sagittal direction while the other singular value (which is always zero) refers to lateral direction. We can simply calculate corresponding singular vectors of D_{passive} and find a null space manifold. Note that there is only one lateral solution as the value of phase variable s should be ± 1. However, the solution in the sagittal plane can be scaled by any arbitrary positive or negative value to obtain different modulated speeds. Therefore, the manifold of pseudo-passive compass gaits, in this case, is only 1-dimensional. A demonstration of normal pseudo-passive compass gaits can be found in Figure 6.6. From Figure 6.4 also, we simply conclude that for any step time other than T_{relax}, the system cannot demonstrate pseudo-passive gaits.

6.3.2 Actuated gaits manifold

In this part, we are going to find manifolds of motion which can benefit from swing hip actuation and CoP modulation as well. These inputs are of course containing constant and time varying components for both sagittal and lateral dynamics, as discussed in the previous section. With these inputs, we can pump energy into the swing leg and brake at the end of the phase to produce faster swing motions. We can also apply contact torques which modulate the CoP and resemble the fact the CoP in human goes forward from the heel to the toes during the swing phase. Here we only remove columns associated to X_3 and W in D and obtain a reduced matrix $D_{\text{active}} \in \mathbb{R}^{12\times 19}$. The corresponding absolute singular values are shown in Figure 6.5 over time.

Surprisingly, the system does not have a distinct zero singular value at T_{relax} like before. However, it has 7 zero singular values that produce a larger null space at any given step time T_{step}. The corresponding actuated gait manifold is not 7-dimensional however. The phase variable s should always be ± 1, reducing the total dimensions to 6. Besides, one can also choose active actuators and the desired speed to reduce the dimensionality further and find a unique solution.

As demonstrated in pseudo-passive gaits, at T_{relax}, D_{passive} has a certain 2-dimensional null space. Now what if we calculate a 7-dimensional null space using T_{relax} and D_{active} instead of D_{passive}? Could we still find a pseudo-passive gait out of this larger null space? In fact, it is possible, even though no distinct zero singular value is observed in Figure 6.5. The reason is that the rank of actuation space at T_{relax} is equal to 5 in the 7-dimensional null space manifold of D_{active}. This means if we constrain all of them to zero for pseudo-passive walking, we only lose 5 ranks. The other 2 ranks could still be dedicated to the phase variable s and the desired speed, like before. So, the null space manifold of actuated gaits already encompasses the one for pseudo-passive gaits and we do not need to calculate them separately. In the next subsection, we are going to show a few examples of walking gaits, using these null spaces.

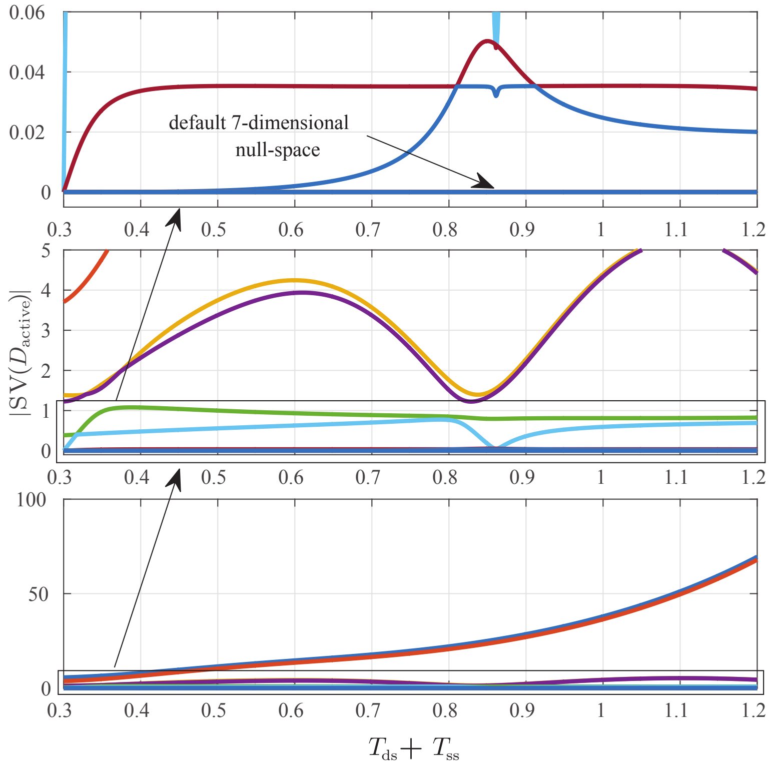


Figure 6.5 – Absolute singular values of the matrix D_{active} with respect to the step time T_{step}, plotted for an adult-size model. In these plots, we have fixed the double support time $T_{\text{ds}} = 0.3$s. It is notable that the system does not show a unique zero singular value anymore. However, there are always 7 default zero singular values that can produce gaits for any choice of T_{step}. These gaits are indeed actuated with arbitrary swing hip or stance ankle torque profiles.

6.3.3 Numerical examples

In addition to the pseudo-passive gait which has a certain timing T_{relax}, we are going to show other same-speed gaits with different timings and actuation patterns. From the singular value analysis, we have 7 singular vectors $n_i, 1 \leq i \leq 7$ for the matrix D_{active}. Each of these vectors have similar dimensions with Y in (6.38), consisting of an initial state, with the contact point (X_3) at origin, a resting swing foot ($\dot{X}_2 = 0$), a certain actuation pattern (encoded in U and V) and no disturbance ($W = 0$) of course. We pack them together in a matrix $N = \begin{bmatrix} n_1 & n_2 & \dots & n_7 \end{bmatrix}$. We also select a smaller yet more human-like choice of $T_{\text{ds}} = 0.1$s (Cappellini et al., 2006), and $T_{\text{ss}} = 0.6028$s calculated by the pseudo-passivity root finding procedure. The choice of speed will be $v_{\text{des}} = 1$m/s.

Given the constant matrix N, we want to find a vector of coefficients $\delta = \begin{bmatrix} \delta_1 & \delta_2 & \dots & \delta_7 \end{bmatrix}^T$ that combine the columns in N and produce unique solutions. We consider minimizing squared torques as an estimate of the mechanical power. The quadratic minimization is formulated as:

$$\min_{\delta,U,V} \; U^T U + V^T V$$
$$\text{s.t.}$$
$$\begin{bmatrix} S_{X_{2,x}} \\ S_U \\ S_V \\ S_s \end{bmatrix} N\delta = \begin{bmatrix} -v_{\text{des}} \times T_{\text{step}} \\ U \\ V \\ \pm 1 \end{bmatrix} \tag{6.41}$$

where $S_{X_{2,x}}$ selects the row corresponding to the sagittal swing foot position ($X_{2,x}$), S_U and S_V correspond to the constants U and V and S_s corresponds to the phase variable s. The specific way of encoding v_{des} in the optimization moves the initial swing foot position backward to find gaits with an average velocity equal to v_{des}. Note that the equality constrained quadratic optimization of (6.41) has, in fact, a closed-form solution and there is no need to solve it iteratively.

Now consider the following scenarios:

- **Pseudo-passive walking:** which is calculated as mentioned earlier. In this gait, the hip and ankle torques are all zero.

- **Long double support:** in this gait, we enforce ankle torques to zero by adding more constraints to the optimization:

  $$\begin{bmatrix} S_{U_a} \\ S_{V_a} \end{bmatrix} N\delta = 0 \tag{6.42}$$

  where S_{U_a} and S_{V_a} select rows corresponding to the stance ankle torques. Keeping the same T_{step}, we double T_{ds} and decrease T_{ss} accordingly. Note that now, the walking cannot be pseudo-passive anymore. The optimization, therefore, finds nonzero hip torques to produce the same speed and step length.

- **Stage walking:** here, we constrain the ankle torques to zero like before. Instead of optimizing the hip torques, however, we optimize lateral velocities in the cost function. In this case, the biped walks on a straight line without lateral bounce.

- **CoP modulation:** given the length of the feet, the total weight and the timing of single support, we can calculate a constantly increasing ankle torque $\frac{t}{T_{\text{ss}}}\tau_{CoP}$, acting in the sagittal plane to move the CoP forward. In this scenario, we force the time-increasing sagittal component of the ankle torques to τ_{CoP} (and other components to zero) by adding the following constraints to the optimization in (6.41):

  $$\begin{bmatrix} S_{U_a} \\ S_{V_a} \end{bmatrix} N\delta = \begin{bmatrix} 0 & 0 & \tau_{CoP} & 0 \end{bmatrix}^T \tag{6.43}$$

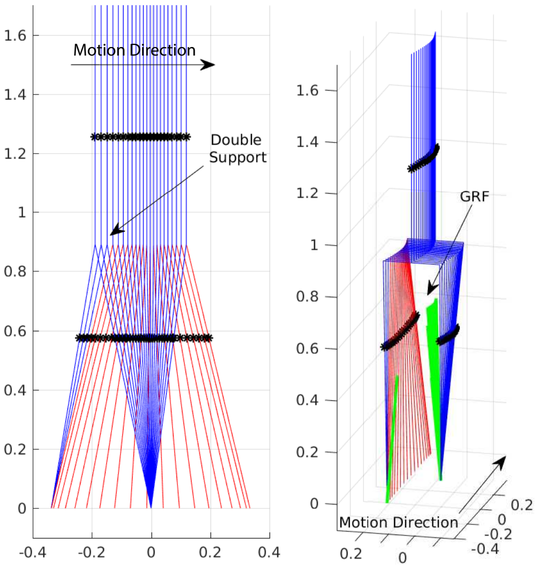


Figure 6.6 – A detailed demonstration of a full step phase in pseudo-passive walking where snapshots are taken every 30ms. Black arrows show the direction of motion, and the swing leg is shown in red. In this figure, lateral bounces could be seen on the right while velocities can be inferred from the snapshots on the left. The swing leg speeds up and slows down during a step phase while the torso has minimum speed when the swing foot is at maximum speed. It can also be observed that the swing foot approximately follows a straight line while the swing hip bounces laterally.

The result is a gait with time-increasing ankle torque profiles and minimal hip actuation.

- **LIP-like:** in this case, keeping the original timing, we change the model of the robot. We transfer most of the weight of each leg to the torso, and also move the three masses closer to the pelvis by decreasing h_2 and h_3. Again, we disable all ankle torques as well.

The 3D geometry of resulting gaits are shown in Figure 6.7 while a detailed diagram of each step is shown in Figure 6.6. The accompanied video demonstrates different features of 3LP and the five previously mentioned scenarios. It can be concluded from the Figure 6.7 that changing different parameters does not have a major effect on the overall geometry of walking. However, since it is important to match dynamics of the full model, we investigate dynamic properties of these walking scenarios as well.

Although CoM trajectories look similar in Figure 6.7, they have very different characteristics regarding velocity variations, shown in Figure 6.8. The LIP-like model shows a significant variation in the sagittal velocity. It is not so obvious how swing and torso dynamics affect this motion at first glance. By torso dynamics, we mean the torques required by the stance hip to keep the torso always upright (Hodgins and Pollard, 1997; Yin et al., 2007; Coros et al., 2010; Mordatch et al., 2010). These torques are not

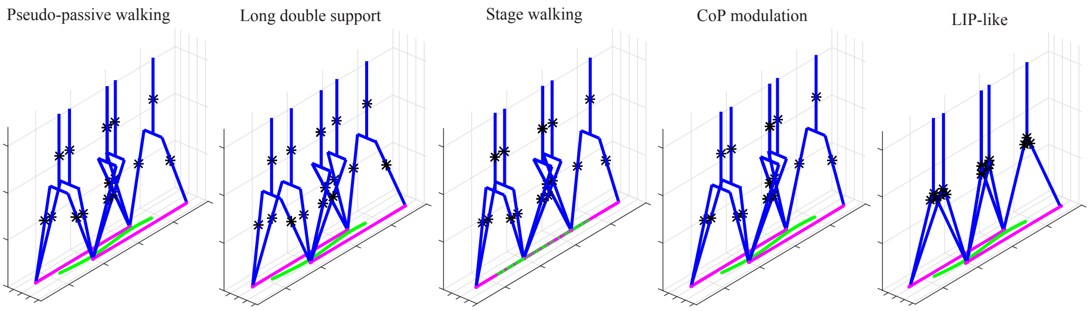


Figure 6.7 – Snapshots of different walking scenarios at 1m/s and approximately 1.5 step/s. These snapshots are taken in phase switching moments. Feet trajectories are plotted along with the projection of the CoM trajectory on the ground. In pseudo-passive walking, there is no actuation. However, one can clearly see that the model can produce CoM trajectory, lateral bounces, and swing motions. In the long double support case, the motion is geometrically quite similar. Stage walking produces no lateral bounce by using proper hip torques to let the model step on a single straight line only. On the real robot, however, one should avoid self-collision, and this motion is thus infeasible. CoP modulation also leads to a geometry similar to pseudo-passive walking, though CoM trajectory (the green line on the ground) starts a bit further from the trailing leg. The influence of CoP modulation is mainly reducing variations in the sagittal CoM speed (Figure 6.8). Finally, the motion of LIP-like model is rather similar to the pseudo-passive case, but with higher CoM speed variations (Figure 6.8). In this case, we enforce lateral footstep distances to mimic other scenarios, since the pelvis width is minimal. All corresponding walking movies could be found in the accompanying video.

necessarily zero since the pelvis has nonzero accelerations. Therefore, torso-balancing hip torques can affect the CoM motion considerably, especially since the torso is relatively heavy. Moreover, although the swing foot has a smaller weight compared to other parts of the body, it can be seen from Figure 6.9 that the swing leg has relatively large velocities during single support. Such motion increases the kinetic energy quadratically and therefore results in a significant workflow. In our model, we have described these effects in a simplified and linear fashion, yet capturing important couplings between the three pendulums.

Taking a closer look at Figure 6.8 reveals that even maximal CoP modulation still does not change velocity profiles considerably. This means the difference between pseudo-passive and LIP-like walking is way larger than that between pseudo-passive and CoP-modulated walking. In other words, CoP authority can at most convert the pseudo-passive gait to the CoP-modulated gait. The available CoP authority is hardly enough to convert the pseudo-passive gait to LIP-like gait, and this gap increases substantially at faster walking speeds. Although here the speed is moderate, we can easily infer that LIP as a template model can only operate in a very limited range of walking speeds. Remember that in fact an inverse dynamics or kinematics approach eventually realizes the template motion with the full model by exploiting all control authorities of the robot

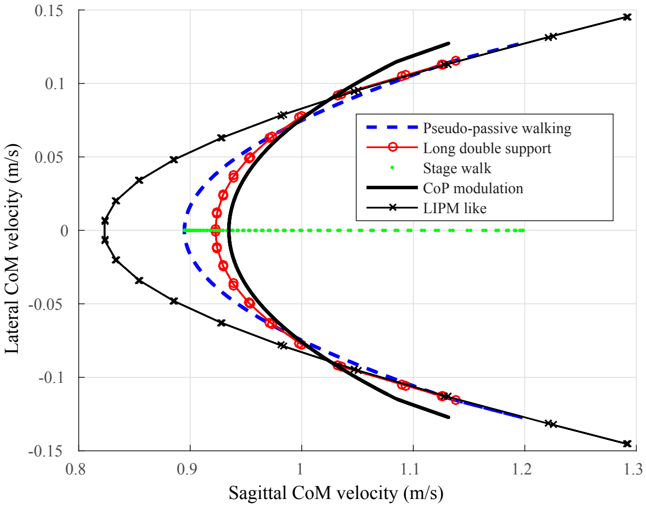


Figure 6.8 – Sagittal vs. lateral CoM velocity trajectories for different scenarios discussed. Note that the LIP-like model produces significant sagittal variations. The pseudo-passive gait shows moderate variations, however, indicating that swing and torso dynamics clearly reduce these variations. Long double support also reduces variations in both directions. By modulating the CoP, although lateral motions remain similar to the pseudo-passive gait, sagittal variations reduce even more, and the motion becomes smoother. Finally, one can see that the stage walking has no lateral motion compared to the other gaits. In general, increasing the double support duration and CoP modulation both have a similar smoothing effect on CoM velocities. However, this does not induce any argument on energy efficiency.

(including the CoP). This motivates therefore not to modulate the CoP in a template level and leave the control authority free for the underlying full-body controllers to mimic the template motion as precisely as possible.

In this section, we introduced an easy method to find manifolds of periodic motions without any numerical forward simulation of the system. Once these manifolds were found, we also showed how to find individual solutions, based on the type of actuation and timing desired. We only considered gaits with minimal hip torques here. However, to go further, we would like to investigate the effect of timing and walking speed as well. Such investigation reveals interesting energetic properties of 3LP, discussed in the next section.

6.4 Comparison with human data

Compared to LIP, the 3LP model is much more similar to human locomotion, because it describes falling dynamics, swing motion, torso balance, lateral stepping and double support features altogether. In addition to geometric similarities, we investigate ground reaction forces and joint torques to compare the underlying dynamics that result in such geometric similarity. For this purpose, regarding available data from human subjects

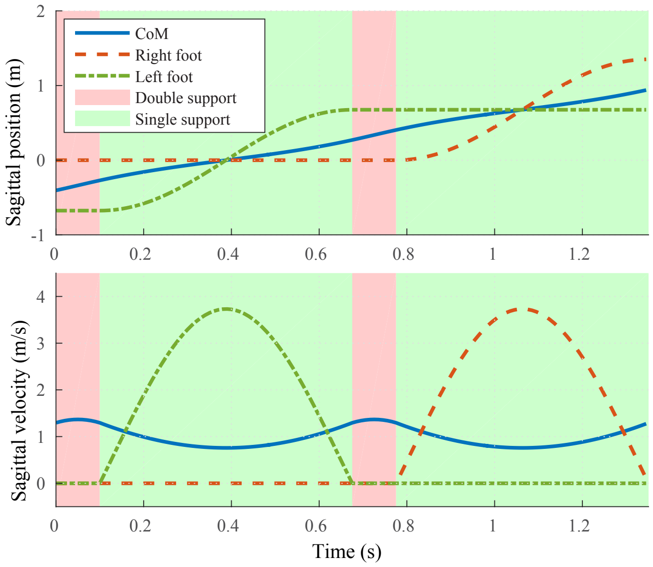


Figure 6.9 – CoM and feet positions plotted together with velocities over a 2-step motion. The average speed is set to 1m/s, the frequency is about 1.5 step/s and the pseudo-passive gait is used to obtain these trajectories. It can be observed that maximum foot velocities are up to three times larger than the average speed of the CoM. Despite a smaller mass, therefore, the swing leg can have a considerable kinetic energy.

(Eng and Winter, 1995), we selected similar model parameters and timing, calculated periodic manifolds and found solutions with the same speed and CoP modulation pattern. The resulting trajectories are demonstrated in Figure 6.10 together with average human profiles. In the following, we discuss various similarities observed in this Figure

6.4.1 Sagittal dynamics

From the last column of Figure 6.10, one can observe a good match of hip extensor and ankle plantar flexor torques as well as Anterior-Posterior ground reaction forces. This is despite a relatively fast walking speed and large step sizes (about 80% of the leg length). Note also that the constant and time-increasing components of hip/ankle torques are roughly enough to describe major trends in human curves. Nonlinear profiles in 3LP are, however, related to the stance leg and those degrees of freedom which are not directly controlled by desired input torques. The LIP model does not have hip torques and produces larger A/P GRF, because of different CoM trajectories shown in the Figure 6.8.

6.4.2 Vertical GRF

By model construction, the CoM height is constant, and we do not expect two peaks in the vertical GRF profiles, similar to SLIP-based models (Rummel et al., 2010; Sharbafi and Seyfarth, 2015). However, the general trapezoidal shape is preserved, thanks to our double support phase and its linear transition rules. Note that the LIP model can produce a similar profile too. The main consequence of a constant-height profile is walking with

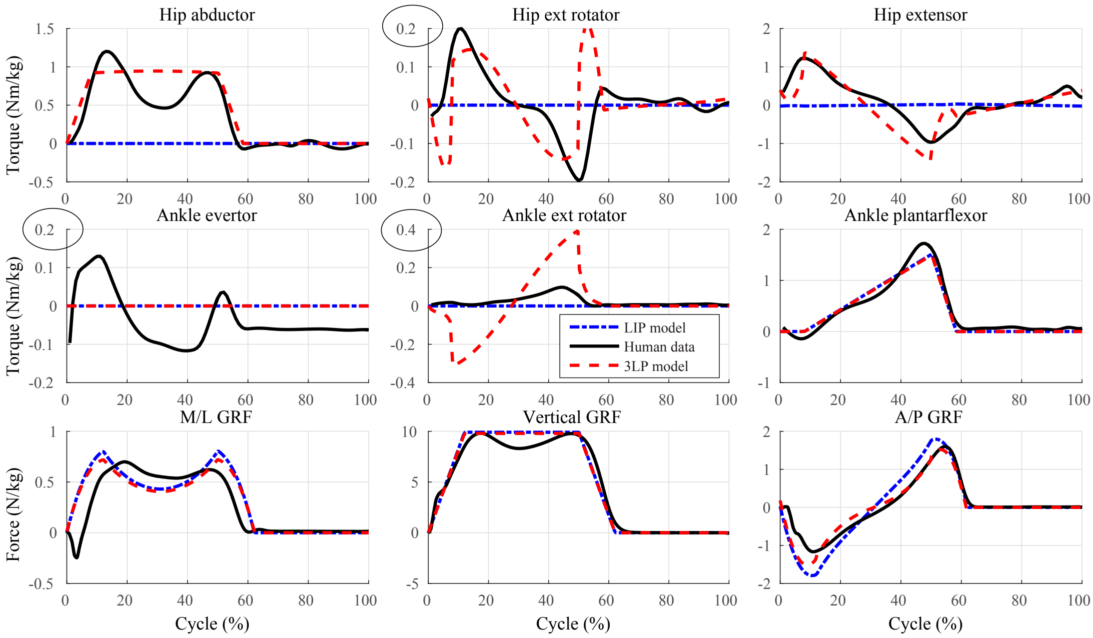


Figure 6.10 – Comparing dynamic profiles of 3LP and LIP with normalized human data, taken from (Eng and Winter, 1995). This data is for male subjects with average mass of 77.2kg and height of 1.8m, walking at 1.6m/s and 108 steps/min. Here we use the same CoP modulation for both LIP and 3LP for better comparison. In these curves, we have demonstrated hip/ankle torques as well as ground reaction forces. Note that our model does not have any knee and we consider ankle torques to approximate contact wrenches. Most of the profiles in 3LP match the human data quite well, although lateral and transversal dynamics have some discrepancies. The LIP model is, however, unable to describe hip torques as it does not include swing and torso dynamics. It is also remarkable that our specific assumptions in (6.16), (6.17) and (6.20) move the discontinuity on variables with smaller magnitudes (shown with circles).

crouched knees which looks less human-like compared to many other template models listed in Figure 6.1.

6.4.3 Lateral dynamics

In 3LP, we only minimize hip/ankle torques to find a unique solution out of the large manifold of all symmetric periodic gaits. This minimization is not necessarily realistic and human-like, as it leads to wider lateral steps, more bouncing, and often larger transversal ankle torques. A better cost function on energy might produce more human-like gaits, although it is doubted in (Workman and Armstrong, 1986) that optimal gaits merely depend on energy terms. There might be terms related to balancing performance as well, at least in lower speeds. In faster speeds also (like the human profiles demonstrated here (Eng and Winter, 1995)), humans take closer steps laterally compared to our model.

In Figure 6.7), we see that 3LP keeps general trends like double peaks in the Medio-Lateral (M/L) GRF, but cannot precisely describe other torque profiles. Humans normally tend to step as close as possible to minimize lateral motions and energy (Kuo, 1999) (Similar to stage walking in Figure 6.7). However, humans swing their foot over an arc shape to avoid self-collision. Such a fine motion is feasible in 3LP but requires a better objective function in (6.41). Note that the sagittal swing motion can influence lateral dynamics as well (Kuo, 1999; Collins et al., 2005), possibly through transversal moments. This might be another reason for the discrepancy observed between different lateral curves since 3LP completely decouples the lateral and sagittal dynamics. In the LIP model, though, there is no pelvis included. However, we consider a gait with the same step-width as 3LP. Although LIP does not have hip torques, M/L GRF forces are yet similar to 3LP, shown in Figure 6.7.

6.4.4 Transversal rotation torques

3LP preserves the general trend of transversal torques observed in human, but not matching precisely, especially in the ankle. One major reason is that arm motions, and pelvic rotations are not considered in the model. Another important reason is wider lateral steps in 3LP compared to human which require larger transversal moments. Note that transversal torques are needed to keep the torso upright and straight ahead during swing phase, compensating the moment produced by the swing leg. In LIP, however, since there is no pelvis and swing leg, we do not expect transversal torques.

6.5 Discussion

Compared to most of other template models listed in Figure 6.1, our proposed model considers swing and torso dynamics in a linear formulation. On the other hand, it is computationally similar to LIP which is vastly used in the literature to control humanoids over a range of relatively slow walking speeds (Sakagami et al., 2002). Nonlinear models are also popular in controlling simpler robots (Collins et al., 2005), but again over a limited range of speeds.

Template models describe major dynamics of the robot in an abstract way used for motion analysis or synthesis, probably in a hierarchy with more complex full models. In such control paradigms, it is important to keep computational costs as minimal as possible, favoring future prediction. On the other hand, template models should match the full models dynamically. 3LP can describe many features of human walking, and consequently, it is more precise for controlling humanoids, compared to many other single-mass models. The energy flow in 3LP is also more similar to the human, providing more natural motions for humanoids which have similar anthropomorphic features.

In 3LP, the pelvis width parameter has a linear effect on the lateral motion. This parameter is used to find lateral ankle/hip torques and to determine a natural lateral foot placement. This is compared to many other methods, e.g. the MPC controller of chapter 2, where the two feet are forced to be apart to avoid self-collision. In the literature, the timing and footstep locations are imposed without enough knowledge of internal dynamics. In our model, however, since swing dynamics is included and a zero final velocity is assumed for the feet, natural periodic gaits automatically come out of equations. The assumption mentioned relieves the need to calculate impact forces and determines the timing and periodicity conditions as well.

3LP can predict human walking profiles quite well, even in relatively fast speeds where the linearity assumption might not be realistic. Although IP-based models can demonstrate CoM excursions quite well, a nonlinear nature makes them less suitable for highly complex robots that require online planning. There are more advanced versions of IP-based models in the literature, including torso and swing dynamics. However again, nonlinear equations cannot be used in a per-time-step MPC control. The proposed model is based on a reasonable trade-off between geometric and dynamic matching, favoring fast computation properties. We would like to mention that the vertical excursion of CoM, even in a very fast walking gait at 2m/s is still about 5cm (Gard et al., 2004) which is quite negligible compared to a step-size of about 1m (pelvis excursion is about 7cm however). CoM excursion depends on step-size which does not increase linearly with the walking speed. Humans increase the frequency as well, which in consequence affects swing dynamics and demands more energy from the hip muscles.

We did not set up control frameworks in this chapter. Rather, we focused on biomechanical analysis and similarities to human locomotion that can be inspiring for generating more precise abstract plans, used to control humanoid robots. In brief, 3LP provides:

+ Swing dynamics.
+ Torso balancing torques.
+ Double support phase.
+ Hip/ankle actuation possibilities.
+ Natural lateral motion.
+ Natural periodic gaits.
+ Pseudo-passive compass gait.
+ Computational advantages.
+ Possibility to consider hip torque limits.

Despite limiting factors such as:

- Constant CoM height.
- Flat vertical GRF profiles.
- Stretched legs.
- No steering capabilities yet.
- No arm motion.
- No torso pitch/roll DoF.

3LP can be extended to have two more degrees of freedom for the torso. It can also include quadratic actuation terms to produce more accurate torque profiles. It should be noted that without pelvis, steering is still possible as demonstrated in chapter 2 which is based on LIP. Steering makes 3LP nonlinear, but one can compromise the lateral motion and let inverse dynamics find proper actuation patterns. Without the pelvis in 3LP, one actually needs to impose the lateral bouncing as we did in the LIP-like scenario. An important role of pelvis is, therefore, to produce a natural lateral motion which automatically emerges from the optimization of (6.41). This makes the model more generic without the need to impose the lateral bouncing. Note that the natural step width is not easy to determine, since it depends on the leg length, stepping frequency and available ankle torques.

It is worth mentioning that an imposed step width which is similar to the natural step width (found when including the pelvis) might not produce very different GRF profiles. This is induced from Figure 6.10 by comparing GRF profiles of the LIP-like scenario and the full 3LP model. As a result, one can easily remove the pelvis and allow for steering with the cost of imposing the step width. This can be realized by finding a natural step width with the pelvis and then imposing it in another version of 3LP which does not have the pelvis anymore. The second model remains natural (in terms of GRF), linear and of course suitable for producing steering motions.

In future work, we are going to replace LIP with 3LP in our MPC-based control framework of chapter 2. 3LP can be used in both state estimation and planning levels. It can possibly exchange information about the CoM or the feet with the full model. Dynamic equations of 3LP can, therefore, predict future states in a MPC framework or help to filter sensory noises by Kalman filtering. The linear equations of 3LP can indeed provide a non-periodic formulation of the system as well, where footsteps might act as inputs to the system instead of the hip torques. This reformulation is similar to planning footsteps in MPC control of chapter 2 or Capturability frameworks (Koolen et al., 2012).

Among all advantages offered by 3LP, we favor its capability to produce more natural motions. In this regard, we can expect our inverse dynamics layer to track the template model more precisely and therefore, being able to produce more human-like motions. Including torso and swing dynamics in template models make them computationally more complicated, but closer to reality. In the literature, complex models are mostly used for offline trajectory generation. In 3LP, however, we partially include these additional dynamic effects while keeping the model yet suitable for online control and future prediction. The focus of this chapter was to introduce the model and to explore different capabilities. In future, we integrate it with our hierarchical walking controller to achieve more dynamic motions. This chapter is accompanied with a video, demonstrating general features of 3LP together with the five gait scenarios discussed.

Contributions in Control Part III

7 Time-Projection Control

Continuing the previous part of the thesis which aimed at developing better template models, this part will focus on walking control and application to the real hardware. We start this part by presenting our theoretical contribution to walking control in this chapter and then present the hardware experiments next in chapter 8. Remember from chapter 2 that the MPC controller was able to adjust footstep locations online. This was achieved thanks to closed-form equations of the linear inverted pendulum model as well as an efficient optimization implementation. The new 3LP model described in chapter 6 provides the same linear equations with better mechanical properties. However, we are interested in the output of MPC optimizations in terms of footstep adjustment strategies. In other words, we look for simpler control rules that provide such optimality and robustness in online push recovery and gait generation scenarios. In this chapter, we aim at finding such universal footstep adjustment controller. We use the 3LP model, find discrete motion equations, derive discrete LQR controllers and propose a time-projection method that maps discrete errors to continuous errors. The time-projection method enables us to perform online reaction to perturbations while being able to modulate the motion speed at the same time. This chapter formulates the new control paradigm, applies it on walking stabilization and extensively analyzes the resulting push recovery performance. Later in chapter 8 for real robot experiments, we derive simple and generic look-up-table control rules that suggest online footstep adjustments based on deviations from the 3LP gait[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji, Philippe Muellhaupt, and Auke Jan Ijspeert. "Time-projection control to recover inter-sample disturbances, application to walking control." arXiv preprint arXiv:1801.02150, 2018.

The first author developed the method and wrote the manuscript. The second author helped with the mathematical proof, development of a simple example and formatting of the manuscript to make it smooth and easy to understand.

---

[1] All the videos of this chapter could be found at https://youtu.be/SdhjV-adClM

7.1 Background

Performing bipedal locomotion for humanoid robots is a challenging task regarding many different aspects. On one hand, the hardware should be powerful enough to handle the weight and fast motions of the swing legs. On the other hand, controllers require accurate perception and actuation capabilities for better stabilization and predictability of the system. Power and precision are two different and sometimes conflicting requirements. From the perspective of geometry also, complex chains in each leg of the robot make the control problem multi-dimensional. Due to an under-actuated nature and geometric complexities, controlling humanoids in dynamic walking is not trivial. The hybrid nature makes such control paradigm even more complicated since the system model changes in each phase of the motion (i.e., left and right support phases).

In a model-based controller which benefits from mechanical models of the system, it is common to break the complexity into multiple levels. Actuator dynamics are identified and compensated for in a separate block of position or torque controller. The kinematic or rigid body dynamic model is placed in an interface block to translate Cartesian variables to joint variables and vice versa. Then, the whole complex system is approximated with a simpler template model to describe walking dynamics (Kajita and Tani, 1991; Faraji and Ijspeert, 2017a). The fewer dimensions of such a template model compared to the actual system makes it possible to apply classical control theories in real time, within capabilities of computation units and agility of the motion. The under-actuated hybrid nature of walking is then handled with template models. In this regard, hierarchical model-based control approaches can handle complexities in different levels and use models to capture main dynamics of the actuators, limbs or the full body.

7.1.1 Template models

Simplified models can speed up the calculation of footstep plans, Center of Mass (CoM) and foot Center of Pressure (CoP) trajectories. A good walking controller should stabilize the under-actuated part of the system (also called falling dynamics) either by modulating the CoP through ankle torques, or by regulating the angular momentum through rotation of the torso, or finally by taking proper footsteps (all discussed in (Stephens, 2007)). The first two strategies provide continuous control, but with strict bounds on the available ankle torques (determined by the foot size) or body rotations. The third strategy, however, uses the hybrid nature of walking and stabilizes the system in successive phases by adjusting footstep locations. Inverted Pendulum (IP) (Kuo et al., 2005) and its linear version (LIP) (Kajita and Tani, 1991; Koolen et al., 2012) are probably the simplest models used for these three strategies, concentrating the whole mass of the robot in a point and modeling the legs with massless inverted pendulums. In these models, swing dynamics is absent, and therefore, the timing and the final swing-leg attack angle is imposed by the controller. A proper attack angle which directly translates to footstep location can stabilize the system through the third strategy, but it might require a precise full-model inverse dynamics to ensure CoM and swing motion tracking. There are more complex versions of inverted pendulum with masses in the legs (Byl and Tedrake, 2008),

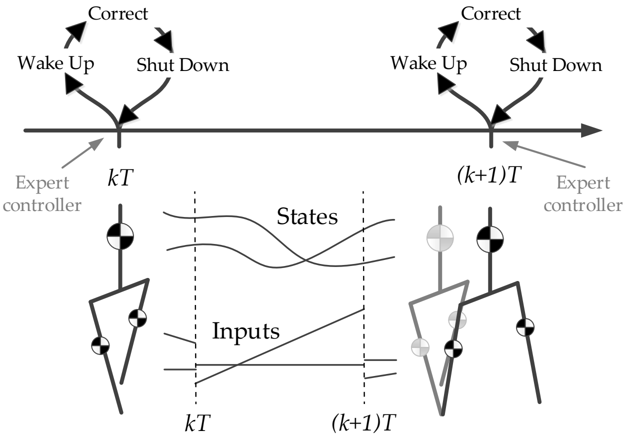


Figure 7.1 – The role of a discrete controller in regulating inputs at the beginning of each phase to stabilize the robot and to track the nominal periodic solution. After one reaction, this controller has to wait until the next event to correct for the accumulated error of all inter-sample disturbances.

torso (Westervelt et al., 2007) and knee for the swing leg (Asano et al., 2004). In all advanced versions of IP, due to non-linearity, optimizations and numerical integrations are needed to obtain periodic walking gaits. These gaits are indeed more natural than simple IP-based gaits, due to inclusion of swing/torso dynamics.

7.1.2 Discrete control

Linearization of template systems around periodic walking gaits offers a linear model for control and stabilization using the third stabilization strategy (Rummel et al., 2010). Such a linear model describes the evolution of possible deviations from nominal trajectories and the effect of inputs, particularly footstep locations. Discretization of this linear model between specific gait events, e.g., touch down or maximum apex moments can form the basis for a discrete controller that adjusts inputs when the event triggers. This approach is shown in Figure 7.1 where an expert controller provides corrective inputs based on deviations observed at each event. The linear inverted pendulum (LIP) provides analytical solutions in this regard and is widely used in slow-walking locomotions (Feng et al., 2013; Faraji et al., 2014a; Herdt et al., 2010b; Koolen et al., 2012). In this model, similar to IP, the next footstep location and timing should be imposed as the original model does not include swing dynamics.

7.1.3 Limitations of discrete control

The discrete control paradigm of Figure 7.1 benefits from a very simple discrete map and the control rate is adjusted to the frequency of phase-switches. The synchrony of control with the hybrid phase-changes is of particular interest in walking. In the frame of third stabilization strategy through footstep locations, once a swing phase starts, the falling dynamics keeps growing until the next contact point is established to absorb

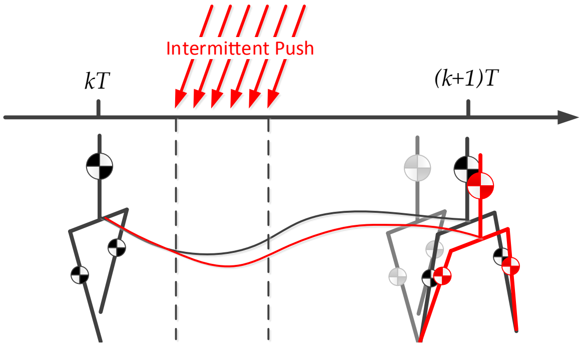


Figure 7.2 – Demonstration of an intermittent push that appears shortly during a continuous phase and influences the system. The normal and disturbed trajectories are shown in black and red respectively. Here, the final footstep location of this phase is not adjusted online. A delayed reaction to disturbance only at the beginning of the next phase might produce a large overshoot in the next footstep locations.

the extra energy. An effective stabilizing correction is therefore applied only at hybrid phase-change moments (Zaytsev et al., 2015; Kelly and Ruina, 2015; Byl and Tedrake, 2008). The first strategy (CoP modulation) offers continuous stabilization authority through modulation of CoP, though the effect is insufficient due to small foot dimensions. This strategy seems more suitable for slow walking speeds and short footsteps, popular for controlling humanoid robots. The second strategy (angular momentum) might lead to unwanted torso oscillations in practice and can not be used alone unless a power-full momentum wheel is mounted on the robot. The third strategy (foot stepping) is more effective in faster speeds and larger footsteps which explains the synchrony and motivates for discretization of the linearized model at hybrid phase-changes, but one should create a library of controllers to handle linearization around different gait conditions (Kelly and Ruina, 2015; Manchester and Umenberger, 2014; Gregg et al., 2012).

Although a discrete model can predict the future very rapidly in terms of computation, there might be intermittent disturbances that shortly act on the system at any time and disappear. As mentioned earlier, discretization of walking models is usually synchronized with phase-change moments. However, due to unstable falling dynamics, the phase period (i.e., stance period) is long enough for even moderate intermittent disturbances to accumulate and result in a considerable deviation. Such destructive effect (shown in Figure 7.2) requires taking a large step in the next phase whereas the footstep location of that same phase could be adjusted online to stabilize with less effort. This raises an interesting control problem in which the discretization rate is reasonable but too slow to handle inter-sample disturbances.

7.1.4 Continuous control

As mentioned earlier, the next footstep has a vital stabilization role, but effective only once the next contact is established. The interesting control problem is in online adjustment of this footstep location, although it does not have a considerable effect instantaneously in the middle of the phase. Without such online adjustment and keeping the swing destination fixed, the second next footstep (which is found in the next event) might become very large to compensate for the accumulated error. This problem might naturally happen in a hybrid system where a deviation of one state (e.g., swing location) might not be important in one phase due to a weak mechanical coupling, but becomes very important in the next phase where it has a strong mechanical coupling (e.g., when it becomes stance foot location). This effect is due to a hybrid change of the system model through motion phases. In normal continuous systems without hybrid phases, however, the coupling between variables remains the same. In other words, since the continuous model remains the same, it is possible to discretize at any rate and apply standard controllers such as Discrete Linear Quadratic Regulators (DLQR) (Ogata, 1995). If inter-sample disturbances were found to be significant for these normal systems, one could naturally increase the control rate. In our hybrid system, however, this is not straightforward and leads to a complicated model-variant DLQR design. Therefore, DLQR controllers are not used very often for online control of footstep adjustment, although they are extensively used in the first stabilization strategy (CoP modulation) and traditional discrete Poincaré based methods (Month and Rand, 1980) which linearize the whole motion phase.

Despite complications of DLQR design in hybrid systems such as walking robots, there are alternative ways to realize online adjustments with very simple laws. Raibert used a known yet simple approach for hopping where the footstep adjustment was a function of CoM forward speed (Raibert et al., 1984). This intuitive law with a hand-tuned coefficient moved the footstep location further when a faster forward speed was detected. This idea was later re-formulated for walking with the linear inverted pendulum in (Stephens, 2007; Koolen et al., 2012) where the coefficient had a more physical meaning. The idea was to capture the motion, i.e., to find a footstep location where the CoM ends up on top of this point with zero velocity. In both frameworks, the footstep location can be adjusted online in a fast reaction to intermittent pushes and disturbances.

In chapter 2, we used analytical solutions of LIP and a Model Predictive Controller (MPC) to predict CoM trajectories and adjust footstep locations. At each instance of time, we used closed-form solutions of LIP to predict state evolution until the end of the same phase as well as few successive phases. Considering footstep locations as inputs for this discrete system, MPC was able to adjust the next footsteps to stabilize the robot, recover from pushes and converge to a specific walking gait. This numerical optimization could extend the capturability framework (Koolen et al., 2012) while providing the same online adjustment capabilities. We were also able to generate lateral bounces and turning gaits naturally inside the optimization while the capturability framework does not have this flexibility. The idea of capture point helps stabilizing the robot when the ankle torques are not enough, i.e. when the instantaneous capture point falls outside the support polygon. The capturability framework, however, requires an intuitive proportional gain to move the desired capture point away from the robot when a walking behavior is desired (Pratt et al., 2012). Besides, swing trajectories are artificially designed to reach the destination, because the underlying template model (LIP) does not have swing dynamics. In contrary, our proposed controller automatically generates walking gaits and swing motions while the stabilizing controller is designed for both in-place and progressing walking gaits. However, inequality constraints of (Koolen et al., 2012) are not yet considered in our method which would be discussed later in the section of viable regions.

7.1.5 A better template model?

In our MPC controller of chapter 2, the resulting Cartesian CoM trajectory and an artificial swing trajectory (which terminates in the optimized footstep location) are given to an inverse dynamics interface to find required joint torques for the robot. Although MPC provided a robust performance by finding proper footstep locations, the practical and feasible range of motions strictly depended on the choice of timing, artificial swing trajectories, foot dimensions and the maximum step-length which should not violate the constant CoM height assumption. This motivated us to develop a more complicated yet linear model that captures falling, swing and torso dynamics altogether. This model, called 3LP is composed of three pendulums for the torso and the two legs with masses that all remain in constant-height planes (refer to chapter 6). Closed-form solutions make 3LP suitable for online MPC similar to the framework we introduced in chapter 2. In this regard, 3LP could relieve the need to produce artificial swing trajectories added to LIP. The inclusion of swing dynamics, however, changed the gait generation paradigm completely.

Remember that with the LIP model, the controller has to impose the attack angle and phase time. This can create a library of gaits with different speeds and frequencies, but not necessarily feasible to track with inverse dynamics. According to early bio-mechanics studies (e.g., (Zarrugh and Radcliffe, 1978)), human's walking frequency increases with speed due to certain mechanical and muscle properties of the body. The particular ascending relation observed in human is the result of optimizing Cost of Transport (CoT) (Bertram, 2005) defined as the energy spent over the unit of traveled distance. This quantity is due to a trade-off between falling, swing and leg lift dynamics. When translated to humanoids, the concept approximately holds in the sense that the ankle, knee and hip joints have to realize push-off, leg lift, and swing works. Thanks to the inclusion of swing dynamics, 3LP uses joint torques to describe the swing motion while LIP is blind to joint torques entirely. Gait symmetry concepts applied to the closed-form equations of 3LP can produce a class of gaits which could be optimized energetically, based on their required hip torques. Such optimal gaits have some dynamic torque and ground reaction profiles similar to the human, despite the constant CoM height assumption.

3LP gaits are obtained by combining eigenvectors of a particular matrix linearly without a numerical integration. Thanks to linear equations, the linearization of system equations around these gaits is very easy to calculate. The tracking problems of inverse

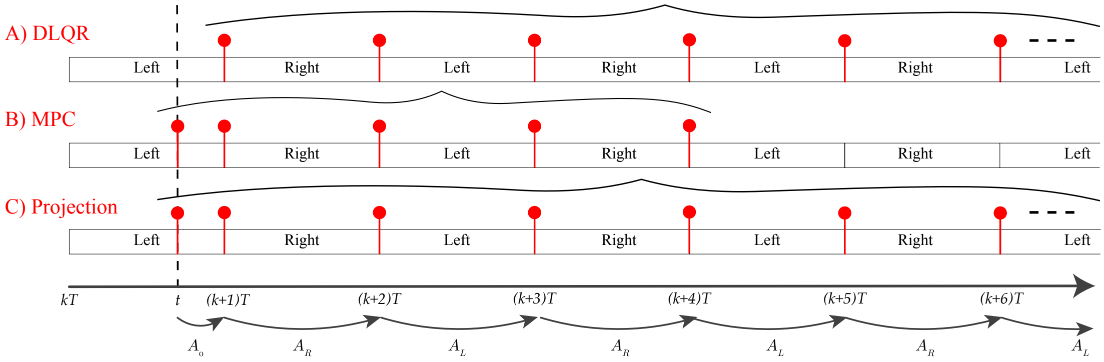


Figure 7.3 – Conceptual comparison of A) DLQR controller, B) receding horizon MPC and C) the proposed projection controller in dealing with the state observed at current inter-sample time t in the left support phase. The DLQR controller only updates inputs at certain discrete events. The receding-horizon MPC controller of chapter 2 used the remaining-phase transition matrix A_0 and the next transition matrices A_L, A_R to optimize the footsteps at each time-step and provide online reaction. The proposed projection controller provides the same online updating feature, but uses the same DLQR expertise without any numerical optimization of MPC.

dynamics arising with the choice of gait timing and artificial swing trajectories with LIP are now improved with dynamics of 3LP, and finally, we can still find analytic transition matrices between any two instances of time in the phase with 3LP. With these new features, is MPC still needed? As we will see next, the answer is no.

7.1.6 Time-projection controller

Intermittent disturbances can have different magnitude, duration and timing. In walking, early-phase pushes can disturb the system more severely due to the unstable nature of falling dynamics. Traditional Poincaré based methods (Month and Rand, 1980) which linearize the system around a pre-optimized gait cannot capture such continuous effects. In other words, all disturbances happening between the two discrete events are accumulated and observed only at the next event. To set up a continuous control, however, there should be a measure to evaluate the error in the middle of a continuous phase. Also, one also needs to know the effect of available continuous inputs (like hip torques/footstep locations). Any-time transition matrices of LIP and 3LP make this evolution prediction possible.

Although LIP can produce gaits used for DLQR control (i.e. event to event control of Figure 7.1), we used MPC in chapter 2 to handle intermittent pushes. The difference between these two frameworks can be seen in Figure 7.3. A DLQR controller could be used on top of LIP gaits to find discrete adjustments (Figure 7.3.A). However, when performing continuous online control, we needed MPC to handle the remaining-phase transition matrix A_0 (in Figure 7.3) and the hybrid nature of phase-switches in a receding horizon fashion (Figure 7.3.B). An infinite-horizon MPC without constraints translates

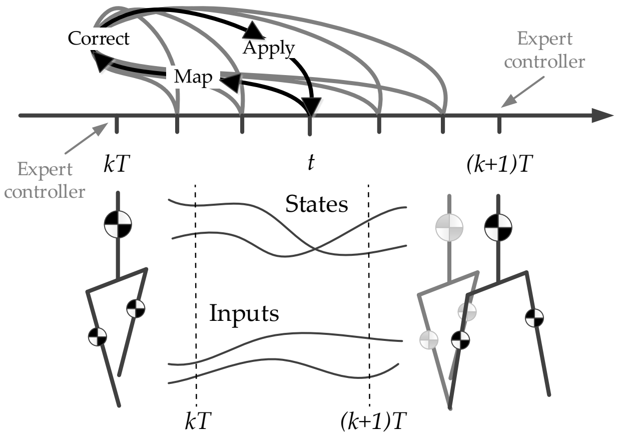


Figure 7.4 – A demonstrative schematic of time-projection idea. This controller in fact relies on the expertise of the DLQR controller. At each time-step t, the measured state is mapped to the beginning of the phase, optimal inputs are calculated using the DLQR controller and then, these inputs are used at the current time-step t.

to DLQR, but numerical optimizations of MPC with limited horizons offered flexibility to model switches and the inclusion of inequality constraints.

In this work, we propose a new online control approach based on the DLQR controller. In discrete control architectures (Kelly and Ruina, 2015; Byl and Tedrake, 2008; Manchester and Umenberger, 2014; Sharbafi et al., 2012), a certain event is considered to decide the new angle of attack or push-off force. Similarly, we use the DLQR controller in our proposed architecture as a core stabilizing expert. For any time instance t in the middle of a continuous phase, we project or map the currently measured error back in time to the previous touch-down event, where the DLQR controller knows best how to handle it. We take the output of DLQR then and apply it to the system in the middle of the phase at time t. Such online policy makes sure that the future evolution of the system given the calculated input will be optimal if seen in a longer time-span over multiple future steps (Figure 7.3.C). Because of unstable falling dynamics, a push of certain magnitude and duration might have different effects if applied early or late during single support. Our continuous projection controller is therefore expected to handle such sensitivity to timing, i.e., handling continuous disturbances.

Figure 7.4 demonstrates the time-projection idea. This new method is simple to execute, and it does not need a redesign of the controller, offline optimizations or numerical MPC optimizations. Back projection idea is also used in (Byl and Tedrake, 2008) where a post-collision state should be rewound to a pre-collision state where an additional impulse should be added. Then the system is forward simulated again to find the contribution of the additional impulse. Here we use a similar projection to map the continuous error at any time back to the discrete event, but then use the LQR controller to find control inputs to be used at the current time-step which is conceptually different.

7.1.7 Novelty

Time-projecting control for a walking system is based on three main motivations:

1. DLQR control rate is normally synchronized with the walking frequency. Due to the unstable falling dynamics, this controller is sensitive to the timing of inter-sample disturbances.
2. The DLQR controller can already stabilize the system through footstep adjustments. There is no need to redesign this part of the controller to capture inter-sample disturbances.
3. An online controller is needed. We want to benefit from DLQR's stabilization property, but avoid numerical optimizations of a receding horizon MPC.

The contribution of this work lies in formulating the time-projection idea and applying it to walking with 3LP in simulations. The aim is to achieve the same performance of online MPC (Faraji et al., 2014a; Feng et al., 2013; Kuindersma et al., 2014) in push recovery scenarios, but without numerical optimizations which might need a considerable computational power. We use 3LP for generation of more feasible gaits compared to LIP, although all control concepts introduced could be applied to LIP as well. An advantage of MPC could be the incorporation of inequality constraints on input torques, the center of pressure, friction cones and footstep length to ensure feasibility of the plan. The proposed method does not consider inequality constraints, but thanks to an analysis of viable regions (Zaytsev et al., 2015), we provide a simple criterion that indicates whether projection controller fails or not, i.e., whether the robot should switch to a more complex controller or even emergency cases. Further analysis of viable regions indicates that extreme conditions rarely happen in normal frequencies and stride lengths. Therefore, our simple projection controller is enough most of the time, and if not, all other controllers including MPC can hardly improve stabilization. We would also like to remark that 3LP already describes the pelvis width and thus avoids internal collisions in natural walking. In extreme cases where a lateral crossover is needed, an MPC controller with non-convex constraints or advanced collision avoidance algorithm might be needed.

The rest of the chapter starts with formulation and discussion of time-projection idea for general linear systems. Demonstration of intermittent push recovery and speed tracking scenarios follow afterward with an extensive analysis of system eigenvalues, push recovery strength and viable regions. We conclude the chapter by comparing the proposed controller with other candidate controllers in the literature and discuss broader control applications which could benefit from the time-projection idea.

7.2 Time-projection

Continuous-time LQR controllers aim at optimizing system performance over an infinite horizon in future. In discretized and digitized controllers, this translates to a DLQR controller with similar discrete system evolution formulas. We mention that for walking applications, the discretization rate is often set to the step frequency. This is mainly because over the hybrid model switch, the final swing location which weakly couples with other system variables (i.e., CoM state) in the swing phase will have a strong coupling in the stance phase. Since footstep adjustment becomes effective only when the next contact is established, the actual control update events are set to the start of hybrid phases in discrete controllers. We challenged this paradigm by inter-sample pushes and motivated the necessity of updating footstep locations online, in the middle of the swing phase. At the same time, we want to keep the same optimality of DLQR controllers over an infinite horizon.

Time-projection is more suitable for systems that have hybrid phases with a variable model in each phase. In ordinary systems with fixed models and continuous phases, a simple increase in discretization frequency is more straightforward and probably more effective. We explain the time-projection idea with continuous models though, aiming at simpler and more comprehensible formulations. The 3LP model and hybrid phase-changes are discussed later in the 3LP section, and appropriate time-projection formulas are included in the Appendices D and E for further information.

7.2.1 Linear system

To present the idea behind time-projection in a simple way, we consider a linear time-invariant system in which an error should be regulated to zero at certain sampling times. Define a state vector $x(t) \in \mathbb{R}^N$ and a control vector $u(t) \in \mathbb{R}^M$. The system can be described by:

$$\dot{x}(t) = ax(t) + bu(t) \tag{7.1}$$

where $a \in \mathbb{R}^{N\times N}$ and $b \in \mathbb{R}^{N\times M}$ are constant matrices. The closed-form solution of this system at time t is obtained by:

$$x(t) = e^{at}x(0) + \int_0^t e^{a(t-\tau)} bu(\tau)\,\mathrm{d}\tau \tag{7.2}$$

For simplicity, we consider a constant input here, although this can be extended to linear or quadratic forms without loss of generality. With a constant input, parametrized by the vector $U \in \mathbb{R}^M$, the equation (7.2) takes the form:

$$x(t) = e^{at}x(0) + (e^{at} - I)a^{-1}b\,U \tag{7.3}$$

where we assumed a is invertible. If a is singular, a similar expression can be obtained by considering the Jordan form of a. We consider a period time $T > 0$ at which the behavior of this system could be described discretely:

$$X[k+1] = A(T)X[k] + B(T)U[k] \tag{7.4}$$

where $X[k] = x(0)$, $X[k+1] = x(T)$ and:

$$\begin{aligned} A(T) &= e^{aT} \\ B(T) &= (e^{aT} - I)a^{-1}b \end{aligned} \tag{7.5}$$

7.2.2 DLQR controller

Assume this system has a nominal solution $\bar{X}[k]$ and input $\bar{U}[k]$. Due to linearity, we can define error dynamics as follows:

$$E[k+1] = AE[k] + B\Delta U[k] \tag{7.6}$$

where $E[k] = X[k] - \bar{X}[k]$ and $\Delta U[k] = U[k] - \bar{U}[k]$. Here we dropped function inputs (T) from $A(T)$ and $B(T)$ for simplicity. Assume this system is controllable and a DLQR controller can be found to minimize the following cost function and constraint:

$$\begin{aligned} \min_{E[k],\Delta U[k]} & \textstyle\sum_{k=0}^{\infty} E[k]^T Q E[k] + \Delta U[k]^T R \Delta U[k] \\ s.t. \;\; & E[k+1] = AE[k] + B\Delta U[k] \quad k \geq 0 \end{aligned} \tag{7.7}$$

where Q and R are cost matrices. The optimal gain matrix calculated from this optimization is called $K \in \mathbb{R}^{M\times N}$, producing a correcting input $\Delta U[k] = -KE[k]$.

Conceptual trajectories plotted in Figure 7.5.A demonstrate the set point nominal solution $\bar{X}[k]$ and $\bar{U}[k]$ without any disturbance. As shown in Figure 7.5.B, in presence of disturbances, this open-loop system might be unstable and easily diverge from nominal trajectories. The nature of DLQR controller designed earlier is shown in Figure 7.5.C. This controller is only active at time instants kT and provides corrective inputs to the system until the next sample $(k+1)T$. In this period, the effect of any disturbance on the system is cumulative, without any correction. Because of an exponential nature in (7.2), even a small intermittent disturbance might create a substantial error at time $(k+1)T$. This issue might be fixed by increasing the resolution and redesigning the DLQR controller depending on disturbance dynamics. However, we can take advantage of the already designed DLQR controller and increase the control samples easily.

7.2.3 Projection controller

Thanks to closed-form system equations (7.3), we can map the state measured at any time t to the samples before and after. Figure 7.5.E shows projection controller idea between samples kT and $(k+1)T$. In brief, to apply time-projection:

1. Measure the current state $x(t)$ at time t.

2. Project $x(t)$ back in time with an unknown $\delta\hat{U}_t[k]$ to find a possible initial state $\hat{X}_t[k]$.

$$x(t) = A(t-kT)\hat{X}_t[k] + B(t-kT)(\bar{U}[k] + \delta\hat{U}_t[k])$$

Here, hat notation means that these variables are just predicted at time t and they are not actual system variables. The subscript t also indicates dependency on t.

3. Now, given $\hat{X}_t[k]$ and $\bar{X}[k]$, find a projected error $\hat{E}_t[k]$ at the beginning of the phase.

$$\hat{E}_t[k] = \hat{X}_t[k] - \bar{X}[k] \tag{7.8}$$

4. The expertise of DLQR controller can be used here to apply a feedback on $\hat{E}_t[k]$.

$$\delta\hat{U}_t[k] = -K\hat{E}_t[k] \tag{7.9}$$

5. Assuming that the output of DLQR is the same unknown $\delta\hat{U}_t[k]$, one can solve a system of linear equations to find $\delta\hat{U}_t[k]$. Defining $\bar{x}(t) = A(t-kT)\bar{X}[k] + B(t-kT)\bar{U}[k]$ and $e(t) = x(t) - \bar{x}(t)$ we have:

$$\begin{bmatrix} A(t-kT) & B(t-kT) \\ K & I \end{bmatrix} \begin{bmatrix} \hat{E}_t[k] \\ \delta\hat{U}_t[k] \end{bmatrix} = \begin{bmatrix} e(t) \\ 0 \end{bmatrix} \tag{7.10}$$

6. The input is then directly transferred to time t without any change and applied to the system:

$$\delta u(t) = \delta\hat{U}_t[k] \tag{7.11}$$

This simple procedure can be done for any time instance t between the coarse time-samples kT and $(k+1)T$ to update the control input. The projection loop involves solving a linear system of equations whose dimensions depend on the number of control inputs and states in the system. Figure 7.5.F demonstrates three different inter-sample times t_1, t_2 and t_3 before, in the middle and after the disturbance respectively. When following nominal trajectories, the projection controller produces zero adjustments $\delta u(t)$ indeed. During the disturbance, the controller keeps updating $\delta u(t)$, and after the disturbance, it produces the same adjustment every time until the end of the phase. The resulting conceptual trajectories are shown in Figure 7.5.D which naturally outperform the DLQR (Figure 7.5.C) due to a higher control rate.

A detailed demonstration of projection control is provided in Appendix F for a simple system. We discuss continuous and discrete control designs for this simple system and provide a numerical example. DLQR feedback gains in our case always fall within the range of allowable gains which stabilize the closed-loop discrete system. Projection control, however, results in a time-variant continuous feedback which is absent in continuous and discrete LQR paradigms. This property slightly shrinks the allowable range to avoid infinite continuous feedback gains. We precisely discuss this issue in Appendix F to provide further insights. For our walking model and the choice of dimension-less DLQR cost design, this property is always satisfied through simulations. Within the allowable

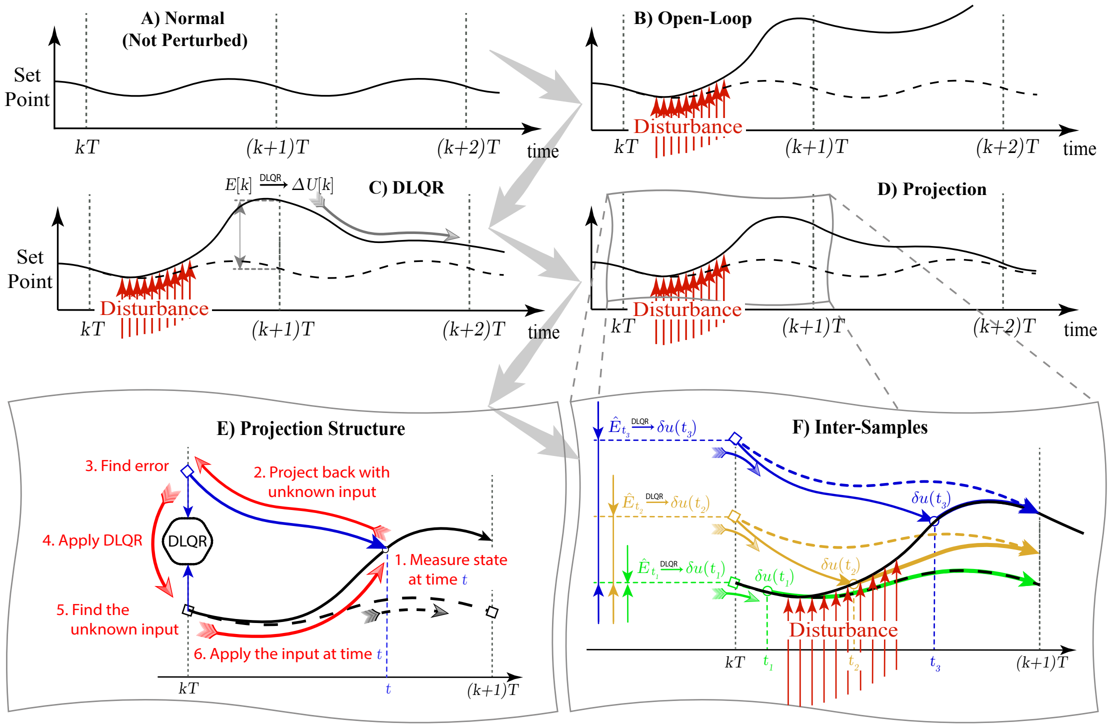

A) Normal (Not Perturbed)
Set Point
kT
(k+1)T
(k+2)T
time
B) Open-Loop
Disturbance
kT
(k+1)T
(k+2)T
time
$E[k] \xrightarrow{\text{DLQR}} \Delta U[k]$
C) DLQR
Set Point
Disturbance
kT
(k+1)T
(k+2)T
time
D) Projection
Disturbance
kT
(k+1)T
(k+2)T
time
E) Projection Structure
3. Find error
2. Project back with unknown input
4. Apply DLQR
DLQR
1. Measure state at time t
5. Find the unknown input
6. Apply the input at time t
kT
t
(k+1)T
F) Inter-Samples
$\hat{E}_{t_3} \xrightarrow{\text{DLQR}} \delta u(t_3)$
$\hat{E}_{t_2} \xrightarrow{\text{DLQR}} \delta u(t_2)$
$\hat{E}_{t_1} \xrightarrow{\text{DLQR}} \delta u(t_1)$
$\delta u(t_3)$
$\delta u(t_2)$
$\delta u(t_1)$
Disturbance
kT
t_1
t_2
t_3
(k+1)T

Figure 7.5 – A comprehensive schematic of time-projection idea. Panel A) shows an open-loop system with nominal trajectories. B) This system might not be stable when subject to disturbances. C) A DLQR controller designed for events kT can stabilize this system but provides corrections only at times kT. Such a delayed reaction might produce large state deviations. D) The projection controller reduces this deviation by reacting to the disturbance immediately. Such reaction is done through an online control scheme where the control rate is faster. E) The projection controller is based on projecting inter-sample errors back in time to the discrete events. The DLQR controller can find appropriate corrective inputs for the system based on the projected errors. These inputs are then applied to the system in the inter-sample time instance. F) The projection controller is very much similar to the DLQR controller. Without disturbance at t_1, it produces no adjustment. During the disturbance, e.g., at t_2, this controller keeps updating the input. When the disturbance is finished, e.g., at t_3 the inputs remain constant until the next event.

range of feedback gains, a stability proof is straight-forward for the simple system as provided in Appendix F. We plan to extend this proof to higher dimensions in a follow-up work with more detailed mathematics discussions.

7.3 Simulations

In this section, we demonstrate walking scenarios which show advantages of using the projection controller over the DLQR controller. These scenarios include simulating intermittent pushes and speed tracking where periodic system trajectories change over time. All simulations are done with a human-sized 3LP model in Matlab which has

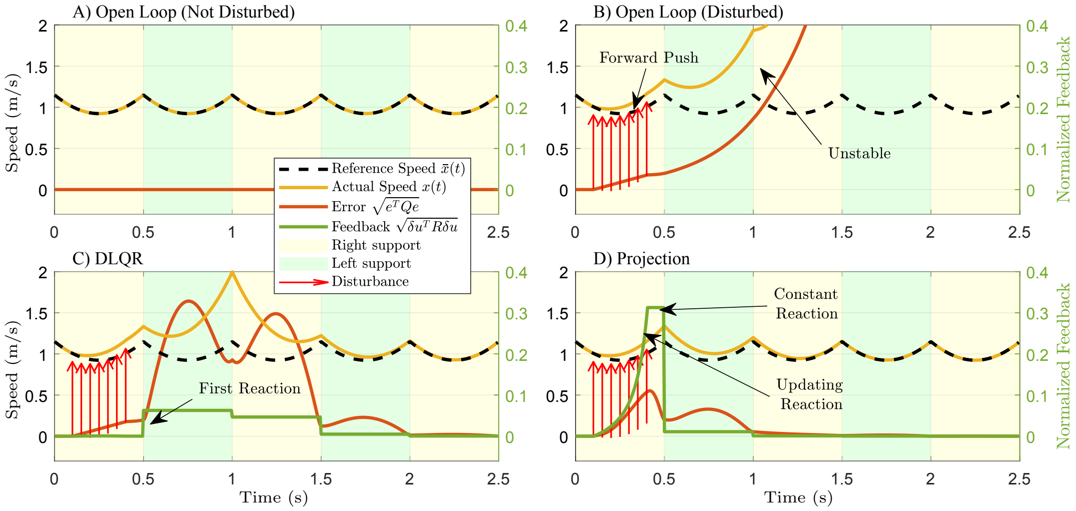


Figure 7.6 – Trajectories of A) open-loop normal, B) open-loop disturbed, C) DLQR and D) projection controllers. Due to falling dynamics and unstable system modes, a moderate disturbance can lead to falling in few steps. The DLQR controller updates inputs only at phase-change times which produces large deviations due to a delayed reaction. The projection controller, however, resolves this issue by a fast online reaction. Note that the projection controller is only using the knowledge of the DLQR controller and uses projection to translate continuous online errors to discrete errors. The expertise of DLQR controller on discrete errors is then used to produce online corrections.

a mass of $m = 70$kg and a height of $h = 1.7$m. Our choice of DLQR cost coefficients are dimensionless diagonal cost matrices $Q = I_{8\times 8}$ and $R = (mg)^{-2} I_{4\times 4}$ for states and inputs, where g is gravity. All results and qualitative conclusions of this section are in fact valid for different 3LP models matching humanoid robots like Atlas with $m = 150$kg and $h = 1.88$m (Feng et al., 2015), COMAN with $m = 30$kg and $h = 1$m (Moro et al., 2011) and Walk-Man with $m = 120$kg and $h = 1.85$m (Tsagarakis et al., 2017). We present simulation results of a human-sized 3LP only for simplicity.

7.3.1 Intermittent pushes

Consider a gait with a frequency of $f = 2$ step/s and a speed of $v = 1$m/s. We only use the sagittal pelvis velocity as a periodic signal for demonstrations. The open-loop velocities are shown in Figure 7.6.A where the reference nominal trajectory $\bar{x}(t)$ matches the open-loop system behavior. Now, applying an inter-sample forward push in the first phase of the motion results in an unstable deviation from nominal trajectories which eventually lead to falling as shown in Figure 7.6.B. This is because of unstable open-loop eigenvalues due to the natural falling dynamics. The DLQR controller can stabilize the system by adjusting the footstep at time $t = 1$, shown in Figure 7.6.C. This adjustment is a result of adjusting swing hip torques at the beginning of the phase (at $t = 0.5$), where the accumulated deviation was first detected. As a result, corrective inputs ΔU are nonzero after the time $t = 0.5$ until the system stabilizes completely.

The projection controller has a different behavior, however. Without deviations, this controller produces no corrective input. Once a deviation exists, if there is no active disturbance applied to the system, the output of projection controller remains the same. However, once an external disturbance is present, the projection controller keeps updating the inputs which can be seen in Figure 7.6.D. Notice the first phase where the external push starts acting on the system at $t \approx 0.1$. At this time, corrective inputs start increasing until the disturbance disappears at $t \approx 0.4$. After, the correction produced by the projection controller remains the same until $t = 0.5$. In the next phase, the new stance foot location is already adjusted, and thus less effort is needed by the controller. In other words, online adjustment of the swing hip torque in the first phase places the swing foot in a proper location so that the second phase starts in good condition. Such footstep adjustment produces a larger error norm in the first phase compared to the DLQR controller, however, reduces this norm in the next phases considerably. As a result, overall deviations from nominal trajectories are less with the projection controller, thanks to an online updating scheme.

A better understanding of time-projection behavior for the walking application can be obtained through a direct plot of actual footstep locations. For this purpose, we reduced the speed to $v = 0.5\text{m/s}$ to plot more footsteps in a single figure and simulated pushes of same magnitude and duration but applied at different times during the phase. The resulting footstep plans are demonstrated in Figure 7.7 for both DLQR and projection controllers. As expected, late pushes have less impact while early pushes can cause large corrective steps with the DLQR controller, shown in Figure 7.7.A. This can be confirmed by the exponential effect of sensory or model errors in the forward integration of our equations (Bhounsule et al., 2015). With the DLQR controller, the reaction is only taken in phase-transition events which can introduce delays if the push is applied much earlier, at the beginning of the phase. Although the controller can still stabilize the system, large input torques or step lengths are not desired in practice. The projection controller adjusts the first footstep location online during the phase when the push is being applied. The swing leg has, in fact, a negligible influence on the system and thus, this adjustment does not provide immediate correction. However, since the swing foot becomes the next stance foot which has a strong coupling to system variables, an early adjustment of its position can considerably stabilize the system in the next phase.

7.3.2 Speed modulation

Remember that 3LP equations provided a class of gaits with different speeds, but a fixed frequency. In this test scenario, we consider changing the reference gait occasionally and observe the tracking performance of the projection controller. We apply these changes only at phase-transition moments, i.e., at every ten steps to make sure the controller has enough time to converge. In Figure 7.8, resulting trajectories are demonstrated. In fact, since there is no inter-sample disturbance, the projection controller performs similarly to the DLQR controller. Once a new desired speed is commanded at the beginning of a phase, the reference gait changes and thus, the controller adjusts the swing hip torques.

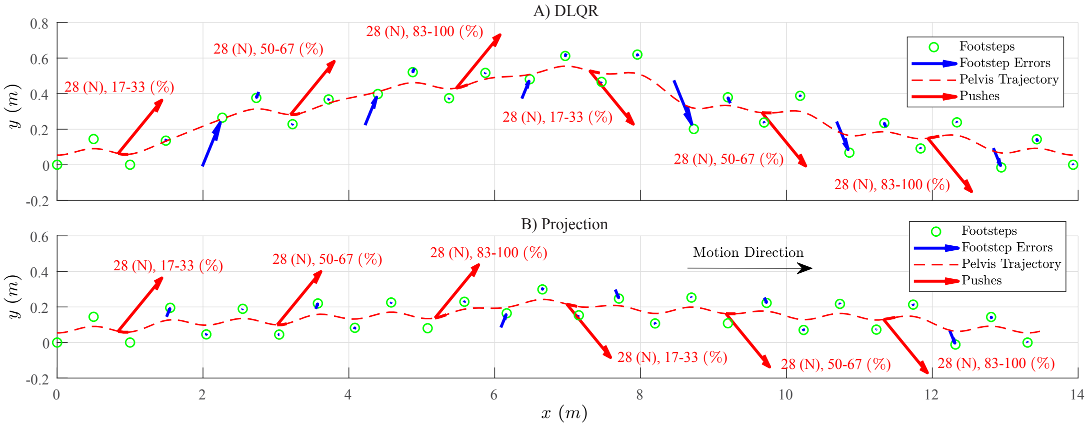


Figure 7.7 – Performance of A) the DLQR and B) the projection controllers in rejecting intermittent external pushes. In this scenario, we apply a push with the same magnitude and duration, but at different times during the phase. We also apply the same pushes in the opposite lateral direction when the other leg is in swing phase, and the CoM has a different lateral motion direction. Along with the force magnitudes, we show by percentage, the period in which the push is applied. Like Figure 7.6, the reaction of DLQR controller is only taking place after the push with a delay. It is also notable that a pushing force at the end of the phase has a relatively small impact. However, the same push at the beginning and in the middle of the phase can result in larger corrective steps. The lateral direction of the push does not influence the size of corrective steps, since the 3LP equations are linear and symmetric in both directions. Likewise, a backward push results in the same footstep adjustment magnitude. The corresponding movies could be found in the accompanying video.

This adjustment leads to modifying the footstep location at most in the following next two steps. Therefore, our controller tracks new velocities in only two steps. Likewise, in Figure 7.7, disturbances are captured at most in two steps following the analysis provided in (Zaytsev et al., 2015).

7.4 Analysis

As motivated earlier in Figure 7.3, the goal of projection controller is to achieve online control performance based on a DLQR controller. The projection controller is comparable to the MPC controller of chapter 2 in the sense that it can react to inter-sample disturbances immediately whereas the DLQR controller has to wait until the end of the phase. The advantage of projection controller to MPC is relieving the need for online numerical optimizations while considering an infinite horizon. In practice, however, MPC can handle inequality constraints which are absent in the projection controller. In this regard, we are interested to know how well our projection controller can cover the set of viable states. In this section, we provide this analysis together with a quantification of open-loop and closed-loop eigenvalues as well as the sensitivity to the timing of external disturbances.

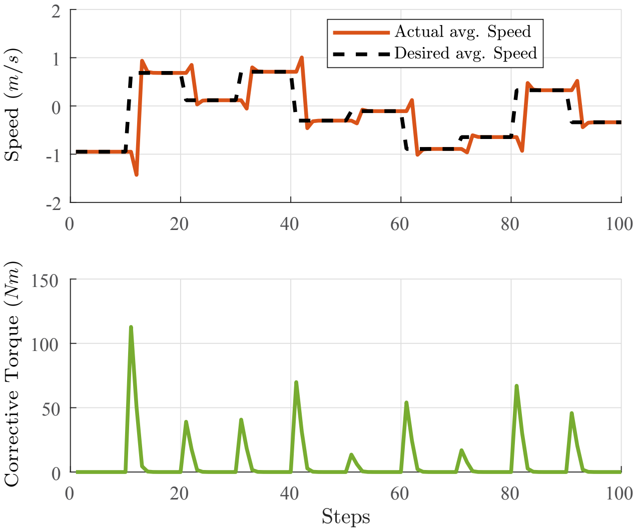


Figure 7.8 – Performance of projection controller in transitioning between different speeds. The original gait is simply scaled to modulate the speed within the null-space of 3LP periodic gait equations. Since the instantaneous pelvis velocity has variations, we reported the average speed at each discrete event by dividing feet distance with stride time. On the top graph, speed tracking performance is shown for the desired profile of speeds which changes randomly at every ten steps. The bottom graph is showing the root-mean-square of controller outputs with respect to the desired gait at each step. One can see that the controller performs the transition in almost two steps. Since we change desired speeds only at phase-change moments, the DLQR controller would perform exactly the same as the time-projecting controller. An example movie of this scenario could be found in the accompanying video.

7.4.1 Eigen values

Since the 3LP model is similar in sagittal and frontal directions, we only consider the sagittal direction. Also, since the foot velocity is forced to zero in phase-transition moments, this dimension is further reduced to three. Therefore, we only consider evolution of errors in three dimensions (i.e. sagittal directions of $s_1 = x_{Swing} - x_{Stance}$, $s_2 = x_{Pelvis} - x_{Stance}$ and $\dot{s}_2 - \dot{s}_1$, shown in Figure 6.2). To find eigenvalues, we added errors to these three quantities in a phase-transition moment and measured them after one step in open and closed-loop systems. Eigen-values of the resulting Poincaré map are shown in Figure 7.9 for the open-loop system, DLQR, and projecting controllers. Due to linearity, such analysis is independent of gait velocity but dependent on the model properties and the gait frequency. Therefore, these plots are repeated for different walking frequencies in a typical range observed in human gaits (Bertram, 2005). It is observed that both the DLQR and projection controllers can stabilize the unstable open-loop system. Since both perform the same in the absence of inter-sample disturbances, their closed-loop eigenvalues are always similar.

The walking system considered in this chapter is interesting regarding eigenvalues.

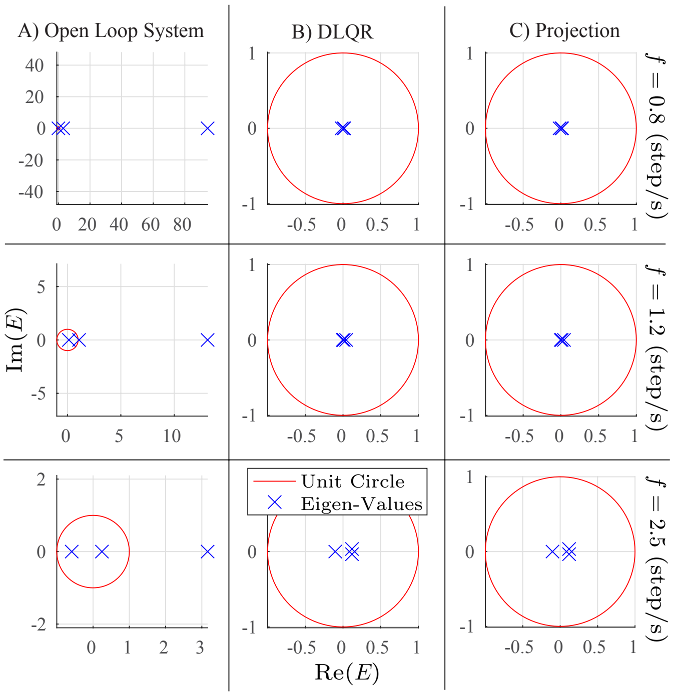


Figure 7.9 – Eigen-values of the walking system in the sagittal direction of s_1, s_2 and $\dot{s}_2 - \dot{s}_1$, calculated over one step. Due to the similarity of lateral and sagittal dynamic equations, lateral eigen-values are duplicates of the plotted sagittal eigen-values here. A) The open-loop system has a large unstable mode depending on the walking frequency. The close-loop systems (with B) DLQR and C) projecting controllers) both stabilize the unstable system through foot-stepping strategy. Note that the DLQR and projecting controllers perform the same in absence of inter-sample disturbances.

Because of falling dynamics, this system has a large unstable mode which highly depends on frequency. As observed in Figure 7.9, the open-loop system at $f = 0.8$ step/s has a very large eigenvalue, indicating that the system deviates more if given a longer phase time. In slow walking speeds, humans use a relatively slow frequency determined by a trade-off between falling and swing costs (Bertram, 2005). Based on our eigenvalue analysis, the third stabilization strategy of adjusting footsteps might not have enough control authority in slow frequencies. Therefore, the first continuous control strategy of modulating the CoP might be more effective in these walking conditions. The second eigenvalue in the open-loop system is small and stable while the third one might be stable or not, depending on the frequency.

7.4.2 Push recovery strength

In addition to stability analysis, we would like to characterize how well the projection controller can recover intermittent pushes. In particular, the sensitivity of footstep adjustments with respect to the timing of external pushes is important. To this end, we simulated the open-loop system, DLQR and projection controllers with pushes of the

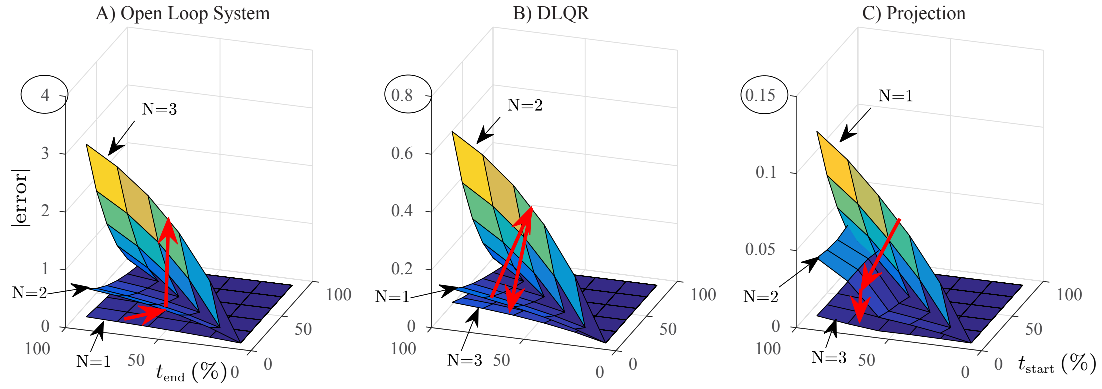


Figure 7.10 – Demonstration of intermittent push recovery performance for the open-loop system, DLQR and projection controllers. In these plots, we show the start and end times of the push application period as a percentage of the phase (like Figure 7.7). Surfaces show error norms, calculated at touch down events over three consecutive steps. The open-loop system is unstable, the DLQR controller overshoots and the projection controller reacts immediately when the push is being applied. The projection controller is therefore much stronger in rejecting even long-lasting intermittent pushes which span throughout the whole phase.

same magnitude, but variable timings and durations like Figure 7.7. The results are shown in Figure 7.10 over three consecutive steps. We applied the push between certain times of the phase, shown by percentage on the two horizontal axes. Error surfaces demonstrate the norm of error with respect to the nominal gait, calculated at touch down events and plotted along the vertical axis.

Figure 7.10 is demonstrating that a push of same magnitude and duration might have a more severe effect on the system if applied earlier in the phase. While the open-loop system is unstable, the other two closed-loop systems recover the push successfully, but with certain dynamics. At the end of the first step, the DLQR controller produces an error similar to the open-loop system. However, due to a delayed reaction, it overshoots in the second step. The projection controller, however, adjusts the footstep online during the first step to avoid the overshoot. Figure 7.10 also indicates that longer pushes have more severe effects, especially if they start earlier in the phase. If a push continues until the end of the phase, the projection controller performs slightly worse compared to itself, yet much better than the DLQR controller.

7.4.3 Viable regions

Computations in the projection controller involve solving a linear system of equations (7.10) at every fine control time-sample. However, other online controllers like MPC might require more computations due to multiple iterations required to optimize a particular cost function with constraints. The advantage of using MPC over projection is mainly the inclusion of inequality constraints. Without such constraints, the viable region for our linear setup is in fact unlimited. Although the proposed framework ignores these constraints, we would like to characterize the valid set of states in which our controller produces feasible actuator inputs and leads to feasible states, within limitations of the actual hardware.

In this work, we considered constraints on hip torque limits and footstep distances in an adult-size 3LP model. Torque limits are physical limitations, and footstep distances are artificially introduced to prevent violation of constant height assumptions on the real hardware. The torque limits are represented by simple boundaries (± 80Nm) while next footstep locations should lie on a diamond region centered at the stance foot location (with equal diameters of 1.7m, compared to a leg length of 0.9m in the model). The latter constraint could be more complex like using a circle around the stance foot, but here we used diamonds to preserve linearity of constraints and our analysis. Considering realistic regions indeed require non-convex constraints, especially when foot cross-over and self-collisions are considered.

Due to the coupling of frontal and sagittal motions when adding diamond regions, we cannot split the reduced 6-dimensional discrete error system into 3-dimensional subsystems anymore. Therefore, it is not possible to completely visualize viable regions. Additionally, these regions are now depending on the forward velocity as well. In other words, over faster speeds and thus longer stride lengths, possible foot-adjustments are more limited compared to slower speeds. Therefore, we need to consider the effect of stepping frequency, speeds and dimensionality altogether. With the inequality constraints mentioned earlier, the goal would be to find the maximum set of viable states for each controller (Zaytsev et al., 2015) as well as the maximum set for all possible controllers.

Convex polyhedrons of viable regions are calculated for six consecutive steps which seem enough to give an approximate of their complex shape. We also divided each phase into five shorter sub-phases where time-projection and arbitrary input profiles (for maximum viable set) can provide continuous input modifications. The maximum viable set includes all states which are capturable in 6 steps, i.e., with feasible arbitrary inputs and states over these six steps. We observed that increasing steps and resolution exponentially increases the dimensionality and polyhedron computation times while having almost no effect on precision. Since the resulting polyhedrons typically had thousands of vertices, we used a ray-casting method and a linear programmer to search along ray directions and find intersection points with the boundaries. For 2D visualizations, we used a resolution of 100 rays per 360 deg.

Geometric constraints are active at faster speeds while torque limits are active at faster frequencies. It is impossible to view the full polyhedron from all perspectives, but spatial projections on important subspaces are shown in Figure 7.11. We take error vectors e_1, e_2 and e_3 added to the symmetry vectors s_1 and s_2 and $\dot{s}_2 - \dot{s}_1$ as error measures respectively and plot viable sets with respect to these error dimensions. The diamonds and torque boundary shapes are implicitly observable in some spatial projections while others provide a less intuitive shape. Compared to the DLQR controller, the projection controller has a larger viable set due to an online update scheme. The maximum viable set is slightly larger in some regions though, indicating that there are

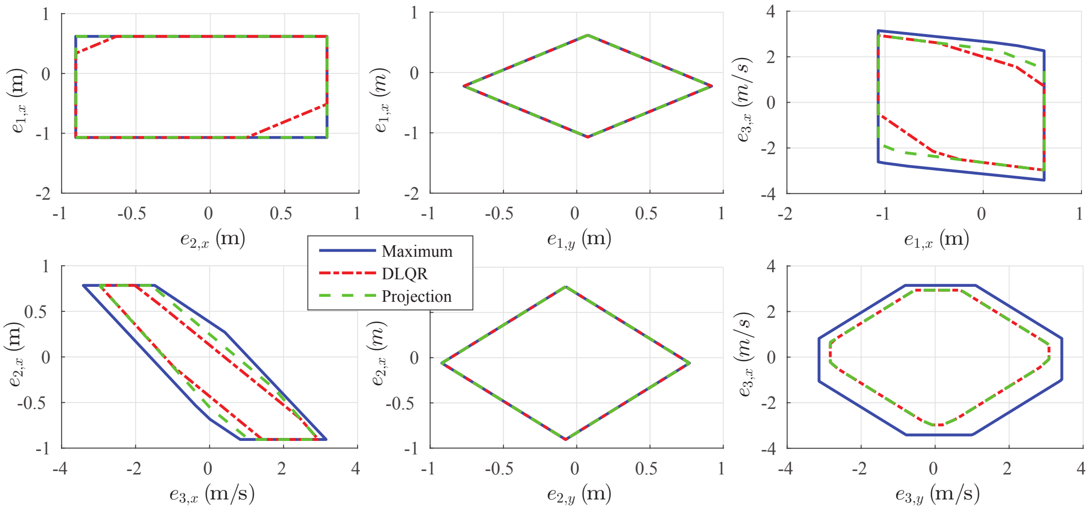


Figure 7.11 – Different spatial projections of 6-dimensional viable regions calculated for the DLQR and projection controllers together with the maximum viable region. The reference gait has a velocity of $v = 0.5$m/s and a frequency of $f = 3$ step/s. Due to an online update scheme, the projection controller always has bigger viable regions compared to the DLQR controller. In some dimensions, however, the maximum viable region is slightly larger.

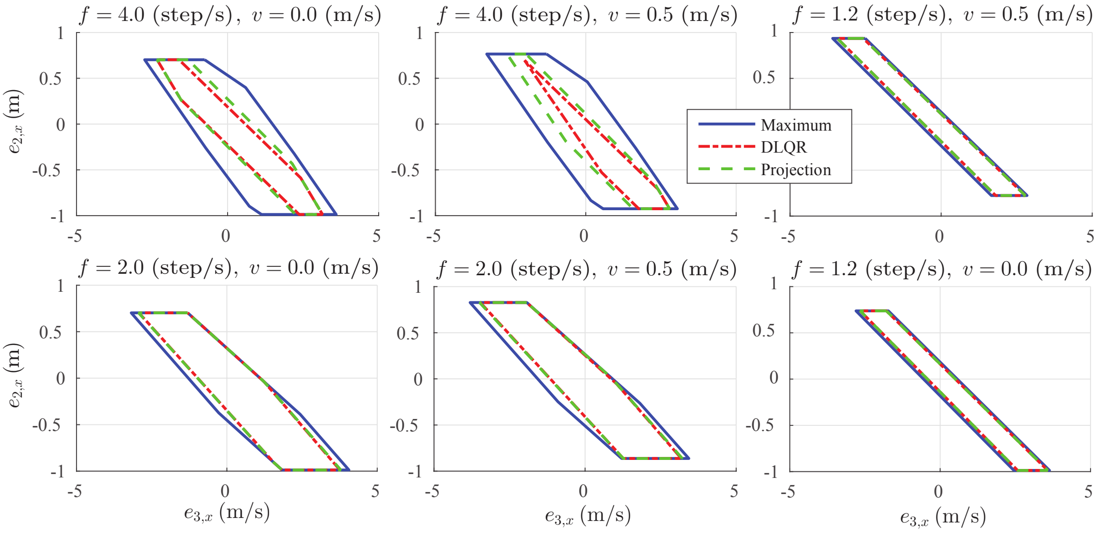


Figure 7.12 – The effect of walking speed and frequency on viable regions. This figure only demonstrates spatial projections of 6-dimensional polyhedrons on one subspace of position-velocity errors. As demonstrated over few choices of velocities and frequencies, the gait velocity mainly shifts the region while faster frequencies can shrink it considerably.

other types of controllers that can stabilize a slightly bigger set of erroneous states, compared to the projection controller.

Further, we are also interested to see the effect of walking speed and frequency on these regions. Again, it is hard to inspect the full-dimensional polyhedron, though we take a position/velocity spatial projection which is more insightful. Figure 7.12 demonstrates viable regions for different choices of walking frequencies and velocities. The velocity mainly shifts the viable region while the frequency can shrink it due to torque limitations. In Figure 7.12, the maximum choice of frequency is unrealistic regarding the human data (Bertram, 2005) just for demonstration purposes. In normal walking conditions, however, the DLQR and time-projection controllers can cover most of the maximum viable region.

7.5 Discussion

In this chapter, motivated by the hybrid nature of walking systems, we formulated a new control framework that can be used for online control. This controller is based on a DLQR controller which works at a rate synchronized to the system's phase-change rate. Such synchrony and a simple derivation of error-dynamic equations which is blind to hybrid phases allow us to use the standard DLQR framework with a fixed model of error evolution. However, in dealing with inter-sample disturbances, this controller cannot be simply used at a higher rate, because of the hybrid phase-change nature. In Appendix C, we show that for normal systems without hybrid phases, increasing the DLQR rate up to a continuous control is always possible. In our hybrid system, however, we introduced the time-projection idea which maps inter-sample errors to discrete errors where they could be easily treated with the DLQR controller. Although time-projection can be used for simple systems as well (refer to Appendix C), increasing DLQR rate and continuous control seem easier for these type of systems.

7.5.1 Physical explanation

Through extensive simulation and analysis, we showed that time-projection could adjust the footstep location online which naturally outperforms the DLQR controller that has a delayed reaction. This has a benefit in hybrid systems like walking robots since the swing state has little influence on the stability of the system, but as soon as it becomes the next stance foot location, it tightly couples with other system variables. Therefore, investing control effort in adjustment of this state in an online fashion does not improve stability instantaneously (unlike modulating the CoP), but saves a lot of control effort in the next phase of motion. Time-projection helps to translate (projecting) phase-dependent inter-sample errors to phase-free discrete errors where a standard DLQR knows best how to stabilize them.

7.5.2 3LP advantage

We used 3LP as a walking model to facilitate gait generation and produce more human-like gaits. This was an improvement to the LIP model we used in chapter 2. However, the main novelty of this work lies in introduction of the projection controller which replaces the MPC controller of chapter 2. Footstep adjustment is a known strategy in robotics hopping (Raibert et al., 1984) and walking (Koolen et al., 2012). Our projection controller generalizes this idea into a DLQR control paradigm where controllers can be designed more systematically. Besides, using 3LP as a physical model helps to find physically more meaningful control laws for footstep adjustment. In 3LP, in fact, footstep adjustment is made through swing hip torques whereas, in other models without a swing leg, this is done directly through adjustment of the final attack angle which translates to footstep location. We believe that the combination of 3LP with time-projection provides a unified control framework in which, in addition to the CoM states, swing location and swing dynamics are also considered. Constraints on modulating the CoP and ankle torques are often used in the literature during trajectory planning (Herdt et al., 2010a). However, hip torques are also important to be considered, especially for swing and torso dynamics. 3LP is the first linear template model that can provide information on hip-torques at an abstract level. In (Byl and Tedrake, 2008) for example, since swing dynamics are absent, the hip controller stiffly tracks the desired attack angle only after the mid-stance event. If a hip torque is applied earlier, it can sometimes cause the robot to fall backward.

7.5.3 Comparison to MPC

The added-value of using time-projection over MPC would be the reduction of computations while considering infinite horizons instead of MPC's receding horizon. However, this framework cannot handle inequality constraints like MPC. Thanks to the analysis of viable regions, we have now a more clear view of other possible controllers. We demonstrated that for normal walking gaits and reasonable constraints imposed by the hardware, time-projection could stabilize the system most of the time. Other controllers can only cover a slightly larger range of disturbed states. If the robot is severely disturbed, inequality constraints of viable regions can simply determine feasibility of a projection control. With such criteria, the algorithm can easily switch to other complicated controllers or even emergency cases. The analysis of viable regions further shows that other controllers can rarely recover if time-projection fails. So, most probably, the robot needs emergency actions. In this regard, MPC controllers cannot do much better than time-projection either, despite considering inequality constraints in their numerical optimization. However, we acknowledge that exploring non-convex inequalities for footstep regions as well as timing adjustment could be possibly handled by nonlinear and non-convex MPC controllers.

7.5.4 Different from intermittent control

As we talk about intermittent disturbances, to avoid confusion, it is worth mentioning that the class of intermittent controllers (Gawthrop and Wang, 2007) are much different from our projection controller. In this work, we propose an architecture based on per time-step measurement and feedback paradigm. Inspired by certain experiments, however, scientists propose a different control architecture for humans (Gawthrop et al., 2011), successfully applied to simple systems too (Bhounsule et al., 2015). Instead of traditional observer-predictor-feedback paradigms, intermittent controllers use a kind of feedback which occasionally modifies certain parameters of the low-level feed-forward or feedback controller. Such architecture can better deal with systems in which high-level information is available with lower frequencies or longer delays, like humans. Compared to recent inverse dynamics methods (refer to chapter 3), intermittent control is computationally less demanding of course, but versatility of this approach is yet questionable. Indeed, intermittent control in mixture with Neuro-Muscular model (Geyer and Herr, 2010) might provide a reasonable explanation of the control system in humans, especially at slow walking speeds. It is, therefore, a very interesting candidate for the control of 3LP in our future works.

7.5.5 Measurement requirements

In this chapter, we assumed the robot state is observable. In other words, the relative pelvis and feet positions and their time derivatives are perfectly measurable. For rigid robots without series elastic elements and precise encoders, this assumption can be realistic. On most of the humanoids, however, a complex Kalman filtering is needed to calculate these states (refer to chapter 3). In this work, the instantaneous error is calculated based on measured positions and velocities while there is no need for a disturbance (external force) observer. However, such information about disturbance can further improve the performance. Knowing that a disturbance exists, the controller can take preventive actions to reduce deviations from nominal trajectories ahead of time.

7.5.6 Performance

Various analyses show that time-projection has similar stabilization and controllability properties with the DLQR controller. However, it performs much better in recovery of intermittent pushes. Overall, advantages of our proposed architecture combined with the 3LP model are:

+ Resistant against intermittent perturbations.
+ No need for offline optimization.
+ Computationally light compared to MPC.
+ Online policy refinement.
+ Slightly expanded viable regions compared to the DLQR.
+ Covering most of the maximum viable region.
+ Speed-independent stability analysis.
+ Optimal future behavior, thanks to the seeding DLQR.
+ Generic design for different model sizes, walking speeds and frequencies.

And disadvantages would be:

- Fixed timing.
- Lack of inequality-constraint support.

7.5.7 Other applications

The new projection controller can simply replace our previous MPC controller of chapter 2, bringing many advantages. In future works, we want to use the proposed controller in combination with inverse dynamics and compare it with our previous MPC controllers.

We would also like to exploit fast computational properties of the 3LP model to set up nonlinear MPC controllers that can adjust the timing. The inclusion of non-convex constraints for avoiding self-collision is indeed another interesting future extension. Finally, in this chapter, we only explored external pushes and not other types of disturbances like uneven terrain. The formulation of 3LP is yet limited to impact-less locomotion which could be extended to handle impacts and height variations. However, the concept of time-projection is yet applicable to other complex models, computationally challenging though if closed-form solutions are not available. Applying time-projection to quadruped locomotion would also be interesting, given that these systems might have more than two distinct phases of motion. This chapter is accompanied with a video, demonstrating walking motions of speed tracking and intermittent push recovery scenarios.

8 Walking and Push Recovery

In this chapter, we use the 3LP model and time-projection control of chapters 6 and 7, and apply them on the real hardware in the simplest form. The discussions of this chapter mainly focus on hardware issues and limitations, especially delays, backlashes, velocity saturations and imperfect series elastic elements. Our hierarchical controller in fact proposes a systematic way of handling these issues without being sensitive to any tuning parameter. Forward and inverse kinematic models translate full-body joint positions to equivalent 3LP states and vice-versa. We use the 3LP model to generate walking gaits at different speeds. We also apply our time-projection control at the same level of abstraction, i.e. comparing desired and actual 3LP states and suggesting footstep adjustments. The experiments of this chapter cover a wide range of applications including walking at different speeds, extreme push recovery scenarios and emergent walking conditions in which an external force pulls the robot. Although 3LP and time-projection equations might look complex, they encode walking mechanics in a simple way which can be applied on the real hardware without complex calculations. It should be noted that we rely on the physical springs on the robot to absorb impacts in the position control mode of this chapter. A better solution is to use the inverse dynamics of chapter 3. However, we could not achieve a convincing tracking performance without using joint positions on the real hardware. Position control provides a better tracking but leads to certain issues, especially systematic spring deflections that require compensation. This chapter provides extensive discussions on all these issues which may happen in many other robots as well[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji, Hamed Razavi, and Auke Jan Ijspeert. *"Push recovery with stepping strategy based on time-projection control."* arXiv preprint arXiv:1801.02151, 2018.

The first author developed the method, performed experiments and drafted the manuscript. The second and third authors gave helpful comments and corrections.

---

[1] All the videos of this chapter could be found at https://youtu.be/AULSwcesQqo

8.1 Background

Humanoid robots are designed to perform different locomotion and manipulation tasks like humans by exploiting similar kinematics and mass distribution properties. In this work, we aim at walking gait generation with the emphasis on push recovery properties and stepping strategy. We discuss critical ingredients needed in perturbed walking conditions together with requirements that the particular hardware platform imposes on the control design. Based on these concepts then, we introduce a very simple controller that benefits from well-established theories in previous works but implements look-up-table control laws to stabilize the robot and recover external disturbances. The proposed controller is very simple to implement, computationally very fast and yet generic with no critical parameter to tune. We will continue this section by introducing key concepts required for perturbed walking. We explain our walking model and control theories developed in previous works in the following two sections briefly. All low-level control strategies that tackle hardware limitations, including velocity limits, delays, noises, backlashes, spring deflections and ground clearance strategies, are explained in a separate section afterward. We conclude the chapter then by a wide range of experiments that demonstrate the effectiveness of our method to perform dynamic walking and push recovery.

8.1.1 Compliance in walking

Virtual or physical leg compliance is a determinant feature of humanoid walking on uneven terrain or perturbed conditions. In the absence of severe impacts, e.g., on perfectly known terrains or in static locomotion conditions (Rebula et al., 2007), compliance does not play an important role. In periodic flat-ground walking also, the control method can be adapted to minimize the impacts and smoothen the locomotion. Non-periodic walking conditions, however, require more compliance due to an unexpected timing of contact phase changes. Many walking controllers rely on swing leg compliance and contact force sensors to provide a phase-based or adaptive contact switch (Pratt et al., 2012; Gehring et al., 2013; Kelly et al., 2016; Lim et al., 2001). Phase detection through contact force measurement helps to choose the right swing/stance controller for each leg which improves stability (Aoi and Tsuchiya, 2006; Faber and Behnke, 2007; Collins and Ruina, 2005), despite a fixed timing in the desired periodic gait. A simple strategy could be early termination or polynomial extrapolation of trajectories in case of early or late phase transition respectively (Faraji et al., 2013).

Compliance and damping can prevent non-smooth changes in velocities over impacts (Park and Chung, 1999) and make state estimation easier. Besides, adding Series Elastic Elements (SEA) in the joints help to reduce impacts and to protect the actuators. However, one requires an acceptable level of precision in state estimation and actuation. This translates to a proper spring deflection (Hutter et al., 2017) or force gauge measurement (refer to chapter 3) for a closed-loop torque control. With SEA actuators, a stiff position control is also possible, but the physical springs typically determine the effective stiffness due to softer properties. Actuators with SEA elements can improve the efficiency of walking too. Depending on the spring size and actuator's isometric force generation efficiency, robots or prosthetic devices can absorb energy during impacts and rerelease it when redirecting the Center of Mass (CoM) velocity (Sreenath et al., 2011; Collins et al., 2005; Grimmer et al., 2016). In some robots like COMAN (Moro et al., 2011) or MIT's Spring Flamingo (Pratt and Pratt, 1998), however, the SEA elements are not big enough to store a considerable amount of energy. Overall, the walking controller should not fight against SEA's functionality. In other words, the controller should demonstrate at least a passive behavior to let the springs absorb impacts or store energy smoothly (Sreenath et al., 2011). In this work, since our robot COMAN has small springs, we are mainly interested in transient push recovery conditions and stability rather than energetics and efficiency.

8.1.2 Swing dynamics

Due to a small size (height of ≈90cm and mass of ≈30kg), COMAN requires a relatively high frequency for stepping. The stepping time in a child of about the same stature (age range of 3-4 years old, height of 105±2cm and mass of 17.3 ± 0.7kg) is about 0.45s while that in a child with about the same mass (age range of 6-7 years old, height of 125±1cm and mass of 25.3 ± 0.9kg) is about 0.47s at self-selected speed (Hausdorff et al., 1999). On the one hand, COMAN is more massive than an average child of the same height. On the other hand, the leg mass in COMAN is about 22.5% of the body mass while this ratio is about 17.3% in human (de Leva, 1996). Stepping times longer than 0.5s in COMAN lead to large lateral bounces and static motions (e.g., 1s in (Kryczka et al., 2015)). Besides, unlike MABEL (Sreenath et al., 2011) and many other biped robots, ankle actuators are distal to the knee in COMAN which further increases the leg inertia around the hip.

With a small size, yet relatively heavy legs and high inertias, the swing dynamics in COMAN has a considerable influence on CoM dynamics. Walking at a high frequency could be challenging due to actuator tracking delays, extra torques needed in the hip joints, fast ground clearance and knee motions, filter delays in state estimation and impacts in case of perturbations. However, at the same walking speed, increasing the frequency reduces the step length which makes application of linear constant CoM height models easier. Besides, SEA elements help to reduce intrinsic impacts of such a high cadence, which makes dynamic walking possible with COMAN too. This goal requires an advanced gait generation method which considers swing dynamics and internal coupling between lower limbs and the upper body.

8.1.3 Gait generation

In chapter 2, we used the Linear Inverted Pendulum (LIP) model (Kajita et al., 2001a) with closed-form equations in a Model Predictive Control (MPC) framework to adjust footsteps in an online fashion. We also introduced an artificial swing trajectory which terminated at the optimized footstep location, hoping that the underlying inverse dynamics can track the template motion. The available control authority for inverse dynamics was modulating the Center of Pressure (CoP) to track the CoM and swing trajectories.

Such hierarchical controller worked at a walking frequency of 2 step/s and speeds up to 0.35m/s in a perfect simulation environment.

Tracking issues in the inverse dynamics layer (due to a limited foot size) motivated us to extend LIP with two other pendulums to model the swing leg and the torso. The new linear model, called 3LP (introduced in chapter 6), assumes ideal actuators in the legs to keep limb masses and the pelvis in constant height planes. It also assumes a perfect stance hip actuator to keep the torso upright. 3LP allows different stance ankle and swing hip torque profiles as inputs which can create various walking gait patterns. Thanks to inclusion of swing and torso dynamics, 3LP can produce more human-like CoM trajectories compared to LIP which facilitates tracking for the underlying inverse dynamics controller.

In this work, we focus on the sagittal plane push recovery performance while leaving the robot bounce left and right with a fixed frequency of 2.5 step/s (step time of 0.4s) to reduce lateral bounces. We assume decoupled lateral and sagittal dynamics (Kajita et al., 2001b, 2003) which requires smaller step lengths (or large feet to provide transverse torques) in practice. Equations of 3LP are also decoupled in lateral and sagittal directions which further supports this assumption. Due to inclusion of pelvis, 3LP is restricted to straight walking unlike the 3D LIP model we used in chapter 2 for steering. However, the pelvis produces a natural lateral bouncing in 3LP that does not need to enforce artificial feet separation like chapter 2.

For open-loop gait generation in the present work, we use Cartesian trajectories of 3LP and convert them to joint trajectories. Such unstable walking gait in the sagittal plane can be stabilized with the time-projecting controller of chapter 7. We only focus on stabilization of sagittal plane dynamics in this work and rely on intrinsic limited stability of lateral plane bounces through a combination of waist roll (Omer et al., 2005) and leg lift (Collins et al., 2001) clearance strategies without any ZMP control (Zhao et al., 2008), momentum control (Kuo, 1999), lateral footstep adjustment (Kryczka et al., 2015) and variable timing (Pratt et al., 2012). This combination is particularly chosen to reduce knee joint motions (motivated by limited maximum joint velocities of our particular hardware) during ground clearance and produce more stretched-knee postures (Omer et al., 2005; Ogura et al., 2003).

8.1.4 Control approach

Due to the fast nature of falling dynamics, the walking cycle has an unstable mode whose strength depends on the robot size and walking frequency (refer to chapter 7). A popular way of stabilizing the gait is to consider specific events (e.g., touch-down or mid-stance) and obtain Poincaré maps (Teschl, 2012) as discrete models. Classical control techniques (e.g. Linear Quadratic Regulators (LQR) (Ogata, 1995)) can then stabilize the system by proper control input adjustment (Zaytsev et al., 2015; Kelly and Ruina, 2015; Byl and Tedrake, 2008; Rummel et al., 2010). The idea of phase space constant time to velocity reversal planner (Kim et al., 2016) also falls in the same category where a new footstep location is calculated only once per step. Although the computations involved are CPU intensive and the update rate is relatively slow, Kim achieved 18 successful steps on a foot-less robot called Hume (Kim et al., 2016). This is considerable given transversal slippages of the point-feet, but an external motion capture system helped the state estimation in this work. Other key elements that helped Hume achieve this performance were a relatively high CoM, a high stepping frequency, and relatively light-weight legs. In chapter 7, by considering the effect of strong inter-phase disturbances, we motivated the need for an on-line adjustment of the next footstep location. Due to a relatively weak coupling of swing and CoM dynamics in slow walking conditions, adjusting the next footstep location becomes effective only in the next phase where this position becomes the new stance position which is tightly coupled to other system variables (Raibert, 1986; Koolen et al., 2012). Modulating the CoP through ankle torques can provide immediate stabilization, but is only useful in small disturbance conditions (Asano et al., 2004; Moro et al., 2011).

On-line adjustment of footstep locations is widely studied in the literature to recover strong pushes (Faraji et al., 2014a; Feng et al., 2013; Herdt et al., 2010b; Koolen et al., 2012) during walking. However, the underlying simple template models (like LIP) require inverse dynamics or ZMP controllers to use ankles for a better tracking. Therefore, a combination of stepping and CoP modulation strategies is usually used (Faraji et al., 2014a; Kryczka et al., 2015; Feng et al., 2013) for humanoid walking. Intuitive attack angle adjustment rules which capture the extra CoM velocity in hopping (Raibert, 1986) are extended to walking conditions too through capturability framework (Koolen et al., 2012). That framework aims to find a footstep location that captures the motion and brings the CoM to rest conditions. One significant strength of the capturability framework is considering certain limitations, particularly the step length which is typically due to the use of crouched knees and the LIP model. Given such constraints, capture regions can be calculated for multiple steps as well when disturbances are too large. For walking generation, however, a proportional gain is practically needed to move the desired capture point away systematically to let the robot progress forward (Pratt et al., 2012). Using the 3LP model for walking generation potentially solves this issue. Besides, we use the time-projection controller of chapter 7 which works for arbitrary walking speeds. Capturability offers an efficient start/stop state machine (Pratt et al., 2012) which is not implemented in our work yet and considered for future improvements.

Previously in chapter 2, we extended the capturability idea and formulated a MPC controller that implemented a similar concept and allowed for automatic gait generation (based on LIP) inside the MPC. At each instance of time, this MPC controller simulated evolution of the measured state over a receding horizon and optimized footstep locations based on a given desired velocity. A simple arc trajectory then determined instantaneous foot accelerations which, together with the CoM accelerations (according to LIP) were converted to joint torques through inverse dynamics. The IMU placed on the pelvis also helped to regulate the torso orientation which leads to a natural falling with the given CoM accelerations. This framework was later extended by Feng (Feng et al., 2016) and applied to Atlas for push recovery and walking generation which featured robustness.

The control approach we use in the present chapter is very similar to the mentioned MPC controllers (Faraji et al., 2014a; Feng et al., 2016) regarding performance, but much simpler in structure.

High-level stabilization:

On top of the 3LP gaits, one can derive transition equations like Poincaré maps and use an infinite horizon discrete LQR (DLQR) controller for footstep adjustment. This architecture offers the same optimality as our MPC, but due to a discrete nature, it is sensitive to intermittent short disturbances. Therefore, we introduced the simple time-projection scheme which maps measured errors at any time to the previous discrete event and uses the expertise of DLQR to find on-line footstep adjustments. In the present work, we use gait generation of the 3LP model, optimality of LQR and online properties of time-projection to achieve the same performance as the MPC (Faraji et al., 2014a; Feng et al., 2016). The new framework offers more natural gaits and faster computations because of the 3LP model and time-projection control respectively. It takes both the CoM and swing errors (positions and velocities) into account to ensures stability for a high-frequency gait and considerable swing dynamics. Compared to Raibert's law (Raibert, 1986) and the capturability framework (Koolen et al., 2012), it effectively adjusts the attack angle by variable gains (on CoM and swing errors) which are found systematically through the DLQR design and time-projection. These gains are also consistent with 3LP's falling and swing dynamics.

Low-level joint control:

Motivated by the specific anthropomorphic features of our COMAN robot, we use the 3LP model instead of LIP in this work which provides more natural CoM and swing trajectories and improves tracking. Although we have achieved a convincing balancing performance through pure torque control and inverse dynamics in chapter 3, we found the pure torque control less precise in the absence of position or velocity tracking for faster-walking tasks. In the literature also, a combination of torque, velocity and position controllers is typically used to ensure compliant and precise tracking (Pratt et al., 2012; Feng et al., 2015). In chapter 2, the inverse dynamics layer could compensate for the inconsistency of LIP trajectories with the real system. In this work, however, thanks to a better consistency of 3LP trajectories, we found inverse kinematics and the physical compliance enough for the range of walking speeds and push recovery scenarios considered. Unperceived locomotion on uneven terrain might require a considerable leg compliance; therefore, we limit the experiments of this work to flat-ground gaits. We only rely on the physical compliance of the robot to absorb impacts and restrict our simple framework to position control. This only requires a simple yet adaptive foot lift strategy to avoid foot scuffing.

8.1.5 Novelty

The novel contribution of this work lies in application of the previously-developed time-projecting controller on the real robot in push recovery and stabilization of walking gaits. We also use the previously-developed 3LP model to generate more natural gaits while keeping the same linear properties of LIP for computations. An extensive set of results demonstrate that the effective control authority is footstep adjustment in our experiments rather than CoP modulation. A considerable portion of the present article is also dedicated to the description of hardware limitations in our robot and the compensatory control strategies which help better realize the 3LP trajectories and the time-projection control. All these control ingredients together motivate application of the proposed method on robots with small feet, soft structural compliance, heavy legs, fast walking frequencies and extreme push recovery conditions.

To implement these ideas on the hardware, we have used well-established methods in the literature for lateral bouncing, leg lift strategies, and torso regulation. The present work is limited to the application of time-projection control on the 3LP gaits. Therefore, we decided to keep a fixed timing instead of a phase-based control, which improves the stability considerably. We also relied on robot's SEA properties while virtual compliance can further smoothen the motion and improve stability. In the next section, we introduce COMAN, discuss sensor qualities and extensively analyze issues arising with SEA elements together with our specific control blocks to compensate them. Open-loop and closed-loop control details are also described in the same section. Different scenarios of in-place walking, intermittent push recovery, continuous push recovery, walking gait and ankle stiffness measurements are presented in the results section. Finally, the chapter is concluded by a discussion of results and possible future improvements.

8.2 Hardware control

In this section, we explain implementation details of our controller on the real hardware, describe limitations and motivate individual control blocks used to track 3LP gaits and stabilize them via time-projection. As mentioned earlier, COMAN has relatively short and heavy legs which motivate considering swing dynamics at the gait generation level. However, SEA actuators and hardware compliance improve stability and performance in perturbed walking conditions. We believe performing torque control might bring additional compliance, which is beneficial, however low signal qualities and the hardware's mechanical status to date do not allow a convincing tracking performance. Therefore, we limited our focus to a position control paradigm.

Figure 8.1 shows the big picture of our entire control framework with all individual components. The 3LP Cartesian gaits are augmented by foot lift strategies and converted to joint angles through inverse kinematics. Hip switching and SEA compensation blocks further augment joint-space trajectories and make them suitable for position control. Actual link encoder and IMU signals are then used for a state estimation whose output goes to the time-projecting controller. This controller simulates the instantaneous measured state until the end of the phase and takes the resulting footstep adjustment to

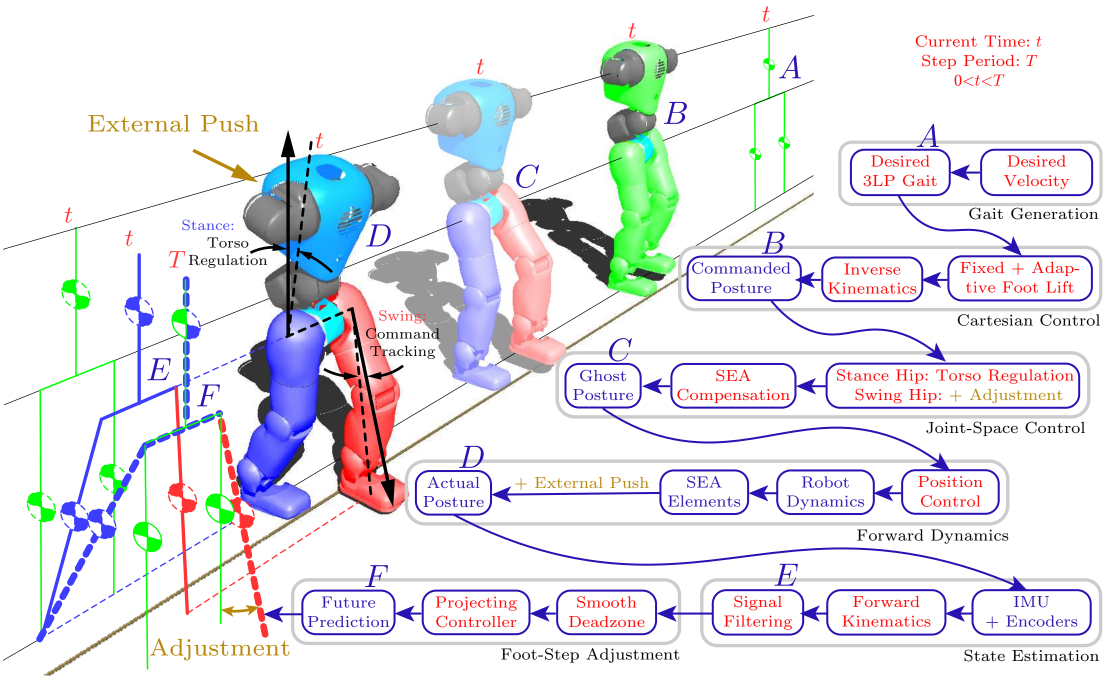


Figure 8.1 – The entire control architecture proposed in this paper. Given a desired walking speed, 3LP produces a nominal gait A which describes the motion of pelvis and both feet in the Cartesian space. We take these trajectories, add fixed and adaptive foot lift components to the swing foot and pass them through an inverse kinematics layer which finds desired joint angles B. The swing hip follows the desired joint angle while the stance hip follows the desired torso angle which is realized by a smooth switch between swing hip encoder and IMU on the pelvis. A time-projection based adjustment is also added to the swing hip trajectory. We add a fixed small angle to the stance knee trajectory to compensate compression of SEA springs in the knee. Joint trajectories are tracked by relatively stiff position controllers on motor shafts before the springs. This creates a posture C which is called the ghost robot, referring to pre-spring motor positions. The actual link positions D are defined by post-spring encoders though, passed through a forward kinematics layer to obtain actual Cartesian feet and pelvis positions. The IMU signal is also used at this stage to orient the robot properly. After a relatively strong filtering of Cartesian positions to remove noise, we apply a dead-zone function too which makes them clean for time-projection control. We take the currently filtered state E, simulate it until end of the phase by a closed-loop time-projection controller to obtain the posture F. We measure the final attack angle adjustment (with respect to the nominal gait) and add it to the swing hip trajectory then. If an external push is applied, the instantaneous error (after filtering) produces a proper attack-angle adjustment through time-projection which captures the push in few steps.

augment the swing hip angle found in earlier control stages. Such adjustment stabilizes the motion, especially in the presence of external disturbances. In the rest of this section, we discuss each control block in details.

8.2.1 Low-level challenges

COMAN (height of $\approx$ 90cm and mass of $\approx$ 30kg) has 23 actuated joints and is equipped with pre-spring and post-spring encoders, strain gauge torque sensors, a high-end IMU and two 6D contact force sensors (Colasanto et al., 2012). It offers a proper level of compliance with the SEA elements which are soft enough to absorb impacts and stiff enough to keep the posture upright. However, the quality of sensory data, actuation limitations, joint backlashes and extra spring deflections make the control of walking at a relatively high-frequency choice of 2.5 step/s more challenging.

Sensor quality:

COMAN has incremental optical encoders to measure pre-spring motor shaft positions and absolute differential magnetic encoders to measure post-spring link positions. The optical encoder signal has a good quality due to a division by harmonic drive's reduction ratio ($N = 101$). The absolute encoder, however, covers the whole span of 360 deg by only 12 bits which produce a considerable quantization error. Dedicated actuator control boards are connected to local LAN dispatchers which are all connected in a tree-shaped topology whose root goes to the external controlling PC. This network architecture together with internal low-level motor controllers lead to a delayed position tracking of at least 20ms in the stiffest and aggressive conditions. Due to the quality of link encoders, delays, backlashes and the fact that we do not want to "fight" against the physical compliance to benefit from impact absorption properties, we perform position control only on the motor shaft (and not the link position) with proper signals from the optical encoder. Figure 8.2 shows sensor qualities and delays in an in-place walking task and important joints involved.

The ghost robot:

As mentioned earlier, SEA compliance in COMAN is enough to let it keep balance to some extent, but depending on the commanded knee angles, spring compressions produce an error in tracking. Imagine we command a crouched standing posture shown in Figure 8.3 with the CoP in the middle of the feet. Due to the ankle pitch compliance, the robot tends to lean forward while due to the knee compliance, the robot leans backward. These two effects slightly cancel each other, but in practice, the knee deflection systematically perturbs the state estimation more. This is because the relative horizontal feet-pelvis position highly depends on the knee angle (with a radius equal to the shank length), but much less on the ankle angle. Thus, the mismatch between actual and ghost postures is small but needs to be considered in control design. Note that we solely rely on the pelvis IMU for state estimation. An alternative would be to assume a flat foot on the ground and to build the kinematic chain upwards. We avoid this approach to minimize the effect of ankle spring deflections, slippages and foot tilt/roll effects. Because of a particular actuator design, a backlash appears in the joints of COMAN over time which requires regular maintenance. At the time of experiments, we had significant backlashes in the knee and ankle joints which were partially removed by external rubber bands (as shown

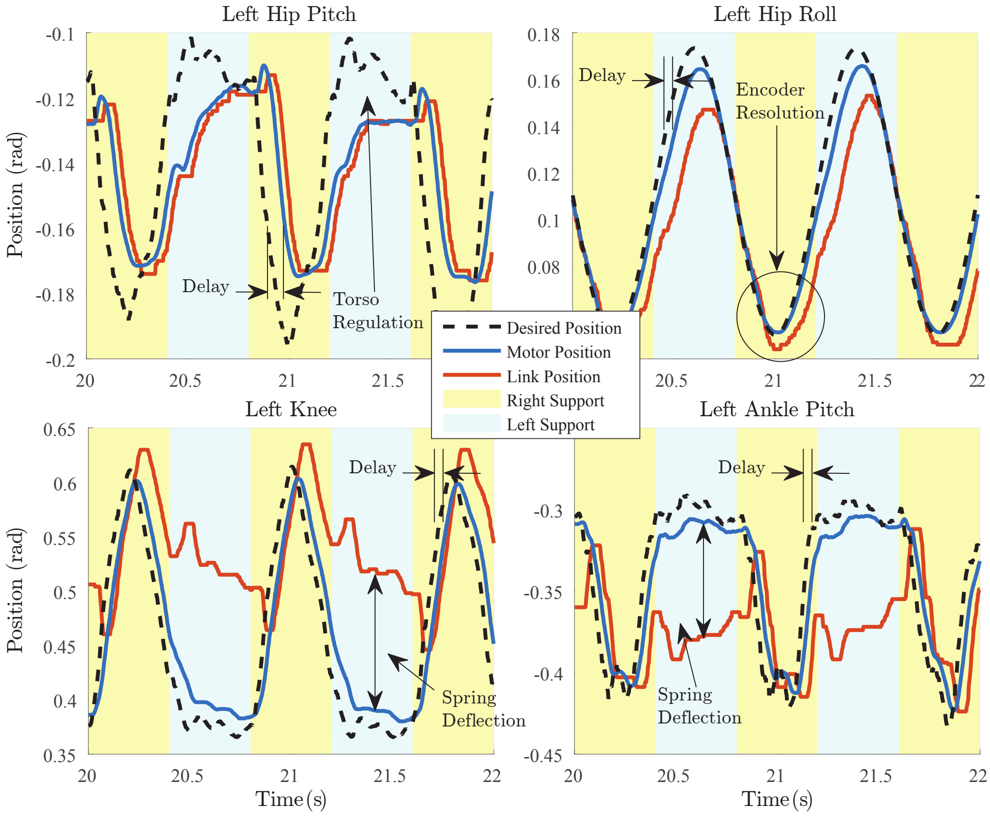


Figure 8.2 – Typical trajectories of joints involved in an in-place walking gait. Link encoders produce noisy signals which cannot be used for stiff position control directly, but motor encoders are precise enough for this purpose. In the hip pitch joints, we use a switching control rule in swing and stance phases which applies footstep adjustment and corrects for IMU angles respectively. Due to the noise on these signals, we use a smaller position gain in hip pitch joints which increases the tracking delay. Considerable spring compressions are also observed in the knee and ankle pitch joints which increase the mismatch between actual and ghost robots in practice.

in the accompanying video). However, these inaccuracies further complicated the state estimation described later in this section.

Actuator limitations:

In chapter 3, to perform whole-body inverse dynamics and torque control, we attempted to identify a model for COMAN actuators. We designed training trajectories with static and dynamic profiles and used a simple proportional controller to track these trajectories by producing actuator voltages. By measuring the output torque, we were able to relate actuator voltages, velocities and output torques together:

$$\begin{aligned} v(t) &= Ri(t) + kN\dot{\theta}(t) \\ kNi(t) &= \tau_{\text{out}}(t) + A_v\dot{\theta}(t) + A_c \,\text{sign}(\dot{\theta}(t)) + J_m\ddot{\theta}(t) \end{aligned} \tag{8.1}$$

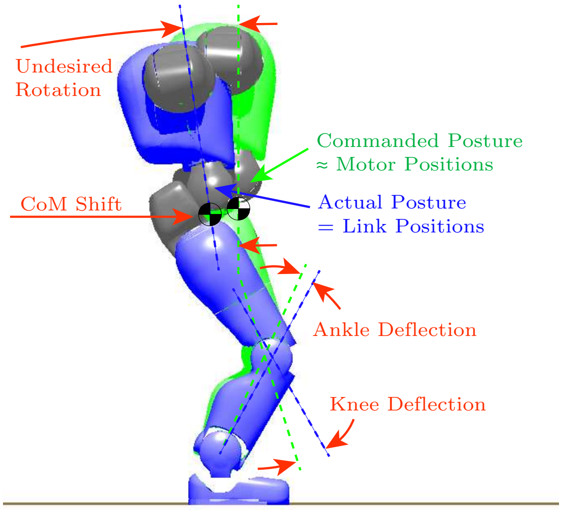


Figure 8.3 – When commanding a resting posture with the CoP in the middle of support polygon, spring compressions lead to a different posture. Ankle spring compressions make the robot lean forward while knee spring compressions make it lean backward. These two effects cancel each other to some extent, but in practice, the torso would have an undesired rotation depending on desired knee angles and CoP locations. The actual CoM might also move backward for 1-2cm.

where $v(t)$ denotes motor voltage, R denotes terminal resistance, $i(t)$ denotes motor current, k denotes torque constant which is assumed to be equal to the back-emf constant, N is harmonic drive ratio, $\tau_{\text{out}}(t)$ is output torque, $\theta(t)$ is shaft position, J_m is reflected rotor inertia and A_v and A_c are coefficients of a simple Coulomb-Viscous friction model. Since current measurements were not reliable enough, we combined equations of (8.1) and obtained:

$$v(t) = \alpha\dot{\theta}(t) + \beta[J_m\ddot{\theta}(t) + \tau_{\text{out}}(t) + A_c \operatorname{sign}(\dot{\theta}(t))] \tag{8.2}$$

where:

$$\alpha = \frac{RA_v}{kN} + kN, \quad \beta = \frac{R}{kN}$$

In COMAN actuators on average $\alpha \approx 5.42$ Vs/rad, $\beta \approx 0.45$ V/Nm, $J_m \approx 0.23$ Nms^2/rad and $A_c \approx 1.66$ Nms/rad. This means voltages are converted to joint torques with a coefficient of $\beta \approx 0.45$ V/Nm and there is a damping of $\alpha/\beta \approx 12$ Nms/rad in practice. A challenge in our experiments was to operate the robot in a reduced maximum voltage of 15V which leads to a theoretic velocity bound of $\dot{\theta}_{\text{max}} = \pm\ 2.43$ rad/s. Regarding the choice of walking frequency (2.5 step/s), the knees do not have enough time to bend fast which makes the knee-bending foot lift strategy limited. We have demonstrated this effect over some perturbed gait trajectories in Figure 8.4 where fixed and adaptive leg lift strategies (described later) command a fast bending angle to the right knee.

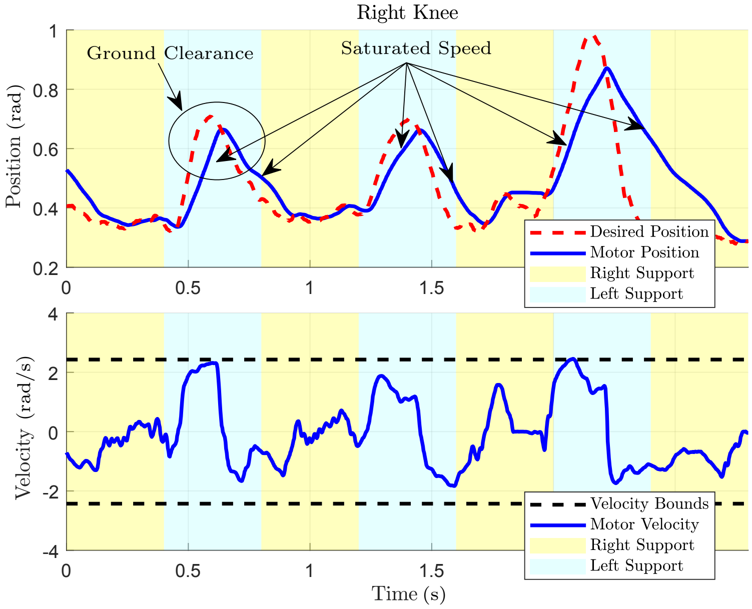


Figure 8.4 – In normal or perturbed walking conditions, a small leg lift of 2-3cm cannot be easily achieved due to a saturation of actuator velocities. The maximum velocity is determined by power supply voltage and motor properties. Due to the targeted perturbed walking conditions in our experiments, we operate the robot with a slightly lower but safe voltage to avoid high currents and unwanted shut-downs. This limits the leg lift performance which might lead to falling when a fast swing progression is commanded, but the foot is not lifted enough, and the toes still touch the ground.

We normally operate the robot with knee angles of 0.4rad ≈ 23 deg while, as depicted in Figure 8.4, a leg lift of 2-3cm cannot be completely realized due to saturated knee velocities. In practice, we cannot operate the robot in straight-knee postures either, although our inverse kinematics (appendix B) is able to handle singularities. Limited velocities motivated us to combine a hip-roll foot lift strategy with the knee-bending strategy to provide an extra leg left without overloading the knee joints. A delayed leg lift is hazardous in practice, for example when the hip is pushing for a fast swing trajectory, but the foot is yet not lifted enough to clear the ground. In this case, the toes might touch the ground much earlier than expected which may lead to an immediate fall.

SEA compensation:

As described earlier, due to the spring compressions, the ghost robot might mismatch the actual robot considerably. This mismatch is hard to compensate in the ankles since the CoP moves back and forth easily in practice and the compression angle is variable. In the knees, however, this compression is quite constant assuming relatively short step lengths. Due to heavy duty daily operations, the knee springs get permanently deflected in COMAN over time which requires frequent replacement. In this regard, we perform an open-loop in-place walking gait before experiments to identify steady-state knee spring

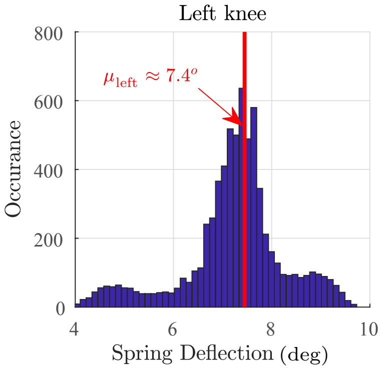


Figure 8.5 – Histograms of typical knee spring compressions during an in-place walking gait when experiments where done. These plots depend on the status of springs because they permanently deflect over time and lose their original stiffness which needs a regular replacement. Assuming short step lengths, we compensate these deflections by a feed-forward angle added to the desired knee trajectories. The compensating angles μ_{left} and μ_{right} are constant all the time, but multiplied by the contact forces to make them effective only during the stance phase in each leg.

compressions which are plotted as histograms in Figure 8.5. This test can be easily repeated to identify the current state of the robot before experiments. In our algorithm, the statistical average of these histograms is used as a constant, multiplied by normalized contact forces and added to the desired knee angles. In other words, assuming $F_{\text{left}}(t)$ and $F_{\text{right}}(t)$ to represent instantaneous filtered contact force measurements, mg to be the total weight and μ_{left} and μ_{right} to represent mean deflections, we add $\delta\theta_{\text{left}}(t)$ and $\delta\theta_{\text{right}}(t)$ to the desired knee trajectories:

$$\begin{aligned} \delta\theta_{\text{left}}(t) &= -\frac{|F_{\text{left}}(t)|}{mg}\mu_{\text{left}} \\ \delta\theta_{\text{right}}(t) &= -\frac{|F_{\text{right}}(t)|}{mg}\mu_{\text{right}} \end{aligned} \tag{8.3}$$

These adjustments can be a function of time as well, but we found this adaptive compensation smoother. Besides, the magnitude of adjustment is so small that it does not cause shaking when directly combined with contact forces and given to the stiff position controllers. However, the compensatory influence is considerable. Given a shank length of ≈ 25cm, $\delta\theta_{\text{left}}(t)$ and $\delta\theta_{\text{right}}(t)$ each compensate about 3.2cm and 1.4cm of errors on relative horizontal pelvis-feet positions respectively.

8.2.2 Open-loop control

This section introduces simple strategies used to generate a periodic rhythm of motion based on 3LP gaits. The components of our open-loop control are gait generation by 3LP, foot lift strategies and hip switching rules.

Lateral plane:

As motivated earlier, a simple foot lift strategy through rotation of hip pitch, knee and ankle pitch joints brings the robot close to actuator velocity boundaries. However, we can use the hip and waist roll joints as well to lift the entire leg. This strategy is inspired by pelvic roll observations in human walking, and used by WABIAN-2 robot earlier (Omer et al., 2005) to achieve straight-knee walking gaits. Figure 8.6 demonstrates both foot lift strategies side-by-side. In this work, we use a combination of the two to make sure that knee actuators stay far from their limits in normal conditions. This enables the actuators to reach fast yet feasible velocities in perturbed walking conditions. Lateral bounces in 3LP are as small as 2.2cm total displacement of the pelvis, given the desired step width of 25cm and step time of $T = 0.4$s. These bounces induce sinusoidal trajectories of about $0.03\text{rad} \approx 1.7\ \deg$ amplitude in the hip roll joints. However, a much larger pelvis roll of $0.1\text{rad} \approx 5.7\ \deg$ leads to a foot lift of only 1.4cm (pelvis width is about 14cm in COMAN). Since the hip-roll strategy requires much larger motions in the joints, it becomes dominant, and there is no need to use lateral bounces of 3LP anymore. In practice, we command a sinusoidal pelvis roll trajectory of:

$$\delta\theta_{\text{roll}}(t) = 0.1\ \sin(\frac{\pi t}{T}) \tag{8.4}$$

and a half-sine vertical Cartesian trajectory of:

$$\delta z_{\text{fixed}}(t) = 0.01\ \sin(\frac{\pi t}{T}) \tag{8.5}$$

to the swing foot to lift it about 2.4cm in total. This makes the robot converge to a natural lateral bouncing, similar to WABIAN-2 robot (Omer et al., 2005). In this paper, we do not use any specific lateral footstep adjustment, momentum regulation or variable-timing phase-based control. Also, we set the ankle roll gain to a small value to let the foot adapt to the ground and establish at least a border contact. This increases the effective contact area in perturbed conditions (Collins and Ruina, 2005), provides more realistic ground reaction forces (Hamner et al., 2013) and reduces unwanted transverse slippages during walking.

Sagittal plane:

Given a desired velocity, open-loop 3LP gaits $q(t)$ are generated in the Cartesian space and assigned to the robot through inverse kinematics. Both foot lift strategies augment the desired Cartesian positions and orientations of 3LP and given to our inverse kinematics controller (appendix B). We consider Cartesian pelvis and feet positions for tracking and state estimation in this work. The pelvis point is precisely in the middle of the two hip joints, and the foot position is on the ankle joint axis, slightly (3cm) shifted forward to be precisely in the middle of the foot. The advantage of considering the pelvis instead of CoM is avoiding the influence of upper-body arm oscillations which can highly perturb the CoM state. In COMAN, the two arms are rather heavy, and shoulder springs are

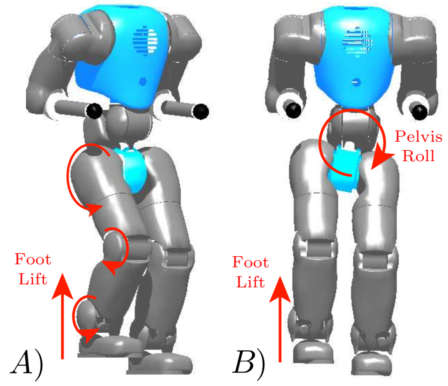


Figure 8.6 – A) Knee-bending and B) hip-roll strategies used to lift the foot in our walking gait generation. In practice, an open-loop foot lift of about 2.4cm is achieved through a combination of both strategies at the same time.

much softer than those used in the legs. This can highly perturb the state estimation in perturbed walking conditions.

Position control:

In this work, similar to MIT's Spring Flamingo (Pratt and Pratt, 1998) and many other robots, we use stance hip joints to regulate the torso angle instead of tracking the desired stance hip angle. This can be perfectly done with inverse dynamics and torque control and by using IMU orientations as feedback on pelvis orientations (chapter 3). In 3LP also, ideal stance hip actuators ensure an upright torso posture by calculating the required torque to be applied. Since we do not have torque control available in the present work, we only rely on position control. We use a simple time-based transition law that changes the desired and actual angles from swing command and hip encoder to zero torso angle and IMU pitch signal during swing and stance phases respectively. The hip actuator then tries to reduce the error by applying a proportional voltage. Denoting the IMU pitch by $\theta_{\text{pitch}}(t)$, the desired hip angle (in both phases) by $\theta_{\text{des}}(t)$ and the measured hip angle by $\theta_{\text{act}}(t)$, our transition law for the hip is formulated as:

$$v(t) = k_d[\gamma(t)(\theta_{\text{des}}(t) - \theta_{\text{act}}(t)) + (1 - \gamma(t))(-\theta_{\text{pitch}}(t)] \tag{8.6}$$

where $\gamma(t) = e^{-(t/t_1)^2}$ in stance phase and $\gamma(t) = 1 - e^{-(t/t_1)^2}$ in the swing phase, t is the time since beginning of the phase and $t_1 = 0.2T$. The choice of t_1 is based on the force transfer rate during contact switch time (double support $\approx 20\%$), $v(t)$ is commanded actuator voltage and k_d is the proportional gain. Figure 8.9 demonstrates the switching rule for different moments of the gait. The position control for all other joints is simply defined by:

$$v(t) = k_d[\theta_{\text{des}}(t) - \theta_{\text{act}}(t)] \tag{8.7}$$

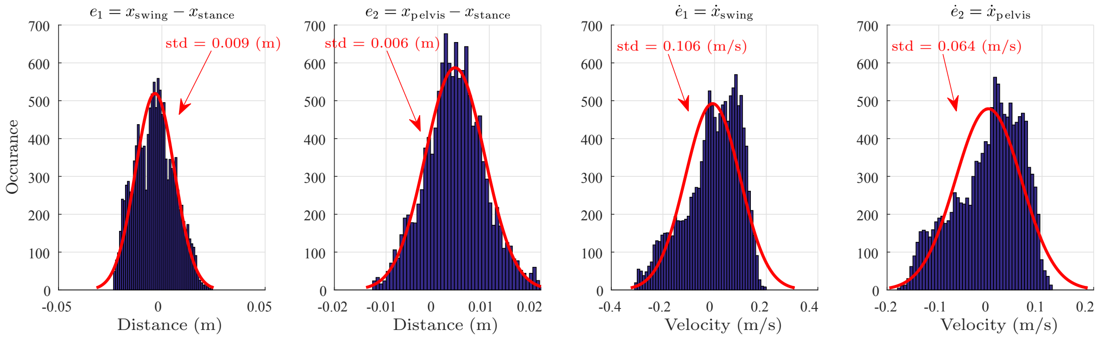


Figure 8.7 – The quality of filtered state errors over a task of in-place walking. These values should be zero in ideal conditions. However, joint backlashes, calibrations and model mismatches in forward kinematics produce systematic errors that cannot be filtered out with a simple time filter of (8.8). We simply fit a Gaussian model on these measurements and estimate the typical standard deviation which is later used to determine the threshold for a dead-zone function.

where $\theta_{\text{des}}(t)$ and $\theta_{\text{act}}(t)$ are the corresponding desired and actual angles. The control gains are set to 10 V/rad for ankle roll joints, 100 V/rad for hip pitch joints and 500 V/rad for all other joints.

8.2.3 Closed-loop control

Our proposed control paradigm is based on planning and stabilization in the Cartesian space. We use inverse kinematics to convert Cartesian 3LP gaits to joint angles and then forward kinematics to calculate the actual Cartesian positions at each instance of time. In this section, we are going to discuss our closed-loop control which stabilizes the robot by dynamic footstep adjustment. The closed-loop controller has three main blocks: state measurement and filtering, adaptive lift strategy and footstep adjustment.

State estimation:

We use link encoder (post-spring absolute encoder) and IMU measurements to build the kinematic chain and then to find relative horizontal pelvis-feet vectors $s_1(t) = x_{\text{swing}} - x_{\text{stance}}$, $s_2(t) = x_{\text{pelvis}} - x_{\text{stance}}$ (shown in Figure 6.2) and their derivatives. Due to a quantization error, however, these vectors have a considerable noise to be filtered. Besides, joint backlashes and geometric model mismatches also contribute to a more systematic erroneous state measurement. We cancel the quantization error by a simple Infinite Impulse Response (IIR) filter of the form:

$$s(t) = (1-\zeta)s(t-\delta t) + \zeta\hat{s}(t) \tag{8.8}$$

where $\hat{s}(t)$ is the observed signal at time t, $\delta t \approx 2$ms is our effective control rate and $\zeta = 0.1$ is tuned to provide a right level of signal filtering and delay. In practice, this parameter was tuned together with position control gains on the hip pitch joints to ensure fast tracking and to avoid high frequency shakes due to actuation delays. To estimate

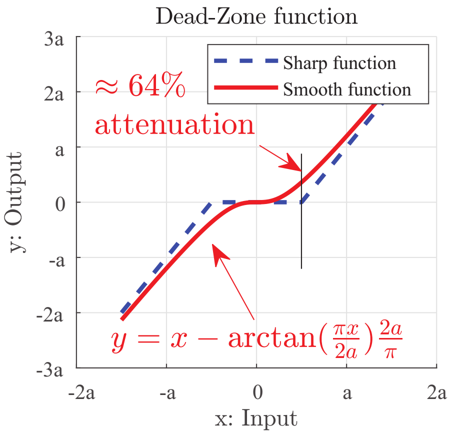


Figure 8.8 – The smooth dead-zone function used to filter systematic errors due to backlashes, calibration and model mismatch problems. Compared to the sharp dead-zone function, the smooth function provides a considerable attenuation (≈ 64%) which makes it suitable for error reduction.

velocities, we simply use time differentiation over the filtered $s_1(t)$ and $s_2(t)$ signals, and another averaging filter afterward with a window of 30 samples. More systematic errors due to backlashes and model mismatches are canceled by introducing a dead-zone function. We denote the errors of $s_1(t)$ and $s_2(t)$ with respect to the nominal gait by $e_1(t)$ and $e_2(t)$. We run an open-loop in-place walking in which the robot only bounces left and right without any forward or backward progression. This gait is stable in a very limited region of states, depending on the stiffness of ankle joints and the feet size. Figure 8.7 shows statistical values of filtered errors $e_1(t)$ and $e_2(t)$ and their derivatives in the sagittal direction. With ideal actuation and measurements, these values should be zero all the time. However, they have a certain standard deviation which we measure by fitting a Gaussian model on the data. A smooth dead-zone function:

$$y = x - \arctan(\frac{\pi x}{2a})\frac{2a}{\pi} \tag{8.9}$$

can attenuate the signal by 64% at the threshold value a, shown in Figure 8.8. We apply such dead-zone on each of the signals in Figure 8.7 and set the threshold a to their standard deviation. Our two-stage filtering helps to stay in the limited basin of attraction produced by CoP modulation, but smoothly and automatically switching to foot-stepping strategy whenever disturbances get large.

Adaptive foot lift strategy:

One of the key control components in perturbed walking conditions is foot lift which should provide enough ground clearance to allow for a complete swing motion and a precise tracking of next footstep locations, which then play important roles in stabilization. However, due to actuation limitations and especially velocity limits (shown in Figure 8.4), we cannot lift the foot in COMAN as much as desired. Previously, we introduced

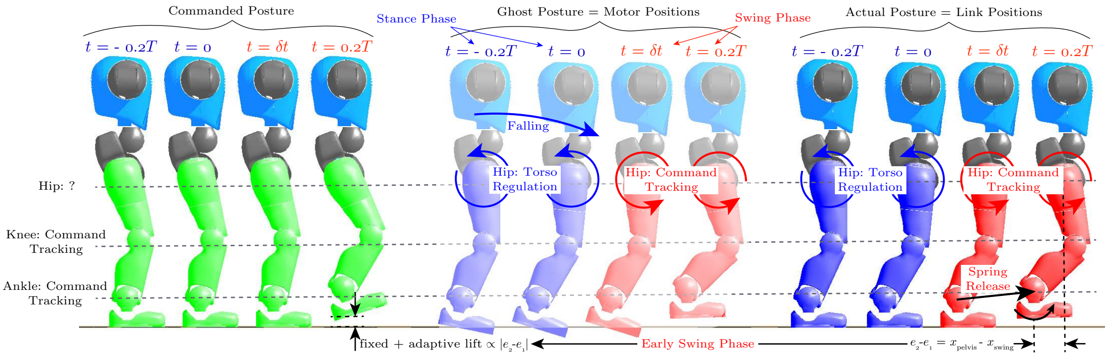


Figure 8.9 – Phase transitions in commanded, ghost and actual postures defined in Figure 8.1. The focus of this figure is to investigate hip switch rules and adaptive foot lift strategies, demonstrated here for example in an in-place walking gait. When the right leg is stance phase at time $t = -0.2T$, the commanded posture implies a pelvis position on top of the stance foot position. Due to external pushes, the robot might fall where the pelvis moves forward, getting away from the stance foot. This naturally happens since the stance hip is regulating the torso angle and not the desired hip angle. At time $t = 0$, a phase transition happens while the robot is still falling. The toes of the ghost robot penetrate in the ground while ankle springs compress and move the CoP forward in an attempt to resist against the push passively. When the swing phase starts at $t = \delta t$, the actual hip angle is far from desired, the ankle spring is still compressed, and the toe in the ghost robot penetrates the ground. An adaptive foot lift strategy is therefore needed on top of our fixed strategies which are only designed for the commanded posture. The adaptive lift, depending on the current posture, provides an extra lift on the commanded posture which leads to enough ground clearance shown at $t = 0.2T$. It also applies a corrective orientation to compensate leg rotation and ankle spring compressions (described better in Figure 8.10).

a mixture of knee-bending and hip-roll fixed foot lift strategies which already provide few centimeters of ground clearance. However, since the stance hip pitch is assigned to regulate the torso angle, the stance foot might get away from the pelvis, depending on external pushes and falling dynamics. Since in stance phase, we lose track of desired stance hip pitch trajectories in favor of torso regulation, the amount of leg lift needed in the early next swing phase should depend on the actual relative swing foot position. To better understand this effect, imagine the robot is performing in-place walking, and a strong push is applied from behind. The pelvis naturally moves forward with respect to the stance foot in the same phase while the torso is kept upright by the stance hip joint. In the next phase, the stance leg goes to swing mode, where the desired gait is only in-place walking. The desired swing hip angle is almost zero while the actual angle is such that this leg is left behind the robot in very early moments of the swing phase that has just started. Figure 8.9 demonstrates this effect in details. At time $t = -0.2T$ for example, the right leg is in stance mode, and the robot is falling forward. Assume the phase transition happens at $t = 0$ where the robot is still falling forward.

At time $t = \delta t$, the right leg has started the swing motion, but its actual position is far from the commanded position. The fixed foot lift strategies are designed for the commanded posture (shown in Figure 8.9) while the actual swing foot position needs an extra lift to clear the ground safely. Our Adaptive foot lift strategy is straightforward. It calculates the relative horizontal pelvis-swing position out of the filtered error signals (i.e., $e_2(t) - e_1(t)$) and approximates the actual hip angle:

$$\hat{\theta}_{\text{act}}(t) \approx \arctan(\frac{e_2(t) - e_1(t)}{z}) \tag{8.10}$$

where z is the constant pelvis height in 3LP model. This approximation is already filtered and registered in the inertial frame thanks to the IMU orientation while the actual post-spring hip encoder angle θ_{act} does not have these properties. The amount of adaptive leg left would be:

$$\delta z_{\text{adaptive}}(t) = z[1 - \cos(\hat{\theta}_{\text{act}}(t))] \sin(\frac{\pi t}{T}) \tag{8.11}$$

which compensates the mismatch between commanded and actual postures, shown in Figure 8.9. The use of a sinusoidal signal is to ensure extended legs at phase transition moments and to prevent the robot form collapsing in practice. At time $t = 0.2T$, the adaptive lift strategy induces an extra lift in the commanded posture (shown in Figure 8.9) which goes to inverse kinematics and produces an extra foot lift, whenever the actual swing foot is unexpectedly far from the desired swing foot position.

Due to hip switching rules and in addition to the foot height mismatches explained, the orientation of the foot might also not be horizontal in early swing phases (shown in Figure 8.9). This is purely due to rotation of the swing leg in the previous stance phase which has to be corrected in the current swing phase. This effect is shown in Figure 8.9 on the ghost robot. Note that the commanded posture requires a flat foot, however in stance phase, the robot falls forward and the whole stance leg rotates, leaving the foot behind the pelvis. Now, since stiff position controllers are tracking the desired ankle angles, the ghost foot penetrates the ground. However, the SEA elements compress in this case and bring the CoP to the toes. This has a positive stabilization effect during the fall and slows it down. However, in the early next swing phase, although the robot attempts to lift the foot through fixed and adaptive strategies, the ankle spring is still releasing, causing the toes to keep touching the ground at time $t = 0.1T$, shown in Figure 8.10. Therefore, there are two distinct effects: I) foot orientation mismatch because of our hip switch rule, II) spring release which happens whenever the CoP goes towards the toes, regardless of the control algorithm. Figure 8.10 provides a closer perspective aiming at a better clarification of spring dynamics in the early swing phase. To compensate both effects, we simply add feedback on the foot orientation:

$$\delta\theta_{\text{pitch}}(t) = -\hat{\theta}_{\text{act}}(t) \tag{8.12}$$

where $\hat{\theta}_{\text{act}}$ is coming out of the forward kinematics model. This feedback which is added

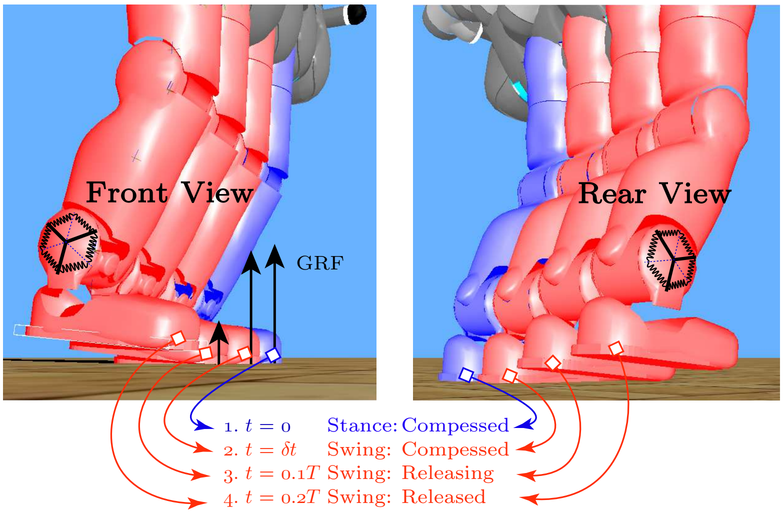

Front View
Rear View
GRF
1. $t = 0$ Stance: Compressed
2. $t = \delta t$ Swing: Compressed
3. $t = 0.1T$ Swing: Releasing
4. $t = 0.2T$ Swing: Released

Figure 8.10 – The effect of ankle spring compressions in early swing phases. Consider a phase change happens at $t = 0$ where the stance leg switches to swing phase. Due to external pushes, the pelvis might move forward during the stance phase which brings the CoP forward too, since a fixed ankle angle is commanded. The movement of CoP is in fact due to the resistance of ankle spring against rolling which indeed provides positive stabilization properties, depending on the ankle spring stiffness. Now, when the swing phase starts at $t = \delta t$, the ankle spring is still compressed which continues to release during the leg lift. At some point, the spring releases completely which is when the actual toe-off happens, and the swing can start. In our adaptive lift strategy, we add some orientation correction feedback to speed up this effect.

to the desired foot orientation before inverse kinematics keeps the foot orientation always horizontal with respect to the ground. Therefore, our adaptive lift strategy is composed of both foot height and foot orientation compensations.

Footstep adjustment:

The second part of our closed-loop control is footstep adjustment based on time-projection control. Remember that the adaptive lift strategy was introduced to handle perturbed walking conditions and to make sure that desired footstep locations can be realized by providing enough ground clearance. After filtering errors $e_1(t)$, $e_2(t)$ and derivatives, we have an estimate of robot's current state error which is further passed through dead-zone functions to reduce systematic errors. This state variable is used in the time-projection controller to find footstep adjustments. In the absence of disturbances, the time projecting controller produces the same corrective input when the error evolves in time according to the natural system dynamics. In other words, once the external push is released, it leaves an error in the system which follows system's dynamical equations. In this regard, assuming short and bounded forces, we expect the time-projecting controller to produce the same final footstep in all time instances after the intermittent push until the end of the phase. Due to IMU and encoder noises, we have to decrease position controller

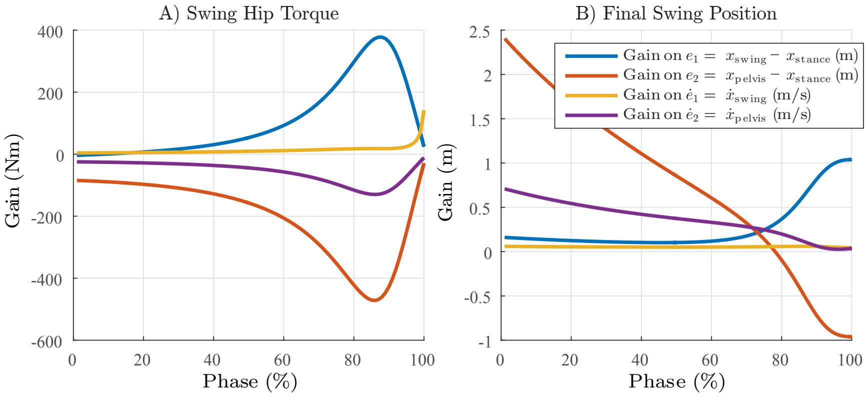


Figure 8.11 – A) The instantaneous output of time-projecting controller on the 3LP model in terms of hip torques. B) The resulting footstep adjustment by simulating the current measured error until the end of the phase. In our position-controlled framework, we measure and filter errors at any time t, perform an inner product by the corresponding gain vector of these figures at time t and truncate the result to find a safe final footstep adjustment. This value is then divided by the 3LP pelvis height z to obtain an attack angle adjustment.

gains in the hip joints to avoid shaking. This inevitable policy introduces more delay in the tracking of the desired hip trajectories, yet is enough to stabilize the system and realize commanded footstep locations. Because of these practical reasons and to set up a simple controller, we directly use the final footstep location to let the swing leg reach it on time. At each control tick (every 2ms approximately), we take the current Cartesian state and simulate it until the end of the phase in a closed-loop 3LP simulation where the time-projecting controller adjusts the swing hip torques. The resulting footstep adjustment at the end is extracted (from posture F in Figure 8.1) and converted to a proper attack angle.

In practice, thanks to the linearity of 3LP and time-projection control, we can simply extract control laws offline and interpolate them in the form of a look-up-table on-line which is computationally very efficient. Note that the effective sagittal error vector used here has 4 dimensions (sagittal components of $e_1(t)$, $e_2(t)$ and their derivatives). The output of projecting controller has two dimensions in the form of constant and time-increasing torques which are added together at each instance of time t and produce a single swing hip torque values. On the other hand, when simulating the current error until the end of the phase, the time projecting controller produces a footstep adjustment in the sagittal direction which has one dimension. Therefore, two different look-up-tables can be produced, one for a torque-controlled paradigm and one for our position-controlled paradigm. Each control table maps a four-dimensional error into a single output. The control gains of individual error dimensions are shown in Figure 8.11 at different phase times. These gains depend on model properties and the walking gait frequency, but not the speed (refer to chapter 7). In practice, we only take filtered errors and perform inner product with an interpolation of the final swing position control gains in Figure 8.11. A truncation of 15 $cm \approx 0.35z$ is also applied to avoid huge steps and breaking the hardware when too strong pushes lead to failure during experiments. Results are finally divided by the 3LP pelvis height z to approximate the equivalent attack angle adjustment.

Having explained both open-loop and closed-loop simple control policies, we are interested in assessing the effect of each control block as well as push recovery and walking scenarios in different conditions. These results are presented in the next section.

8.3 Results

The results presented in this section are categorized into five groups of in-place walking, intermittent push recovery, continuous push, normal walking and stiffness measurement scenarios. We explore the functionality of closed-loop and open-loop control configurations where the foot-stepping mechanism is just disabled in the second case. We further continue with the closed-loop controller and in-place walking and show how the algorithm can take corrective steps to recover pushes in different directions. Next, we disable the footstep adjustment and demonstrate the limited stability of open-loop controller which is due to passive CoP modulation. This controller can easily follow a continuous external push and pull while behaving almost on the margin of stability. Finally, we show how different walking gaits can be generated with this framework where we only change the desired speed in the 3LP gaits. Finally, we perform a simple standing experiment and demonstrate the effective level of ankle stiffness in COMAN which further clarifies dynamic walking features of our controller.

8.3.1 In-place walking

As motivated in the beginning, CoP modulation provides a fast yet limited control authority for immediate stabilization. Foot-stepping, on the other hand, provides stronger stabilization, but only over phase changes. SEA elements and internal joint backlashes lead to a relatively compliant ankle joint, shown at the end of this section. The passive ability of ankle springs in resisting against falling together with the fact that during in-place walking, the swing foot comes under the pelvis all the time make the open-loop gait stable in a limited region of states without any time-projection control. Figure 8.12.A shows foot positions with respect to the pelvis during this open-loop gait which is also used in Figure 8.7 to design the dead-zone functions. In Figure 8.12.B, we show how the complete closed-loop system behaves when dead-zone functions reduce systematic errors of Figure 8.7.

Without dead-zone:

Without dead-zone functions, the closed-loop system gets perturbed completely. This is shown in Figure 8.12.C where the robot is still stable, but stumbling considerably. In these conditions, the lateral stability of the robot is also systematically influenced because of unexpected touchdown events shown in Figure 8.13. Here, the adaptive foot

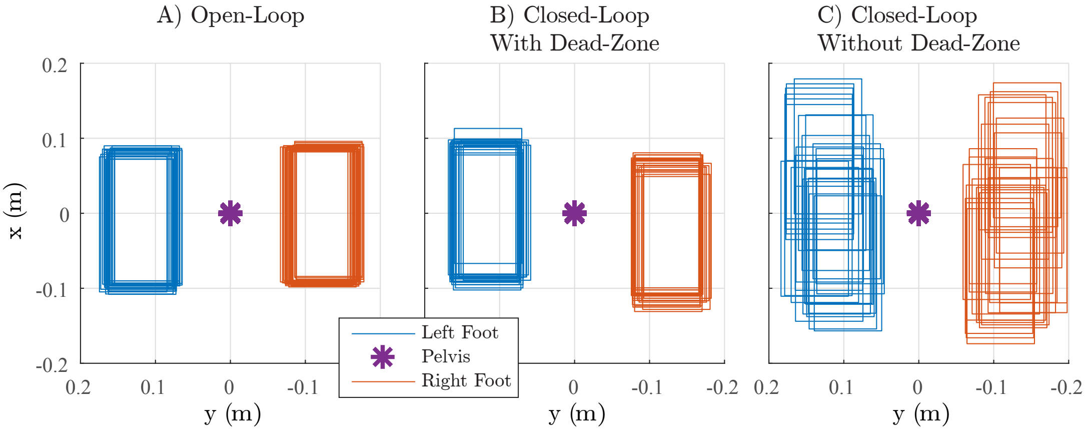


Figure 8.12 – In-place walking in A) open-loop, B) closed-loop and C) closed-loop without dead-zone functions. This figure shows horizontal foot locations only at phase change moments over 10s of walking, i.e., 25 steps. The open-loop gait is stable in a very limited region of states thanks to the passive CoP modulation of ankle springs and the natural swing trajectory which always comes under the pelvis during an in-place walking gait. When applying dead-zone functions in closed-loop control, the in-place walking gait is only slightly perturbed in figure B). Without the dead-zone functions, however, the walking gait is completely perturbed. Here, the footstep still stabilizes the system, but the timings and trajectories are all perturbed systematically, which is not desired. The corresponding movies of all the three scenarios could be found in the accompanying video.

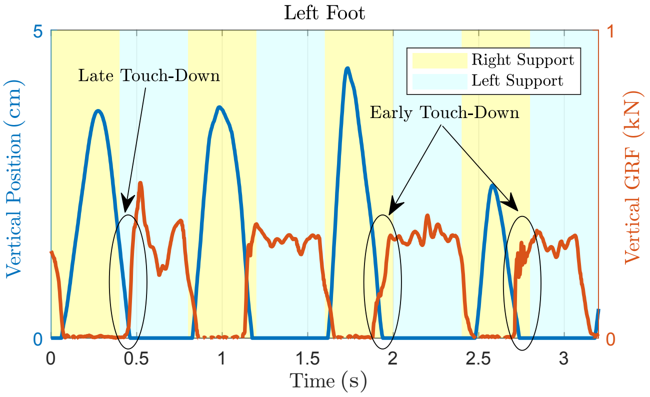


Figure 8.13 – Closed-loop in-place walking when the dead-zone functions are disabled. In this case, the robot stumbles considerably due to the systematic errors in state estimation. The time-projecting controller stabilizes the system despite an erroneous measured state. However, actual phase change moments might happen earlier or later, depending on adaptive foot lift trajectories which are also subject to an erroneous measured state. This illustrates the effectiveness of dead-zone functions in improving the repeatability of closed-loop in-place walking gaits.

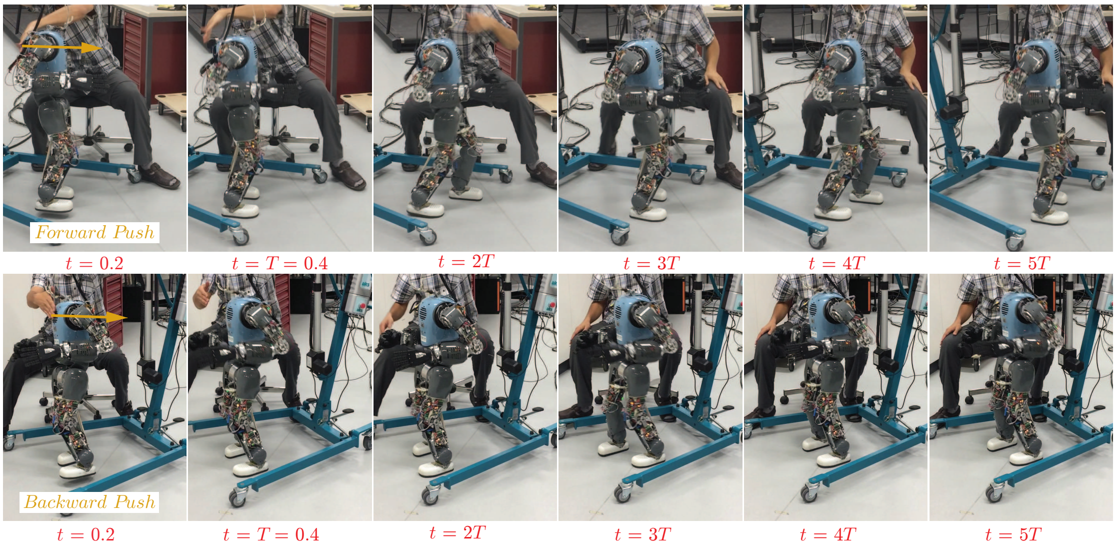


Figure 8.14 – Short intermittent pushes during an in-place walking gait in different directions. On top, the robot almost takes three corrective steps to recover the forward push while it only takes two steps the recover a lighter backward push. Snapshots are taken at phase change moments every $T = 0.4$s. The corresponding movies could be found in the accompanying video.

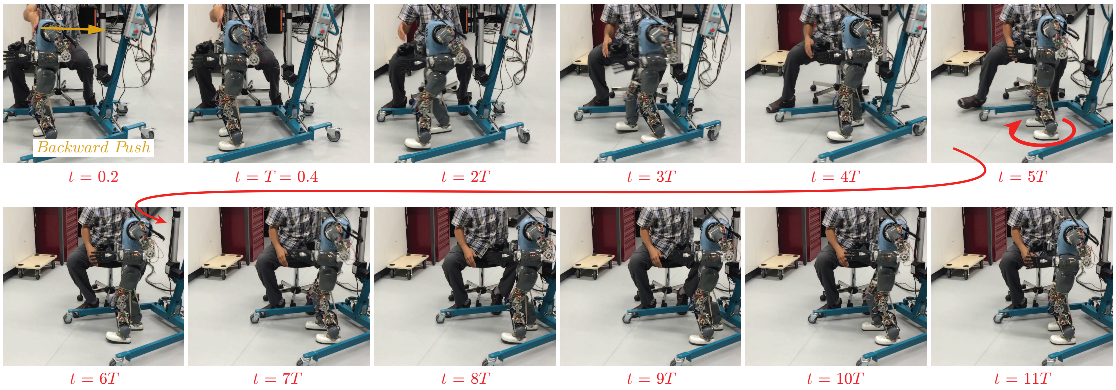


Figure 8.15 – Recovery process of a strong backward push during in-place walking. Snapshots are taken at phase change moments every $T = 0.4$s. Since the passive CoP modulation is almost ineffective in the backward direction, the stepping strategy takes multiple steps to recover the strong push. Despite a safety threshold which limits the step size, the push is so strong that it violates decoupling assumptions, leading to a sudden rotation at $t = 5T$. The new undesired perturbation requires the robot to take more backward steps to recover completely and go to a rest condition. The corresponding movie could be found in the accompanying video.

lift strategy commands different clearance heights due to erroneous measured states. This can lead to unexpected touchdown events which prevent a complete swing when happening earlier than expected or produce an early stance phase when the leg is still in a swing phase.

8.3.2 Intermittent push recovery

Although the passive CoP modulation provides little stability over in-place walking in the absence of considerable disturbances, we would like to know how effective the foot-placement strategy is. We use the same closed-loop in-place walking gait of the previous part and apply short intermittent pushes on the torso.

Moderate Pushes:

Figure 8.14 shows a sequence of walking snapshots taken at phase change moments when a forward and backward push is applied to the system. Depending on the timing of the push, a small footstep adjustment takes place in the same phase with the push, but the next footstep locations recover the push gradually. Due to the safety threshold (= 15cm) applied on footstep adjustments and depending on the push strength, the recovery process might take more than 1-2 steps. Note that we typically command the pelvis in the middle of the support polygon. Ankle SEA springs, therefore, compress slightly as shown in Figure 8.3 due to a non-zero ankle torque when the CoP is not exactly under the ankle joint. Because of this default compression, the passive CoP modulation strategy resists more against forward pushes than backward pushes. Although most of the stabilization comes from the stepping strategy, overall, recoverable forward pushes can be slightly stronger than backward pushes.

Strong pushes:

Our control design and safety thresholds target small, yet dynamic footstep adjustments which do not violate the linearity of 3LP, sagittal-lateral decoupling, and actuator limitations. When a strong push is applied, some of these assumptions might not be valid anymore. The robot might be still able to recover, but over multiple steps or with an extra effort. Figure 8.15 shows such strong backward push during in-place walking. As mentioned earlier, the passive CoP modulation is almost ineffective in the backward direction which further complicates the recovery for the stepping strategy. The robot takes multiple backward steps to recover the strong push, but suddenly, a transverse slippage happens in the right foot which rotates the robot slightly. This happens due to the violation of sagittal-lateral decoupling assumptions. After this slippage, the robot takes a few other backward steps to stabilize completely.

8.3.3 Continuous pushes

Apart from impulsive pushes, we are also interested in investigating the performance when the robot is subject to a moderate continuous pushing/pulling force. In these conditions, bounded-push assumptions behind the time-projecting controller are violated. As mentioned earlier, the open-loop controller is also stable due to a passive CoP modulation, but in a very limited region of states. This controller weakly reacts to perturbations, almost on the margin of stability which makes it compliant in following

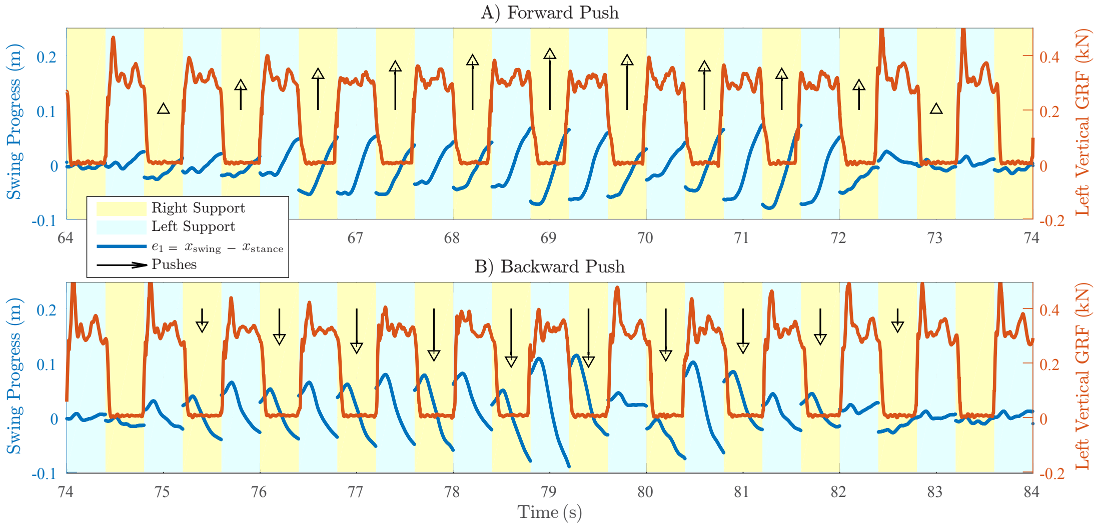
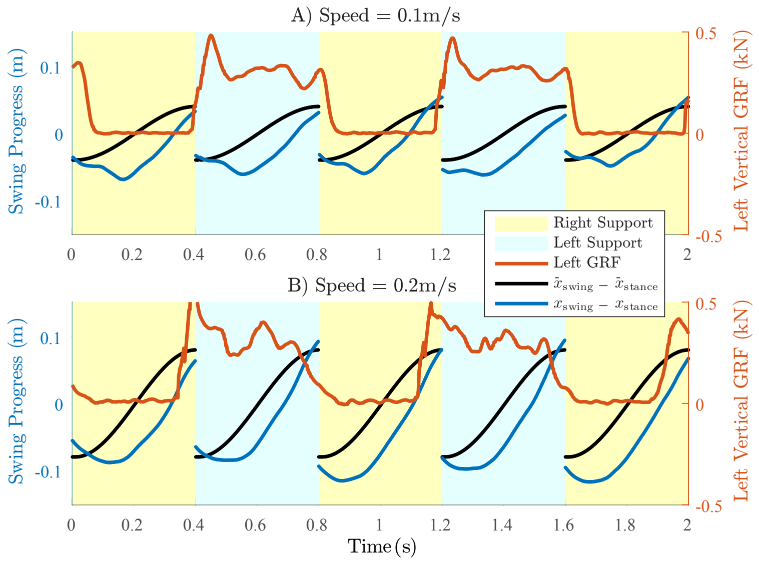


Figure 8.16 – The emergent walking of the open-loop controller when an in-place walking gait is commanded and a moderate yet continuous external A) forward and B) backward force is applied. In-place gait requires swing foot to come under the pelvis while the external drag causes a continuous falling. Passive CoP modulations in the ankle SEAs slightly resist the force and provide a weak stability. However, the gait does not have a large basin of attraction and requires a smooth profile of external forces. The important property of this controller is compliance to external pushes and emergent speed modulations, although no specific desired velocity is commanded. In this scenario, the robot can sometimes reach a peak velocity of ± 0.2m/s. The corresponding movie could be found in the accompanying video.

the pushing/pulling force. Figure 8.16 demonstrates this behavior in the presence of continuous pulling and pushing forces applied to the right hand. In these scenarios, we disable the stepping strategy while still keeping the adaptive foot lift active. Depending on actuator capabilities, the robot can easily reach walking speeds of up to ± 0.2m/s. When enabling the stepping strategy, however, the robot naturally uses footstep adjustments to slow further down. This leads to a stumbling behavior and a less rhythmic motion compared to the open-loop controller. Movies of both scenarios are included in the accompanying video.

8.3.4 Walking

Remember that due to the linear properties of 3LP and the time-projecting controller, with a fixed walking frequency, the controller remains unchanged at different walking speeds. While keeping the same closed-loop control architecture, we now change the desired speed in 3LP to produce variable trajectories. Figure 8.17 shows two different cases of walking at 0.1m/s and 0.2m/s together with snapshots from the robot. The same dynamic stabilization mechanism can be seen in small variations of swing foot trajectories where the controller provides adjustments for stabilizing the gait. One can notice that,



C) Speed = 0.2m/s

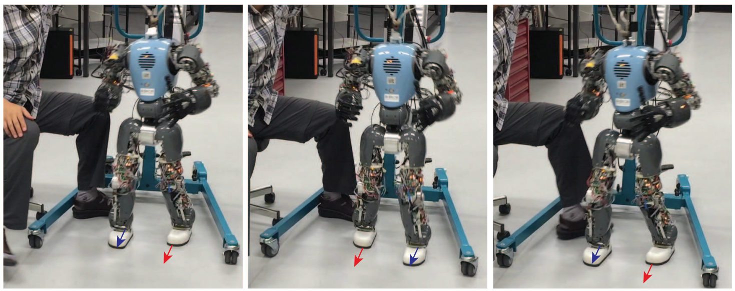


Figure 8.17 – Walking gait generation with the same controller used previously for in-place walking. In these scenarios, we command a desired velocity of A) 0.1m/s and B) 0.2m/s which produces different 3LP gaits and Cartesian profiles. Here, the passive CoP modulation has little stabilization role while the time-projecting controller plays an important role, observed in small trajectory variations. Figure C) shows snapshots of the faster gait on the real robot, taken at phase transition moments. Movies of both scenarios could be found in the accompanying video.

however, the actual phase transition takes place slightly earlier than expected in faster walking speeds. This can be due to a tracking delay in the hip joints or extra spring compressions in the stance leg which limit the ground clearance and influence the swing phase timing. The time-projecting controller can still stabilize the system, handling these timing issues.

8.3.5 Effective ankle stiffness

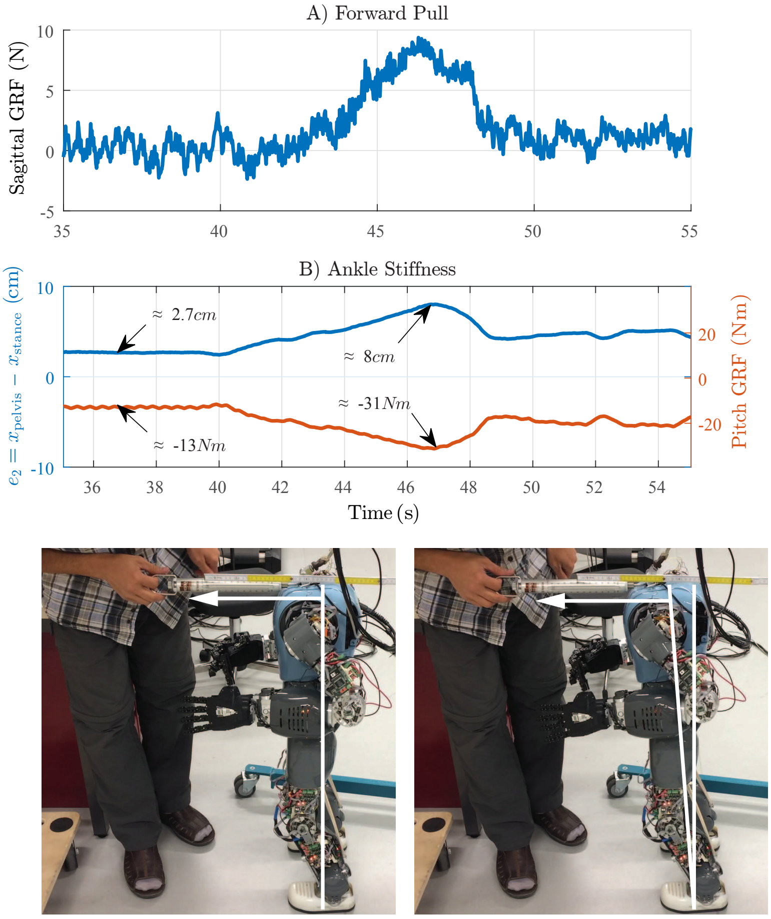
The last experiment presented in this section aims at quantifying the actual passive stiffness of ankle springs in COMAN. To this end, we command a stand-still posture to the robot with stiffest position gains everywhere, the CoP in the middle of support polygon and disabled feedbacks. In this scenario, as depicted in Figure 8.18, both ankles contribute a resisting force against external pushes applied to the neck. As Figure 8.18.A and Figure 8.18.D show contact force readings and external gauge values respectively, a minimal pulling force of $\approx$ 7N can easily move the pelvis by $\approx$ 5cm (and the torso by $\approx$ 9cm) which implies an effective stiffness of only $\approx$ 31 Nm/rad per ankle. Note, however, that the position controller with a high gain of $k_d = 500$ and an effective voltage-torque coefficient of $1/\beta \approx 2.2$ leads to a stiffness of $\approx$ 1100 Nm/rad in series which is much larger than the overall stiffness. Besides, we also have small backlash problems in the joints which further limit the influence of ankle springs. This experiment illustrates that, despite having relatively large feet in COMAN, our controller is mostly relying on the footstep adjustment strategy rather than the passive CoP modulation in recovery from disturbances.

8.4 Discussion

In this paper, we proposed a very simple but robust control architecture that can stabilize the robot against external pushes. Motivated by specific properties of our hardware, including heavy legs, we used a model called 3LP which can capture swing and torso dynamics and produce more dynamically consistent gaits compared to LIP. We also used the rich literature of Poincaré maps and DLQR controllers for walking control together with our time-projecting controller that resolves the continuous-discrete mapping and offers an on-line control scheme. This controller, in a very simple position-controlled architecture, can suggest footstep adjustments that stabilize the robot against external disturbances. By using the 3LP model, we are performing a model-based control. However, thanks to a close dynamical match of this model to the real robot, both gait generation and on-line control in our architecture take falling and swing dynamics into account. This knowledge is encoded in a very simple control look-up-table that stabilizes the robot easily without using any advanced state estimation algorithm like Kalman filtering or advanced inverse dynamics.

8.4.1 Dynamic walking

Over an extensive range of results and analysis, we showed that both passive CoP modulation and foot-stepping strategies are involved in our controller. The CoP modulation alone has a limited basin of attraction while the foot-stepping strategy is sensitive to the sensory data but more powerful in recovering strong disturbances. We resolved the sensitivity with simple filters and dead-zone functions on the sensory data. However, the effective ankle stiffness was determined by the physical hardware and much smaller compared to the virtual compliance realized through position controllers. Our architecture is not designed for an active CoP or Zero Moment Point (ZMP) control, and we only rely on the passivity of ankle springs to resist or at least damp falling dynamics. CoP modulation has, therefore, a very limited influence regarding COMAN's foot size and effective ankle stiffnesses. Our controller provides stability against very small disturbances through a passive CoP modulation and recovers more substantial disturbances dominantly through



C) Rest Condition D) Forward Pull

Figure 8.18 – Standing robot subject to external pulling forces applied to the neck. A) Filtered sagittal forces measured by contact force sensors of the robot. B) Measured contact moments and the pelvis displacement reported by forward kinematics. C) The actual rest condition. D) The maximum forward lean due to the external pulling force. One can observe that with a minimal external force, the pelvis in COMAN can easily move forward which induces an effective stiffness of ≈ 31 Nm/rad in each ankle spring. The virtual compliance realized by the position controller on the motor shaft is much stiffer, effectively negligible in series with such compliant physical springs. The corresponding movie could be found in the accompanying video.

an active foot-stepping strategy. Our proposed algorithm can be suitable for taller robots like Atlas or robots with even smaller feet sizes like Marlo, Cassie or Humo as well. This is an advantage for our model-based controller which encodes robot and gait properties all in the 3LP model. It produces then a consistent time-projection look-up-table control which only depends on the gait frequency. Position control gains and filtering constants are simply chosen to provide the stiffest control without shaking which could be tuned for other robots easily during a simple standing task.

8.4.2 Decoupling assumptions

We focused on sagittal plane dynamics only while producing lateral bounces through simple open-loop foot lift strategies existing in the literature. The relatively wide choice of step width improved the lateral stability, though limited the sagittal-lateral decoupling. Our controller is nevertheless limited to the constant-height assumption of 3LP as well which further limits step sizes in practice. Besides, velocity limitations of actuators also do not allow for fast swing motions. Inspired by numerous works on simple passive walkers with wide feet and human observations, we decreased the ankle roll stiffness and allowed for a wider foot contact in practice which increases available transverse torques and prevents unwanted turning around the yaw axis. For most of the experimental scenarios presented in this paper, the decoupling assumption was valid. When applying large pushes, however, the robot takes longer steps which might lead to small rotations, yet capturable with our powerful foot-stepping algorithm.

8.4.3 Constant CoM height

This assumption is initially used in the 3LP model to derive linear equations of motion which computationally facilitate gait generation and control. The idea of time projection can be extended to nonlinear models too, as long as one can linearize the system around a certain periodic trajectory and find any-time transition matrices $A(t)$ and $B(t)$, used in time-projection. We believe that the simple control look-up-tables found here can be used in other robots as well through the adjustment of model properties and the desired walking frequency. All the knowledge of falling and swing dynamics together with the optimality of LQR design are encoded in these look-up-tables which produce very simple swing hip torques or desired attack angles. A fixed frequency is an underlying assumption behind 3LP and time-projection. However, we showed that the architecture could tolerate significant phase mismatches too. This makes time-projection suitable for phase-based bipedal walkers and robots taller than COMAN. The algorithm might also work on smaller robots. However, they usually have larger feet which make them statically very stable.

8.4.4 Hardware limitations

The present manuscript features discussions of COMAN hardware in a very deep level. We presented actuator models, velocity limits, sensory data qualities, actuation delays, backlashes and systematic model errors. We used a mixture of fixed foot lift strategies to maximize the lift and to avoid overloading actuators in terms of velocities. The dynamic matching of 3LP with the robot is so close that our controller does not need a good quality of state estimation and control. We only read encoder and IMU positions while relying on internal actuator properties for a high-frequency damping. We also use very simple filtering policies that are all tuned based on actual robot data, position control gains, and control loop delays. The unified control framework has only a few parameters to tune and does not include tunning of any critical parameter. A higher amount of foot lift, of course, pushes actuators towards their velocity limits but improves the stability.

A stronger filtering provides cleaner signals which allow for higher gains in the hip joints, but in practice increase the delay too which is not desirable in our relatively high walking frequency.

8.4.5 Compliance

Our control architecture includes specific components to compensate for systematic errors arising from hardware compliance, but not fighting against this advantageous property that protects the hardware in perturbed walking conditions. We used a simple compensation in the knee springs to adjust unwanted rotations and pelvis shifts. We also used a simple orientation feedback to compensate ankle spring deflections in early swing phases. These simple policies improve stability considerably. We also employed a switching control rule for the hip joints which is also inspired by the vast body of literature on planar walking robots. This policy regulates the torso orientation while providing natural falling dynamics which is originally encoded in the stance hip joints of 3LP too. We believe a torque control paradigm can further increase compliance, but the available level of compliance in the physical SEAs of the robot was enough for our perturbed walking conditions.

In future, we would like to extend this framework to the lateral direction too by finding lateral foot-placement suggestions. Since dynamic equations are the same in 3LP for both sagittal and lateral planes, the same control look-up-tables can be applied in the lateral direction. However, one should carefully handle internal collisions which might happen in case of large lateral pushes. We hope to reduce the step width by an active control of lateral stability which can then help reducing the overall walking frequency and provide a more human-like walking. We also aim at upgrading our low-level joint controllers with feed-forward velocity and torque terms calculated by 3LP and the time-projecting controller. This can improve compliance and tracking delays. Finally, we consider improving low-level control-board issues, renewing springs and improving backlashes on COMAN to increase the voltage, to reduce mismatches and to improve sensory precision respectively. This can further enlarge the operational region of actuators and the overall controller. Hardware imperfections and possible model mismatches always exist, but dynamic foot placement is strong enough to stabilize the robot through hybrid walking phases. This motivates us to add more human-like features like CoM excursions, straight knees, and toe-off phases to achieve faster speeds in future work. This paper is accompanied with a video demonstrating walking and push recovery scenarios.

Application to Human Walking Part IV

9 Energetic Model

In the previous parts of the thesis, we discussed all the hardware experiments and robot control scenarios. This part of the thesis goes beyond robot control and uses our new walking model to analyze human walking energetics and kinematics. Remember from chapter 6 where we argued that the 3LP model can describe the trade-off between falling and swing dynamics. This energetic trade-off partially describes the optimal choice of speed-frequency relation in human walking. However, the 3LP model is missing important mechanical effects in the vertical direction which are necessary to reconstruct the entire energy landscape of human walking. In this chapter, we aim at building a generic walking cost model in which 3LP describes the cost of horizontal effects while additional components predict vertical costs. This cost model can predict the entire cost landscape of human walking and therefore, replicate the speed-frequency trend as well. In this chapter, we derive this cost model and demonstrate its application in a number of other walking conditions. While this chapter is mainly motivated by explaining the human cost landscape from a mechanical viewpoint, it gives us essential hints on the mechanical components missing in the 3LP model. Including these components may improve efficiency and control, but with the cost of possible nonlinearities. In the next chapter 10, we aim at a better understanding of human lower-limb kinematics by studying walking asymmetries and then in chapter 11, we propose a method to include the missing vertical effects in the 3LP model.

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji, Amy R. Wu, and Auke Jan Ijspeert. *"A simple model of mechanical effects to estimate metabolic cost of human walking."* Nature Scientific Reports, 8(1):10998, July 2018.

The first and second authors have equal contributions. The second author drafted the manuscript, and the first author helped draft the manuscript. The first and second authors carried out simulations and data analysis together.

9.1 Background

Energetic economy has been shown to have a large influence on human walking behavior. For example, at a given speed, humans tend to walk with a preferred step length that coincides with minimum metabolic cost (Bertram, 2005; Zarrugh et al., 1974). Despite the complexity of relating walking mechanics to energetic expenditure, past studies have determined important contributors towards the overall energetic cost of walking, such as the work performed during step-to-step transitions to redirect the center of mass (CoM) velocity and the cost of generating muscular force for body weight support and for leg swing (Donelan et al., 2002; Grabowski et al., 2005; Doke et al., 2005; Gottschall and Kram, 2005). To the best of our knowledge, however, no study to date has used walking mechanics to present a unified cost landscape that can predict metabolic cost under various walking conditions. Without understanding the major energetic contributions, it would be difficult to identify the energetic consequences of compensatory movement in abnormal gait or prescribe effective treatment. Likewise, reducing the metabolic cost of impaired walking towards normative levels may contribute towards the efficacy of prosthetic and orthotic devices (Ferris et al., 2007).

The metabolic cost of walking is the overall energy consumption from many different mechanisms in the body, including muscle dynamics, blood circulation, and aerobic processes (Workman and Armstrong, 1986). In human gait experiments, this cost is typically calculated from measurements of oxygen consumption and carbon dioxide production minus the basal metabolic rate of standing to yield net metabolic power (Brockway, 1987). Metabolic cost is conventionally expressed in two different ways: the metabolic energy consumed per unit of time (metabolic rate or power) or the metabolic energy consumed per unit of distance (Cost of Transport, CoT).

Several papers have tried to relate the metabolic cost of walking to walking mechanics. Despite a measurable energetic cost, the average mechanical work per stride during steady-state level walking is near zero. In early biomechanical studies, Saunders identified various geometric determinants of walking (e.g. torso rotation, lateral torso tilt, midstance knee flexion) and postulated that all these determinants are the result of body's best effort to flatten and smoothen the CoM trajectory, i.e. minimizing accelerations (Inman et al., 1953). To the contrary, numerous studies have identified the energetic benefits of non-flat pendular dynamics of the CoM due to potential energy and kinetic energy exchange (Cavagna et al., 1976). These simple inverted pendulum walking models demonstrate that no work is needed during each step. Rather it is the positive work performed to restore lost collisional work during the step-to-step transitions that dictates metabolic cost (Donelan et al., 2002). Other mechanisms, such as swing leg dynamics and torso balance, are also important contributors to energetic cost during walking. While leg swing can be explained by passive pendular dynamics (Mochon and McMahon, 1980) and thus is sometimes unmodeled in simple walking models (Srinivasan and Ruina, 2006), experimental studies have shown that it contributes approximately 10% of the net metabolic cost (Gottschall and Kram, 2005). Despite its relatively small weight, leg swing oscillations forced at frequencies above the natural frequency can have a significant energetic consequence due to muscle force production at the swing hip (Doke et al., 2005). Furthermore, due to pendular falling dynamics, the pelvis also has a considerable acceleration in walking direction. This requires the stance hip muscles to apply torques to the torso, which are comparable in magnitude to swing torques (Liu et al., 2008).

Modeling the energetic cost of walking is challenging due to the complexities of the musculoskeletal system and its intricate relationship with neural locomotor mechanisms (Ijspeert, 2014). Translating from chemical processes at the molecular level to muscle force production and metabolic consumption is nontrivial and difficult to measure. The indeterminate relation between muscle work and joint mechanical work further complicates biomechanical analysis and modelling. Despite these difficulties, both sophisticated neuromusculoskeletal models (e.g. (Anderson and Pandy, 2001)) and simple inverted pendulum models (e.g. (Kuo, 2001)) have shown that minimizing some form of metabolic cost of transport leads to patterns similar to those of healthy human gait. While the complexity of the former models precludes further insight, the simple walking models permit linear separation of cost factors, such as pendular and swing dynamics (Kuo, 2001). Experimental studies suggest that a large portion of the energetic cost of walking can be attributed to a few factors (e.g. $\approx 28\%$ for body weight support, $\approx 45\%$ for CoM work (Grabowski et al., 2005)), but it is still unclear if and how these determinants can be combined to predict metabolic cost under various walking conditions. We propose a simple metabolic cost model to provide meaningful estimates of the human metabolic rate under general walking conditions. First, to encapsulate walking dynamics in a simple manner, we utilized a 3D linear model that can describe major sagittal plane and frontal plane dynamics (i.e. pendular falling, swing and torso-balancing effects) over a wide range of walking speeds and step frequencies (refer to chapter 6). While this model can capture horizontal energetic contributors during normal walking, it cannot fully capture the empirical CoT data as a function of speed and step frequency from Bertram (Bertram, 2005). Therefore, we needed additional components to capture such unmodeled costs.

We demonstrate that the total energy expended by the body is largely determined by a linear combination of four main mechanical components (i.e. linear separation premise). These are (i) sagittal and frontal dynamics (termed “3LP dynamics”), (ii) CoM vertical velocity redirection, (iii) ground clearance, and (iv) body weight support. We chose these four components because previous studies have shown that they contribute greatly to the metabolic cost of normal walking. They were also the minimum number of components needed to reproduce Bertram’s cost surface, especially at the extreme locations. To scale mechanical work components to metabolic cost, we used both a constant efficiency of 25% and a variable efficiency dependent on step frequency.

After building the model with fits to Bertram’s data, we tested model validity and generality by replicating walking conditions from six other experiments in literature and comparing model predictions with reported energy measurements. The simulated experiments from literature were intended to investigate each cost component. CoM redirection cost was examined with step width studies (Donelan et al., 2001), and the swing component was evaluated with added mass to the leg (Browning and Kram, 2007).

We tested the cost of additional foot lift during swing (Wu and Kuo, 2016) to isolate and further assess the ground clearance component. Comparisons with simulated reduced gravity conditions (Farley and McMahon, 1992) further evaluated the effect of gravity on the vertical cost components. The data from flat walking experiments (Ortega and Farley, 2005) were used to isolate weight support component. Finally, to test how the model fares with varying anatomical properties, we also investigated the metabolic cost of walking for obese individuals (Browning and Kram, 2005).

While these experiments can induce any number of biomechanical changes, we were interested in the choice of step frequency within the experimentally-modified cost landscape. We chose step frequency over other gait parameters because we can use the speed-step length surface of our cost model to estimate the optimal step frequency from the known experimental walking speed. Therefore, the model evaluated the cost landscape to predict the optimal step frequency (with respect to minimizing CoT), without any knowledge of experimental data other than imposed test conditions of walking speed, experimental parameter change, and subject mass and height. We focused specifically on how well the model could predict reported trends by comparing coefficients from a linear or quadratic fit (depending on the original experiment). We also estimated the contributions of each of the four mechanical components and how they change under different walking conditions, insights which cannot be easily obtained from empirical testing.

9.2 Methods

Our cost model is composed of four mechanical components: sagittal and frontal dynamics, CoM velocity redirection, ground clearance, and body weight support (see equations and schematics in Figure 9.1, cost curves in Figure 9.3). The overall metabolic cost is composed of the energies of these mechanical effects, scaled by the inverse of muscle efficiencies (Equation (9.5)). We evaluate our cost model by comparing model predictions with experimental measurements from different walking conditions in literature (Figure 9.4).

9.2.1 Metabolic cost model

Model development and choice of components

The four components were determined as follows. We started with the 3LP model, our first component, which was developed in chapter 6. As a great simplification of human dynamics, we did not expect 3LP alone to be able to predict the empirical CoT, and we used Bertram's cost surface to conceptually help us identify most important components missing in 3LP. One advantage is that 3LP can describe pendular and swing dynamics together. This combination can naturally explain the trade-off between push-off and swing costs (Kuo, 2001) but seems insufficient to explain the overall cost of transport surface in different walking conditions (Bertram, 2005). In particular, 3LP does not account for changing vertical dynamics with large step lengths and does not demonstrate high CoT at slow speeds, as found in humans.

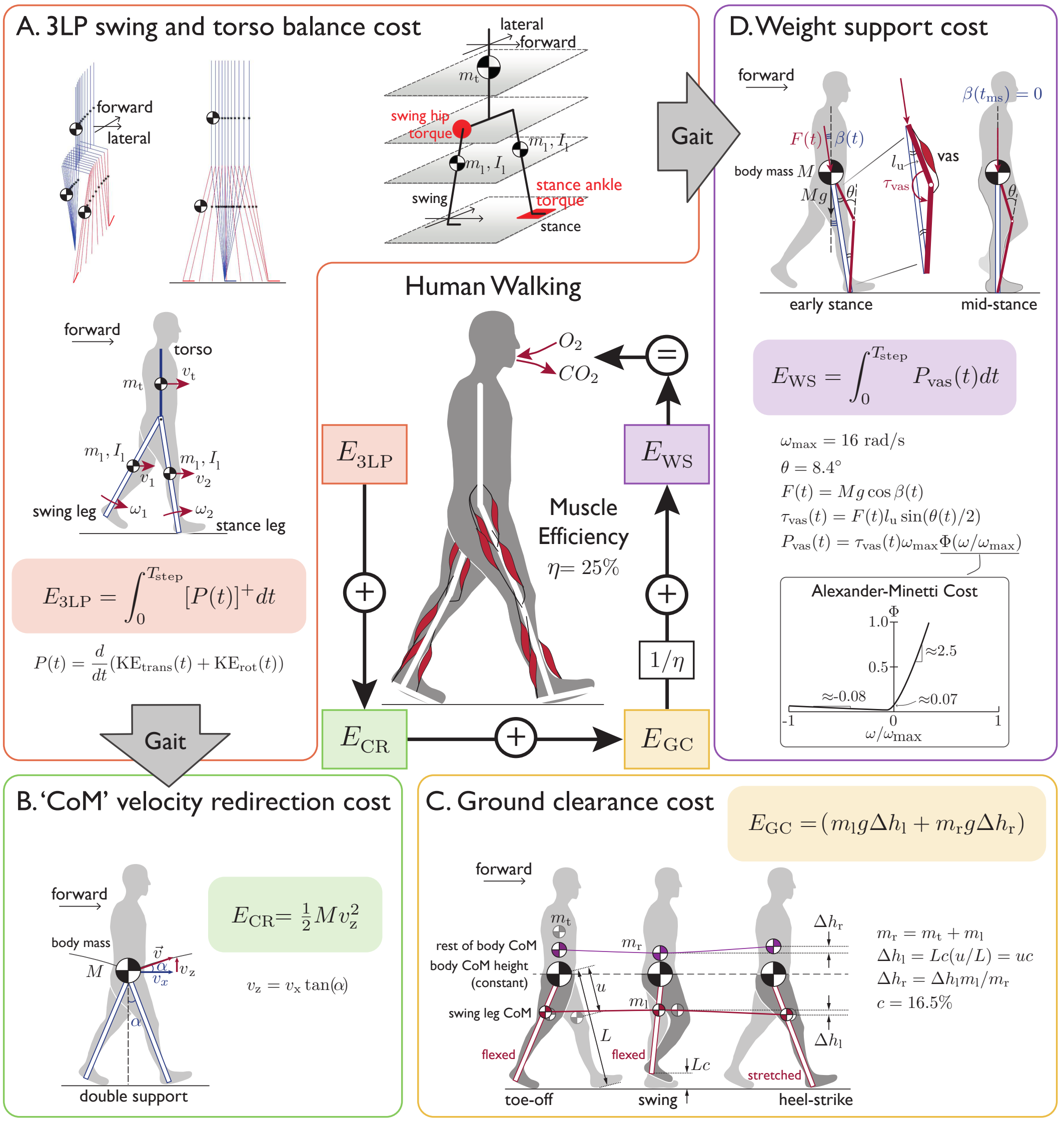


Figure 9.1 – The four energy components of the metabolic cost model and their formulations. The cost components are from (A) 3LP dynamics, (B) CoM velocity redirection, (C) ground clearance, and (D) weight support. (A) 3LP is composed of three linear pendulums (blue), two represent legs and one for trunk. Translational and rotational kinetic energies are calculated from the linear and angular velocities of each segment. The 3LP cost is the integral of the positive component of the kinetic energy change rate. (B) CoM velocity redirection cost accounts for the vertical work to change CoM velocity at the step-to-step transitions, which is not accounted for by the 3LP model. Similar to Kuo (Kuo, 2002), the magnitude of the velocity redirection, and thus kinetic energy, depends on geometry (i.e. the angle α between the legs). This angle comes from 3LP geometry (represented in blue). (C) Ground clearance cost is the potential energy to lift the leg. We used a constant c of 16.5% of leg length for lift height. Since the vertical CoM displacement must be constant, there is a corresponding penalty to move the 'rest of body' mass down. (D) Supporting the body during stance requires extensor muscular force to keep the leg from collapsing. The metabolic cost of the vasti muscles performing leg extension is calculated from the Alexander-Minetti curve (Alexander, 1997), following the work of Srinivasan (Srinivasan, 2011). The leg angle $\beta(t)$ is derived from 3LP geometry (blue), and we used a constant knee angle $\theta = 8.4$ deg.

This suggests that the model may be missing cost components, especially in the vertical direction, and we strived to add a minimal number of components to produce the cost surface. Vertical CoM excursions and the associated push-off cost seemed most relevant for the cost of greater step length. Leg lift could possibly explain the high CoT in slow speeds and less dynamic walking conditions (Ivanenko et al., 2002). The constant non-zero knee angle reported by (Zelik and Kuo, 2010) at different walking speeds, which could be related to the significant contribution of weight support on energetic cost (Grabowski et al., 2005), was not captured in any of the previous three components. Therefore, we added a fourth component to our cost model for body weight support. This component captures the metabolic cost of producing an isometric force at non-fully stretched knee angles. At very slow speed and frequency conditions, where the first three components together underestimates Bertram's CoT surface, this weight support component could explain the mismatch. We used both a constant efficiency and a variable efficiency (Massaad et al., 2007) to scale mechanical power to metabolic power.

The free model parameters are minimum stance knee angle, amount of ground clearance, center of pressure (CoP) profile, and muscle efficiency. Values for all four were determined by calculating the best fit to Bertram's data, within the constraint of existing data (see Figure 9.6). Best fit was determined by the smallest average p-value between model and human CoT across the entire cost surface. We found that constant values in the middle of reported ranges generally provided a good fit and thus used these values. However, we also found that a variable efficiency parameter had the best fit, especially as a function of frequency. We decided to show estimates for both constant efficiency and variable efficiency to demonstrate the effects of added complexity.

3LP dynamics: swing cost, torso balance cost

To generate gait as well as measure the swing and torso balancing dynamics, we needed a dynamic walking model that can describe active torques at the hip. We used the previously developed 3LP model of chapter 6 for this purpose. The 3LP swing and torso balance cost is the positive component of overall mechanical power over one simulated step of the 3LP model. Given step time T_{step} and translational and rotational kinetic energy KE_{trans} and KE_{rot}, the 3LP cost is calculated by:

$$E_{\text{3LP}} = \int_0^{T_{\text{step}}} [\frac{d}{dt}(KE_{\text{trans}} + KE_{\text{rot}})]^+ dt \tag{9.1}$$

This integration is actually done for variations of translational and rotational kinetic energies in the sagittal and lateral directions separately. There is no variation in the potential energy due to constant mass heights. The cost of all vertical motions is captured in the other three cost components.

CoM redirection cost

An immediate mechanical effect missing in 3LP is the consequences of pendular falling dynamics in the vertical direction. At the end of each step, negative work is performed to redirect the body CoM velocity, and positive work must be performed to recover collisional losses. Both in simulation and in human experiments, the ideal time is to provide a push-off force by the trailing leg right before collision (Kuo, 2002; Donelan et al., 2002). To address the vertical component of this push-off cost, we introduced the velocity redirection work, calculated using 3LP gait geometry. Considering the position of the model's legs at push-off and the pelvis horizontal velocity vector, we calculated an augmented 3D pelvis velocity vector, orthogonal to the leading leg. The energetic consequence E_{CR} was then calculated similarly to the method proposed in Kuo (Kuo, 2002) but using the vertical component of the velocity. Assuming horizontal pelvis velocity v_x at the push-off moment and attack angle α, both given by 3LP gait, the vertical CoM velocity change due to collisional impact is given by $v_z = v_x \tan(\alpha)$. The push-off work needed to compensate this loss is:

$$E_{\text{CR}} = \frac{1}{2} M v_z^2 \tag{9.2}$$

We used body mass M at the pelvis as a proxy for CoM. Only the vertical component was considered because the forward and lateral costs have already been included in the 3LP model.

Ground clearance cost

During the swing phase, humans nominally walk with a nonzero amount of leg lift, possibly to avoid foot scuffing or obstacles. The maximum toe lift over the entire swing phase is few centimeters (Winter, 1992), whereas during the swing initiation, the heel is already lifted due to the action of rolling on the toes and flexing the knee. Although an extra passive shank lift (due to the leg inertia) happens shortly after push-off, it does not increase the maximum heel height significantly (Ivanenko et al., 2002). We assumed prismatic legs which simplify the complex knee mechanism but provide a good approximation of leg CoM trajectories (Figure 9.1).

The cost of foot lift is partly attributable to mechanical work, which increases with lift height (Wu and Kuo, 2016). Therefore, to associate a cost to foot lift, we simply considered the mechanical work to lift the heel to a fixed maximum height c of 16.5% of leg length, the middle of the range of reported maximum heel lift heights over different speeds (Ivanenko et al., 2002) (Figure 9.6-2). Based on average anatomical data (de Leva, 1996), we calculated the vertical displacement of the leg's CoM accordingly and associated a potential energy cost which should be provided by the leg muscles. Assuming a heel lift of c (as percentage of leg length L), a mass m_l for each leg, and a leg CoM located u units below the hip joint, the leg CoM lifts $\Delta h_l = Lc(u/L) = uc$ units with a heel lift of c. The energetic consequence is:

$$E_{\text{GC}} = 2 m_l \Delta h_l g \tag{9.3}$$

where the gravity is denoted by g. While leg swing encompasses a cost to both swing the leg and clear the ground, 3LP already encodes the swing cost. The ground clearance cost simply accounts for the work to lift the leg vertically. Also note that the lift of the swing leg displaces the body CoM vertically, but vertical motions of the CoM were already considered in the previous cost components. We assume that during the lifting of the swing leg, the rest of the body moves in the opposite direction, in order to keep the CoM at the same vertical level and to avoid interference between cost components. Therefore, the cost E_{GC} is the sum of the mechanical work to lift the foot first and then, to lift the rest of the body back to the initial CoM height.

Weight support cost

During the stance phase of walking, leg extensor muscles must act to prevent the stance leg from collapsing under the weight of the body. This cost is not captured in the 3LP model and indeed not straightforward to calculate directly based on a mechanical work, especially since muscles are not ideal actuators and consume energy when applying forces isometrically. As a simple model of weight support, we calculated this cost from the knee torque required to maintain a constant knee angle $\theta = 8.4$ deg (Figure 9.6-1). This angle was derived from the minimum knee angle at mid-stance, which we observed to be relatively constant over a range of walking speeds (Zelik and Kuo, 2010). Calculated in a similar manner as Srinivasan (Srinivasan, 2011), we converted the knee torque to an isometric muscle force applied by vasti group muscles and then calculated the metabolic cost E_{WS} from this force production using muscle-specific parameters and Alexander-Minetti metabolic rate curves (Alexander, 1997) (see Figure 9.1). Assuming thigh length l_u, body mass M, gravity g, and stance leg angle with respect to gravity $\beta(t)$ (determined by 3LP gait geometry, see Figure 9.1), the torque required in the knee is approximated by $\tau_{\text{vas}}(t) = Mg\cos(\beta(t))l_u\sin(\frac{\theta(t)}{2})$, $\theta = 8.4$ deg is the knee angle assumed to be constant in the model. Given vasti group muscles' maximum rotational velocity $\omega_{\max}$ and the Alexander-Minetti's cost curve $\Phi(\frac{\omega}{\omega_{\max}})$, where $\omega(t)$ is the time derivative of $\theta(t)$, the weight support cost is calculated as:

$$E_{\text{WS}} = \int_0^{T_{\text{step}}} \tau_{\text{vast}}(t)\omega_{\max}\Phi(\frac{\omega(t)}{\omega_{\max}})dt \tag{9.4}$$

This simplified method neglects co-activation of the antagonist hamstring muscles during early stance. However, previous simulations have found that hamstring muscles contribute little to support the body (Anderson and Pandy, 2003).

Total cost: scaled by muscle efficiencies

We propose that the overall energetic cost can be approximated by the sum of all aforementioned costs. While the fourth cost E_{WS} already accounts for the conversion from mechanical energy to metabolic cost, the first three costs are expressed as mechanical work and need to be scaled by muscle efficiency η to be converted from positive mechanical work to metabolic input.

$$E_{\text{walking}} = (E_{\text{3LP}} + E_{\text{CR}} + E_{\text{GC}})/\eta + E_{\text{WS}} \tag{9.5}$$

For whole body behavior, muscle efficiency parameter η could vary widely depending on walking conditions (e.g. from approximately 20% to 33% at different speeds, (Figure 9.6-3, (Massaad et al., 2007)). We chose to apply $\eta = 25\%$, derived from isolated muscle (Heglund and Cavagna, 1987) and inclined walking (Margaria, 1968) studies and typically used in biomechanics studies (Donelan et al., 2002; Grabowski et al., 2005).

Variable muscle efficiency

Inverse dynamics calculations on recorded kinematic data of subjects walking at different speeds result in a variable overall muscle efficiency (Figure 9.6-2) when compared to actual oxygen measurements (Massaad et al., 2007). Since walking frequencies were not originally reported in (Massaad et al., 2007), we obtained them using the speed-frequency relations reported in (Snaterse et al., 2011). A variable efficiency function was then defined by interpolating muscle efficiencies reported by Massaad as a function of these frequencies. We recalculated the cost model (see Figure 9.3) by this variable efficiency function and observed a better match in different regions of the speed-frequency CoT surface, especially in normal walking conditions. Using speed as interpolation variable for efficiency instead of frequency only worked around the optimal walking regions, but perturbed the surface completely in other regions (see Figure 9.3).

Experiment Replication

Six experimental conditions were replicated in simulation with little modifications to the model. Step width (Donelan et al., 2001), added mass (Browning and Kram, 2007), extra foot lift (Wu and Kuo, 2016), and obesity (Browning and Kram, 2005) experiments were recreated by simply imposing the specific parameter varied in the study (step width, additional segment mass, foot lift height, and body mass, respectively). Simulated reduced gravity conditions (Farley and McMahon, 1992) was imposed by applying a constant upward force to the 3LP model and scaling half of the ground clearance and weight support costs by gravity reduction factor (see Equation (9.3) and Equation (9.4)). Note that the leg experiences full gravity (to be comparable with the actual experiment) while the other half of the body is vertically moving in reduced gravity conditions, when calculating the ground clearance cost for this particular case. For all experiments, the model was scaled by the average body mass and height of subjects participating in the experiments, and gaits were found at the experimentally imposed walking speeds.

In flat walking condition (Ortega and Farley, 2005), the CoM height was kept constant which, imposed by the constant CoM height trajectory of 3LP, required the knee angle to change with time. To calculate this knee angle trajectory, we solved a simple inverse kinematic problem between the fixed stance foot point on the ground and the pelvis location at each instance of time in 3LP. We superposed a 2-segment leg model composed of thigh and shank segments only. Since the pelvis height is constant in 3LP, these two segments can capture the peak knee angle difference between flat and normal walking conditions during stance phase (Ortega and Farley, 2005). Weight support cost was simply calculated by considering the force required for the new knee profile (see

Equation (9.4)).

Analysis

To evaluate the speed-step frequency predictions, we used a paired t-test to determine if the mean of the cost of transport at a given speed and step frequency pair was not statistically significantly different from the model's prediction (significance defined as $p < 0.05$). We estimated the cost of transport for each subject from Bertram's study using subject mass and height. The average and standard deviation of the p-values over all speed and frequency conditions are reported. Since $p \geq 0.05$ does not indicate similarity, we also calculated the 95% confidence interval at each reported speed-step frequency and evaluated whether model prediction was contained within this interval. For the other scenarios, we did not have access to individual subject data and therefore could not perform similar statistical tests. To assess the model under these conditions, we performed a linear fit of model estimates against measured data to determine how well they correlated. A trend of unity represented perfect agreement. Additionally, we fit a linear or quadratic curve to our model estimates, depending on the original fitting equation used in the respective papers (Table 9.2). We then compared the trend and offset values with those reported.

Given experimental condition and walking speed, the estimated metabolic cost was the minimum with respect to step frequency. This limits model predictions to walking conditions that do not overtly induce strong preferences for other objectives, such as safety or robustness. Model estimates (Figure 9.4) and optimal frequencies (Figure 9.2) are compared with the reported frequencies. Given the cost surface for different speeds and frequencies, optimal trends for speed constrained, frequency constrained and step-length constrained walking conditions (Bertram, 2005) (see Figure 9.5) were determined.

We also evaluated the sensitivity of our results to the model's four free parameters (mid-stance knee angle θ, max heel lift height c, muscle efficiency η, and center of pressure distance CoP), which were derived from existing measurements of human data (Figure 9.6). We repeated the replicated experiments using the minimum and maximum values reported for those parameters, instead of the average, and compared the modified variable efficiency predictions.

9.3 Results

Our model produced a metabolic cost surface by the superposition of four components, each with its own energetic penalty profile, over a range of speeds and step frequencies (shown as cost of transport in Figure 9.3). We observed from our model that the metabolic cost landscape is composed of energy tradeoffs. Walking at low speeds with high frequencies is penalized by leg lifting effects. Taking larger steps increases costs due to CoM redirection, as was found previously with simulation and empirical data in (Donelan et al., 2002). Walking at very slow speeds and slow frequencies is costly due to weight support.

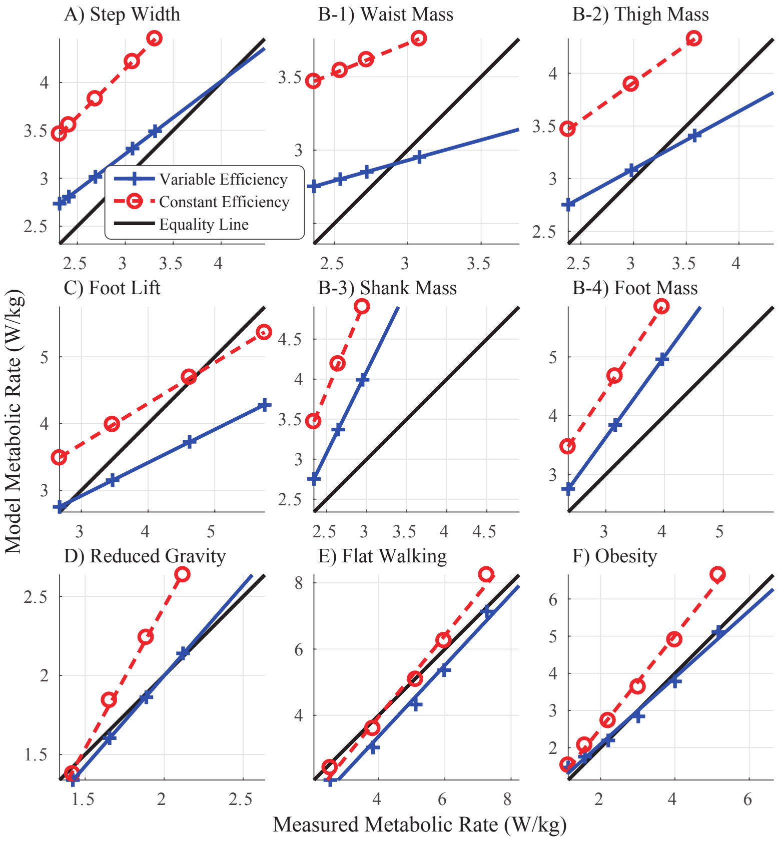


Figure 9.2 – Correlation between metabolic rates from model estimates and actual measurements for the six experiments under study. Model estimates (blue crosses for variable efficiency, red circles for constant efficiency) are plotted against empirical data, along with corresponding linear fits (solid blue for variable efficiency, dashed red for constant efficiency, see Table 9.1 for fit values). A slope of unity with zero bias means perfect agreement with empirical data.

We reproduced Bertram's speed-step frequency study in simulation and compared his measurements with the output of our cost model. As expected due to model fitting, the simulated cost surface corresponded well with empirical data (Figure 9.3), and a variable muscle efficiency term provided a better match. Model outcomes were not statistically significantly different from data for constant ($p = 0.152 \pm 0.234$, mean±s.d.) and variable efficiencies ($p = 0.377 \pm 0.273$, mean±s.d.). Model outcomes fell within the data's 95% confidence interval (47% for constant, 92% for variable efficiency), indicating that our linear separation premise could potentially provide an explanation for the real data. Model's optimal cost of transport was 2.13 J/kg/m at 1.03 m/s with a step frequency of 1.88Hz with variable efficiency. Using a constant efficiency yielded an optimal cost of transport of 2.62 J/kg/m (at 0.925m/s). For both gaits, approximately 50% of the costs were due to ground clearance. Without changing model parameters (other than the experimental variable), we were able to reproduce experimental data from six different

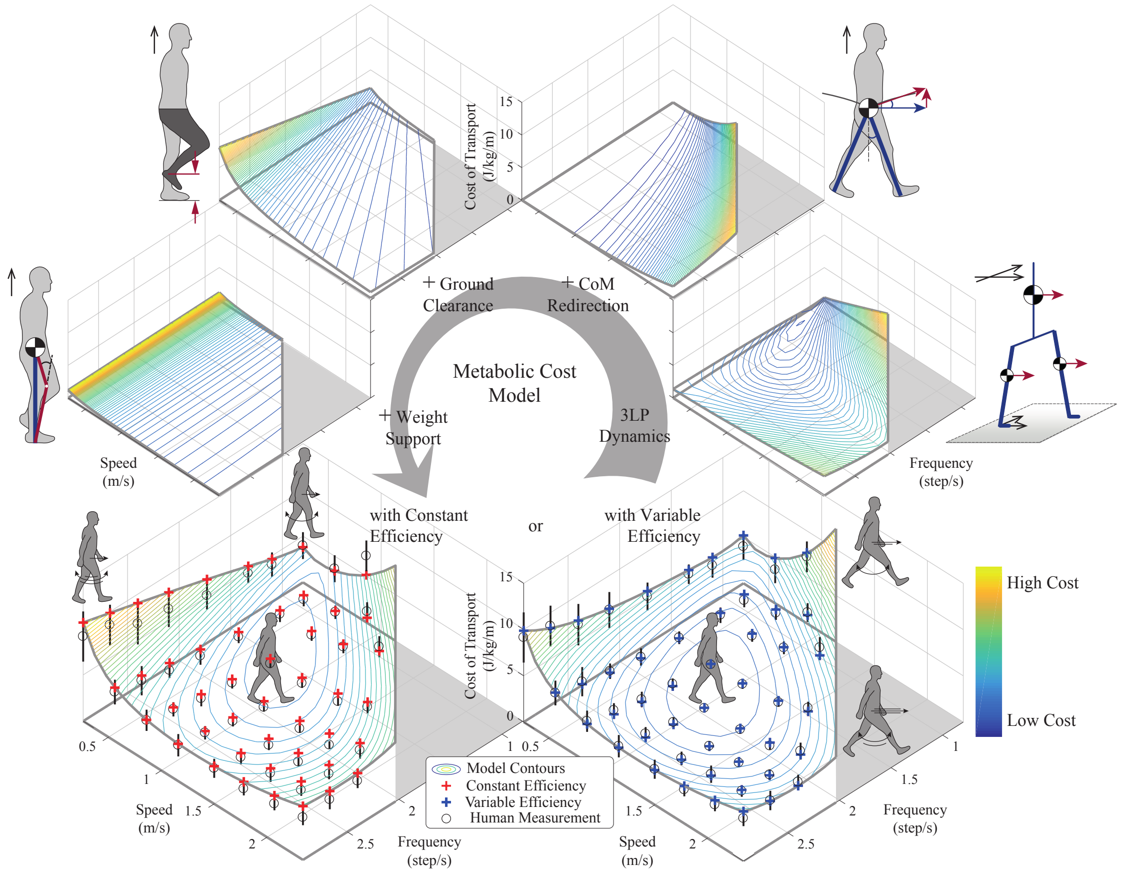


Figure 9.3 – The metabolic cost model and its four components, shown as cost of transport, at different walking speeds and step frequencies with experimental data reported in (Bertram, 2005) for comparison. The overall cost of transport is composed of the swing and torso cost from sagittal and frontal dynamics (3LP dynamics), CoM velocity redirection, ground clearance, and weight support costs. Each component is dominant at different speed-step frequency combinations. CoM redirection is costly at long step lengths, foot lift at slow speeds and high frequencies, and weight support at slow speeds. These components can be combined with constant muscle efficiency (red crosses) or variable efficiency (blue crosses) to yield costs more similar to experimental data (mean represented by black circles, standard deviation by vertical lines).

walking conditions and compare model estimates with empirical data (shown in Figure 9.4, quantified in Table 9.1 and Table 9.2). The model generally could estimate changes related to gait configurations (e.g. step width, flat walking) but not other changes (e.g. anatomical, added mass at the shank). The model was also better at estimating trends (metabolic cost as function of the experimental variable) than offsets.

9.3.1 Step width

To examine 3LP's ability to produce motions in the frontal plane, we compared model predictions against step width experimental data (Figure 9.4A). Like the original study,

Table 9.1 – Trends, offsets, and goodness-of-fit values from linear fitting of correlations between metabolic rates from model estimates and metabolic measurements. For the measurements, we used polynomial equations reported in the original papers since we did not have access to individual subject data. The goodness-of-fit values are close to one, indicating that model estimates follow similar polynomial trends found in the original papers. Both constant efficiency estimates and variable efficiency estimates are shown. A slope of unity with zero bias means perfect agreement with empirical data (see Figure 9.2) for visualization). If fits were not reported in the original paper (indicated by †), we performed fits on the empirical data for model comparison.

Experimental Parameter (x)	Trend a		Offset b		R^2	
	Variable	Constant	Variable	Constant	Variable	Constant
Step Width (m)	0.758	0.998	0.981	1.15	1.000	1.000
Added Mass: waist (kg)	0.278	0. 404	2.097	2.516	1.000	1.000
Added Mass: thigh (kg)	0.547	0.716	1.451	1.763	1.000	1.000
Added Mass: shank (kg)	2.038	2.36	-2.016	-2.053	1.000	1.000
Added Mass: foot (kg)	1.378	1.49	-0.504	-0.045	1.000	1.000
Extra foot lift (m)	0.497	0.611	1.425	1.858	1.000	1.000
Reduced Gravity (g)	1.145	1.796	-0.293	-1.157	1.000	0.998
Flat Walking † (m/s)	1.074	1.228	-0.936	-0.959	0.986	0.989
Obesity † (m/s)	0.897	1.25	0.303	0.021	0.989	0.995

estimated metabolic costs varied quadratically with step width ($R^2 = 1.000$ for both constant efficiency and variable efficiency). Metabolic cost matched empirical data fairly well in trend (0.758 for variable efficiency and 0.998 for constant efficiency estimates against metabolic data, where 1.000 indicates a good fit) but not in offset (0.981 and 1.150, variable efficiency and constant efficiency). Step width costs were dominated by increases in the 3LP and CoM velocity redirection costs.

9.3.2 Added mass on the leg

Additional mass was added to the model equivalent of a foot, shank, thigh, and waist to evaluate leg dynamics in the sagittal plane. These added masses led to a linear increase in metabolic cost ($R^2 = 1.000$). As also observed in the original experiment, added mass was more costly when placed at distal location than proximal ones (Figure 9.4B). Added mass to the foot increased metabolic cost the most while adding mass to the waist had the least effect. However, the model also overestimated the distal mass cost increase rates (shank: 2.038 and 2.360, foot: 1.378 and 1.490, variable efficiency and constant efficiency) and underestimated the proximal mass cost increase rates (thigh: 0.547 and 0.716, waist: 0.278 and 0.404, variable efficiency and constant efficiency). Swing dynamics (3LP) and ground clearance played a greater role with distally located added mass than proximal mass placement.

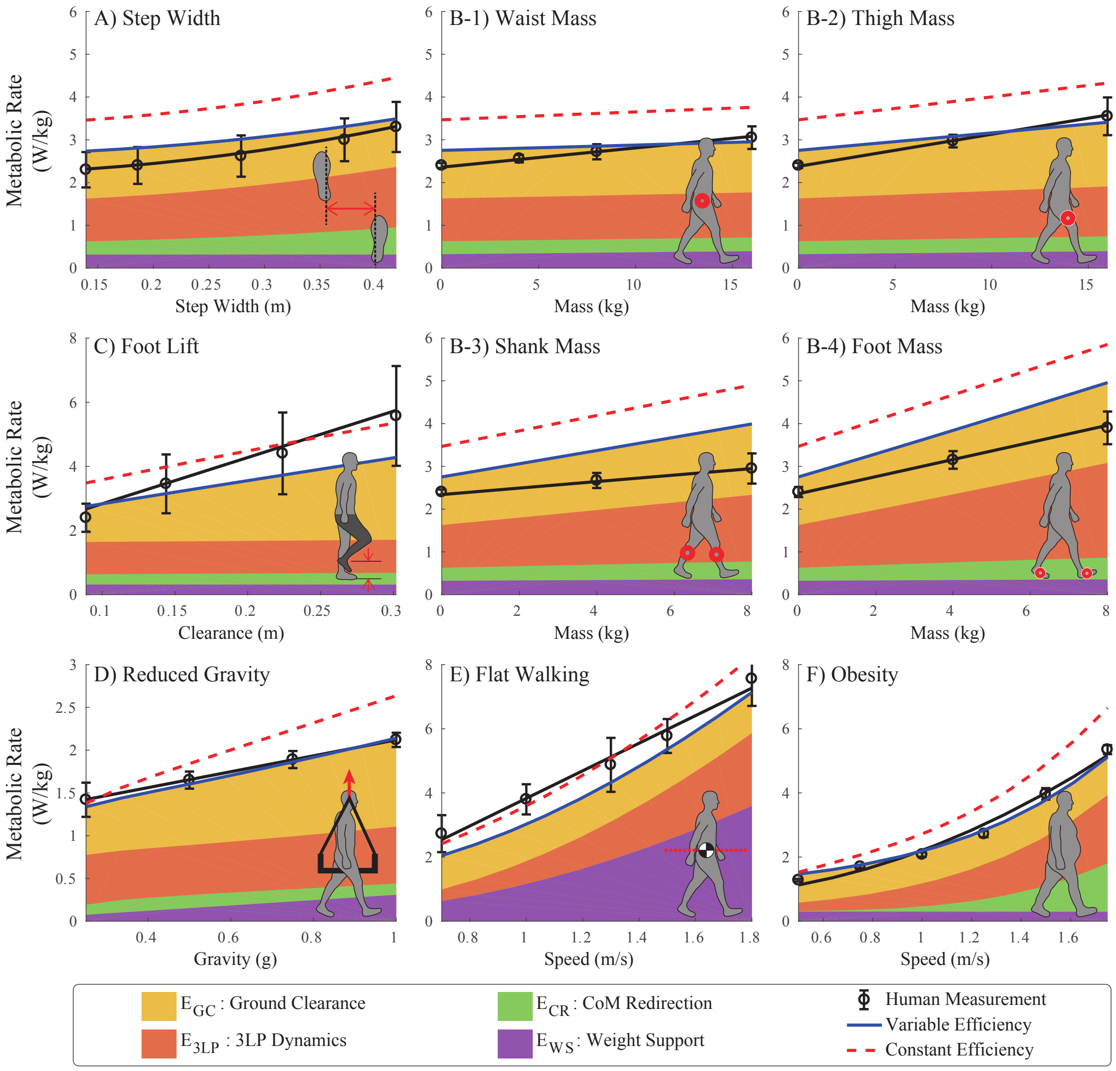


Figure 9.4 – Comparison of model metabolic rate (red lines) with data from six walking experiments from literature (black circles, solid lines). Model predictions include with variable efficiency (solid blue line) and with constant efficiency (dashed red line). The six comparisons were on A) step width (Donelan et al., 2001), B) added mass to the leg (Browning and Kram, 2007), C) extra foot lift (Wu and Kuo, 2016), D) simulated reduced gravity (Farley and McMahon, 1992), E) CoM flat-trajectory walking (Ortega and Farley, 2005), and F) walking with obesity (Browning and Kram, 2005). Fitting equations, from the original experiments when possible, were used to investigate trends (see Table 9.2). Patch layers represent the contribution of each cost component (yellow: ground clearance, orange: 3LP dynamics, green: CoM redirection, purple: weight support).

9.3.3 Extra swing foot lift

We applied various foot lift heights to estimate the metabolic cost of clearing the ground during swing. The model, predicting a linear increase ($R^2 = 1.000$), underestimated the metabolic cost in trend (0.497 for variable efficiency, 0.611 for constant efficiency) with an offset in magnitude (1.425 and 1.858, variable efficiency and constant efficiency). Not surprisingly, the leg lifting cost accounted for the majority of the increase with some contribution from swing dynamics (Figure 9.4C).

9.3.4 Simulated reduced gravity

We conducted two comparisons to further investigate the cost of body-weight support. For the first study, we investigated simulated reduced gravity, where a counter weight force is applied to the upper-body, not to the entire body as under true reduced gravity conditions. The model estimated a linear increase of energy with gravitational acceleration ($R^2 = 1.000$ for variable efficiency, $R^2 = 0.998$ for constant efficiency, Figure 9.4D). In trends, the variable efficiency estimate (1.145) provided a better match to empirical data than the constant efficiency (1.796). As gravity decreased, weight support cost reduced as expected (Farley and McMahon, 1992; Grabowski et al., 2005), along with CoM redirection and ground clearance. 3LP costs also indicated that in very low gravities, leg swing motion becomes costly again.

9.3.5 Flat-trajectory walking

The second evaluation for weight support cost replicated flat walking (i.e. minimal vertical CoM movement) studies, which demonstrated that reducing COM displacement does not lead to reduced energetic cost (Gordon et al., 2009; Ortega and Farley, 2005). Here we used experimental data from Ortega (Ortega and Farley, 2005). Here we varied the stance knee angle with simple inverse kinematics as a function of stride (see Methods) to model a constant CoM height trajectory. The cost of flat walking increased linearly with speed ($R^2 = 0.986$ for variable efficiency, $R^2 = 0.989$ for constant efficiency, Figure 9.4E). The cost model estimated the main trend reasonably well (1.074 for variable efficiency, 1.228 for constant efficiency) with some offsets (-0.936 and -0.959, variable efficiency and constant efficiency). We observed a substantial increase in weight support cost. Thus flat walking creates unfavorable muscle-related changes, which agrees with reports of increased muscle activation and co-contraction during flat walking (Massaad et al., 2007).

9.3.6 Walking with obesity

Finally, to investigate changing anatomical properties, we estimated the metabolic cost of walking for obese individuals. We did not expect to be able to reproduce this condition due to the simple scaling of body mass in our model. We found that this cost increased quadratically with speed ($R^2 = 0.989$ for variable efficiency, $R^2 = 0.995$ for constant efficiency, Figure 9.4F) but overestimated trends (0.897 for variable efficiency, 1.250 for constant efficiency). The majority of the cost increases were due to CoM redirection and 3LP dynamics.

9.4 Discussion

We sought a simple but unified model that could predict the metabolic cost of walking at various speed and step frequency combinations, as well as be generalizable to a range of different walking situations and anthropometric dimensions. We proposed that this could be achieved with a linear combination of four main components: the costs of swing and torso dynamics, CoM velocity redirection, ground clearance, and weight support. To test the model's linear separation premise, we used a combination of these components to reproduce empirical speed-step frequency data and then tested the model with six different experiments. Overall, this simple model was able to predict some of the energetic trends and magnitudes reported in biomechanically important experiments, demonstrating that linear combinations of these four components could constitute the main metabolic determinants of walking.

Model composition suggests that the optimal metabolic cost is moderated by tradeoffs among different component surfaces. This is comparable to the tradeoffs (Bertram, 2005; Kuo, 2001) producing the optimal speed-step frequency curve found in humans (Zarrugh and Radcliffe, 1978; Bertram and Ruina, 2001). As other studies have found, CoM redirection costs penalize longer step lengths (Donelan et al., 2002), and swing costs penalize fast step frequencies (Doke et al., 2005). The model also suggests that weight support costs are more prominent at slow walking speeds, and increasing the frequency at slow speeds incurs high ground clearance costs due to leg lifting. Our model estimates were similar to those of other energetic models and human studies aimed at decoupling the cost of walking. At the optimal (preferred) speed and step frequency, we found that 24% of the optimal cost can be attributed to swing and 76% to stance. In comparison with computational approaches, neuromuscular models with muscle models also estimated approximately 30% for swing and 70% for stance (Umberger, 2010). We found that swing costs increased with greater step frequency, to the contrary of Umberger's model (Umberger, 2010) but in agreement with experimental results from Doke (Doke et al., 2005).

The cost model was able to estimate energetic cost under various speed-step frequency combinations in both trend and magnitude. The model only differed at very large step lengths, where the linear decoupling of horizontal and vertical motions is weaker, due to larger CoM vertical excursions. The relative accurate predictions elsewhere imply that the four cost landscapes do encompass the overall energetics (Figure 9.3) and thus can be used to yield further insight. Our dimensionless cost of transport (energy over body weight and distance) at the model's preferred walking speed (1.03m/s) and step frequency (1.88Hz) was 0.217, a difference of 8.7% in comparison to the average experimental value of 0.2 using net metabolic cost at 1.25m/s (Wu and Kuo, 2016). More physiological models and energy calculations had similar error magnitudes. In comparison with net cost, Endo and Herr's model had $\approx 10\%$ error (Endo and Herr, 2014). Compared with human gross cost of transport of 0.3 (Wu and Kuo, 2016), sagittal plane walking models from Umberger et al. (Umberger et al., 2003) and Song and Geyer (Song and Geyer, 2012), who used the same muscle energy formulations, had approximately 15% error and 5% error, respectively. Our model did not greatly overestimate costs like other 3D models (e.g. by 63% (Anderson and Pandy, 2001)), and Roberts et al. (Roberts et al., 2016), who included measured kinematics and kinetics, had a 12% error.

We tested if the model could estimate metabolic cost of six experiments without knowledge of the step parameters chosen by subjects. Using only the reported subject mass, subject height, experimentally-fixed walking speed, and experimental variable, the model estimated an optimal step frequency (Figure 9.5) which was very close to the measured frequency in most of the six experiments. Interestingly, despite large differences between predicted and empirical step frequencies for the reduced gravity and flat walking condition, model cost estimates are not very different from measured costs, indicating that perhaps the model is not sensitive to the choice of step frequency. Since the model was fit to Bertram's CoT surface, we expected our model to be better at estimating the energetic consequences of changing speed and/or frequency. This is partially reflected in the flat walking and obesity experiments, where walking speed was the main experimental variable (see Figure 9.4). Model predictions in flat walking experiment were reasonable, likely also due to the use of the Alexander-Minetti curve for the weight-support component (Alexander, 1997). Predictions in increased step width and reduced gravity conditions were also relatively good, perhaps because energetic changes were small, and the model roughly stayed within linear regions. Of the six experiments, we did not expect the model to estimate the metabolic cost of obese walking. While the model obtained a decent fit, the obesity prediction was similar to those for normal walking. The model cannot differentiate well between obese and normal walking.

Added mass and extra foot lift experiments extend beyond linearity and decoupling assumptions, which may explain why model predictions failed. The human knee-ankle mechanism is also much more complex than in our model. We can attribute some of the estimation errors to the efficiency of muscles. For example, using a lower walking efficiency of 19.5% improves trend estimates for foot lift (Figure 9.6). However, with the 3LP component, it is unclear whether the errors are due to decoupling assumptions, muscle efficiencies, or some other unknown parameter that we did not consider. We found that prediction trends were relatively insensitive to the choice of free parameters—mid-stance knee angle θ, max heel lift height c, muscle efficiency η, and center of pressure CoP (Figure 9.6). Except for reduced gravity and flat walking, changes in knee angle mainly changed biases and not trends. Heel lift and center of pressure variations produced minor changes in cost estimation. Muscle efficiency seemed to have the largest influence, affecting the trend for normal subjects and obese subjects. Therefore our cost model is more robust to parameter variations in predicting trends, but less precise when estimating exact magnitudes. More accurate trends are arguably more important than magnitudes when estimating human energetic consumption because relative changes between nominal and new conditions can be detected.

Our cost model successfully demonstrates the speed-constrained optimization hypothesis (see Figure 9.5) proposed by Bertram (Bertram, 2005) as well as provides reasonable trend estimates of human energy expenditure under varying anatomy and

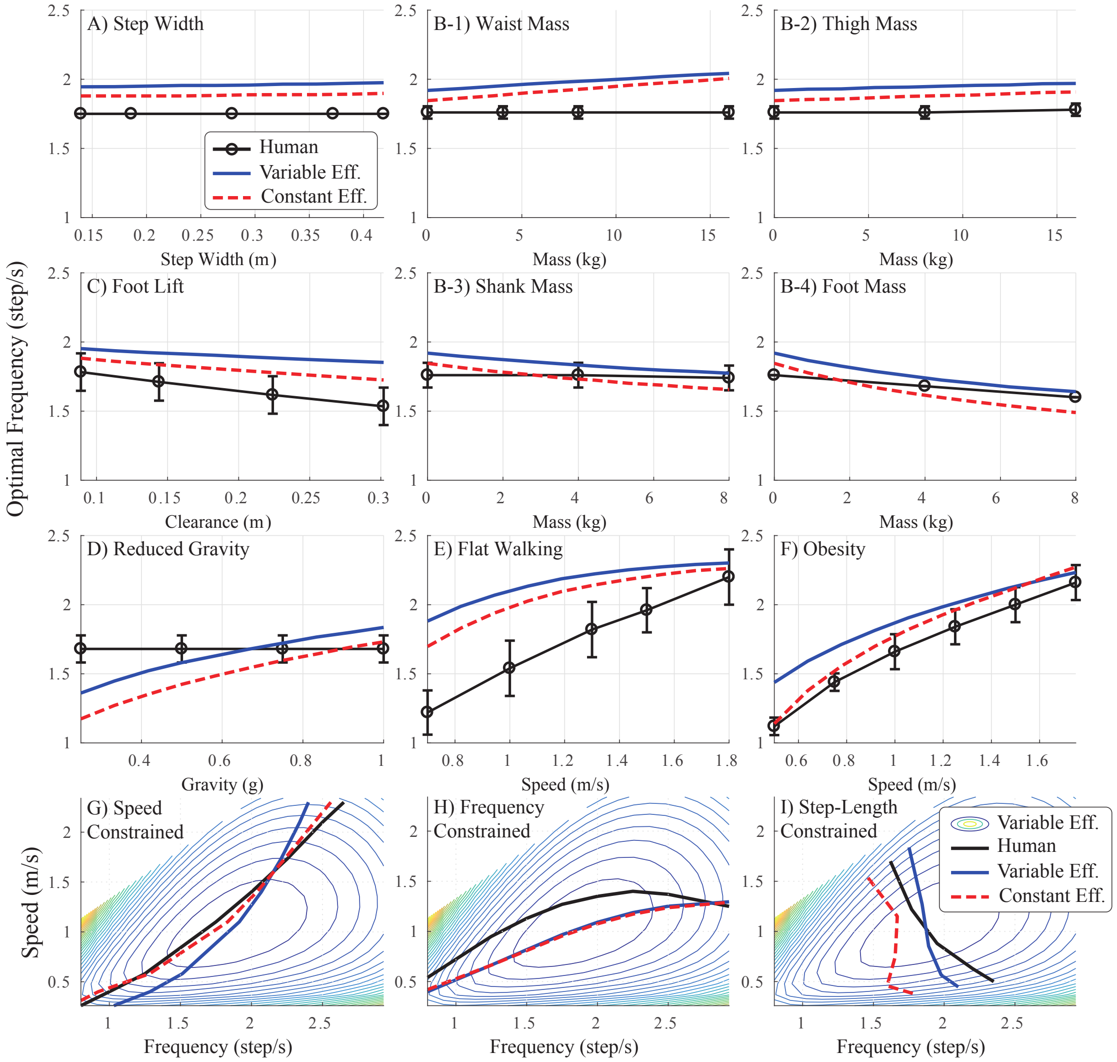


Figure 9.5 – (A-F) Comparison of experimental step frequencies and model-optimal frequencies by minimizing the cost of transport for each experiment and (G-I) predicted optimal trends by constraining different walking parameters (Bertram, 2005). We repeated optimizations for both constant and variable efficiencies. Increasing and decreasing trends are captured in all conditions despite a bias between the curves. The constant efficiency yields optimal frequencies typically closer to human. The variable efficiency seems unreliable, since there is no evidence for how it actually changes in different experimental conditions. Our model fails at predicting human frequency behavior in reduced gravity, flat walking, frequency-constrained and step-length constrained conditions. We consider muscle efficiencies to be the main reason behind such discrepancy.

walking conditions. The intrinsic power of this method comes from the separation principle—the decoupling of different phenomena in a linear fashion and studying each separately. Linear separation of energy-related components is not uncommon, as evidenced by both simple models (e.g. summation of push-off, hip actuation, swing leg costs in Kuo (Kuo, 2001)) and more complicated musculoskeletal models (e.g. addition of various heat rates and mechanical work in Anderson (Anderson and Pandy, 2001)). Linear separation implies more than an addition of energy costs, suggesting that the decoupling of highly complex and interconnected human locomotor functions could still encompass the major costs of walking. As is evident from the added mass experiment and extra lifting conditions, the vertical and sagittal dynamics can, to some extent, be separated. This may be similar to how the control of walking in the sagittal plane and the frontal plane can be considered separately, as demonstrated in simulations (Kuo, 1999) and in human experiments (Bauby and Kuo, 2000). Similarly, the separation of stance, swing, and balance control, with limited sensory exchange despite their inherent interconnectivity, has been shown to simplify gait coordination on robots and assistive wearable devices (Sharbafi et al., 2017).

The six validation experiments were chosen to isolate cost components. For example, foot lift experiments were to study foot-to-ground clearance and reduced gravity to study weight support. The swing cost in 3LP could also be isolated to some extent by the addition of distal masses to the legs. Due to gait geometry, however, vertical CoM redirection cost highly correlates with the horizontal falling dynamics in 3LP. Comparisons with the flat walking experiment attempted to cancel the vertical component but significantly increased the weight support cost. Additional experiments and analysis are needed to further separate each cost component. For example, a modified flat walking experiment with some weight support could possibly isolate the 3LP cost. We can further challenge 3LP and CoM redirection costs by investigating asymmetric walking gaits (e.g. on inclined terrains, with constant pulling forces or with extra torso bending). We can also investigate lateral swing dynamics in 3LP with swing foot circumduction experiments. These extra validations may require extensions of 3LP or experiments with human subjects, which we consider for future work.

Our use of mechanical measures to estimate metabolic cost is limited by their rather abstruse relationship. Metabolic cost can be incurred without net mechanical energy, such as during cyclic locomotion or muscle co-contraction. Observed mechanical work at a single joint could entail not only positive muscular work, but also contributions from elastic tendons or bi-articular muscles, which act across multiple joints. Positive and negative muscle work also contribute differently towards metabolic cost (Margaria, 1968). Thus it is unsurprising that the ratio of whole body metabolic cost to mechanical work can vary widely depending on walking condition (Massaad et al., 2007; Williams, 1985). Nonetheless, simple mechanical models and biomechanical experiments have shown that mechanical measures could largely account for changes in metabolic cost (Donelan et al., 2002; Tzu-wei and Kuo, 2014).

We were positively surprised by favorable energetic trend predictions. Not surprisingly, there were also inaccuracies in estimated magnitudes. Multi-segment leg motions and internal muscle properties were highly simplified. Telescoping actuators, meant to reproduce knee-ankle energy pumping and absorption mechanisms, are not physiological but can produce human-like pendular dynamics in the sagittal plane. Passive coupling between knee flexion and leg swing is also missing, which implies that the hip actuator contributes more to swing the leg. Unmodeled changes in leg inertia during knee flexion could also explain trend differences for the added shank mass and extra foot lift experiments. Indeed, we have omitted several features of human walking, including a non-infinitesimal double-support phase, a non-constant muscular efficiency, arm and transversal pelvic motions, and more anthropometric features. For example, while 3LP does include the mechanical work to balance the torso, our simplified upper body model neglects upper body angular momentum with no arms and a torso that remains vertical with respect to gravity. We believe these missing features do contribute to the observed differences between estimated and empirical data in some of the experiments, especially the use of muscular force to regulate whole body angular momentum (Herr and Popovic, 2008; Neptune and McGowan, 2011). Arm dynamics, for instance, affect metabolic cost rates, with an increase of 26% if swinging anti-normally (Collins et al., 2009).

More complex (e.g. nonlinear) or physiologically complete models (e.g. neuromuscular, multi-segmental model) could provide more realistic predictions in different walking conditions. For example, this model uses whole body mechanical work to capture the work performed by the muscles, instead summing energy consumptions at the muscle level (Wang et al., 2012). Accounting for muscle dynamics could provide better estimates and predict muscle-related effects that our model cannot capture. Additional features such as muscle co-contraction and realistic mass distribution could provide a better energy estimate in experiments, such as obesity, but might overfit the general CoT surface. More complicated models may also require potentially time-consuming optimization routines contending with more tuning parameters and appropriate objective functions to find periodic gaits.

Here we have proposed a minimalistic model to capture main trends in the CoT curve. The proposed cost model is based on a linear walking model, for which periodic gaits can be easily found. Such computational advantage makes our model suitable for prediction of transient walking conditions, such as accelerations and decelerations in walking speed. The proposed cost model can also be easily tailored to subject height, leg length and pelvis width. Its effectiveness in estimating changes with body weight (due to obesity results) requires further investigation. By decomposing the overall cost landscape into different components, our model suggests the dominant physical effects of different walking conditions. Quantifying these components for some gait condition can be performed empirically and would require clever but possibly laborious experimental procedures and apparatuses. Here separate experiments are not needed to study the effect of each component. While we acknowledge the difficulty in translating additional gait assistance to reduction of metabolic cost, the resulting decomposition could still help physiotherapists or biomechanists improve assistance or promote rehabilitation by targeting components that contribute the most towards whole body measurements, such as metabolic power. Thus this simple cost model creates insights not easily obtainable in human experiments and potentially valuable towards improving or augmenting human performance.

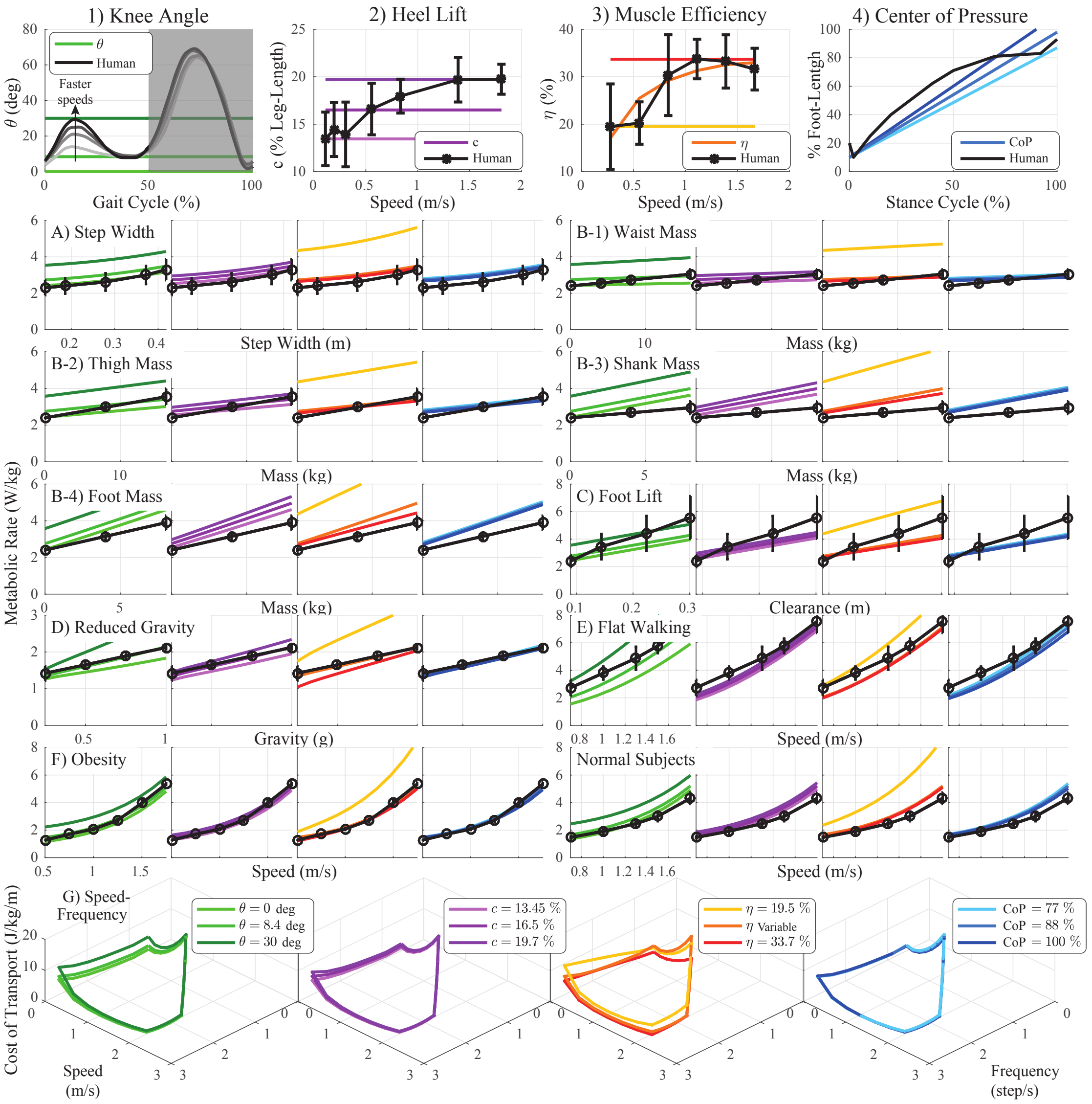


Figure 9.6 – The choice of free parameters in our model based on human measurements and the sensitivity of the variable efficiency results to parameter changes. (Top Row) Model parameters were based on (1) knee angle trajectories (Zelik and Kuo, 2010) at speeds ranging from 0.9m/s to 2m/s, (2) heel lift heights (Ivanenko et al., 2002), (3) overall muscle efficiencies (Massaad et al., 2007), and center of pressure (Browne, 2016). Parameter variations about the nominal value varied from the maximum (darkest line) to minimum (lightest line). (A-F) Sensitivity of metabolic predictions in the six experiments (nominal result shown by middle curves). Using the maximum parameter value (darkest line) to the minimum value (lightest line) produced variation mostly in magnitude and less in trend. Except for muscle efficiencies, parameter variations led to offsets in magnitudes but not changes in trends.

Table 9.2 – Comparison of trends, offsets, and goodness-of-fit from curve fitting of the original energy measurements reported in the studied experiments against our model predictions. Both variable efficiency and constant efficiency estimates are shown, parenthetically for the latter. Trend a is in units of W/kg over the experimental variable, which is linear or quadratic depending on the fitting equation. Offset b is in W/kg. Model R^2 values indicate the goodness-of-fit of predictions to fitting equation. Unless reported otherwise, experimental values were derived from the referenced paper (first column). If fits were not reported in the original paper (indicated by †), we performed fits on the empirical data for model comparison. Equations used for the step width and obesity experiments are quadratic of the form $ax^2 + b$. For other experiments, we used a linear equation of the form $ax + b$. N/A, not available.

Experimental Parameter (x)	Trend a		Offset b		R^2	
	Model	Human	Model	Human	Model	Human
Step Width (m)	4.848 (6.382)	6.4	2.639 (3.333)	2.19	1.000 (1.000)	0.91
Added Mass: waist (kg)	0.012 (0.018)	0.045	2.752 (3.469)	2.36	1.000 (1.000)	0.65
Added Mass: thigh (kg)	0.041 (0.054)	0.075	2.752 (3.466)	2.38	1.000 (1.000)	0.72
Added Mass: shank (kg)	0.155 (0.179)	0.076	2.752 (3.470)	2.34	1.000 (1.000)	0.61
Added Mass: foot (kg)	0.276 (0.298)	0.2	2.743 (3.476)	2.36	1.000 (1.000)	0.85
Extra foot lift (m)	7.173 (8.817)	14.43	2.118 (2.710)	1.39	1.000 (1.000)	0.81
Reduced Gravity (g)	1.054 (1.646)	0.93	1.078 (1.003)	1.19	1.000 (0.998)	0.55
Flat Walking † (m/s)	4.643 (5.299)	4.299	-1.512 (-1.600)	-0.468	0.988 (0.990)	0.984
Obesity † (m/s)	1.273 (1.774)	1.437	0.991 (0.992)	0.77	0.989 (0.995)	0.541

10 Walking Asymmetries

As mentioned earlier, the focal point of this thesis is the 3LP model which contributes to different chapters summarized in Figure 1.2. 3LP is a walking model purely encoding falling, swing, and torso balancing dynamics. This mechanical model is scalable in different dimensions and can be used to model and analyze human walking as well. 3LP, of course, does not include all geometric and mechanical details, but it can approximately describe important dynamics and energetics with acceptable precision (refer to chapters 6 and 9). In this chapter, we study lower-limb kinematics in human walking while approximating the overall dynamics by the 3LP model. When rotating the torso considerably or walking on inclined surfaces, the triangular coordination of the two legs becomes less symmetric. This is due to extra forces applied on the body in different locations which could be easily modeled in the 3LP equations. We study these effects and perform a comparison between human data and the 3LP model. This chapter continues the comparisons of chapter 6 to explore further kinematic similarities. It provides a useful explanation of how asymmetries in walking can compensate each other. Also, it gives us insights on how the lower-limb joints adapt to the overall asymmetries and how the vertical motions are produced. We use these insights in the next chapter 11 and propose a human-like walking trajectory generator based on the 3LP model.

Publication Note: The material presented in this chapter is adopted from:

- Amy Roning Wu, Salman Faraji and Auke Jan Ijspeert. *"Effects of trunk lean and ground slope on leg angle asymmetry during human gait."* in preparation, 2018.

The first and second authors conceived of the study, designed the study, and conducted the experiments. The first author drafted the manuscript, and the second author helped draft the manuscript. The second author carried out the data analysis, and the first author helped with the data analysis.

10.1 Background

The human locomotor system is capable of performing stable bipedal walking in various conditions. Humans can carry heavy backpacks (Caron et al., 2013), negotiate inclined grounds (Kawamura et al., 1991), pull objects when walking (Lee et al., 1991), and bend our trunk to reduce height in low-seam coal mines (Gallagher et al., 2011). The complex neuromuscular system tolerates these additional constraints and compensates for them to achieve stable locomotion. Sometimes these constraints can be internal as well as external. Muscle length and joint limits in hip extension can change gait kinematics and dynamics at faster walking speeds (Kerrigan et al., 1998) or on steep inclines (Kozma et al., 2018). In these conditions, once the constraint is reached, the locomotor system has to adjust other joint angles to maintain walking stability. For example, subjects with lack of hip extension compensate with excessive lumber extension or plantar flexion (Godges et al., 1993). Considering single-mass models for bipedal walking (Hemami et al., 1973), a stable gait may have symmetric kinematics in level walking. More specifically, the relative position of leading and trailing feet with respect to the Center of Mass (CoM) is similar during (instantaneous) double support (Kuo et al., 2005) (as demonstrated in Figure 10.1A). This effectively keeps the average CoM position on top of stance foot in periodic walking (Lugade et al., 2011).

Bipedal walking stability is easy to understand theoretically with simple models. However, given the kinematic and dynamic complexities of lower-limbs in human, it remains hard to extend simple theories to these complex systems. In unusual walking conditions such as fast speeds, stooping, inclined walking or external interactions, the hip, knee and ankle joints follow different trajectories and sometimes apply considerably larger forces to the system to maintain stability. Despite such complexities, the CoM to ankle vector remains the same in different backpack loading conditions for example, suggesting dynamic stability (Caron et al., 2013). McIntosh et al. also report that the pelvis-foot angle is proportional to slope in inclined walking (McIntosh et al., 2006). For subjects that have a limited range of motion in some joints (for example limited hip in elderly, or limited ankle in amputees), other joints in the leg have to increase their range of motion to compensate and keep the pelvis-foot angle the same (McIntosh et al., 2006). These compensatory mechanisms are very interesting because of their crucial roles and applications in control of bipedal robots (Pratt et al., 2001) and powered lower-limb assistive devices (Sawicki and Ferris, 2009). While individual limbs can be approximated by spring-damper systems in symmetric and asymmetric walking conditions (Aminiaghdam et al., 2017), understanding the interactions between them remains difficult in the joint-level.

In stoop walking for example, since the torso is always tilted forward, the hip joints apply large torques to prevent the trunk from falling (Gallagher et al., 2011). In consequence, these torques change the geometry of stance leg by increasing the attack angle at heel-strike. Because of such asymmetries, the muscles have to operate outside their usual functional regions, which is costly. A 20% height decrease in stoop walking increases the gross metabolic cost by 31-49%, while a 40% height decrease results in a

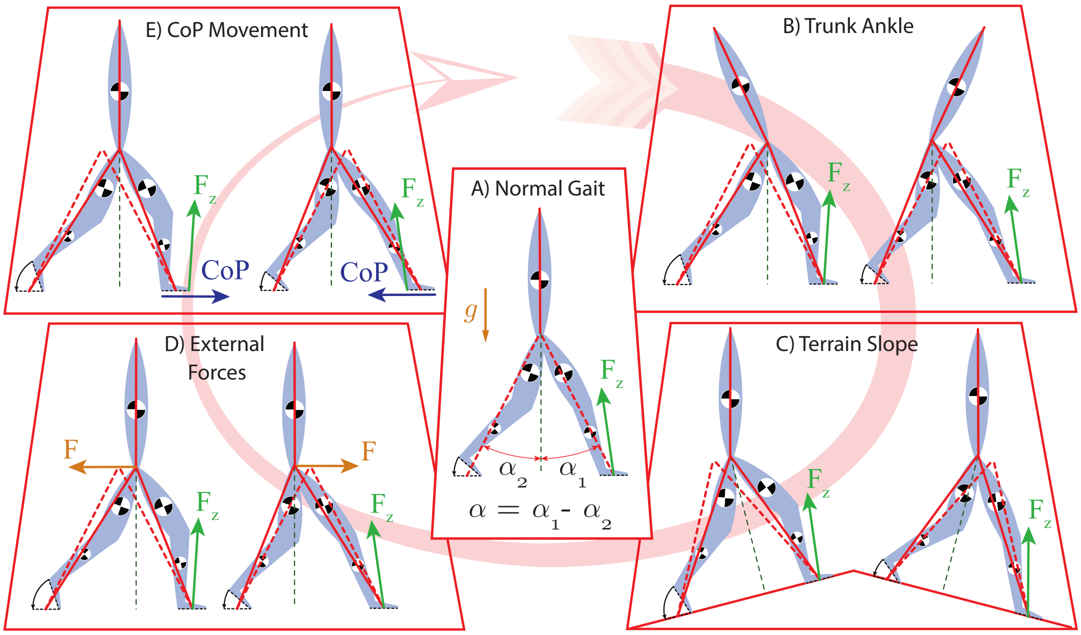


Figure 10.1 – Possible sources of asymmetries in walking. A) In normal walking conditions, the triangle formed by geometric center of the foot pad and the pelvis is almost symmetric at phase transitions. Symmetry index α is defined as the subtraction of individual leg angles α_1 and α_2 with respect to the vertical line (orthogonal to the ground). B) Tilting the trunk forward or backward (e.g., in stoop walking) can shift the pelvis position to keep the CoM roughly unchanged to maintain balance. C) Inclined walking and gravitational pulling and pushing forces along the slope can induce similar shifts. D) Likewise, external forces can induce the same effect in level walking. E) While it is uncomfortable, humans can walk on the heels and toes only, which shifts the CoP and the pelvis accordingly. By combining these asymmetry sources, one can reduce the overall asymmetry and get close to symmetric normal walking conditions. In each block, the left character shifts the pelvis forward while the right character shifts it backward. The dashed triangle is equivalent to the one of normal walking.

drastic increase of energy up to 350% (Bedford and Warner, 1955). Besides, the knees are also more flexed in stoop walking compared to normal walking (Gallagher et al., 2011), although the level of vertical CoM excursions are found to be the same in both conditions (Saha et al., 2008). Possible reasons could be gaze stabilization or avoiding large fluctuations in the overall potential energy which might be costly for the muscles again (Saha et al., 2008). In inclined walking, on the other hand, there is more stress on the knee joints to bear the weight (Leroux et al., 2002). The hip muscles work a lot in uphill and the ankles absorbs a lot of energy in downhill (McIntosh et al., 2006). Due to these complex mechanisms and various other factors in inclined walking, head stabilization becomes difficult (Cromwell, 2003), and gait stabilization becomes more challenging evident by Center of Pressure (CoP) variations (Kawamura et al., 1991), and all spatio-temporal parameters of walking and Ground Reaction Forces (GRF) change as well (McIntosh et al., 2006). In consequence, extreme uphill and downhill locomotion has different energetics from level walking.

In unusual conditions like stooping or inclined walking, as mentioned earlier, different sources of asymmetry result in different limb functionalities. In stooping (Figure 10.1B), the large hip torques required to keep the trunk fixed induce reactive torques in the limbs (Müller et al., 2017). In inclined walking (Figure 10.1C), because of the slope angle, the gravity accelerates or decelerates the motion as if an external force (Figure 10.1D) was applied to the body along the locomotion direction (Kawamura et al., 1991). Likewise, walking on the toes or on the heels (Figure 10.1E) can change limb geometries whereas in normal walking (Figure 10.1A), the CoP naturally moves forward from the heel to the toes (Francis et al., 2013). However, in knee-flexed or trunk-flexed walking conditions (stooping), an increased level of activity in gastrocnemius lateralis muscles is reported (Grasso et al., 2000). This indicates that the CoP moves faster towards the toes during the stance phase (Kluger et al., 2014). A similar effect is observed in downhill walking, although the CoP behavior is almost normal in uphill walking (McIntosh et al., 2006). Figure 10.1 summarizes some asymmetry sources and their geometrical consequences.

While the musculoskeletal system is very complex, we can still include few more details to single-mass models and achieve a much better understanding of dynamic walking geometries. Bipedal systems can be simplified by three segments as well: two lower-limb and one upper-body segments attached together through the pelvis. These three segments might by rigid in length (Gomes and Ruina, 2011), elastic like a spring (Maufroy et al., 2011) or stretchable with prismatic actuators (Hasaneini et al., 2013). These models have certain benefits and drawbacks, but they all share symmetric leg coordinations in periodic walking. Favoring faster computation properties, we developed 3LP, a linear model with three pendulums to represent the trunk, swing and stance leg segments in chapter 6. This model can easily simulate compass walking gaits (without feet) to achieve symmetric kinematics. However, it is possible to apply ankle and hip torques as well to modulate the motion. We can also tilt the trunk or gravity to simulate stooping or inclined walking conditions. The 3LP model simplifies limb mechanisms with simple prismatic actuators and keeps the CoM height constant (to achieve linearity), yet it can generate meaningful walking dynamics.

To simulate asymmetric walking conditions, Spring Loaded Inverted Pendulum (SLIP) models are predominantly used in literature and produce more realistic GRF profiles than the 3LP model (Iida et al., 2009). Extensions of SLIP models assume inertia for the trunk and allow for hip torques to balance it through Virtual Pivot Points (VPP) control (Maus et al., 2010). VPP refers to a point above CoM at which the GRF vectors roughly intersect. Andrada et al. observed VPPs in the GRF profiles of quails and used extended SLIP models to replicate similar kinematics (Andrada et al., 2014). The work of Andrada et al. is interesting in the sense that in quails like many other birds, the trunk is titled for almost 90 deg. In human stoop walking, therefore, similar VPP points exist (Müller et al., 2017), and similar extended SLIP models could be applicable (Aminiaghdam et al., 2017). Compared to these extended SLIP models, our 3LP model can simulate swing dynamics, inclined walking and CoP modulation in addition. Therefore, it seems a better candidate to model various asymmetric walking conditions of Figure 10.1 within the range of parameters where the underlying linearity assumptions remain roughly valid (step lengths not larger than the leg length).

Three configurations could provide a forward bias of the CoM to aid in asymmetric walking conditions. First, the trunk could lean forward but maintain symmetry between the legs. Second, the trunk could remain aligned with gravity atop of a CoM shifted forward (i.e. anteriorly shifted pelvis) relative to the feet. A third option would be combination of trunk flexion and asymmetry. We hypothesize that this combined strategy is employed by humans and that asymmetry is a function of speed, inclination angle, and trunk angle, possibly with opposing contributions. We use the 3LP model to simulate mechanical interactions of joint-level mechanisms in asymmetric walking conditions –fast speeds, torso flexion and inclined walking– at a more abstract level. By quantifying asymmetry as the triangle formed by the two feet and the pelvis (see Figure 10.1), we compare human gaits with our model at different walking speeds, trunk angles and ground inclinations. This study potentially quantifies the interplay between asymmetry sources and mechanically explains why some asymmetries can compensate for each other.

10.2 Methods

This section describes our mechanical model, human experiment protocols, and analysis procedures. We refer to chapter 6 for derivation of 3LP walking model equations and details. We determined asymmetry from the model walking at different speeds, frequencies, inclination angles, trunk angles, step width and double support times. These parameters as well as anatomical properties are taken from human walking experiments described next. The model therefore finds trajectories and inter-leg postures that satisfy periodicity and dynamic balance. Our definition of asymmetry index α reflects the shift in the triangle formed by the two legs and the pelvis during the middle of double support (see Figure 10.1). In case of symmetric triangles like Figure 10.1A, the index α is zero while shifting forward and backward makes it negative and positive respectively. Therefore, α is a signed variable with our definition.

10.2.1 Experiment

To compare model with human data, we measured the inter-leg asymmetry at different trunk angles, ground slopes, and walking speeds from healthy, young adults subjects walking on a treadmill. Asymmetry α was defined as the difference between the angle of the trailing leg and the angle of the leading leg relative to a vector orthogonal to the walking surface during double support (Figure 10.2B). Positive asymmetry values mean the inter-leg angle is biased posteriorly, and negative asymmetry values indicate an anterior bias (Figure 10.1). Prescribed trunk angles were imposed by real-time visual feedback of trunk angle measured through an inertia measurement unit (Xsens MTi-30, Enschede, Netherlands). Ground slope and walking speed were set by the treadmill (Forcelink N-Mill, Culemborg, Netherlands). Five subjects ($N = 5$, 3 male, 2 female) walked on three slope grades with three trunk angles and three walking speeds. The trunk angles imposed were 0 deg, 15 deg, and 30 deg relative to gravity. The slopes tested were 0%, 7%, and 14% grade, corresponding to 0 deg, 4 deg, and 8 deg, respectively. The speeds were 2 km/hr (0.56 m/s), 4 km/hr (1.11 m/s), and 6 km/hr (1.67 m/s), yielding a set of slow, normal, and fast speeds. Subjects walked with a combination of these trials. Subjects also performed an additional trials of normal walking at each slope and speed (with self-selected trunk angles). Each of the 36 trials were 2 min in duration and performed in randomized order. To determine the maximum hip extension angle, we also included a static condition where subjects placed their feet at a self-selected distance apart and leaned their trunk back as far as possible. Subjects' age ranged from 25 to 29 years of age and had body masses M of 68.1 ± 14.6 kg (mean±s.d.) and leg length L of 0.925 ± 0.073 m (measured from the greater trochanter to the ground during standing). All subjects were healthy and had no known gait impairments. All subjects provided written informed consent according to Institutional Review Board procedures.

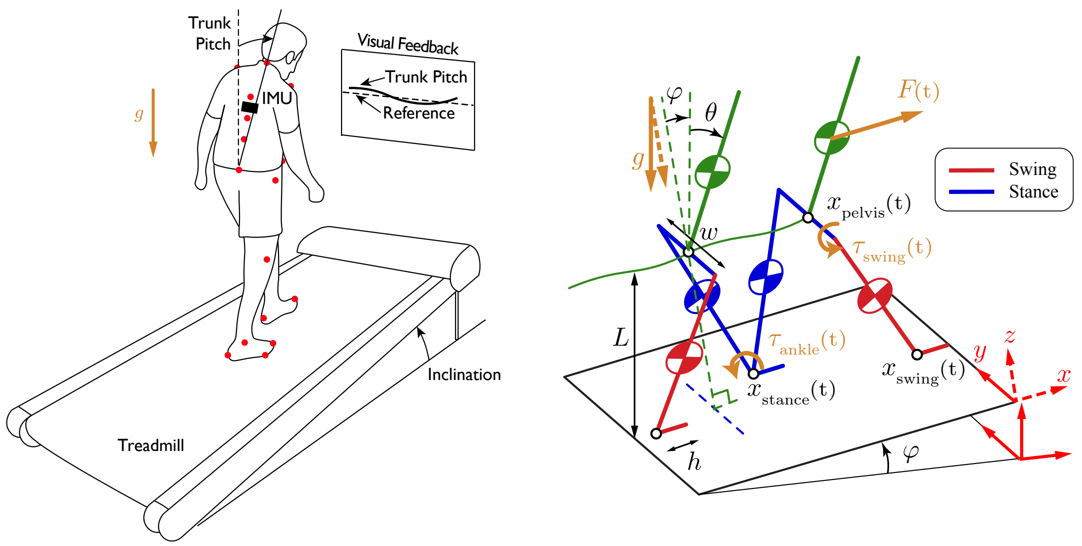
We measured lower and upper body kinematics with motion capture (Optitrack, NaturalPoint, Inc., Carvallis, OR, USA, collected at 120 Hz). Passive reflective markers were placed bilaterally on the ankle (lateral malleolus), knee (lateral epicondyle), hip (greater trochanter), and shoulder (acromion). We also placed three markers on each foot (third metatarsal, fifth metatarsal, calcaneus), three on the pelvis (sacrum, left/right anterior superior iliac spine) and four on the trunk along the spine (three spanning mid to upper back, C7). Asymmetry, joint angles, and step parameters were computed from the kinematic data. The trailing leg was the vector from mid-foot (between calcaneus and third metatarsal) to the greater trochanter of the trailing leg. The leading leg was defined similarly. The trunk was defined as the vector from the average of pelvis markers to the average of the shoulder markers. We reported dimensionless measurements using base units of body mass M, standing leg length L, and gravitational acceleration g. Walking speed was normalized by $\sqrt{gL}$ (mean 3.011 m/s), step length and step width by L (mean 0.925 m), and step time by $\sqrt{L/g}$ (mean 0.307 s).

10.2.2 Analysis

We analyzed asymmetry and kinematic data with respect to inclination angle, trunk angle, and walking speed to test our hypothesis. We used a linear regression model to determine the relative contributions of each parameter to asymmetry α.

$$\alpha \quad = C_v v + C_\theta \theta + C_\phi \phi + C_0, \tag{10.1}$$

where v is walking speed, θ is trunk angle, ϕ is inclination angle, and C_0, C_v, C_θ, and C_ϕ are scalar coefficients. C_0 represents constant offsets. Linear fits were performed for all data of each subject simultaneously while allowing each subject to have an individual constant offset. Additional fits were performed on gait parameters, such as step width and step time, to determine if they changed with experimental conditions. We also performed t-tests on trunk angle conditions to verify that subjects achieved significant



A)

B)

Figure 10.2 – The 3LP model and human walking experiment. A) In the experiment setup, we attached an IMU to the subject and calibrated it during normal standing. During the walking trials, we asked the subject to follow a reference angle on the screen by adjusting the actual trunk angle plotted in real time. The red dots indicate the markers used to record kinematics with a motion capture system. B) The 3LP model can simulate similar walking conditions. Model properties were matched to the subject's height and weight while walking speed and slope angles were set to the desired values. The trunk angle, walking frequency, step width and double support times are also set to the actual average values recorded from the subject.

differences in trunk angle relative to 0 deg for each of the commanded trunk positions.

10.3 Results

For both subjects and the model, we found that each parameter affected asymmetry, with the inclination angle inducing the greatest change (Figure 10.4). In the model, inclination angle had a negative relation with asymmetry while the trunk angle had the inverse relation. The speed, however, does not influence asymmetry considerably. In the human data, only the slope parameter changed asymmetry similarly to the model. The trunk angle only had small changes in asymmetry (towards reducing asymmetry magnitude), and walking speed asymmetry decreased in value (towards increasing asymmetry magnitude). On average, subject asymmetry changed at a rate of -3.77 deg/(m/s) (speed), -1.54 deg / deg (slope), and 0.14 deg / deg (trunk) with offset 1.72 deg. In comparison, the model's asymmetry increased at a rate of 0.39 deg/(m/s) (speed), -1.81 deg / deg (slope), and 0.54 deg / deg (trunk) with offset -2.29 deg. To produce a change in asymmetry of 14.5 deg relative to normal walking (4 km/hr, 0 deg inclination, 0deg trunk angle), the model would either have to increase the slope by 14% or bend backwards by 27 deg. For the same change in humans, subjects would have had to walk on a slope of 17% or bend

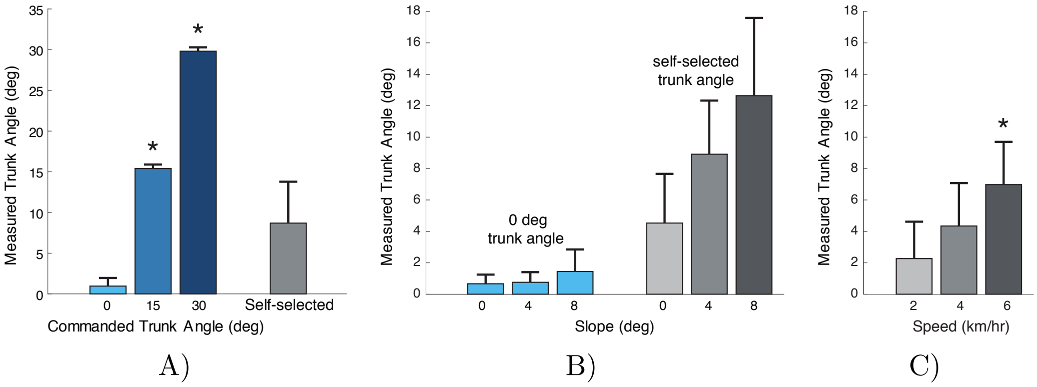


Figure 10.3 – Measured trunk angles for different commanded trunk angles, inclination, and speed. A) Commanded and measured trunk angles (mean and s.d. of all speeds and slopes). All three levels of imposed trunk angles were significantly different from each other ($P<0.05$). B) For the 0 deg trunk condition over all speeds, subjects were able to maintain the commanded trunk angle ($P>0.05$), which was significantly different from their self-selected trunk angle ($P<0.05$). C) Self-selected trunk angle changes with speed (0 deg inclination) are significantly greater at the fastest speed compared with normal speed (4 km/hr). Asterisks (*) indicate statistical significance ($P<0.05$) from 0 deg in A) and from 4 km/hr in C).

backwards by 104 deg.

Each experimental condition affected subjects' gait. The treadmill imposed the walking speed and slope. Visual feedback of desired trunk angles yielded significant differences in trunk angles (Figure 10.3, all $P<0.05$ relative to 0 deg condition). The greatest trunk level (30 deg) resulted in measured angles about 45 times greater than the lowest level (0 deg). Even at 0 deg trunk angle on 8 deg slope, subjectively the most difficult condition, subjects maintained upright trunk relative to gravity ($P>0.05$ relative to 0 slope condition).

The walking conditions had little effect on step parameters and expected effects on joint kinematics. For the inclined walking conditions, subjects exhibited slightly longer step length, step width, step period, and double support (Table 10.1), while for non-zero trunk flexion conditions, these parameters were shorter. Speed elicited expected changes in spatio-temporal parameters (Grieve and Gear, 1966), increasing both step length and step frequency. Walking with greater trunk angle resulted in smaller changes in kinematics (Figure 10.5) than for speed or inclination changes. At higher inclinations, we observed more knee and hip flexion at heel-strike.

10.4 Discussion

We sought to understand inter-leg symmetry as a function of different walking speeds, inclination angles, and trunk angles. From experimental data, we found that asymmetry increased in magnitude with speed and inclination but decreased in magnitude with trunk angle. In contrast, speed had a small effect on the model asymmetry, and trunk

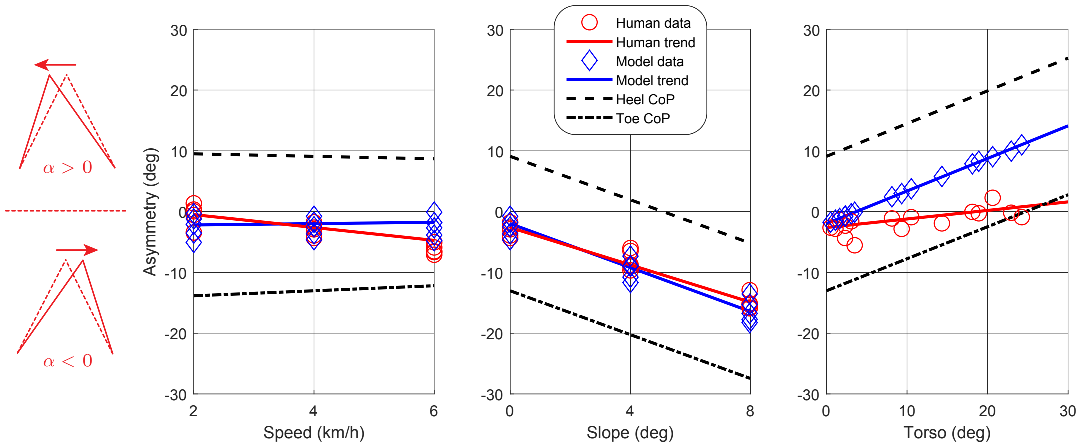


Figure 10.4 – Asymmetry values obtained from simulations and experiments as a function of different experiment parameters. The three columns in this figure each correspond to changing one parameter while keeping the other two fixed at their nominal values: speed of $v = 4$ km/hr, trunk angle of $\theta = 0$ deg and slope angle of $\phi = 0$ deg. Model predictions are calculated based on scaling the 3LP model by average subject anatomies and simulating walking with gait parameters obtained from human experiments. The two lines plotted for heel CoP and toe CoP are also obtained by running the same walking model, but keeping the CoP constant at the heel and toe. Coefficients and quality of fit are reported in Table 10.1.

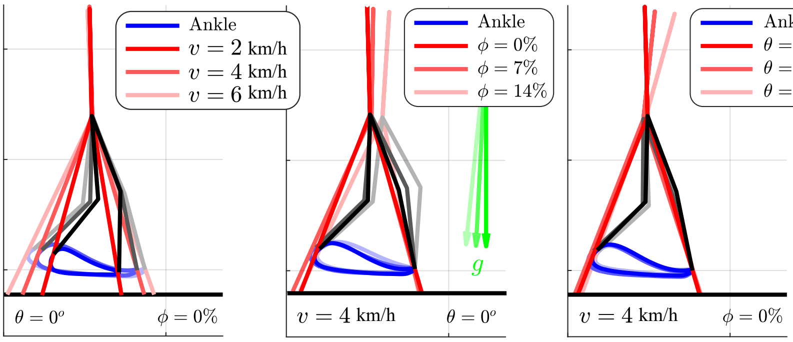


Figure 10.5 – Mean kinematics from a set of experimental conditions (same as shown for asymmetry in Figure 10.4). The three columns in this figure each correspond to changing one parameter while keeping the other two fixed at their nominal values: speed of $v = 4$ km/hr, trunk angle of $\theta = 0$ deg and slope angle of $\phi = 0$ deg.

had a greater effect than found with humans. For both model and data, inclination changes induced the largest changes in asymmetry.

While predicted asymmetries followed a similar trend to human data in inclined walking, there was mismatch for walking speed and trunk angle changes. According to Table 10.1, human walking frequencies and double support times are very sensitive to the choice of speed, which might explain why asymmetry trends are slightly different

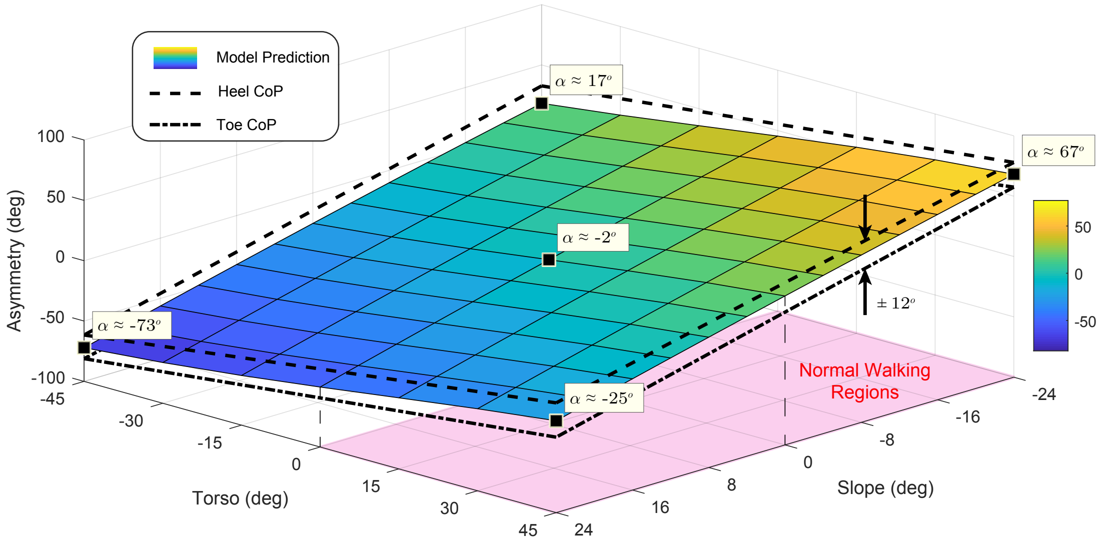


Figure 10.6 – The asymmetry index α by model predictions over a wide range of inclination angles and trunk angles at 4 km/hr. We determine walking frequencies, double support times and step width by interpolating the data presented in (Snaterse et al., 2011), (Cappellini et al., 2006), and (Donelan et al., 2002) respectively as a function of walking speed. Normal and feasible angles are shown by red regions, since bending back may not be comfortable for humans. The model is still able to simulate these infeasible walking conditions, indicating the possibility to compensate other sources of asymmetry by wearing heavy backpacks for example and maintaining erect postures. Note that the asymmetry is close to zero in neutral conditions, but the current definition is sensitive to double support duration. Ideally calculating the asymmetry index at the middle of the double support phase would remove this subtle bias at least in the model.

Table 10.1 – Quantitative results for fits to asymmetry and step parameters. Fit parameters include trend value (means±95% confidence interval, CI) and offsets (means±s.d.) of linear fits. Fit variables include normalized speed v, trunk angle θ and slope angle ϕ. R^2 values indicate the goodness of fit, and P-values indicate statistical significance of the trend (*P<0.05). Quantities are reported in dimensionless form, with body mass, gravitational acceleration and leg length as base variables.

	Trend			Offset	R^2
	Speed (c_v)	Slope (c_ϕ)	Trunk (c_θ)	c_0	
Human asym.	-0.20 ± 0.04	-1.54 ± 0.11	0.14 ± 0.04	0.03 ± 0.01	0.83
Model asym.	0.02 ± 0.01	-1.81 ± 0.02	0.54 ± 0.01	-0.04 ± 0.00	1.00
Step length	2.06 ± 0.07	0.26 ± 0.19	-0.15 ± 0.07	0.60 ± 0.02	0.95
Step width	0.04 ± 0.02	0.03 ± 0.06	-0.01 ± 0.02	0.05 ± 0.01	0.08
Step period	-5.58 ± 0.32	0.84 ± 0.87	-0.48 ± 0.31	6.09 ± 0.07	0.87
Double support	-2.59 ± 0.21	0.54 ± 0.56	-0.37 ± 0.20	2.05 ± 0.05	0.78

between model and human when changing speeds (Figure 10.4). In the model, the CoP moves linearly from the heel to the toe. To test the sensitivity of the model to CoP location and progression, we determined asymmetry trends for when the CoP is fixed at the heel and at the toe (Figure 10.4). We observed that CoP can introduce a bias which might bring the predictions closer to human data for the $\theta = 30$ deg trunk flexed conditions. Kluger et al. (Kluger et al., 2014) reported that CoP moves forward slightly faster during normal walking than in extreme trunk flexing conditions, which suggests the CoP might help compensate for the asymmetry caused by trunk flexion. A similar effect of increased forward progression of CoP was also reported for downhill walking in (McIntosh et al., 2006). While both the 3LP model and previous results point to the importance of CoP on asymmetry, we did not record CoP displacements and thus cannot experimentally verify this effect.

Both model and empirical data indicate that leaning the trunk forward increases asymmetry while walking uphill and walking faster decreases asymmetry. Therefore, trunk lean could compensate the asymmetric effect of speed and inclination. A possible mechanical explanation for the need for compensation in humans could be the discomfort of reaching hip extension limits in very negative asymmetries (i.e. shifting the pelvis forward). In our experiments, when walking at maximum speed ($v = 4$ km/hr) and inclination ($\phi = 14\%$), we measured a maximum hip extension angle of -9.9 ± 7.1 deg (mean ± s.d.) at self-selected trunk angles and -12.6 ± 4.0 deg at imposed zero trunk angles. Maximum hip extension angles during static conditions were found to be -17.2 ± 7.6 deg. Performing t-tests with respect to static angles, walking at zero trunk angle was not significantly different ($P = 0.259$) while walking with self-selected trunk angle was significantly different ($P = 0.032$). This indicates that walking with imposed zero trunk angle was close to the hip joint limits, perhaps leading to discomfort for our subjects.

The kinematic and spatio-temporal changes with speed, inclination angle, and trunk angle were similar to those previously reported. Speed-induced changes in both step length and step time non-linearly, as found in (Grieve and Gear, 1966). Walking uphill leads to increased trunk flexion (Rosa et al., 2018), as also seen in our data (see Figure 10.5). We also observed similar knee and hip changes in flexion presented in (McIntosh et al., 2006) with inclined walking. Similar to a neuromuscular reflex-based model, which found that the trunk favored more flexion at faster walking speeds (Song and Geyer, 2012), subjects exhibited increased trunk lean with speed (Figure 10.3). Walking with greater trunk angle resulted in small changes in kinematics (Figure 10.5), given the limited range of angles we explore in our experiments. At higher trunk angles up to 90 deg, more kinematic changes in the joint angles are expected (Aminiaghdam et al., 2017; Müller et al., 2017; Leroux et al., 2002).

We observed that the model predicts rapid asymmetry changes with inclination and trunk angles, but not with the walking speed (see Table 10.1 and Figure 10.4). Our model also did not have a notion of "comfort" and therefore the hip limits reached by human subjects do not explain model behavior. Humans can also use pelvis rotation and lumbar flexion to compensate for reaching their maximum hip extension limits (Vogt and Banzer, 1999; Leroux et al., 2002). In our model, however, these flexibilities are not included. Our model can also approximate adjustments in the pelvis position due to different sources of asymmetry but, with only telescoping legs, can not explain or produce joint-level adjustments (e.g. knee bending) .

We used linear fits to describe asymmetry as a function of speed, trunk, and slope angles because we found that a model with cross-terms did not markedly improve the fit (0.999 R^2 vs 0.999 R^2 for model predictions, linear and nonlinear models respectively) and (0.833 R^2 vs 0.862 R^2 for human data, linear and nonlinear models respectively). Model asymmetry is also considerably sensitive to CoP profile, yet we used a simple linear progression of CoP in the 3LP model. For future studies, we plan to conduct similar experiments and explore larger trunk flexion levels while recording ground reaction forces and CoP. This would hopefully provide enough data to find a more complete model of asymmetry.

Our study investigates the changes in inter-leg symmetry with various speeds, inclination angle, and trunk angles. We compared human data to data from a 3D walking model and found that both predicted a decrease in asymmetry with inclination angle and an increase with trunk angle. This leads us to conclude that while forward bias of the CoM is important for forward progression on positive slopes or fast speeds, the resulting hip angle may be uncomfortable or not easy to control. Trunk lean may be used then to compensate by reducing asymmetry. Thus, the role of trunk in walking is an important mechanical component that has whole-body effects on control and energetic costs and should not be neglected.

11 Human-Like Animation

The method presented in this chapter is chronologically developed at the end of the project. We used observations and expertise obtained in all other chapters to propose a more complete model of walking. This model is in fact tailored to an application of animation in which the goal is to produce human-like trajectories. This goal could be potentially achieved with very complex and realistic models of walking that include muscle details and neural circuits. However, our method aims to achieve a similar performance with microsecond calculations only. Such a hard requirement is achieved via derivation of closed-form equations in different levels. The proposed framework of this chapter can produce periodic and perturbed gaits in different walking conditions. We use the 3LP model of chapter 6 as a core to simulate walking dynamics at an abstract level. Besides, we use the time-projection control of chapter 7 to stabilize the gait in case of transitions or perturbations. On top of these two components, we introduce a novel component that produces vertical trajectories for the pelvis as well as human-like leg coordinations. As mentioned in chapter 9, these components have energetic consequences that influence the overall walking behavior. We also saw in the walking experiments of chapter 8 that missing human-like gait features, especially the toe-off motion, can be geometrically very restricting for the robot. The method of this chapter tries to include these features while respecting computational advantages of the previously developed components. Although there was no time left in the project to use the newly developed method on the real hardware, we believe the ideas presented here would make the dream of human-like walking on the robot close to reality[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji and Auke Jan Ijspeert. *"Scalable closed-form trajectories for periodic and non-periodic human-like walking."* arXiv preprint arXiv:1803.10048, 2018.

---

[1] All the videos of this chapter could be found at https://youtu.be/FdNrNeOqBHw

11.1 Background

The musculoskeletal system of human has multiple degrees of freedom and many muscles used to produce a wide range of activities. In particular, human walking features many complex motions in the lower-limbs produced by gravity and muscle forces. This complex system can be simplified to reproduce walking behaviors in simulation environments, depending on the level of details needed. The limbs can be simulated with multi-segment rigid bodies while rotary actuators in the joints can play the role of muscles to some extent. Such huge simplifications are probably enough to produce very realistic locomotion behaviors, however, a powerful controller is needed to stabilize the gait. A unified framework is hard to achieve given different anatomical properties, gait parameters, styles of motion and environment conditions. Besides, a plausible controller in this framework should easily handle transition conditions as well as capturing disturbances to simulate interactions with the environment. Many successful controllers are proposed in (Yin et al., 2007; Tsai et al., 2010; Coros et al., 2010; Mordatch et al., 2010) for example which use motion-capture data, simplified models or dynamic equations to achieve amazing walking behaviors.

In this work, we mainly focus on simulating essential principles of walking in the lower-limbs. We propose a method that combines trajectories of 3LP, a simple walking model developed earlier in chapter 6, with additional features that result in a human-like gait. We use the previously developed time-projection controller of chapter 7 to stabilize the gait and perform transitions. Recorded human data is also used to validate trajectories quantitatively. Thanks to linearity of the 3LP model and simplicity of the controller, we offer closed-form solutions for all lower-limb trajectories of human walking in a wide range of parameters. Our method, therefore, captures the main principles of walking with microsecond calculations. It can simulate walking behaviors multiple orders of magnitude faster than physics-based simulators which work at best in real time. This could be used for video games and animations especially on portable electronic devices with limited computational capabilities.

Animating walking behaviors is primarily inspired by biomechanics studies which quantify human gait properties and explain the mechanics behind. There are multiple techniques used to reproduce the behavior, ranging from pure interpolation of human data to detailed physics-based simulations that implement low-level control rules to mimic human behavior. This section reviews different walking animation methods proposed in the literature to highlight some key inspiring ideas. We classify the existing literature into three main categories: interpolation of recorded data, artificial trajectory synthesis, and physics-based simulation. We are more interested in a trade-off between genericity of the method and computation times. In this regard, each category provides certain advantages and limitations discussed as follows.

11.1.1 Interpolation of recorded data

Using a motion-capture system and markers placed on different body parts, one can capture human locomotion trajectories in different conditions. Choi et al., for example, used a motion library to plan bipedal locomotion in un-structured environments (Choi et al., 2003). They planned probabilistic road-maps which determined a sequence of character configurations based on foothold locations. Similarly, Lee et al. used a human motion database to control animated avatars (Lee et al., 2002). A Markov process was used in this method to plan motion phases while blending transition rules were applied to produce smooth motions based on a relatively large database of recorded data in non-periodic walking conditions. Similarly, a single set of periodic walking trajectories were scaled in (Okada and Miyazaki, 2013) and applied to a robot after fixing dynamical consistency.

Assuming a fast data query from the database, interpolations could be done relatively fast in terms of computations. However, if dynamical consistencies are considered like (Okada and Miyazaki, 2013), a small modification of trajectories is needed. Such modification might not be visually notable, but it might require expensive iterative computations. Besides, the quality of animations might heavily depend on the quality of recorded data. The method in (Lee et al., 2002) for example requires enough data-points to produce transient conditions. It is also sensitive to glitches and slippages. In a broader perspective, generalizing interpolations to characters with different anatomical properties than the dataset is feasible but challenging (Hodgins and Pollard, 1997). Covering a wide range of locomotion parameters such as speed, frequency, and style is also hard with a sparse database (Hodgins and Pollard, 1997). Besides, a realistic simulation of interactions with the environment seems impossible in this approach due to the absence of physics in the interpolations.

11.1.2 Artificial trajectory synthesis

A similar approach is to consider parametric trajectories instead of interpolating a motion database. Given the wide range of kinematic data reported in biomechanics studies, one can design empirical trajectories with appropriate phase and adjustable amplitudes for certain variables in the system. The approach proposed in (Boulic et al., 1990) for example considers sinusoidal variations for pelvis translations and rotations. It also considers similar trajectories for the hip, knee, ankle, thorax, shoulder and the elbow joints. A predefined phase and adjustable amplitude (as a function of velocity) for each of these trajectories can produce walking motions in a wide range of speeds. The relation of step frequency and speed is also taken from human data while an inverse kinematics algorithm slightly modifies trajectories to ensure contact constraints. The parametrization process can be done over key-frames too. The approach proposed in (Li and Liu, 2000) for example interpolates between different postures defined for certain gait events to obtain continuous trajectories. By changing the key-frames only, plausible walking trajectories can be obtained for different inclined terrains and walking speeds. The speed-frequency relation, as well as the double-support time ratio, is taken from human data in (Li and Liu, 2000).

Parametric trajectories have a wide range of applications in robotics. Handharu et al. used parametric trajectories for the foot and the knee to produce walking motions

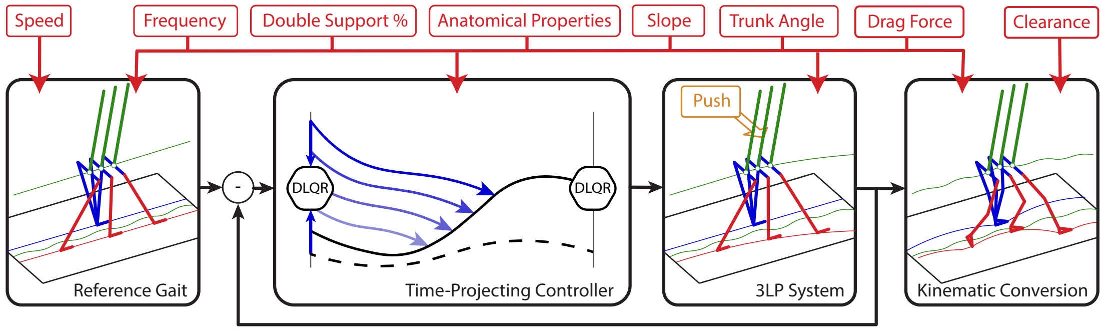


Figure 11.1 – An overview of our proposed simulation architecture. Given certain walking parameters shown in red, a reference 3LP gait is calculated based on which the time-projecting controller suggests footstep adjustments. The resulting trajectories of the 3LP system are then converted to a more human-like posture.

with toe and heel joints (Handharu et al., 2008). Similarly, Ogura et al. used parametric trajectories for the waist roll, foot positions, and the knee joints to produce stretched-knee walking patterns (Ogura et al., 2003). The approach proposed in (Ogura et al., 2006) used similar prescribed knee trajectories and optimized other parametric trajectories to find a ZMP-stabilized gait, validated on the WABIAN-2 robot. ZMP refers to Zero Moment Point around which contact reaction forces produce no moment in the horizontal direction. Manually designed Center of Mass (CoM) height trajectories together with a robust inverse kinematics method could later produce stretched-knee gaits on a refined version of this robot (Kryczka et al., 2011) as well. Although dynamic consistencies were resolved in all these methods (Handharu et al., 2008; Ogura et al., 2003, 2006; Kryczka et al., 2011) by proper regulation of ZMP trajectories (Vukobratović and Borovac, 2004), no online control method was proposed. Heerden, on the other hand, used sinusoidal reference trajectories for the pelvis and optimized jerks in a Model Predictive Control setup to plan CoM trajectories with variable heights (Van Heerden, 2015). This framework offered reactive stepping and online control, though costly in terms of computations, and missing the knee and ankle joints. The method proposed in (Griffin et al., 2017) offers a faster online control based on the capturability control framework (Koolen et al., 2012), though uses manually tuned set-points for the knee joint in different phases of motion to achieve stretched-knee walking.

Parametric trajectories are easy to design and tune for specific gaits, but hard to generalize for a broader range of walking conditions, e.g., at different speeds, frequencies, inclinations, walking styles, foot lifts, character sizes, etc. So far, we have only discussed trajectory generation methods (interpolation and synthesis) that impose kinematics. An alternative approach would be to compromise the animation speed and simulate system dynamics directly by time-integration to let the kinematics emerge automatically, i.e. taking dynamics into account.

11.1.3 Physics-based simulation

Simulating the full dynamical model simply translates to a complicated control problem. The walking system has multiple degrees of freedom while the type of task is hybrid, involving a change of mechanical model in each phase of motion (i.e., left, right or double support). Very similar to the few interpolation approaches mentioned earlier, one can use recorded human data or parametric trajectories to produce animations via position control. However, such open-loop controller might only be stable over a small set of states. To tackle this problem, Yin et al. simplified walking dynamics with a single mass and improved stability by suggesting footstep adjustments (Yin et al., 2007). In their simulation framework called SIMBICON, they obtained desired joint angles by applying a Fourier transformation on the recorded human data and taking only essential harmonics. While replicating the recorded trajectories, they used a simple proportional controller in the stance hip to regulate the trunk orientation. A control rule was also introduced to adjust the next footstep location as a linear function of CoM relative position and velocity (with respect to the stance foot) by tunable gains. Using task-specific parameter tunings, SIMBICON was able to achieve realistic walking gaits with different torso styles, leg lifts, motion directions, and push recovery properties. The method proposed by (Tsai et al., 2010) was also very similar to SIMBICON and relied on motion-capture data. However, Tsai et al. used an inverted pendulum model to adjust the footstep locations instead of the original tunable gains used on the relative CoM position and velocity in SIMBICON.

The free parameters of the SIMBICON framework together with initial conditions were later optimized by (Wang et al., 2009) using mechanical power terms in the objective function to produce more natural walking gaits. The optimized framework removed dependency on the motion-capture data and handled inclined walking as well. However, the optimization procedure had to be repeated for characters with different body shapes. Favoring generalization of the controller, Coros et al. also removed any dependency on the motion-capture data and only used few parametric spline trajectories to allow for human-like knee and ankle trajectories (Coros et al., 2010). They used the inverted pendulum model as a core motion generator with parametric swing leg motions. The resulting trajectories were converted to joint-space via inverse kinematics. In the low-level control, he used small gains to track the desired joint angles in addition to gravity compensation and Center of Pressure (CoP) modulation for better compliance and stability. Assuming a decoupling between horizontal and vertical dynamics, Mordatch et al. used a Linear Inverted Pendulum (LIP) and a Spring-Loaded Inverted Pendulum (SLIP) to describe motions in these directions respectively (Mordatch et al., 2010). They formulated a robust nonlinear Model Predictive Control (MPC) problem to plan the motion, although reaching a reactive online control slowed down their simulations considerably. The complete inverse dynamics formulation used by (Mordatch et al., 2010) could, however, unify the gravity compensation and CoP modulation rules of (Coros et al., 2010) and produced natural upper-body motions. Apart from simplifying the dimensionality problem, inverse dynamics can provide compliance (chapter 3), realize imprecise Cartesian plans (You et al., 2016) and allow for multi-character interactions (Vaillant et al., 2017).

A sub-category of physics-based simulation methods aims at finding task-specific controllers, but not using simplified models. A network of neurons stimulating virtual human-like muscles (neuro-muscular model) can be optimized for example to produce human-like walking gaits at different speeds (Geyer and Herr, 2010). Likewise, a character-specific optimization of control gains and set-point angles for a musculoskeletal model combined with the SIMBICON stabilization rules can also produce realistic walking behaviors (Geijtenbeek et al., 2013). Individual controllers can be composed together to cover a wider range of tasks using support vector machines (Faloutsos et al., 2001) or interpolation of control laws (Laszlo et al., 1996). Offline optimizations can also achieve more versatility by using exteroceptive sources of information. The reinforcement learning method proposed in (Heess et al., 2017) for example can achieve robust locomotion in rich environments by using very simple reward functions. Another promising method of generating locomotion behaviors aims at optimizing a sequence of end-effector trajectories through contact-invariant optimizations (Mordatch et al., 2012). This approach can produce realistic walking gaits (Posa et al., 2014). However, a considerable offline optimization effort is needed to obtain a single walking gait.

11.1.4 The proposed method

Although physics-based simulations can potentially produce various kinds of locomotion scenarios (Vaillant et al., 2017), the control algorithm remains a big challenge. Even for very simple walking behaviors, these simulations can hardly go faster than real-time (Mordatch et al., 2010). However, since dynamic equations are being integrated, interactions with the environment are made possible given stable controllers. Direct integration of multi-body symbolic equations (Docquier et al., 2013) would slightly speed up the animation (Van der Noot et al., 2015), but the effect of interaction forces should be included in the symbolic equations. A much faster speed can be achieved via interpolation or trajectory synthesis methods, however, producing interactions is not possible. In this work, we propose a method that can cover a wide range of walking conditions generated by physics-based simulations (Yin et al., 2007; Coros et al., 2010; Mordatch et al., 2010) while offering 2-3 orders of magnitude faster simulation speeds. While physics-based simulations need sub-millisecond integration times and hardly reach real-time factors, we use closed-form solutions of 3LP (as fast as microseconds only) to update the state only at display frames (e.g., 30 frames per second). This boost of speed easily makes real-time crowd walking simulations possible. It also enables computationally limited or portable electronic devices to simulate interactive walking scenarios easily.

To be precise, the proposed method is a novel hybrid combination of physics-based simulations and interpolation methods summarized in Figure 11.1. The physics of walking in our method is encoded in the 3LP model which is composed of three linear pendulums to model falling, swing and torso balancing dynamics. 3LP supports walking at different speeds, frequencies, double-support times, torso bending styles, terrain inclinations and subject heights and weights. Our symbolic equations also support external forces and torques applied to the torso while the time-projection controller of chapter 7 automatically captures these perturbations by adjusting footstep locations. This controller supports all the previously mentioned walking conditions without any parameter tuning. Since masses in the 3LP model are fixed to constant-height planes, we introduce a kinematic conversion to produce height variations. This part of our method involves adaptive trajectory synthesis without any tuning of parameters unlike the literature (Boulic et al., 1990; Li and Liu, 2000). Given a 3LP state (pelvis, torso and toe positions), our conversion adaptively varies the pelvis height to produce human-like excursions (Gard et al., 2004), lifts the swing toe to provide ground clearance (Wu and Kuo, 2016) and resolves a single Degree of Freedom (DoF) in each leg to produce thigh-shank-foot kinematics. These variations are all adaptive, independent of the previously mentioned walking conditions and calculated in closed-form.

11.1.5 Novelties

While covering a wide range of walking conditions, the proposed method simplifies physics-based simulations favoring faster computations. The 3LP model and the time-projection control are originally developed to control a real robot for walking and push recovery applications explained in chapter 8. Therefore, the novelty of this work mainly lies in adding torso styles and terrain inclination features to the 3LP model and more importantly, introducing an adaptive kinematic conversion to produce human-like gaits from 3LP states. This work mainly focuses on producing lower-limb kinematic trajectories while upper-body and pelvis oscillations can be included via predefined scalable trajectories similar to (Boulic et al., 1990; Li and Liu, 2000). An essential advantage of the 3LP model and time-projection control is in closed-form future predictions, given walking speed and external disturbance profiles. The proposed kinematic conversion method also produces human-like postures while preserving this property. In other words, the current converted posture does not depend on the previously converted postures. The entire method, therefore, enables a fast approximation of future kinematics in few microseconds which makes it suitable for model predictive control too. Knowing formulas of the 3LP model and the time-projection control form previous chapters, in the next section, we formulate our adaptive kinematic conversion method. We continue by demonstrating different walking gaits and conclude by a discussion on the supported range of walking conditions as well as promising aspects for future work.

11.2 Kinematic conversion

In this chapter, we only use swing hip torques in the 3LP model for active control and leave the stance ankle torque profiles fixed, i.e., moving the CoP linearly from the heel to the toe or vice versa, depending on the direction of motion. The model state in 3LP is composed of horizontal pelvis and feet positions. A torso angle θ, slope angle ϕ and constant external force F_{drag} are simply added to the original 3LP equations of Chapter 6. These constant terms can change periodic gaits, but not the derivation of control rules.

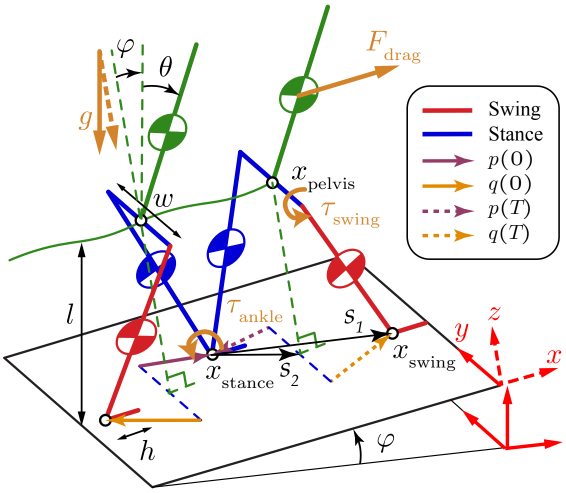


Figure 11.2 – A schematic of the 3LP model used in this chapter for walking gait generation. This 3D model is composed of three linear pendulums connected with a massless pelvis to simulate natural lateral bounces. All masses and the pelvis stay in constant-height planes which make the model linear. The torso also remains upright by an ideal actuator placed in the stance hip joint. The state of this model is described by the pelvis and swing foot horizontal positions while the swing hip and stance ankle torques serve as inputs to generate different walking gaits.

All variables are expressed in a rotated coordinate frame attached to the slope, shown by dashes in Figure 11.2.

The idea of kinematic conversion is to make the 3LP posture more human-like. As mentioned earlier, we focus on adding pelvis vertical excursions, ground clearance and lower-limb motions to the gait. Other human-like features like pelvis rotations and upper body motions are not considered in the present work. Our simple conversion algorithm only needs the current 3LP state and phase time t as inputs. Knowing other constants including phase timing parameters T_{ds}, T_{ss}, the slope angle ϕ, anatomical properties and the ground clearance height, our conversion algorithm performs the following steps:

1. **Pelvis height**: Find a smooth trajectory based on relative feet and pelvis positions.
2. **Ground clearance**: Synthesize a vertical toe position profile.
3. **Knee target points**: Find two target points for the knees that translate to hip angles and solve the inverse kinematics redundancy of each leg.

These small modifications are based on trajectory synthesis and minimally influence 3LP's overall falling and swing dynamics. Here, we simply rely on decoupling assumption between vertical and horizontal dynamics similar to (Mordatch et al., 2010; You et al., 2016). A more precise but computationally expensive approach would be to use the full dynamics equations and perform vertical adjustments only within the null-space of horizontal tasks (Griffin et al., 2017). The remainder of this section describes our adaptive kinematic conversion method in details.

11.2.1 Pelvis height

The horizontal dynamics of walking in our simulations is approximated by the 3LP model assuming constant heights for the pelvis, the two feet, and all limb masses. Given a 3LP state in periodic or transient walking conditions, our kinematic conversion takes the relative foot-hip positions and calculates a smooth pelvis height trajectory. This is done via a simple mixture of geometric variables at each instance of time. Remember that the legs in 3LP are modeled by extensible prismatic actuators. Given certain footstep locations, if we assume fixed-length legs like the normal inverted pendulum model, we obtain arc shapes for the pelvis which sharply intersect together (Kuo et al., 2005). Human pelvis trajectories are similar to smooth sine shapes, however, going to a minimum during the double support phase (Gard et al., 2004). To produce such trajectories, we use two different methods:

1. Fixed mixture: we consider fixed leg-length arcs around the CoP in each foot and introduce a soft weighting between them to produce the final pelvis height profile.

2. Adaptive mixture: we consider variable leg-length arcs around the CoP in each foot and apply a soft minimum function to produce the final pelvis height profile.

While the first method is enough for periodic walking and small perturbations, it cannot support backward walking or extreme toe-off stretching in perturbed conditions. The second method, however, simply produces a feasible pelvis height for both legs.

Remember that all the 3LP equations are solved in a rotated coordinate frame attached to the slope (refer to Figure 11.3). Therefore, the gravity vector is rotated and the nominal leg length (the variable l in Figure 11.2) is reduced by a factor of $\cos(\phi)$. This ensures a stretched leg when walking at zero speed on the slope. For a given 3LP state, we define relative vectors $p(t)$ and $q(t)$ by:

$$\begin{aligned} p(t) &= P\,(x_{\text{stance}}(t) - x_{\text{pelvis}}(t)) + \begin{bmatrix} 0 & \frac{ws}{2} \end{bmatrix}^T \\ q(t) &= P\,(x_{\text{swing}}(t) - x_{\text{pelvis}}(t)) + \begin{bmatrix} 0 & -\frac{ws}{2} \end{bmatrix}^T \end{aligned} \tag{11.1}$$

where the variable $s = \pm 1$ indicates left or right support phases, the parameter w denotes pelvis width and the matrix P is a simple operator to project the quantities on the $x - y$ plane shown in Figure 11.2. Example sagittal components of $p(t)$ and $q(t)$ are shown in Figure 11.4E for an adult person (height of 1.7m) walking at a speed of 1m/s and a frequency of 1.7 step/s. During in-place walking, we assume that the CoPs stay in the middle of each foot (the CoP in swing foot is virtual). Depending on the distance between the two feet, we move CoPs to the toes or heels. This is done via a linear profile

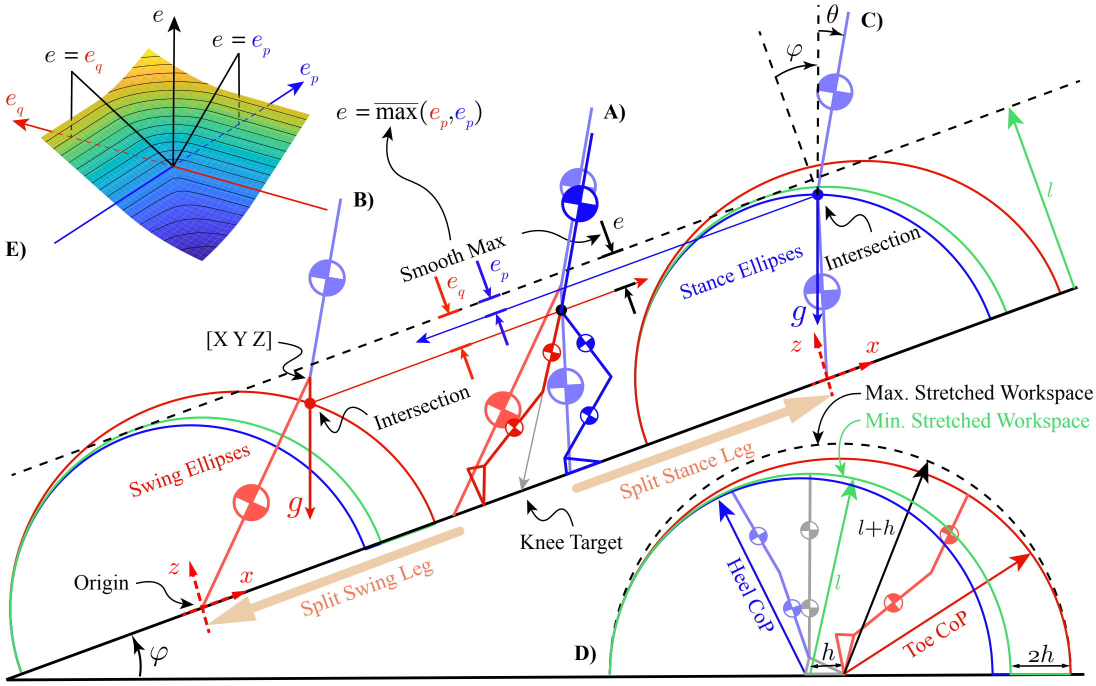


Figure 11.3 – A snapshot of inclined walking in early swing phase. A) The given 3LP state in the background can be split into B) swing and C) stance legs. Depending on the sagittal distance of the two feet, we calculate CoP points (virtual in the swing foot) and create candidate pelvis height arcs around the heel points. For the trailing leg, the CoP is on the toe which produces the red arc. For the front leg, the CoP is on the heel which creates the blue arc. For each leg, by intersecting a vertical line (passing through the pelvis in 3LP) with the arc, we find a candidate pelvis height. Now, compared to the maximum height possible (equals the leg length l), we calculate errors e_p and e_q for each leg and apply a smooth maximum function on them which is shown in E). The resulting maximum e determines the final pelvis height with respect to the maximum height l. Given the pelvis and toe positions for each leg then, we just need to resolve a single degree of freedom in each leg to find the complete thigh-shank-foot posture. This is done by determining a target point for the knee on the ground which implicitly determines the desired hip angle. In case the foot penetrates the ground, this angle is adjusted to keep the foot always flat.

in which a step length equal to the leg length produces a half-foot CoP movement:

$$\begin{aligned} p_{\mathrm{CoP}}(t) &= \frac{h}{2}\left(1+\frac{q(t)-p(t)}{l}\right) \\ q_{\mathrm{CoP}}(t) &= \frac{h}{2}\left(1+\frac{p(t)-q(t)}{l}\right) \end{aligned} \tag{11.2}$$

where the parameters h and l denote foot length and leg length shown in Figure 11.2 respectively. We also assume no lateral movements for the CoP in each foot. As observed in Figure 11.4E, the trajectories $p(t)$ and $q(t)$ have non-zero derivatives at the boundary times (in the beginning and at the end of the step phase). These derivatives produce sharply intersecting arc shapes for the pelvis whereas setting them to zero would produce flat trajectories. To achieve this, we define an additional signal $\alpha(t)\dot{p}(t)$ where the polynomial $\alpha(t)$ has derivatives of -1 at $t = T_{\text{ds}}$ and $t = T$ shown in Figure 11.4A. This signal can correct the derivatives by setting them to zero when added to $p(t)$ and $q(t)$ (shown in Figure 11.4E). Therefore, modified relative positions $\bar{p}(t)$ and $\bar{q}(t)$ can be defined as:

$$\begin{aligned}\bar{p}(t) &= p(t) + \alpha(t)\dot{p}(t)\\ \bar{q}(t) &= q(t) + \alpha(t)\dot{p}(t)\end{aligned} \tag{11.3}$$

which represent smooth relative positions between the hip and the heel in each leg, ensuring zero derivatives at the phase boundary times. Assume we put a rotated coordinate frame (along the slope) on each heel and express the modified relative 3LP pelvis position by $[X\ Y\ Z]$ in this frame where $Z = l\cos(\phi)$ shown in Figure 11.3B. Now, fixed leg-length arcs can be found as:

$$Z^2 = (X + x_{\text{CoP}} + l\sin(\phi))^2 + Y^2 + z_{\text{fixed}}(X, Y, x_{\text{CoP}})^2 \tag{11.4}$$

where the constant $l\sin(\phi)$ is added to compensate the effect of slope. The candidate pelvis heights for each leg are:

$$\begin{aligned}z_{p,\text{fixed}}(t) =&\quad z_{\text{fixed}}(\bar{p}_x(t), \bar{p}_y(t), p_{\text{CoP}}(t))\\ z_{q,\text{fixed}}(t) =&\quad z_{\text{fixed}}(\bar{q}_x(t), \bar{q}_y(t), q_{\text{CoP}}(t))\end{aligned} \tag{11.5}$$

where x and y are sagittal and lateral components respectively. Now, according to the first method, the final pelvis height trajectory could be found by a smooth transition from the arc on the stance foot $z_{p,\text{fixed}}(t)$ to the arc on the swing foot $z_{q,\text{fixed}}(t)$:

$$z_{\text{fixed}}(t) = (1 - \gamma(t))\ z_{p,\text{fixed}}(t) + \gamma(t)\ z_{q,\text{fixed}}(t) \tag{11.6}$$

which is shown in Figure 11.4F. The function $\gamma(t)$ implements a smooth transition shown in Figure 11.4C. The resulting curve $z_{\text{fixed}}(t)$ features zero derivatives at the phase boundary times while it peaks approximately in the middle of the single support phase. Overall, the converted pelvis position is matching 3LP's pelvis position horizontally and shifted down from $l\cos(\phi)$ to the new height $z_{\text{fixed}}(t)$. The first method produces a good approximate of human trajectories, but there is no guaranty that the next touch-down happens on the heel. In backward walking or perturbed conditions, for example, the next step might touch down on the toes. In these cases, the resulting pelvis height (at $t = T$) should be feasible for the other stance leg which is still on the heel. However, the formula (11.6) always converges to the swing arc at $t = T$ which could become infeasible. Therefore, the mixture used in (11.6) is meaningful in terms of producing vertical excursions, but limited to periodic forward walking conditions.

To overcome limitations of the first method, we replace the time-based mixture of (11.6) with a smooth maximum function in the second method to better decide between swing and stance leg arcs. Also, we slightly modify the arcs to allow for more leg flexion and extension in the stance phase. Consider Figure 11.3D which shows different pelvis arcs depending on the position of CoP. When walking in-place, no matter what other gait parameters are, the arc is calculated around the heel position. In maximum step length conditions, however, the CoP moves to extremities according to (11.2). In this case, we use a specific nonlinear function $\delta(x)$ (Figure 11.4D) which maps the linear CoP movements of (11.2) into asymmetric profiles:

$$f(x_{\text{CoP}}) = \Delta = \frac{h}{2}\delta(\frac{2}{h}\ x_{\text{CoP}} - 1) \tag{11.7}$$

When the CoP goes to the toes ($x_{\text{CoP}} = h$), the function f produces a value of $h/2$ and when the CoP goes to the heels ($x_{\text{CoP}} = 0$), this function returns $-\epsilon h/2$. Based on this function, we formulate our modified arcs by following ellipses:

$$(\frac{x - 2\Delta}{l + 2\Delta})^2 + (\frac{y}{l})^2 + (\frac{z}{l + \Delta}) = 1 \tag{11.8}$$

which are shown in Figure 11.3D. The dashed circle in this plot shows the maximum workspace of the pelvis rotating around the toe. We consider smaller arcs (the ellipse shown in red) when the CoP is at the toes to avoid extra lifting and over-extension. The green circle also shows the minimum workspace when rotating around the heel (for in-place walking). When the CoP is at the heel, the blue ellipse produces a small knee flexion at the touch-down moment (like human (Liu et al., 2008)) determined by the choice of $\epsilon = 0.2$ shown in Figure 11.3D and Figure 11.4D. Our specific design of $\delta(x)$ and adaptive ellipses of (11.8) produce convincing human-like trajectories compared to a few human gaits recorded (discussed in the next section). However, they could be tuned further in future work to better match human trajectories in a wider range of walking conditions.

Given a 3LP state, we calculate the CoP points, split the legs and create an ellipse for each of them in the rotated coordinate frame (shown in Figure 11.3). A vertical line coming down from the pelvis in each case intersects with the ellipse and determines the candidate pelvis height of that leg. Denoting the modified relative 3LP pelvis position by $[X\ Y\ Z]$ like before, the vertical line in the rotated coordinate frame is:

$$y = Y,\ x - X = \tan(\phi)(z - Z) \tag{11.9}$$

which intersects with the ellipse of (11.8) and results in the following equation (as a function of z):

$$(\frac{X + \tan(\phi)(z - Z) - 2\Delta}{l + 2\Delta})^2 + (\frac{Y}{l})^2 + (\frac{z}{l + \Delta}) = 1 \tag{11.10}$$

The height of intersection point (denoted by $z = z_{\text{adapt}}(X, Y, x_{\text{CoP}})$) solves the equation (11.10) and therefore, each leg gives a candidate pelvis height:

$$\begin{aligned} z_{p,\text{adapt}}(t) = &\quad z_{\text{adapt}}(\bar{p}_x(t), \bar{p}_y(t), p_{\text{CoP}}(t)) \\ z_{q,\text{adapt}}(t) = &\quad z_{\text{adapt}}(\bar{q}_x(t), \bar{q}_y(t), q_{\text{CoP}}(t)) \end{aligned} \tag{11.11}$$

Unlike the time-based mixture of (11.6), in the second method, we use a smooth minimum function between $z_{p,\text{adapt}}(t)$ and $z_{q,\text{adapt}}(t)$ which is implemented as:

$$z_{\text{adapt}}(t) = l - \overline{\max}(l - z_{p,\text{adapt}}(t),\ l - z_{z,\text{adapt}}(t)) \tag{11.12}$$

where $\overline{\max}(a, b)$ is defined as:

$$\overline{\max}(a, b) = \begin{cases} \sqrt{a^2 + b^2} & 0 \le a, b \\ a & b < 0 \le a \\ b & a < 0 \le b \\ a + b + \sqrt{a^2 + b^2} & a, b < 0 \end{cases} \tag{11.13}$$

and shown in Figure 11.3E. The time-trajectory of $z_{\text{adapt}}(t)$ shown in Figure 11.4G is very similar to $z_{\text{fixed}}(t)$ (shown in Figure 11.4F) in periodic walking conditions. Figure 11.3A also visualizes the mechanism of finding individual pelvis heights and the smooth maximum function $\overline{\max}(a, b)$. Due to the fact that we always choose the minimum pelvis height in the second method (which is of course feasible for both legs), we can support non-periodic and backward walking as well as extremely asymmetric triangular leg coordinations which happen in inclined walking or presence of dragging forces. Therefore, we prefer the second adaptive method over the first fixed-time mixture method. The modified ellipsoid design of the arc shapes also produces realistic knee flexion during the touch-down and ankle extension during the push-off moments.

11.2.2 Ground clearance

Remember that both feet in the 3LP model are constrained to have a zero height. To make them more realistic, we consider simple vertical sinusoid curves that lift the swing toe vertically. These curves are scaled by the ground clearance parameter as a percentage of the leg length. Our simple design of these trajectories produces realistic motions, but cannot simulate foot flapping effects shortly after the touch-down moment (in which the foot completely lands on the ground after the heel-strike). We consider adding this feature in future work.

11.2.3 Knee target points

Given the pelvis and toe positions in the Cartesian space, the task in this stage is to resolve a single degree of freedom in each leg to find a human-like thigh-shank-foot posture. Our strategy is to determine the hip angle based on certain target trajectories on the ground. In each leg, the thigh vector (connecting the hip to the knee joint) points

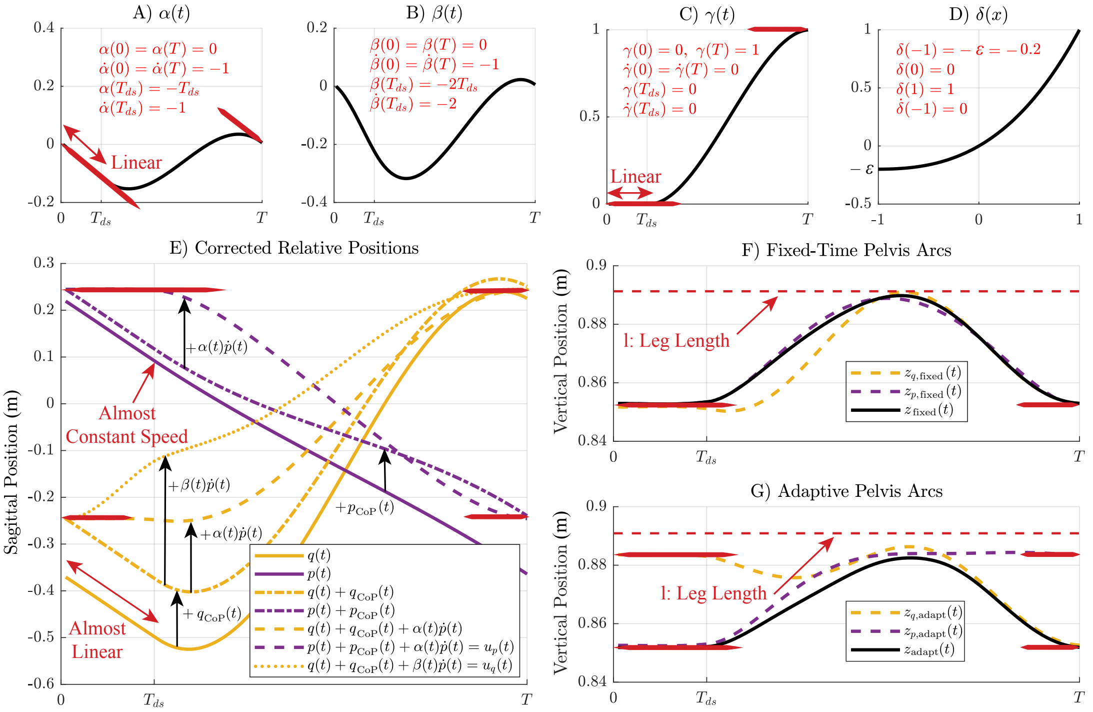


Figure 11.4 – Composition of smooth relative positions and knee target trajectories based on relative foot positions and phase timing of 3LP. A) The polynomial $\alpha(t)$ is used to correct the nonzero velocities of $p(t)$ and $q(t)$ at the boundary times. When multiplied by $\dot{p}(t)$ in and added to $p(t)$ or $q(t)$, the specific design of $\alpha(t)$ can result in zero derivatives at the boundaries. B) The polynomial trajectory $\beta(t)$ is designed to perform a faster progress than $\alpha(t)$. When polynomial $\beta(t)$ is multiplied by $\dot{p}(t)$ in and added to $q(t)$, it produces a trajectory which moves forward even during $0 \leq t \leq T_{\text{ds}}$. C) The polynomial trajectory $\gamma(t)$ used for a time-based mixture of $z_{p,\text{fixed}}(t)$ and $z_{q,\text{fixed}}(t)$ in the first method to produce a soft transition. D) The nonlinear function $\delta(x)$ which maps the linear motions of CoPs into asymmetric profiles Δ for each leg. These profiles adjust the default circular pelvis arcs (shown in green, Figure 11.3D) into different ellipses for swing and stance legs according to (11.8). E) Original relative foot-hip positions $p(t)$ and $q(t)$, addition of smooth CoP trajectories $p_{\text{CoP}}(t)$ and $q_{\text{CoP}}(t)$ and addition of corrective signals $\alpha(t)\dot{p}(t)$ and $\beta(t)\dot{p}(t)$ to produce smooth trajectories $\bar{q}(t)$ and $\bar{p}(t)$ (used for pelvis height trajectory generation) and $u_q(t)$ (used together with $u_p(t)$ as knee target trajectories). The synthesized signals are mere functions of 3LP state and phase timing without any history or dependency on the past. F) The arc trajectories $z_{p,\text{fixed}}(t)$ and $z_{q,\text{fixed}}(t)$ (produced from $\bar{p}(t)$, $\bar{q}(t)$, $p_{\text{CoP}}(t)$ and $q_{\text{CoP}}(t)$) are smoothly combined together with $\gamma(t)$ to generate the final pelvis height trajectory $z_{\text{fixed}}(t)$. G) The adaptive arc trajectories $z_{p,\text{adapt}}(t)$ and $z_{q,\text{adapt}}(t)$ are smoothly combined together by $\overline{\max}(a, b)$ to generate the final adaptive pelvis height trajectory $z_{\text{adapt}}(t)$.

towards a target trajectory on the ground shown in Figure 11.3A. Once the hip angle is determined, the configurations of shank and foot segments are found by solving a simple Inverse Kinematic (IK) problem between the knee and the toe, restricting the

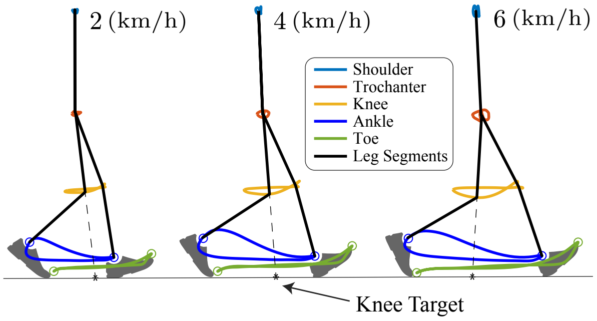


Figure 11.5 – Kinematic configurations of the leg segments in human walking at different speeds (subject height of 1.76m). The black segments indicate body postures at the touch-down and toe-off moments approximately. Given the knee marker trajectories, it is obvious that the thigh segment already starts swinging forward before the entire leg starts its swing phase. This requires target swing trajectories on the ground that progress forward during the double support phase already.

foot segment inside the sagittal plane. Our IK formulation also does not allow for a heel position below the toe position vertically.

We design the knee target trajectories by a similar mixture of geometric variables (foot positions) introduced earlier. In fact, for the stance leg, the relative position:

$$u_p(t) = p(t) + \alpha(t)\dot{p}(t) + p_{\text{CoP}}(t) \tag{11.14}$$

is a good target trajectory. A careful inspection of human trajectories at different walking speeds reveals that the Cartesian swing knee position already starts moving forward before the swing phase starts. This effect is shown in Figure 11.5 at different walking speeds. The relative position $q(t) + \alpha(t)\dot{p}(t) + q_{\text{CoP}}(t)$ is not a good target trajectory for the swing leg, since it remains constant during the double support phase and starts moving forward only in the single support phase (Figure 11.4E). To overcome this issue, we introduce a polynomial $\beta(t)$ shown in Figure 11.4B, a modified version of $\alpha(t)$ which is approximately two times larger, but with similar derivative properties. The swing target point is now defined by:

$$u_q(t) = q(t) + \beta(t)\dot{p}(t) + q_{\text{CoP}}(t) \tag{11.15}$$

which already starts moving forward during the double support phase shown in Figure 11.4E. The two target trajectories $u_p(t)$ and $u_q(t)$ are eventually used to find the hip angles which then determine the leg configuration completely.

11.3 Results

Implementation of the 3LP simulator and the kinematic conversion is currently done in MATLAB with a simple GUI (shown in Figure 11.9) that allows the user to change

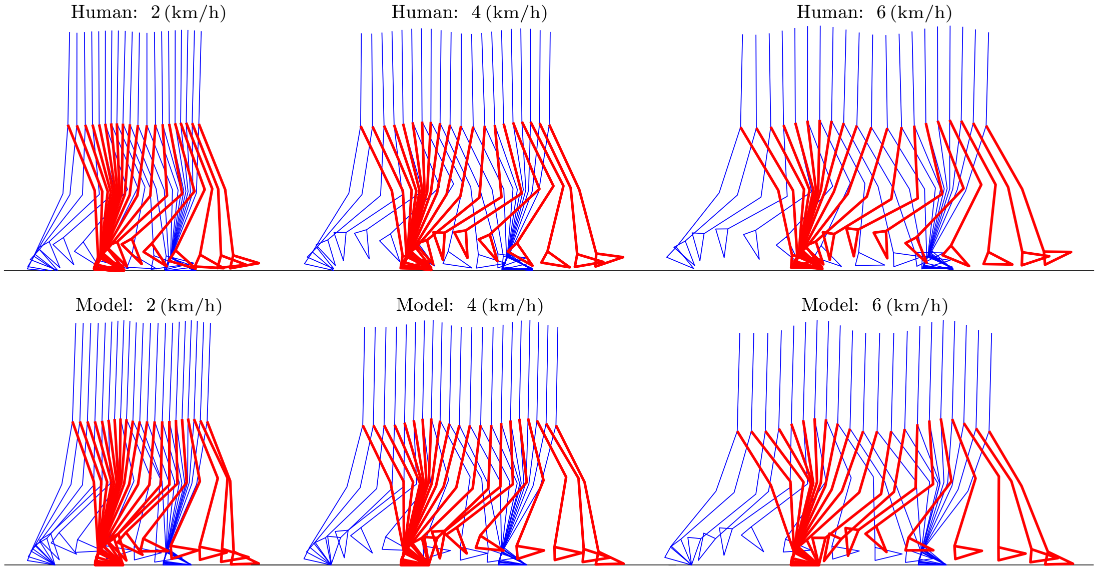


Figure 11.6 – Snapshots of human walking and synthesized walking trajectories at different speeds. The choice of step frequency, double support duration, torso angle and ground clearance parameters as well as body properties are taken from each human experiment and used in the corresponding simulation. The overall horizontal dynamics of walking is encoded in the 3LP model which produces human-like limb motions. On top of 3LP trajectories, our kinematic conversion can produce vertical excursions for the pelvis, human-like knee angles at the touch-down moments and realistic coordinations of thigh-shank-foot segments. However, the current method is unable to produce foot flapping motions after the heel-strike.

subject/gait parameters as well as to test transient conditions [2]. A C++ implementation of the 3LP model and the time-projection controller is also available for robotic applications of chapter 8. These codes only contain pure mathematic formulas in closed-form. The most computationally complex function in our method solves a linear system of equations to find periodic walking gaits. We use the Eigen library (Guennebaud et al., 2010) to perform this operation in microseconds. The inverse kinematic problem at the last stage of the kinematic conversion is also as simple as finding roots of a second-degree polynomial in closed-form. Using the MATLAB interface in this section, we present a collection of various walking trajectories produced with the proposed framework.

11.3.1 Different speeds

We start this section by providing a comparison of synthesized walking trajectories against human trajectories at different walking speeds (2,4,6 km/h). The data presented here is collected by a lab motion-capture system from treadmill walking. Five subjects with average height of 1.76± 0.11m and weight of 68 ± 14 kg participated in the experiment, walking for a minute at each desired speed to collect enough gait cycles. We measured

---

[2] Source codes available online at https://biorob.epfl.ch/research/humanoid/walkman.

gait parameters and replicated each experiment by our model to find corresponding synthesized gaits. Example collected and synthesized trajectories of one subject are demonstrated in Figure 11.6 which visually look very similar. Over all subjects and speeds, we found average correlations of 0.80 ± 0.06 for the hip angles, 0.86 ± 0.04 for the knee angles and 0.72 ± 0.09 for the ankle angles. Our method can produce many features of human walking such as pelvis vertical excursions (Gard et al., 2004), ground clearance (Wu and Kuo, 2016), heel-toe motions (Cappellini et al., 2006) and lateral bounces (Donelan et al., 2002). However, it does not produce pelvis and trunk rotations as well as foot flapping. Apart from scaling with respect to the walking speed parameter, our method supports variation of many other gait conditions as follows.

11.3.2 Model sizes

The 3LP model is scalable with respect to the mass and subject height properties. In this work, we considered average human anatomic proportions (de Leva, 1996) to scale all limb masses and body segments only with the overall body mass and height. However, both the 3LP model and kinematic conversion are independent of body properties without needing any re-tuning. Figure 11.7A shows a child at the height of 1m performing a walking gait similar to a tall adult of 2.5m shown in Figure 11.7B. In these case, we scaled the walking speed proportionally. The gait kinematics in 3LP is independent of the body mass however.

11.3.3 Inclined walking

By increasing or decreasing the terrain inclination, we can produce human-like walking gaits without re-tuning of any other parameter. The resulting kinematics shown in Figure 11.7C,D are very similar to the human data (Leroux et al., 2002). However, our method is not able to simulate extreme climbing cases where the hands are also involved. An interesting feature of inclined walking is extra knee flexion at the touch-down moment on positive slopes (Leroux et al., 2002) which is observed in Figure 11.7D as well.

11.3.4 Walking frequency

Although human walks at a particular combination of walking speeds and frequencies (Bertram, 2005), the frequency can be changed while keeping the speed constant. This directly influences the step length which is increased for example when the frequency is decreased (shown in Figure 11.7E). While the 3LP model easily supports this modulation of frequency, our kinematic conversion method can produce realistic walking gaits in both low frequency and high frequency conditions shown in Figure 11.7E, F.

11.3.5 Backward walking

The 3LP model can easily simulate backward walking by finding solutions in the linear null-space of initial gait conditions. The CoP motion can also be easily reverted to make the motion more realistic. This fact is reflected in the kinematic conversion method as well by automatic reversion of $p_{\text{CoP}}(t)$ and $q_{\text{CoP}}(t)$ trajectories in (11.2). Without

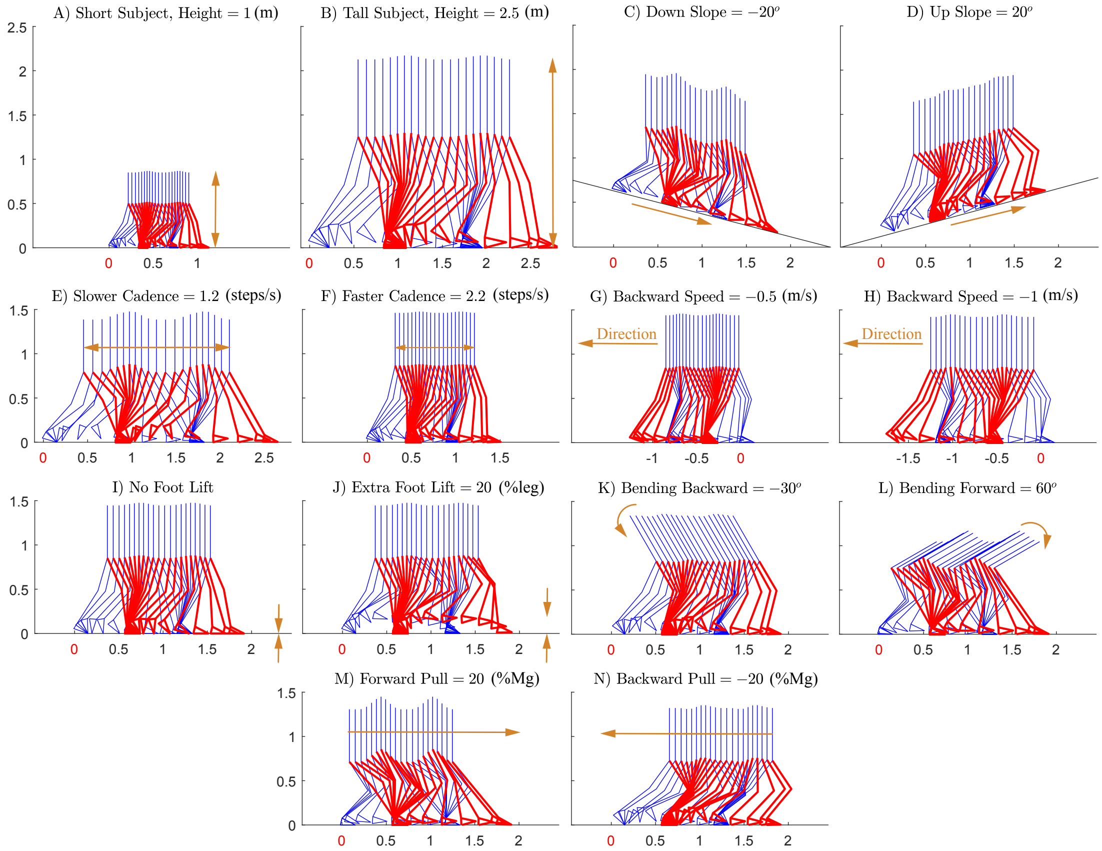


Figure 11.7 – Gait snapshots produced at different walking conditions. By default, we simulate an adult person (height of 1.7m, a weight of 70kg) walking at a speed of 1m/s, a frequency of 1.7 step/s and a ground clearance of 5% leg length. A) and B) show walking gaits for a child and a very tall person. C) and D) demonstrate inclined walking conditions at moderate slopes. E) and F) show the effect of changing walking frequency which directly influences the step length. G) and H) show backward walking gaits. I) and J) demonstrate no foot clearance and extra foot clearance conditions. K) and L) simulate walking gaits with different torso styles. Finally, M) and N) simulate walking gaits with considerable external dragging forces. Generation of walking gaits while combining all these conditions is also possible.

changing other parameters, the 3LP model can easily walk backward at different speeds while the kinematic conversion produces human-like coordinations of lower-limb segments shown in Figure 11.7G, H.

11.3.6 Ground clearance

Our model simulates this motion with a simple sinusoidal curve while the actual curve in human might be slightly different, especially in extra foot lift conditions (Wu and Kuo, 2016). Our simple strategy produces visually plausible walking gaits (shown in Figure 11.7I, J) while the main inconsistency comes from missing the short flapping phase after

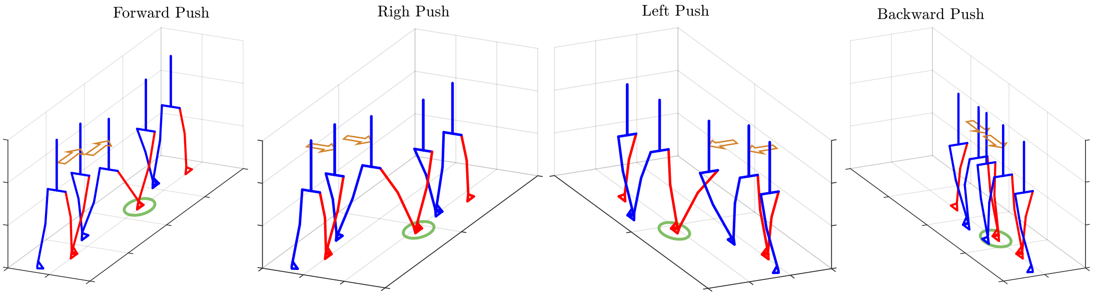


Figure 11.8 – Different scenarios of perturbed walking conditions where the robot is subject to large external pushes of 50N applied continuously during a step phase. Our time-projecting controller can easily stabilize the 3LP model while the kinematic conversion takes the 3LP state and produces human-like trajectories. Pure interpolation methods for walking animation cannot simulate perturbed walking conditions interactively while physics-based animations require a lot of computation power to simulate such interactions with the environment. Our hybrid approach, however, can cover a wide range of transient walking conditions.

the heel-strike. Also, an extra ground clearance might slightly affect the swing dynamics which is not included in the 3LP model.

11.3.7 Torso style

A vast part of the walking animation literature introduces methods to produce walking gaits at different torso angles, referred to as torso styles. This is achieved via a simple proportional-derivative controller in the stance hip to regulate the torso angle while foot-placement algorithms automatically compensate the dynamic effects of such asymmetry (Yin et al., 2007; Mordatch et al., 2010; Coros et al., 2010). Our 3LP model can easily simulate these scenarios while the kinematic conversion adjusts the kinematics automatically, shown in Figure 11.7K, L. Note that bending backward is uncomfortable for human while in simulations, it is theoretically possible. The extra vertical excursion observed in Figure 11.7L is also less human-like. When bending forward, humans damp these vertical excursions by an increased flexion in the knees at the mid-stance moment (Grasso et al., 2000), probably for the sake of comfort or gaze stabilization. Our method, however, produces a peak in the pelvis height trajectory at this moment which results in a stretched-knee posture. This probably prevents our method to simulate extreme torso bending conditions.

11.3.8 Dragging forces

Another interesting scenario is to produce periodic walking gaits subject to constant external dragging forces. This could be useful in a simulation of pulling or pushing heavy objects (Coros et al., 2010). 3LP can easily produce such walking gaits by including the external force in symbolic equations. We considered forces applied to the torso while 3LP formulations can be easily changed to simulate other force application points.

Although the triangular coordination between the two legs in 3LP becomes asymmetric in these conditions, the kinematic conversion can still produce lower-limb coordinations adaptively.

11.3.9 Push recovery

The main purpose of incorporating a physics-based animation in our method is to model interactions with the environment. In addition to simulating constant external forces discussed previously, we are interested in simulating transient conditions due to disturbances as well. This simulation scenario involves time-integration, i.e., considering small time-steps, applying arbitrary disturbance forces at each time-step and finding system evolution through integration. Thanks to the linear equations of the 3LP model, the system evolution can be described by a single closed-form matrix which relieves the need to perform iterations. Also, if the disturbance pattern is known beforehand, we can find closed-form equations and avoid using small time-steps, depending on the precision required. Figure 11.8 demonstrates transient walking conditions due to external pushes applied in different directions. While the 3LP model and the time-projecting controller can produce natural and stable horizontal motions, the kinematic conversion takes the 3LP state and produces vertical motions adaptively.

The strength of our method lies in generating walking trajectories in different combinations of all the previously-mentioned gait conditions. We limit our results section to discuss each gait condition separately. However, thanks to the closed-form solutions available, changing many gait conditions at the same time does not need any re-tuning of trajectory generation or control parameters. The next section will provide a comprehensive discussion of these strengths and intrinsic limitations of the proposed approach.

11.4 Discussion

The proposed method combines physics-based and pure interpolation approaches in the literature for walking trajectory generation. We simulate physics by a linear simplified model called 3LP that has closed-form solutions. On top of this model, in the present work, we propose an adaptive kinematic converter which synthesizes human-like lower-limb postures. The resulting trajectories follow the overall dynamics of 3LP while remaining geometrically feasible in transient conditions. The goal of such a hybrid approach is to achieve faster simulation speeds while offering an online walking control. We can simulate interactions with the environment to some extent and produce transient walking trajectories thanks to a previously developed walking controller called time-projection. This controller together with the 3LP model encapsulates important dynamic properties and control rules needed to stabilize the gait in a wide range of walking conditions. Therefore, the proposed model-based approach does not have any parameter to tune.

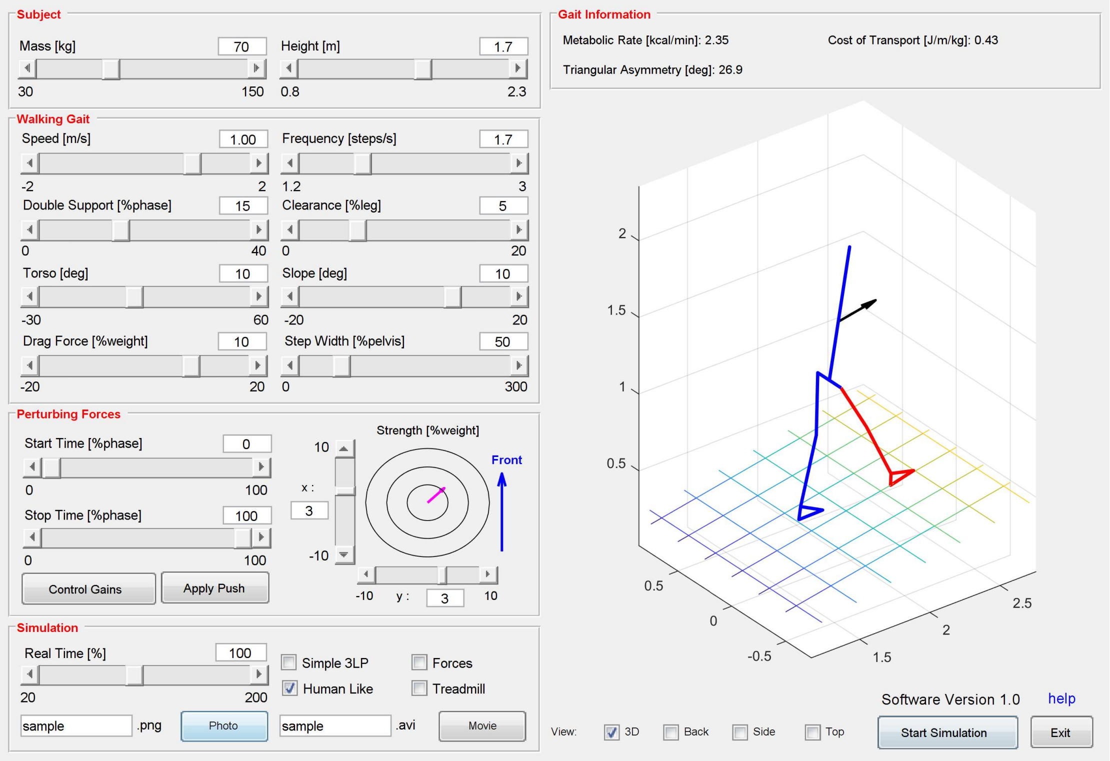
11.4.1 Closed-form solutions

The 3LP model and the time-projection controller were originally developed to extend the LIP model and MPC control paradigm of chapter 2 for humanoid walking applications. The closed-form equations of 3LP or LIP enable MPC controllers to stabilize the system in an online fashion by adjusting footstep locations. Our MPC controller of chapter 2 was able to solve a quadratic optimization problem in less than a millisecond and suggest footstep corrections online. The time-projection control, however, aims at finding closed-form solutions for the numeric optimizations of MPC. Therefore, the combination of a linear model and time-projection control can offer simulation speeds as fast as microseconds. Besides, thanks to all these closed-form solutions, we do not need to use sub-millisecond simulation time-steps like (Yin et al., 2007; Mordatch et al., 2010; Coros et al., 2010) to ensure numerical stability. To generate animations, we only need to consider movie frames (e.g., 30 frames per second) and find the system evolution in between by closed-form matrices.

While walking gait generation and stabilization with a simplified model are developed in our previous work, in the present chapter, we aimed at filling the gap with reality. In other words, we proposed a kinematic conversion method to convert walking trajectories from the 3LP space to a real character with thigh, shank and foot segments. The novelty of this chapter, therefore, lies in the conversion method and the entire architecture that produces periodic and transient human-like walking trajectories. Using an intuitive mixture of the geometric variables in 3LP, we can produce smooth vertical excursions and human-like thigh-shank-foot coordinations. Although we do not simulate dynamics of these leg segments explicitly, each leg follows the approximate dynamics encoded in the 3LP model. Following the same philosophy of developing closed-form solutions in the 3LP model and the time-projection controller, the kinematic conversion is also formulated in closed-form. Various parametric trajectory design or interpolation methods already exist in the literature and offer a similarly fast simulation speed, but they cannot produce transient walking conditions. They also need either a large library of human trajectories to interpolate or a large set of trajectory or control parameters to produce as many walking conditions. The proposed architecture is mathematically involved, but generic and straightforward to be used in walking control, animation or analysis.

11.4.2 Limitations

Walking dynamics in our method is simulated in the 3LP model which relies on linearization assumptions. 3LP is an extension of the LIP model and both assume linear pendular dynamics. 3LP can simulate swing and torso balancing dynamics (in addition to falling dynamics of the LIP model) which allow for simulation of faster walking gaits. However, the resulting motions are valid only where the coupling between horizontal and vertical dynamics is negligible. Our method cannot simulate very large step lengths. Besides, the time-projection controller does not consider such feasibility boundaries. Although this controller always stabilizes the gait, in extreme conditions, it might produce large step lengths that violate decoupling assumptions. These conditions only happen in case of very



Subject

Mass [kg] 70

Height [m] 1.7

Walking Gait

Speed [m/s] 1.00

Frequency [steps/s] 1.7

Double Support [%phase] 15

Clearance [%leg] 5

Torso [deg] 10

Slope [deg] 10

Drag Force [%weight] 10

Step Width [%pelvis] 50

Perturbing Forces

Start Time [%phase] 0

Stop Time [%phase] 100

Strength [%weight]

x : 3

y : 3

Simulation

Real Time [%] 100

☐ Simple 3LP
☑ Human Like
☐ Forces
☐ Treadmill

sample .png

sample .avi

Gait Information

Metabolic Rate [kcal/min]: 2.35

Cost of Transport [J/m/kg]: 0.43

Triangular Asymmetry [deg]: 26.9

Software Version 1.0

View: ☑ 3D ☐ Back ☐ Side ☐ Top

Figure 11.9 – The GUI developed to simulate periodic and non-periodic walking conditions. This picture shows boundaries of different gait parameters within which the synthesized gait stays reasonably human-like. In this GUI, there is a possibility to apply perturbations with different strengths and timing. For steady walking conditions also, the metabolic rate estimates of chapter 9 and the asymmetry index of chapter 10 are printed. The source code of this GUI is available for download at https://biorob.epfl.ch/research/humanoid/walkman.

large disturbances or considerable sudden changes in the desired gait parameters (such as speed or frequency). The linearity assumptions allow for simulation of flat or inclined walking conditions, but not uneven terrains or structured environments. However, if the terrain profile is known in advance, we can design certain height change profiles and solve the new linear time-variant 3LP equations numerically. Besides, the 3LP model does not simulate turning in the current implementation due to nonlinearities. We can remove the pelvis width and allow the 3LP model to turn, but an artificial separation of the two feet is needed (explained in chapter 2). We consider these fundamental improvements for future work.

The present framework can simulate different gait conditions as shown in Figure 11.7, however, some transient conditions are not always easy to model. We cannot simulate torso oscillations unless we linearize the torso and give it a degree of freedom. In this work, the 3LP model assumes a fixed torso angle and finds necessary stance hip torques to realize this assumption. Variations in the external dragging forces are easy to simulate though, since they can be treated as perturbations. A transient change of speed, step frequency and double support time is possible in our current framework. The time-projecting controller can handle them stably.

While the entire framework can simulate a wide and continuous range of walking conditions with different combinations, the natural human-like choice of walking parameters remains un-modeled. Humans can also walk at various gait conditions, but not necessarily be energy optimal or comfortable. When changing the walking speed, humans change the frequency (Bertram, 2005), double support ratio (Cappellini et al., 2006), ground clearance (Ivanenko et al., 2002) and torso orientation (Song and Geyer, 2012). All these parameters change in inclined walking as well (Vogt and Banzer, 1999). Our framework provides the necessary platform to simulate all these walking conditions, but not including human-optimal relations of parameters. Realistic choices of these parameters can be extracted from the related biomechanics literature in walking animations like (Boulic et al., 1990; Li and Liu, 2000). Other human-like walking features such as arm motions or pelvis rotations can be added easily without affecting the overall walking dynamics (Boulic et al., 1990).

11.4.3 Applications and future work

In a trade-off with some of the features offered by physics-based frameworks like rough-terrain locomotion (Mordatch et al., 2010; Coros et al., 2010), we achieved much faster simulation speeds by simplifying the physical model. Our method provides pure mathematical formulas with a minimal dependency on the Eigen library for a matrix inversion (Guennebaud et al., 2010). Our source codes can be easily integrated with other simulators to produce animations on visually more human-like characters. It can be used for crowd-walking simulations as well as animations on portable electronic devices with a limited computational power. Besides, the ideas introduced in this chapter can be used to control humanoids or simulated robots. In particular, our kinematic conversion can be used to produce more human-like pelvis trajectories and thigh-shank-foot coordinations. All the source codes would be freely available online after publication.

12 Conclusion

In previous chapters, we discussed the development of various building blocks in our control architecture (shown in Figure 1.1). It has multiple layers to deal with complexities of a walking system from different aspects. In this chapter, we briefly review the presented material again and conclude simulations and real-robot experiments conducted in the project. We believe it is crucial to fill the reality gap when transferring simulations to the real hardware. Therefore, we first provide a discussion on these challenges, which might be inspiring for other researchers in the field. Going to a higher level, we then continue by discussing our control architecture and the assumptions behind. Further, we analyze the walking gaits from a biomechanical point of view, proceed by enumerating the promising control aspects, and finally discuss possible future directions.

12.1 Low-level hardware complexities

We chronologically started with walking simulations in an ideal environment in chapter 3. However, throughout the project, we noticed multiple physical effects that can potentially make the control setup challenging. The proposed hierarchical architecture helped us identify these challenges in a systematic way and propose appropriate solutions in a modular manner.

12.1.1 Sensory noise

COMAN is equipped with multiple sensory devices (Colasanto et al., 2012). It has a high-end IMU which provides clean signals, but other sensors require further signal processing. COMAN has precise optical encoders that provide clean motor position and velocity signals. However, after the series elastic elements on the link side, the encoders only have 12 bits of resolution which introduce quantization noise when building the kinematic chains. In practice, we used motor shaft velocities with link positions, which together approximated the overall state of the robot with acceptable precision. The velocities are used for state estimation, and for compensation of Coriolis forces in inverse dynamics which become significant in very fast and dynamic motions. Given the ideal stiffness level of springs in COMAN and the limited range of joint velocities it can produce, our assumption practically remains valid. The dead-zone functions we used to filter the quantization noise and backlash errors in the walking experiments of chapter 8 actually produced larger errors in state estimation. Our powerful foot-stepping strategy is, however, able to recover from these errors as shown in chapter 8.

Strain-gauges of COMAN (in the joints and the feet) have considerable noise which is filtered by onboard DSPs (Tsagarakis et al., 2013). This introduces a significant delay, and besides, since the bias point changes quite rapidly during experiments (mostly with the temperature), we had to re-calibrate in each trial. The delays can cause oscillations which could be canceled by extra damping. However, we preferred to tackle this problem by a combination of feed-forward terms and a disturbance observer which limited the influence of our delayed joint-torque measurements (Le Tien et al., 2008). The observer is practically used to compensate Coulomb frictions (only 4% of the maximum joint torque) and provide transparency. This strategy was fine for the dynamic balancing tasks of chapter 3, where the inverse dynamics controller imposed the dynamics and demanded relatively high torques. For the walking experiments of chapter 8, where the choice of frequency was more natural (according to pendulum dynamics), we needed less torques in the hip joints, almost comparable to the Coulomb frictions. While more transparency was needed in this case, technical problems with the torque sensors and the delayed control loop practically prevented us to use torque sensors and torque control for walking. The 3LP model in this case helped by producing more consistent trajectories which were easier to track. We also relied on the mechanical compliance of springs to capture the impacts. Although we used position control for walking, we acknowledge that using torque control with inverse dynamics can improve tracking and compliance together which is considered for future work.

12.1.2 Backlashes and mechanical softness

We expected the series elastic elements to absorb impacts and thus protect the hardware. But soon after starting our walking experiments, significant joint backlashes appeared in the knee joints, and to a lesser extent in the ankle and hip joints. This problem came from permanent spring deflections and torque sensor strain gauges which open space inside the encapsulating flanges. Despite a regular maintenance which is necessary, we believe that the control algorithm should be robust against these problems to some extent.

Inverse dynamics and pure torque control are somehow robust to backlashes when the feed-forward term (of the motor model) is dominant. A small down-scaling of the estimated Coulomb friction (through disturbance observer) can also stabilize the system further. Backlashes and soft springs become more problematic in position control, where we can stiffly control motor positions (via the optical encoders on motor shafts), but not the actual link positions. We do not actually want to control link positions stiffly. This cancels the mechanical compliance of springs which is essential for impact absorption. Absolute encoders provide a good estimate of link positions. However, these values do not necessarily match the desired angles (because of the backlash and soft springs). Depending on the severity of backlash and spring softness, we can have significant tracking errors.

The spring softness can be somehow compensated by a feed-forward bias (as a function of required torque) in the desired motor position trajectories. The backlash, however, could be filtered in a higher level. The discrete behavior of backlash is relatively deterministic at a joint level. It highly correlates with the joint torque, but remains hard to estimate robustly, given that it is mixed with spring deflections in the encoder signals. We have solved this problem for the walking experiments of chapter 8 by using dead-zone functions. We minimally apply them on a few Cartesian variables which mix all the noisy sensory data together (with forward kinematics). At this higher level, all mechanical plays together seem to have a more random effect as shown in chapter 8. Our foot-step adjustment is also strong enough to stabilize the system with and without the dead-zone functions, although we acknowledge that such functions cannot be easily extended to other Cartesian tasks.

12.1.3 Control delays

COMAN has individual motor-board controllers connected through an Ethernet network. While pinging individual joint boards takes at most a millisecond, a continuous data exchange with all the motor controllers at the same time introduces a delay of almost 20ms. Although we can control the robot with an effective rate of 1kHz, in practice, the communication latency limits the performance. Using feed-forward terms in all control stages (for the motors, full-body inverse dynamics, and the 3LP model in footstep planning), we can improve the dynamic consistency of control inputs with the system (Pratt and Williamson, 1995; Mistry et al., 2010). This makes tracking easier and reduces the control effort which is subject to significant delays in our robot.

The above low-level challenges were the most important limiting factors in real hardware experiments. Other issues such as Coulomb frictions, model mismatches, temperature effects and yaw observability are typical in other robots as well. We refer to chapters 3 and 8 for detailed discussions of these issues. Next, we focus on challenges arising from restricting assumptions behind our proposed control hierarchy.

12.2 Hierarchical control challenges

Our hierarchy is based on planning and control of states in the Cartesian space. We have forward and inverse kinematics/dynamics layers to perform such conversion in different types of controllers discussed in previous chapters. While these models are obtained from CAD files and are reasonably precise, there are certain assumptions imposed by the template models we use which limit the conversion precision in different balancing and walking conditions.

12.2.1 Singularities

Both the LIP and 3LP template models used in chapters 2 and 8 are linear, meaning that they keep the CoM in a constant height plane. While this assumption brings computational advantages in our MPC and time-projection controllers, the resulting motion in the robot becomes crouched and less human-like similar to (Pratt et al., 2012; Feng et al., 2016). Such posture has significant energetic consequences and causes hardware damages in the long term. Besides, it is not always easy to plan optimal CoM height trajectories in the Cartesian space unless some information about the full model is available (Inman et al., 1953). We attempted to tackle this problem by using soft constraints in our inverse kinematics/dynamics algorithms. We can command an unfeasibly high CoM while the low-level controller tries to achieve this task as much as possible (i.e., going to the maximum height where at least one of the knees gets stretched completely). We studied this idea in chapter B and identified different challenges including joint position/velocity limits for which we proposed a nonlinear optimization. This solution is validated for balancing tasks on the real hardware, but not for walking motions. However, is the idea of maximum knee-stretching a good solution for walking? In other words, does it produce human-like vertical CoM motions in different walking speeds?

12.2.2 CoM height

Producing human-like limb coordinations with our template models is challenging for many reasons. In the normal fixed-length Inverted Pendulum model (IP), collision and push-off works are needed to redirect the CoM velocity (Kuo et al., 2005), and the resulting trajectories would sharply change directions during phase transitions. In flat-foot walking and stretched-knee conditions, it is almost impossible to apply the vertical push-off force needed for those sharp transitions. Therefore, the idea of maximum knee-stretching is not a good solution for walking. The LIP model solves this problem, though results in a very crouched posture in the stance leg. Using both IP and LIP template models, in fact, we often ignore the feet in the real robot and only rely on the ability to modulate the CoP. The feet segments can make trajectory generation more complicated, though they help to produce smooth vertical motions for the CoM (Adamczyk and Kuo, 2013). This can be done by going to the toes and stretching the leg while effectively lifting the heel during the stance phase. In flat-foot walking, however, the CoM has to go down during phase transitions to keep the heel on the ground. In human-like animations of chapter 11, we attempted to propose a better solution for our trajectory generation method. This new method effectively creates pelvis arcs from the CoP points, not from the heel positions (demonstrated in Figure 11.3). When rolling on the toes, the robot loses controllability (Griffin et al., 2017), though it can have more realistic CoM vertical motions. The idea proposed in chapter 11 works for animations and is not yet validated on the real hardware.

12.2.3 Swing dynamics

The LIP model used in chapter 2 can produce relatively fast locomotion patterns for two main reasons:

- MPC control and foot-stepping strategy are both very powerful in capturing steady-state errors (i.e., model mismatches) (Feng et al., 2016; Pratt et al., 2012).

- Inverse dynamics can use stance-ankle torques to compensate for swing dynamics.

Remember from simulations of chapter 2 that we did not use the CoP modulation strategy to stabilize the motion and left this actuation dimension for the inverse dynamics layer. However, when attempting to realize MPC-based walking with inverse dynamics on the real robot, we noticed that the torque tracking performance is not precise enough to compensate for swing dynamics. In other words, due to torque tracking delays and fast dynamics of walking:

- The real robot was not able to reach the desired footstep locations in time.
- The CoM moved in an unexpected direction most of the time because of such tracking problems.

Inspired by these observations, we took the following steps to achieve walking on the real hardware in chapter 8:

- We developed the 3LP model described in chapter 6 to produce dynamically more consistent Cartesian trajectories.
- We used position control and the low-level firmware of COMAN for faster tracking with shorter delays.
- We directly commanded the final footstep position (calculated through time-projection) to let the hip controller reach it stably at the end of swing phase.

Remember from time-projection control theories of chapter 7 and Figure 7.6 that once perturbation disappears, the projecting controller produces a constant footstep adjustment until the next touchdown event. The MPC controller of chapter 2 also showed a similar property, but we used smooth swing-foot trajectories to reach the final target position over time. In fact, ideal torque tracking in simulations and the powerful inverse dynamics controller (used in chapter 2) could provide a convincing tracking precision. On the real hardware, however, we needed to command the final adjustment directly to compensate for tracking delays.

12.2.4 Torso balancing dynamics

The LIP model used in chapter 2 is blind to torso balancing dynamics, i.e., the effect of stance hip torques that keep the torso upright during locomotion. We used full-body inverse dynamics and IMU feedback to guaranty an upright torso in the simulations of chapter 2. In real hardware experiments of chapter 8, however, we used the 3LP model to produce dynamically consistent Cartesian trajectories. Although the IMU feedback is still used in our hardware walking experiments of chapter 8 like (Pratt and Pratt, 1998), we believe the feed-forward precision provided by inverse dynamics is practically not needed anymore when using the 3LP model.

12.2.5 Arm dynamics

Both the LIP and 3LP models assume a single mass for the entire upper-body of the robot. In COMAN, the arms are comparable to the legs in terms of mass and length, i.e., similar pendular dynamics. The inverse dynamics controller used in chapter 2 could keep the arms fixed by compensating their pendulum dynamics (although this strategy is not natural (Collins et al., 2009)). Therefore, it was convenient to command CoM trajectories directly. On the real robot, due to the mechanical springs in the shoulders which have lower stiffness compared to other joints, the two arms naturally move back and forth during locomotion. We normally keep the upper-body joints fixed in our inverse kinematics or dynamics controllers. However, in state estimation, arm oscillations directly influence the CoM variables at frequencies very close the stepping rate. In our walking experiments of chapter 8 on the real hardware, we measure and control the pelvis positions instead of CoM. Besides, we always lift the arms by bending the elbows. This strategy reduces the length of these limbs and increases their natural frequency.

12.3 Walking gait analysis

The proposed architecture is entirely based on classical control and mechanical models. We decompose the control problem into multiple abstraction levels and propose a precise solution that works in simulation and on the real hardware, though in a limited range of speeds compared to the human. Thanks to various biomechanical studies performed in the final stages of our project, we obtained a better view of hardware and software limitations. This section will cast light on the proposed control architecture from a different perspective.

12.3.1 Missing vertical pelvis motions

Motion planning with linear template models and precise tracking of constant pelvis-height trajectories can successfully produce a walking gait, but less human-like. The robot needs to crouch more to take longer steps while keeping the pelvis height constant. Geometrically, the maximum knee-stretch happens at the touchdown and toe-off moments. However, human walking gait is different. As discussed in chapter 11, at the touchdown moment in Figure 11.6, the pelvis is closer to the front heel than the trailing heel. The forward progression of CoP during the stance phase keeps the walking gait symmetric, but around mid-foot positions instead of the heels. More precisely, the triangle made by the two legs is symmetric around the average CoP position. Since the pelvis is much closer to the front heel at phase transition moments in human, and since the front knee is almost stretched at this moment, we can conclude that the trailing heel can never reach the ground even with a maximum knee-stretch in the trailing leg. In brief, at phase transition moments:

- Because of the feet, the heel-pelvis-heel triangle is asymmetric.
- The front leg is almost stretched, and both legs have the same length (thigh+shank segments).

- Conclusion: Keeping the same pelvis height, even if we stretch the trailing leg completely, the trailing heel cannot reach the ground at this event.

This simple argument suggests that at some point before the phase transition (refer to recorded heel trajectories of Figure 11.5), the stance leg goes on the toes and thus, the stance heel is lifted (shown in Figure 11.6). Apart from controllability issues of standing on the toes, the feet can make the gait asymmetric. At the same time, they can minimize vertical excursions by lifting the pelvis when going on the toes (Adamczyk and Kuo, 2013). Without feet (i.e., compass gait), the heel-pelvis-heel triangle might be symmetric, but vertical excursions are more pronounced (for the same step length) (Gregg and Spong, 2009). Such vertical motions are costly in terms of potential energy and CoM velocity redirection work (refer to chapter 9) (Kuo et al., 2005). In conclusion, even though we can simulate CoP progression in the 3LP model of chapter 6 and produce asymmetric gaits, we can never account for human-like vertical excursions of the pelvis with this simple model. This is a purely geometric problem and less related to the assumption of dynamic decoupling between horizontal and vertical motions in the 3LP model.

12.3.2 Footstep adjustment

In the abstract template level according to Figure 1.1, we use LIP or 3LP models to plan and stabilize the motion. The stabilization part is straightforward, meaning that after all MPC optimizations in chapter 2 or complex calculations of time-projection in chapter 7, the resulting footstep adjustment is, in the end, a simple function of state errors. We use CoM position and velocity errors in MPC control and the LIP model of chapter 2. Later in chapter 8, we consider swing dynamics and include swing foot position and velocity errors as well. In literature, this strategy is very well known and established. The SIMBICON controller used a linear function of CoM errors with tunable gains (Yin et al., 2007). Coros et al. used the LIP model (Coros et al., 2010), Mordatch et al. used MPC on both LIP and SLIP models (Mordatch et al., 2010) and the capturability framework proposed a systematic way of footstep adjustment for capturing external pushes (Koolen et al., 2012). Due to constant horizontal velocities of the CoM during ballistic motions in hopping robots, Raibert was able to only use the CoM velocity for attack angle adjustment (Raibert et al., 1984). In walking gaits due to exponential falling dynamics, one needs to consider relative CoM position errors as well. In this regard, the LIP model provides a good basis for modeling falling dynamics in walking. A reasonable footstep adjustment rule in walking is, therefore, a function of CoM errors probably with time-variant coefficients. The task-specific constant coefficients of SIMBICON controller might not be optimal (Yin et al., 2007). But thanks to the LIP model, the capturability framework (Koolen et al., 2012) and MPC controllers (Mordatch et al., 2010; Faraji et al., 2014a) can implement this time-dependency efficiently. Through the development of 3LP model and time-projection, we attempted to generalize these frameworks by considering swing dynamics and providing automatic gait generation and simple control look-up-tables. Our method is, therefore, well suited for humanoid robots with relatively heavy legs (like COMAN), although it supports normal robots as well. It can also generate high frequency gaits in which swing dynamics becomes important.

12.3.3 Gait parameters

Many traditional walking algorithms used on humanoid robots produce quasi-static gaits with a precise ZMP tracking (Sakagami et al., 2002; Zhao et al., 2008; Kryczka et al., 2015). This can be simply noticed from large lateral bounces in the movies. Compared to human and from the viewpoint of pendular natural frequencies, many robots in literature perform walking in frequencies much different from their human-like frequencies. In quasi-static gaits, adult-size robots might take step phases as long as 1.5s while kid-size robots might spend almost 1s in each step. It is well-known that in human, the gait frequency increases with speed monotonically (Bertram, 2005). However, the mentioned numbers are far from normal walking gaits (about 0.8s and 0.5s in adults and kids respectively (Hausdorff et al., 1999; Liu et al., 2008; Cappellini et al., 2006)), and even different from human-like frequencies in slow walking speeds (Bertram, 2005). Recent control algorithms can produce more dynamic motions while recovering from stronger pushes (Pratt et al., 2012; Feng et al., 2016). This is done through footstep adjustment instead of CoP modulation.

We performed a comprehensive analysis of energetics in chapter 9 to study walking mechanical effects in different combinations of speeds and frequencies. We suggested three determinant cost components for human walking: push-off work, swing dynamics and leg lift costs. These components become important in large step lengths, fast frequencies and short step lengths respectively (refer to figure 9.3). Walking at high frequencies is costly for the hip joints; it produces large transverse torques in the stance foot and also introduces position or torque tracking challenges. To achieve faster walking speeds with linear models, taking larger steps is challenging for the knees (in terms of larger torques in more crouched postures). A better choice is to increase frequency and speed at the same time to keep the step length constant. However, depending on the hardware platform, the swing dynamics becomes more critical in higher frequencies. Few recent methods can reach faster walking speeds by using a step time of 0.5s on adult-size robots. The reason is to take shorter steps to avoid violating linearity assumptions. In human, this frequency is typically used when walking faster than the normal speed (i.e., faster than 1-1.25m/s). In conclusion, there might be ways to compensate for constant and flat CoM height and foot orientations to some extent. However, they may come with a cost of enforcing non-human-like dynamics, i.e., too slow or too fast stepping frequencies (Bertram, 2005). Simple walking models without swing dynamics can discover walking to running transitions, but they do not estimate the energies of high-frequency walking precisely. (Srinivasan and Ruina, 2006).

12.3.4 Flat walking consequences

The fourth cost component introduced in chapter 9 quantifies weight-support costs which are negligible in regular human walking. In flat-CoM walking conditions and crouched postures, however, it becomes significant. We studied flat walking conditions in chapter

9 to get inspiration for humanoid control tasks. We also attempted to formulate a singularity-tolerant inverse kinematics algorithm in chapter B to better prepare our lower level blocks in the control hierarchy of Figure 1.1. This algorithm improves the energy efficiency up to three times for balancing tasks in two scenarios where crouched postures and unfeasibly high CoM positions are commanded (to achieve straight-knee postures). However, if we command infeasible pelvis heights in walking gaits and only rely on linear models, the resulting pelvis trajectory would contain sharply intersecting arcs (at phase transition moments). These arcs are not only human-like but also not realizable because of missing push-off force exertion mechanisms in flat-foot walking conditions. Constant-height pelvis trajectories, on the other hand, may require a small push-off force only in horizontal directions. The typical straight posture of the trailing leg is not considered a limitation in this case. Such analysis indicates a fundamental problem in our trajectory generation method. Our latest animation method of chapter 11 suggests one possible solution in which dynamics and control are handled in the template space, and the kinematic conversion method adds vertical motions, assuming decoupled horizontal and vertical dynamics. This conversion method uses the full model though, i.e., using the feet and knee joints at the same time. Therefore, the resulting trajectories are not considered template trajectories anymore. In other words, we need to consider lower-limb geometries to produce human-like vertical motions for the robot.

12.3.5 Leg lift principles

Remember from walking experiments of chapter 8 that velocity saturation in the knee joints of COMAN limited our capability to lift the leg quickly. Therefore, we introduced a second strategy which used the hip roll joints to rotate the pelvis. In human, however, the situation is different. As discussed earlier, the stance leg goes on the toes sometime before the phase transition moment. At the same time, the stance knee joint starts to flex a bit which is shown in Figure 11.6. In a singular straight-knee posture, geometrically, the knee joint has to rotate a lot to reduce the overall length of the leg by few centimeters. In a flexed posture, however, a much smaller rotation is needed to achieve the same length reduction. When using linear models and flat-foot walking conditions on the robot, we mentioned that the trailing leg is almost straight at the push-off moment. Because of a singular posture, we need to rotate the knee very quickly to achieve some ground clearance. In human, this is somewhat easy because the knee joint already starts to flex before the push-off. Therefore, although human trajectories have inspired the design of COMAN (Colasanto et al., 2012) and the actuators are ideally capable of performing human-like locomotion, control limitations and model simplifications in our control hierarchy introduce such unnecessary difficulties. In conclusion, going on the toes and flexing the knees may facilitate the leg lift mechanism.

12.3.6 Thigh-shank-foot coordinations

As discussed in the walking experiments of chapter 8, we needed adaptive foot-lift strategies to account for tracking errors due to mechanical springs in perturbed walking scenarios. Remember that the commanded gait was in-place walking in these scenarios and the stance hip only regulated the torso angle. Therefore, the pelvis could naturally move forward with an external push while the desired stance hip angle was still zero. In fact, the rest of joint angles were also calculated according to an in-place walking gait. Due to such postural mismatches in case of perturbations (refer to Figure 8.9), we needed to provide an additional leg lift to make sure that no scuffing happens in early swing phase. In general, increasing the amount of leg lift reduces such risks in perturbed walking conditions (Voloshina et al., 2013). In human, however, this is done adaptively depending on the terrain and somehow optimally (Wu and Kuo, 2016). Inspired by various studies in the literature (Yin et al., 2007; Coros et al., 2010; Mordatch et al., 2010; Pratt et al., 2001), we use the stance hip to regulate the torso angle both in 3LP model and hardware experiments. In other words, since we do not use CoP modulation, we do not have enough control over the CoM position, i.e., the robot naturally falls. Therefore, we believe that in a generic walking controller that supports perturbed walking conditions, the entire stance leg posture may depend on the actual relative position of the stance foot (with respect to the pelvis), not the desired position. The same principle could be applied to the swing leg as well, although the swing hip ideally tracks the desired swing trajectory most of the time.

As suggested in chapter 9, the energetic cost of walking is dominantly determined by push-off work, swing dynamics, leg lift motions and weight support torques in normal walking conditions. These mechanical effects relate to the step length, walking frequency, ground clearance and vertical pelvis excursions respectively. The control problem, i.e., the strategy to generate and stabilize the motion, is equally influenced by the choice of these parameters. A large step length breaks linearity assumptions, a fast frequency introduces hip tracking issues, a high leg-lift is hard to achieve with limited knee-joint velocities, and a crouched posture or flat walking is costly in the knee joints. So far in this chapter, we discussed how human overcomes these issues by lower-limb joint mechanisms and dynamic walking, i.e., the ability to stabilize the motion with minimal dependency on ankle torques. In other words, the human can quickly stabilize the motion despite going on the toes for a few moments. This stabilization may be done by footstep adjustment or using the upper-body momentums, but to a lesser extent with CoP modulation.

We wrap up this section by concluding that having feet in the robot is advantageous from various aspects compared to compass gaits (i.e. without feet). In more traditional control algorithms, this body segment and the ankle torques help stabilize the motion. We believe that this is a minimal advantage and probably produces a weak stabilizer depending on feet dimensions. We might damp the falling behavior with CoP modulation like (Coros et al., 2010) in certain parts of the phase. However, it is better to move the CoP on the toes and lift the stance heel whenever needed. With this strategy, we may lose controllability, but gain a nice thigh-shank-foot coordination that helps push-off and leg-lift mechanisms. This would reduce CoM vertical motions and besides helps the hip joint to flex earlier which facilitates the swing motion (refer to Figure 11.5). Given all these discussions, to what extent can our control theories support dynamic walking?

This will be discussed in the next section.

12.4 Promising control aspects

We motivated our control hierarchy by the fact that we require fast and straightforward models to predict the motion and find meaningful control inputs. Among CoP modulation, footstep adjustment and momentum stabilization strategies, we focused on footstep adjustment as a robust method to capture strong perturbations. This strategy is a meaningful way to handle speed transitions since steady-state footstep patterns also change during such transitions. In chapter 7, we extensively discussed the fact that footstep adjustment does not have an immediate effect until the next phase starts. However, we also emphasized that it is necessary to adjust the footstep location online before the next touchdown happens. Therefore, we somehow need to know the future evolution of state variables to find proper footstep locations online, even though they do not have any immediate influence like CoP modulation or momentum regulation strategies.

12.4.1 Capturing systematic perturbation by foot-stepping

The theoretical background introduced in this project through development of the 3LP model and time-projection control translate to very simple control rules. Although we derive them based on linearity and fixed phase-time assumptions, they result in powerful stabilization rules that cover a wide range of walking conditions. In the walking experiments of chapter 8, we showed that our controller could easily comply with constant external dragging forces to produce emergent walking gaits. We also showed that internal perturbations might distort the phase timing (refer to Figure 8.13), but footstep adjustment is yet powerful enough to stabilize the motion. Such robustness comes from dynamically consistent trajectories of the 3LP model as well as dynamically consistent control rules of time-projection. In ideal simulation environments also, as discussed in chapter 2, a robust walking was achieved on uneven terrains and slopes, with model errors, sensory noises and also with control delays. Thanks to the whole-body compliance offered by inverse dynamics and torque control, the robot was able to comply with unforeseen terrain variations without any planned adaptation. More precisely, despite a fixed timing in our gait generation method, early or late touchdown events did not perturb the gait significantly. Therefore, our footstep adjustment methods may go beyond the assumptions used to derive them mathematically. In other words, even un-modeled systematic or steady-state system errors (such as slopes or dragging forces) may not make the system unstable unless they are too strong. Foot-stepping strategy and compliance can together make the control algorithm robust against systematic and occasional perturbations to some extent.

12.4.2 Compliance

In the balancing tasks of chapter 3, we proposed a powerful control architecture in which the robot shows whole-body compliance. Due to hardware maintenance issues though, we were not able to use this framework in the walking experiments of chapter 8 and only relied on the mechanical springs to achieve compliance. The inverse dynamics controller of chapter 3 may:

- Reduce impacts on a robot like COMAN with rigid feet. This would significantly reduce vibrations.
- Help better control the torso orientation through a unified whole-body framework.
- Reduce rotational slippages in the stance foot by adapting to the terrain shape compliantly.
- Prevent the feet from rolling or tilting when external pushes are applied.
- Help to damp the falling behavior by CoP modulation.

We used different control strategies and dominantly relied on the mechanical springs to achieve these goals in our walking experiments. However, the inverse dynamics controller would unify all these strategies together and provide a better control framework. This remains to be tested, ideally on a robot platform with better torque tracking than COMAN.

12.4.3 Swing dynamics

In different parts of the project, we attempted to add a sense of internal limb dynamics to the template models. Various Cartesian motion planning approaches in literature optimize CoM, momentum, contact points and contact force trajectories. Using the point-wise modeling technique of chapter 4, we split the concentrated point-mass into multiple limb masses and re-wrote the equations of motion. Such new model adds no extra free variable to the dynamic equations used in other popular Cartesian planners. However, it allows for modeling of internal momentum exchanges between the limbs which may produce physically more consistent trajectories. Our formulations further support inclusion of approximate limb inertias as well, assuming that limb lengths do not change much during locomotion. This assumption might be realistic in slow walking conditions, but in complex multi-contact locomotion, the case might be different. Through the development of point-wise modeling in chapter 4, the 3LP model in chapter 6 and motion planning in chapter 5, we were looking for approximate ways of modeling important dynamics in the robot. We believe having a sense of some dynamical effects inside the template planner is better than ignoring them completely. However, increasing the precision by including complex geometric and dynamic effects is more against the concept of template planning. In particular, we want to keep a reasonable level of details to assure fast computations.

12.4.4 Model-based control

One major concern throughout the project was portability of the proposed method to different robots. We chose a model-based control framework for various reasons.

The models helped us understand different mechanical effects, not through equations and mathematics, but through multiple handy simulations in which we could change parameters and immediately identify the effect. In the absence of practical models, e.g., in approaches based on learning algorithms, particle swarm optimizations, grid searches or data-driven approaches, it is probably time-consuming to understand different physical effects, since a huge offline computation/data collection effort might be needed. Physical effects do not necessarily refer to walking dynamics; they might cover sensory imperfections, actuator dynamics, frictions, mechanical plays and body softness. More importantly, a quick test of candidate control mechanisms and parameters helps better designing the controller. Mechanical models, at different levels of abstraction, helped us identify and distinguish between walking dynamics and the rest of other effects mentioned. Therefore, we were able to model/simulate these effects easily (in stable walking conditions) and obtain more intuition for the real hardware. Our control hierarchy is easily scalable to different robots. We encode robot-specific information into the models and derive control rules accordingly. This process naturally does not require re-tuning of parameters, offline optimizations or data collections.

12.4.5 Decomposition of planning and control

Another primary concern throughout the project was the ability to control a wide range of walking gait conditions, i.e., we were not interested in finding local controllers. It is popular in literature to train a library of local controllers for different walking conditions and link them via interpolation or transition rules (Kelly and Ruina, 2015; Manchester and Umenberger, 2014; Gregg et al., 2012; Lee et al., 2002; Choi et al., 2003). Instead, we aimed at finding a unified framework in which we do not need to re-tune the controller depending on state variables (or the desired gait). This philosophy can simply translate into a feed-forward and feedback control design in which dynamics and physics are all encoded in the feed-forward trajectories. However, we learned that a generic controller should not be blind to system dynamics. To keep such generality in our latest controller which is time-projection of chapter 7, we used 3LP dynamic equations to derive DLQR controllers. Therefore, our control design is matching dynamics of the desired feed-forward 3LP gaits. Our MPC controller of chapter 2 also features the same property as the LIP model is used in both planning and stabilization at the same time. The resulting footstep adjustment rules depend on robot's anatomical properties and the walking frequency. However, other gait parameters discussed in chapter 11 such as speed, step width, external dragging forces and torso angle appear as constant terms in the 3LP equations, and therefore, they are not considered in derivation of DLQR and time-projecting controllers. This makes our footstep adjustment rules generic for all these walking conditions. The slope angle may have some influence though by changing the pendular dynamic properties.

12.5 Future directions

Our primary goal in this thesis was to perform robust and fast locomotion on a real robot. Although human-like locomotion was not the primary goal, over various biomechanics studies we learned that faster speeds and energy optimality could be achieved in our robot by making the locomotion more human-like. However, limited hardware capabilities, as well as the linearity assumptions in our control algorithm, prevented us from achieving a fast human-like motion with the real robot. In this thesis, we proposed better solutions for dynamic walking and control with footstep adjustment instead of CoP modulation strategy. Our controllers can recover from stronger pushes and perturbations. The maximum walking speed we achieved on the real hardware is very close to the maximum capabilities, however, we believe that this robot is probably capable of achieving faster speeds and more human-like motions with improvements in the control method. In this section, we are going to discuss possible improvement as well as further applications of the models and tools developed.

12.5.1 Better template models

We aimed at improving the conventional LIP model by adding swing and torso balancing dynamics. Various other template models exist in the literature and provide different levels of abstraction (refer to Figure 6.1). However, they do not provide closed-form solutions due to non-linearity and the need for time-integration. The 3LP model proposed in chapter 6 is a linear model that can produce a wide range of walking conditions discussed in chapter 11. There are other three-mass models in the literature which provide partially linear equations, but with imposed swing dynamics (i.e., artificial swing foot trajectories). The 3LP model encodes dynamics by hip and ankle torques which makes it more realistic and generic. Using similar linearity assumptions, we can extend the 3LP model to have arms, limb segments or a moving torso. As explained later, a moving torso may help to implement the momentum regulation strategy. Other features might have negligible contributions in the range of walking conditions we can achieve with the real robot. In complex multi-contact motions of chapter 5, since all the limbs can move, we think it is better to represent the torso by a trapezoid to account for pelvis and shoulder width. This may improve geometric inconsistencies primarily.

During different optimizations, especially in the posture planning tasks of chapter 4, we noticed that having limb segments in the template model introduces various unnecessary numerical difficulties including singularities and joint limits. Our solvers can solve these issues but take more optimization iterations to tackle the nonlinearity. We believe that prismatic length boundaries or simple work-space representations may encode reachability constraints efficiently and simplify optimizations. We prefer to denote each limb by a vector which represents a virtual prismatic actuator (like chapter 5). However, we are looking for better objective functions that quantify the cost of all joint torques in the limb only by these geometric vectors and contact forces, at least in static conditions. With these functions we can plan optimal postures without considering the limb segments individually and thus, avoid numerical difficulties.

12.5.2 Better planning

We argued that the primary limiting factor in achieving faster speeds in our method is probably a geometric constraint. Planning with linear template models assumes prismatic stretchable legs. However, the seemingly complex lower-limb mechanism can physically support more human-like gaits while we do not take advantage of this capability. We believe that our foot-stepping strategy can easily support dynamic walking, i.e., going on the toes or performing compass gaits. This controller shows a compliant walking behavior in the sense that it can handle systematic model errors which translate to steady-state tracking offsets. These errors could be external dragging forces, slopes, control delays or even further, violations of horizontal and vertical decoupling assumptions. In these cases, one probably needs to command a slightly different desired velocity to compensate for such steady-state errors. We believe our controller would be powerful enough to stabilize a human-like fast walking gait. However, 3LP trajectories are widely ignoring lower-limb functionalities and advantages.

We have attempted to produce more realistic motions in the human-like animations of chapter 11. In this method, we rely on 3LP dynamics and stabilization, but slightly modify the constant vertical trajectories of 3LP to add pelvis excursions as well as human-like lower-limb coordinations. This new method is not validated on a full model yet, neither in simulations nor on the real hardware. However, we have used all the observations on real hardware as well as the intuitions obtained through biomechanic studies to properly design such kinematic converter. In this method, human-like lower-limb trajectories are obtained by applying an intuitive inverse kinematic algorithm on the 3LP states. We avoid any time-integration on the full model which keeps the entire method as fast as the 3LP model itself. Besides, we include necessary geometric details and make sure the resulting joint-trajectories are feasible. We hope to conduct extensive human experiments in future to quantify this method and compare it with human. The synthesized trajectories are eventually useful for locomotion on exoskeletons, powered lower-limb assistive devices, humanoid robots and virtual characters in gaming or animation applications.

12.5.3 Better stabilization

The 3LP model ideally supports both CoP modulation and foot-stepping strategies in the current format introduced in chapter 6. We require to free the third pendulum (which represents the upper-body) and let it move to implement the momentum control strategy. Our derivation of DLQR and time-projection controllers is general enough to support addition of such new states and control input dimensions. On the resulting model which would be complex but still linear, we can perform MPC optimizations as well to include inequality constraints and a variable phase timing. We believe that the three CoP modulation, foot-stepping, and momentum strategies can work together to recover from much larger external pushes. Shorter swing times also help the footstep adjustment to capture the falling dynamics faster and prevent large overshoots. These improvements are considered for future work.

12.5.4 Better lower-body control

Our real-hardware experiments with inverse dynamics are limited to the balancing tasks of chapter 3. We found the whole-body compliance offered by inverse dynamics very powerful in the sense that it can improve robustness in perturbed walking conditions. In future, we have to adapt our formulations to support point-contacts, e.g., when going on the toes. However, since our new planning method of chapter 11 involves calculation of time-trajectories for the individual joints, we might need to revise inverse dynamics formulations and adapt them to joint-space instead of task-space. We believe that for faster gaits and perturbed walking conditions, compliance is a vital component that should be integrated into our walking controller.

12.5.5 Further biomechanic studies

We mentioned that our latest trajectory generation method of chapter 11 requires further validation. We are interested in conducting a unified set of human experiments and obtaining human-like geometries for all the different walking conditions mentioned in that chapter. The recording of dynamics would also help to quantify any violation of the linearity assumptions. Besides, we would like to perform a quantitative comparison of our time-projection controller with human behavior. In this regard, we may need an experimental setup in which we can apply controlled and measurable perturbations to humans. Besides, our new trajectory generation method might further improve the energetic model of chapter 9 in the sense that it can simulate more complex link functionalities. The resulting kinematics may also help to perform inverse dynamics to find equivalent muscle activations. The 3LP model, time-projection control, human-like animation, energetic model and the study of asymmetries are all packed in a lightweight walking animation software that hopefully gives useful insights to researchers in robotics, biomechanics and computer graphics communities. This package is distributed along with this thesis.

12.6 Final words

This thesis presented a hierarchy of components for walking control and used different tools to reduce dimensionality and map the complexities of walking into simplified models. The philosophy behind such a control paradigm was the need to predict future due to intrinsic instabilities of the system and lack of controllability over single steps. The model predictive control allows us to plan footsteps and ensure walking stability even though the robot is unstable at each instance of time. Despite mathematical complexities, with this strategy, we can easily generate walking at different speeds and recover from significant disturbances which are not possible with conventional methods. In the proposed hierarchy, our main contribution has three folds:

- Proposing a new simplified modeling technique inspired by molecules to speed up calculations considerably.
- Deriving a simplified model that provides better dynamical matching with the real

system.

- Developing an analytical controller that adjusts footstep locations optimally by a model-based future prediction.

A significant strength of this work lies in inspirations from human locomotion and biomechanics principles. By decomposing the system into smaller components, we could understand different mechanical effects individually and treat them simply by allowing them to play their roles. An alternative was to acquire a better robot and apply some black-box control while being blind to these effects. However, the proposed approach can effectively comply with many of these effects at the same time which makes it flexible and portable to other robots as well. More importantly, because of such inspirations, the tools and methods developed in this project seem useful to analyze human locomotion too. Therefore, going beyond search and rescue which was the primary application of this project, we have developed components that could be useful in rehabilitation, computer gaming, and animation communities. In future, we hope to see extensive applications of our tools and methods in all these fields. Specially with faster computational tools, our modeling techniques can help producing more dynamic and agile motions together with online reactions and control in both bipedal and multi-modal locomotion.

A Force Estimation

This appendix describes an extended application of full-body dynamic equations in balancing tasks. Since the robot is equipped with contact-force sensors in the feet and joint-torque sensors, we can use this information to estimate interactions with the environment without having physical touch sensors on the skin. We present an efficient method to estimate external forces on the robot without having physical touch sensors mounted. We consider simulation scenarios in which the robot performs compliant balancing motions with the inverse dynamics controller, and external forces are applied to different body parts arbitrarily. Since the controller is compliant, these forces can change the posture and at the same time result in different joint-torques. Using the IMU sensor and joint encoders, and also assuming non-sliding contacts, we estimate accelerations in the robot through an optimization method first. Together with joint torques and contact forces, we then use these accelerations in the equations of motion to find a residual vector presumably corresponding to an unknown source of external force. Our method searches over all the body segments and finds the force application point with an analytic method. It can ideally handle dynamic scenarios as demonstrated, although due to hardware issues, we did not find the chance to validate it on the hardware. This part primarily focuses on balancing tasks and interactions with the environment. It is worth mentioning that this method is not used in the walking push recovery scenarios of this thesis in chapters 2 and 8 although it could be very valuable[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji and Auke Jan Ijspeert. *"Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics."* In Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, pages 5564–5571, 2016.

---

[1] All the videos of this chapter could be found at https://youtu.be/Fnws2rtgIR4

A.1 Background

In humanoid robotics control, dynamics-based approaches are becoming more and more popular (Faraji et al., 2015; Feng et al., 2015; Kuindersma et al., 2014). Such trend has many motivations behind, especially compliance and precision. Compliance is a crucial feature of a humanoid robot in the tasks that involve human-robot interaction. However, going further, compliance can be beneficial for the robot as well, mainly regarding safer operation and avoiding self-damage. Compliance has two different types, active and passive (Wang et al., 1998). Passive compliance is mainly inspired by human tendons and their energy storing role in addition to absorbing impacts that often exist in loco-manipulation tasks. However, from the control perspective, certain tasks might require different levels of compliance in the task or joint space. Such compliance cannot be realized by passive elastic elements that are often in series with actuators (Pratt and Williamson, 1995) in most new humanoid robots (Colasanto et al., 2012). Here, the other type of compliance becomes very important, i.e., active compliance. The control algorithm takes advantage of sensory data available on the robot to generate actuator policies that behave as if a real spring is there in the robot. Such virtual elastic element (Pratt et al., 2001) can be used in control paradigms such as balancing, manipulation or locomotion tasks.

In this chapter, based on our previous compliant balancing controller of chapter 3, we propose a complementary component that estimates external disturbances as well, using the available sensory data. This architecture is briefly presented in Figure A.1. By fusing available sensory data, we can estimate global states and take advantage of dynamics equations to control the robot. On the other hand, we can also use same equations and dynamics sensory information to identify mismatches and associate them to unknown sources of error. In the following, we first briefly describe the proposed architecture.

A.1.1 Inverse dynamics

As soon as active compliance is considered, we also need to think about available kinematics and dynamics information. Such information can be used in model-based or model-free control approaches that try to perform the desired tasks while providing a certain level of compliance. Kinematics information gives the current state of the robot and determines how precise the desired task is being executed. On the other hand, dynamics information can be used to determine interactions and compliance. Among different control approaches, inverse dynamics is becoming very popular as it can deal with the complex nature of humanoids efficiently.

A.1.2 State estimation

The complexity of humanoids can be both due to high dimensionality and a floating-based nature. The latter is practically more important in fact, although it seems not if computational power is solely considered. Inverse dynamics requires the full state of the robot including joint configurations and base variables, expressed in the global frame.

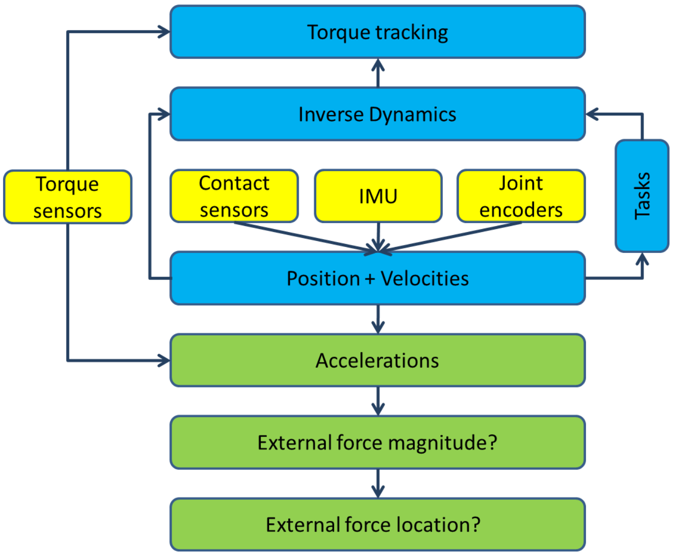


Figure A.1 – Control/estimation architecture proposed to estimate external disturbances. The sensory data available from the robot are fused together to estimate the current state of the system, including positions, velocities, and accelerations. These states are used to perform desired tasks using inverse dynamics and torque tracking blocks. On the other hand, using the redundancy in the dynamics information available, we can estimate external disturbances and their locations.

Since the robot does not have direct measurements of base variables, there is often a geometric filter block that fuses all available sensory data to determine the full state. Therefore, inverse dynamics is always accompanied by another complex routine that has certain limitations.

A.1.3 Torque tracking

Apart from observing the state, inverse dynamics also needs a precise torque tracking implemented at the actuator level. Although models are not very accurate, precise torque tracking can still increase consistency of the model with the real robot. This block requires measurements of either direct joint torques, or corresponding series elastic deflections or the electrical current in the actuators. Compared to the traditional position controlled robots (Sakagami et al., 2002; Moro et al., 2011), torque measurement can provide additional information about interactions or un-modeled phenomena.

A.1.4 Dynamics inconsistency

The idea behind using the model is to generate control policies that realize tasks of certain priorities (Herzog et al., 2014). The knowledge required for many of these tasks is already available in the model. Balancing task, for example, is mainly linked to gravity compensation and keeping the static stability which relates to dynamics and kinematics information respectively. We developed a different formulation of inverse dynamics problem in chapter 2 which considers system dynamics, desired tasks and physical limitations altogether to generate proper joint torques. This general and versatile formulation was later combined with state estimation and torque tracking algorithms and successfully tested on our real COMAN robot in chapter 3. An essential feature of our implementation is pure torque control. We do not put secondary loops over joint velocities or positions that require integration of desired joint accelerations, coming out of inverse dynamics. This simple feature improves precision, compliance, and stability by avoiding interfering stiffnesses.

Our formulation let us define all the tasks in Cartesian space. There are certain trajectories considered for the end effectors (hands, feet, CoM and torso orientation) and followed by Cartesian PID regulators that generate accelerations in the end. The notion of active compliance appears here since our inverse dynamics compromises different tasks by predefined priorities. In case of an external perturbation, for example, the desired acceleration is nonzero, while the robot does not move anymore (refer to (Faraji et al., 2014a, 2015) for real demonstrations). In fact, inverse dynamics is blind to the external disturbance and produces a force that reacts to the perturbation, proportional to the displacement.

A.1.5 External force estimation

In the present work, we want to investigate dynamics errors and infer information about perturbations. This is possible only thanks to the three previously mentioned blocks which estimate the global states precisely, plan consistently and execute the plan with fast torque controllers. Thanks to 6D contact force/wrench sensors and all joint torque sensors in our platform, apart from the control, we can re-write the equation of motion with available sensory data. We also take advantage of other kinematic constraints to determine joint accelerations. Now, we can simply investigate if there is any mismatch in the equation of motion and associate it with an external force. Figure A.2 demonstrates a simple scenario which gives intuition on how joint torques and contact forces can determine external force positions.

This idea is not new however and appears in literature, in the context of model identification (Traversaro et al., 2014) or impedance control (Le et al., 2013). The main motivation behind looking at the equation of motion is to avoid direct measurement due to many reasons.

- In some cases, there are concerns about pricing, weight or complexity. In robotics arms, for example, force estimation by joint torque sensors is popular, either for identification of new objects or better control performance (Colomé et al., 2013; Le et al., 2013). These robots are however much simpler in terms of degrees of freedom and more importantly, fixed to a table. Therefore, it might be reliable enough to estimate contact forces by the equation of motion, as state estimation is simpler. Note that this method can be applied to position-controlled robots as well, as far as they have torque sensors.

- On the other hand, it is less practical to put many small sensors together to form a

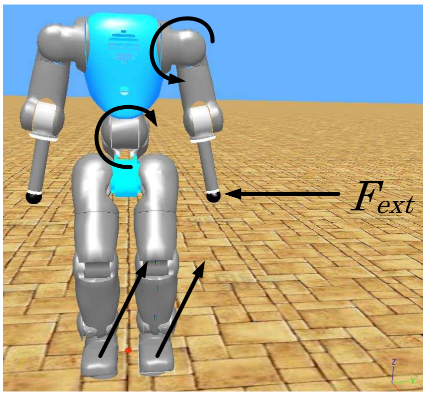


Figure A.2 – An intuitive figure, demonstrating how an unknown external force can be estimated based on the internal sensory data. In this case, the lateral force exerted on the hand tip produces an additional torque in all the joints. However, we can only measure the torque around the joint pivot vector in each joint. Therefore in the scenario of this figure, only the shoulder roll sensor detects a larger torque. After the arm, this torque appears in the waist joint as well, going down to the contact 6D force/wrench sensors. The proposed algorithm is limited to a single external force, although it can potentially explore more forces under certain conditions. We also assume a point-wise perturbation without an external wrench to simplify the problem.

tactile suit with many cables which increase the weight. The artificial skin proposed in (Mistry and Righetti, 2012; Strohmayr and Schneider, 2013) is useful however in certain manipulation tasks. The flexible skin proposed in (Noda et al., 2012) is also very useful in human-robot interaction. However, it is not always easy to anticipate how humans especially elderly and kids would touch the robot.

These shortcomings in the literature of force estimation have motivated us to propose a general approach to estimate external force strengths and locations on the whole body of the robot, without adding any other physical sensor.

The proposed approach uses contact force information to infer the strength of a single external force and then searches over the surface of all links in the robot to find the best matching correspondence. The first part to find force strengths is inspired by a straightforward approach proposed in (Kaneko et al., 2012) which uses the second Newton law, written for Center of Mass (CoM). For the second part, we propose a similar algorithm used in (Likar and Zlajpah, 2014), though computationally more efficient. Searching over body segments of the robot requires exact geometries of the surfaces. Likar approximates link shapes of an ABB manipulation arm by cylinders (Likar and Zlajpah, 2014) and then sets up constrained optimization problems to find the force application point on all cylinders. In our method, however, due to the complexity of our humanoid robot in terms of the number of links, we approximate link shapes by ellipsoids which look realistic and propose an exact alternative solution of similar optimization problems. The core idea behind our approach is to break down unknown Jacobians and to find local positions in closed-form.

We can cover the whole body of the robot, compared to the limited application of explicit physical touch sensors. We are also not restricted to the end-effectors and basically, search for both forces and application locations over the whole body of the robot. Although the presented work is yet limited to a single external force, it can be instrumental in model identifications and refinements as well as human-robot interactions, for example detecting pushing or pulling forces. In this chapter, we do not use the estimated forces to improve the balancing controller. However, this can be another application of the proposed method. In next section, we will introduce details of our search algorithm as well as underlying assumptions. Next we will demonstrate the results and discuss the performance and in the end, conclude the chapter with proposing possible future extensions.

A.2 Methodology

For this chapter, we base the estimator and simulations upon our previous inverse dynamics controller of chapter 3 where a multi-stage controller performs state estimation and balancing using inverse dynamics. A closer look at these publications will indeed give more insight on the application of the proposed method. Note that in this work, we do not feed the estimated external forces back to the controller to improve the balancing performance.

As mentioned previously, the proposed estimation method has three main stages, acceleration, force strength and force location estimation. The general equations of motion for the whole body of the robot are basically written as:

$$M(q)\ddot{q} + h(q,\dot{q}) = \tau + \sum J^T F \tag{A.1}$$

Where $q \in \mathbb{R}^{6+23}$ denotes the full state of the robot (including global position and orientation of the base), $M(q)$ denotes mass matrix, $h(q,\dot{q})$ denotes gravitational, centrifugal and Coriolis forces, τ is joint torques, J denotes translational and rotational contact Jacobians and F denotes contact forces or wrenchs. In this set of equations which follows Kane's convention (Kane and Levinson, 1983), the first six rows are actuation-less ($\tau_{1..6} = 0$) and describe global frame dynamics of the floating based robot. Apart from dynamics equations, we also have to consider fixed contact constraints:

$$\dot{J}(q)\dot{q} + J(q)\ddot{q} = \ddot{x} \tag{A.2}$$

Where $\ddot{x}$ represents contact acceleration which is zero in case of static contacts. A more detailed discussion about these constraints can be found in chapter 3.

We have a direct measurement of joint torques and contact forces as well as q and $\dot{q}$ for the joints. We have also developed another state estimation algorithm in chapter 3 which serves as the basis of this chapter. This algorithm determines full vectors q and $\dot{q}$ by fusing all sensory data on the robot and contact constraints. However, to estimate additional external forces, we need to know joint accelerations as well. So before mentioning the main force-estimating stages, we introduce another stage to determine joint and global accelerations. In the following, these three stages are described.

A.2.1 First stage: acceleration estimation

In addition to contact forces F and M, joint torques τ, joint positions q and velocities $\dot{q}$, our robot COMAN is equipped with an Inertial Measurement Unit (IMU) on the pelvis which provides linear accelerations and angular velocities as well as orientations, internally calculated in the IMU by fusing other two variables. Although most of the sensory information is used in the underlying state estimation of chapter 3, we use dynamics information (forces, torques and accelerations) again in this stage to find the full vector $\ddot{q}$. As a rough estimation of joint accelerations, we differentiate joint velocities to obtain $\ddot{q}_{\text{opt}}$. COMAN has optical encoders to measure the velocity directly. However this differentiation is not accurate, because the velocity is measured before the series elastic elements, not after on the link side. We similarly differentiate the gyro velocity to obtain an estimation of base's angular acceleration $\ddot{q}_{\text{gyr}}$. A quadratic optimization problem is then formulated to fuse these estimations with IMU accelerations $\ddot{q}_{\text{IMU}}$, the equation of motion (A.1) and contact constraints (A.2).

$$
\begin{gathered}
\min_{\ddot{q}} V_{Q_{\delta_E}}(\delta_E) + V_{Q_{\delta_R}}(\delta_R) + V_{Q_{\delta_I}}(\delta_I) + \sum_{i=1}^{n_c} V_{Q_{\delta_i}}(\delta_i) \\
M(q)\ddot{q} + h(q,\dot{q}) = \tau + \textstyle\sum_{i=1}^{n_c} J_i^T F + \delta_E \\
\ddot{q}(1..3) = \ddot{q}_{\text{IMU}} + \delta_I \\
\ddot{q}(4..6) = \ddot{q}_{\text{gyr}} + \delta_G \\
\ddot{q}(7..29) = \ddot{q}_{\text{opt}} + \delta_O \\
\dot{J}_i(q)\dot{q} + J_i(q)\ddot{q} = \ddot{x}_i + \delta_i, \; i = 1..n_c
\end{gathered}
\tag{A.3}
$$

Here, we define $V_Q(\psi) = \psi^T Q \psi$ to represent a quadratic function. The role of slack variables δ in this formulation is to create flexible fusion of sensory data and reject noises, similar to Kalman filtering. In fact optimization approaches are becoming more popular for filtering sensory data in humanoid robots (Xinjilefu et al., 2014a) as they provide the option to add inequality constraints, compared to the conventional Kalman filtering (Bishop et al., 2001). It is also easy to prove that such optimization is equivalent to Kalman filtering, if the quadratic cost matrices are equal to the inverse of covariance matrices in Kalman formulation. In our implementation however we tune these matrices manually and keep them constant. Although we lose optimality, the performance still remains acceptable. The choices for positive definite cost matrices are heuristically diagonal-shape with the following numbers $Q_{\delta_E} : 1$, $Q_{\delta_I} : 10^4$, $Q_{\delta_G} : 1$, $Q_{\delta_O} : 1$ and $Q_{\delta_i} : 10^4$ to ensure constraint satisfaction and give more importance to the IMU accelerations than velocity differentiations. This quadratic problem is solved in less than $100 \mu s$ by CVXGEN QP optimizer (Mattingley and Boyd, 2012). Note also the structure of our generalized floating base coordinate vector where the first three elements are global translation (of the pelvis), the second three are global rotation (of the pelvis) and the rest are joint coordinates.

A.2.2 Second stage: force strength estimation

In this work, we limit the formulation to detection of a single external force, although it can be easily extended to detect an additional external moment at the same time. The strength of external force can be simply calculated by looking at the first three lines of the equation of motion. Since in the previous stage, more importance is given to the IMU reading rather than the equation of motion, the vector δ_E estimates the projection of external force onto the space of floating base generalized coordinates. Regarding the structure of our state vector, the first three elements of δ_E indeed estimate the external force vector itself. Note that the vector δ_E can already be used in the balancing controller to improve the performance without performing the next stages of this estimator. This is more generic, without the limiting assumption of a single force or ellipsoid geometries. In the present work, however, we do not explore this possibility and leave it for future works. Note also that in case of static balance, all accelerations are zero and the external force can be calculated more simply. However, we prefer to keep the mass matrix (and the first stage accordingly) to explore dynamic motions. Disabling the first stage is indeed equivalent to a static algorithm which is compared with the full estimator later in the results section.

A.2.3 Third stage: force location estimation

Once the extra force is calculated, we can consider the equation of motion again. In fact the resulting δ_E in the first stage accounts for an unknown external force F_{ext} that is applied to an unknown point P, located on an unknown body link B_i of the robot:

$$\delta_E = {J_P^v}^T R_{B_i}^T F_{\text{ext}} = {J_P^v}^T \hat{F}_{\text{ext}} \tag{A.4}$$

Where R_{B_i} denotes body's rotation matrix. Note that all Jacobians are originally written in the local frame in Kane's formulation. Therefore we have to transform the globally expressed F_{ext} back to the local frame of B_i to obtain $\hat{F}_{\text{ext}}$.

In the following, we are proposing a procedure to find the body index i and the force location P. Note that P is expressed in the local frame of the body B_i. Therefore, we do not exactly know the Jacobian J_P^v unlike many other approaches that assume known Jacobian (Colomé et al., 2013). Figure A.3 demonstrates B_i, the exact link shape, approximate ellipsoid and the unknown point P. Considering the frame B_i located on the CoM of the body link B_i, we can use known Jacobians to find the translational velocity of the point P:

$$\begin{aligned} x_P &= x_{B_i} + R_{B_i} P \\ \dot{x}_P &= R_{B_i} J_P^v \dot{q} = R_{B_i}(v_{B_i} + S(\omega_{B_i})P) \\ v_{B_i} &= J_{B_i}^v \dot{q} \end{aligned} \tag{A.5}$$

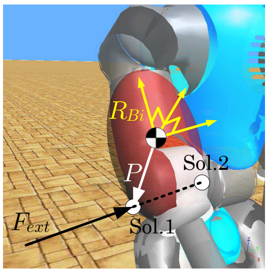


Figure A.3 – Demonstration of the upper arm of the robot, approximated with an ellipsoid of a proper size. The local frame B_i is placed on the CoM of the link, which we consider the centroid of the ellipsoid as well. When an external push is applied on the link at a local point P, the algorithm intersects a line calculated based on the measured dynamics variables with all the ellipsoids over the whole body and then selects the one making the minimum cost.

Where $S(\psi)$ represents skew-symmetric matrix, obtained from the vector ψ. Similarly, we can find the angular velocity of the body B_i on which the point P is located:

$$\begin{aligned} R_P &= R_{B_i} \\ \omega_{B_i} &= J^{\omega}_{P}\dot{q} = J^{\omega}_{B_i}\dot{q} \end{aligned} \tag{A.6}$$

Now, we can link the two sets of equations together:

$$\begin{aligned} R_{B_i} J^{v}_{P}\dot{q} &= R_{B_i}(J^{v}_{B_i}\dot{q} + S(\omega_{B_i})P) \\ &= R_{B_i}(J^{v}_{B_i}\dot{q} - S(P)\omega_{B_i}) \\ &= R_{B_i}(J^{v}_{B_i}\dot{q} - S(P)J^{\omega}_{B_i}\dot{q}) \end{aligned}$$

Therefore, the Jacobian of the point P can be written as:

$$J^{v}_{P} = J^{v}_{B_i} - S(P)J^{\omega}_{B_i} \tag{A.7}$$

Which is written in terms of known Jacobians $J^{v}_{B_i}$ and $J^{\omega}_{B_i}$ for the B_i frame. However we still need to determine the vector P to obtain the full Jacobian. Coming back to the equation (A.4), we can write:

$$\begin{aligned} \delta_E = {J^{v}_{P}}^{T}\hat{F}_{\text{ext}} &= (J^{v}_{B_i} - S(P)J^{\omega}_{B_i})^{T}\hat{F}_{\text{ext}} \\ &= {J^{v}_{B_i}}^{T}\hat{F}_{\text{ext}} - {J^{\omega}_{B_i}}^{T}S(P)^{T}\hat{F}_{\text{ext}} \\ &= {J^{v}_{B_i}}^{T}\hat{F}_{\text{ext}} - {J^{\omega}_{B_i}}^{T}S(\hat{F}_{\text{ext}})P \end{aligned} \tag{A.8}$$

As a result, we can obtain a relation for P:

$$S(\hat{F}_{\text{ext}})P = -({J^{\omega}_{B_i}}^{T})^{+}(\delta_E - {J^{v}_{B_i}}^{T}\hat{F}_{\text{ext}}) \tag{A.9}$$

where $()^+$ denotes pseudo-inversion and $J^{\omega}_{B_i}$ and $J^{v}_{B_i}$ are known. Although there are three equations in (A.9), due to rank deficiency of skew symmetric matrices, we effectively have two equations with three unknown components of P.

Since we also know that the force application point is located on the body surface of the robot, we have to intersect the resulting line obtained from (A.9) with the surface of each link B_i to obtain two points maximally, in case of convex link geometry. Our proposed method, however, approximates each link with an ellipsoid which matches the actual link shape as much as possible. Such ellipsoid is formulated around the CoM of B_i:

$$\frac{P(1)^2}{a^2} + \frac{P(2)^2}{b^2} + \frac{P(3)^2}{c^2} - 1 = 0 \tag{A.10}$$

Where the semi-principal axes have a length of a, b and c, proportional to the diagonal elements of the inertia tensor for each link with slight hand tunings. Our geometric model is based on exact CAD models of the robot (Colasanto et al., 2012). In future, however, we plan to set up optimization routines to adjust the positioning and scaling of the ellipsoids for better matching.

Now, we have the third equation to find all components of the variable P. A simple approach is to take two variables out of (A.9) in terms of the third variable, replace them in (A.10) and solve a polynomial of degree two. In the end, depending on the polynomial, we might have zero, one or two solutions. In case of having no solution, we simply set the discriminant of the second-degree equation to zero to obtain two points close to the ellipsoid. The proposed search procedure takes the following steps:

1. Calculate the full acceleration vector $\ddot{q}$ through the first stage.
2. Extract the external force vector F_{ext} out of the first three lines of (A.1).
3. Search over all body links B_i: solve the closed-form equations for P.
4. Calculate the norm of the error vector $e_1 = \delta_E - J_P^T \hat{F}_{\text{ext}}$.
5. Calculate the distance of P to the ellipsoid by taking the left hand side of equation (A.10), $e_2 = \text{lhs (A.10)}$.
6. Pick the link with minimal error $e_1 + e_2$.

This procedure is in fact very fast compared to optimization based approaches proposed in (Likar and Zlajpah, 2014). All the stages together can run in real-time ($100\mu s$) on a moderate Core i5 CPU. The algorithm can also be extended to other geometric shapes easily as the idea of intersecting lines with the volumes is the same. The key point to obtain closed-form solutions is indeed to decompose the Jacobian into known and unknown parts. In the next section, we demonstrate simulation results over different tasks, followed by discussions on the performance and generality of this approach.

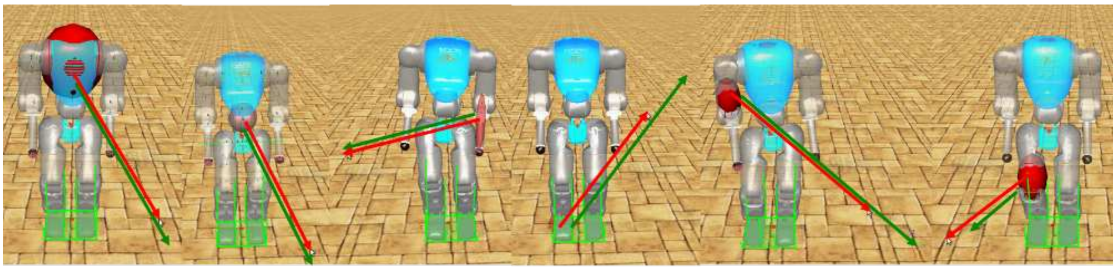


Figure A.4 – Few positions where the balancing robot has stabilized after being pulled by an arbitrary external force. In this figure, red arrows are pulling forces applied manually in the user interface of our simulation software, while green arrows show the final estimation. Ellipsoids are transparently shown in red, demonstrating the body link subject to the pull, determined by the algorithm. One can verify that arrows have minimal shift while the estimated body links are matching the actual links (where the red arrows originate from). In general, detection of a force applied to the upper links is easier as it influences the torque in many joints down to the foot. Also, detecting forces on bigger links is easier too, since finding an intersection is more probable. Therefore, one can deduce that the forces being applied to the torso are the easiest to estimate.

A.3 Results

As explained, at each time step, the search process determines the force vector, body link and the local point P where the external force is applied. In this implementation, we do not set up specific time-filtering algorithms to deal with noises and other uncertainties. Instead, we rely on our state estimation methods to provide a clean and stable fusion of sensory data. It should also be noted that in practice, out of the two solutions obtained for the best link, we only consider the one corresponding to a pulling force, i.e., the point where the external force vector goes out of the body link. On the real robot, however, the other point should be selected as disturbances are mainly pushing forces unless someone can pull the robot.

A.3.1 First scenario: exploring different body links

The first scenario is characterizing the resulting precision for all the links of the robot. Here, we apply forces to various links of the robot, while performing stable and compliant balancing task. We let the robot stabilize and demonstrate the final force estimation and the corresponding body link in Figure A.4. One can obviously see that in static situations, the algorithm finds an acceptable estimate of the external force, although sometimes the body link is very narrow like the lower arm. In these cases, the intersection is hard to find, regarding uncertainties available in the procedure. The estimated force application points P are acceptable too, although there is a shift sometimes due to our state estimation method or passive elastic elements (joint springs). Since on the real robot, link-side encoders (after springs) do not have enough resolutions, we always use the motor-side encoders to build kinematic chains. In the simulations of this work also, we follow the same convention to be more realistic.

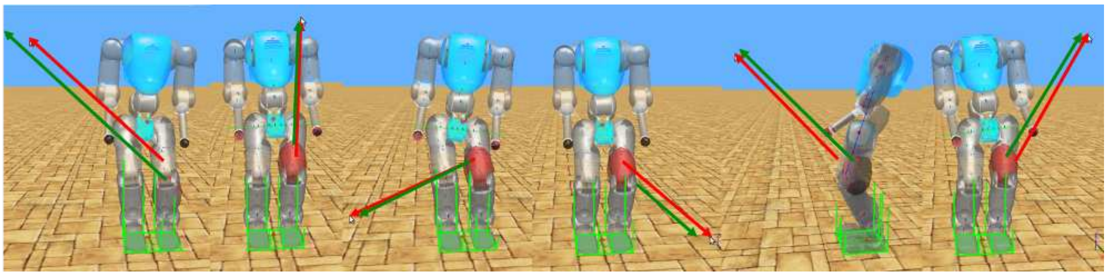


Figure A.5 – Demonstration of different pulling forces applied to the left thigh. Red arrows are actual force vectors while green arrows demonstrate the estimated force. Although the matching is good in most of the cases, the left robot can not estimate the body link correctly.

Although the precision is convincing, there are situations where the force cannot be uniquely estimated. For example, imagine in Figure A.4, there is an external force applied to the tip of the hand, directly upwards. In this case, the algorithm might report any of the links in the arm, as the external force produces the same torque in the shoulder joint. Finding a unique solution is even more challenging in dynamic motions, depending on the force magnitude, agility of the task, closeness of the force to the joint pivot, etc. The external force vector can always be estimated, i.e., the second stage is robust as long as being provided with correct accelerations. However, there should be another high-level method that decides whether the reported link and P variables are correct or not. We leave the design of such complex method for future works. However, in this chapter, we would like to characterize the performance and argue about challenging parts of this algorithm.

A.3.2 Second scenario: exploring force directions

In the second scenario, we investigate the effect of force direction and keep the body link choice fixed. Figure A.5 demonstrates a scenario where different pulling forces are applied to the left thigh of the robot. Detecting forces on the two legs is also challenging as there is redundancy in the loop created when both feet are supporting a portion of the weight. One can observe that in Figure A.5, there is only one case where the algorithm reports a different body link, i.e., shank. This can be due to many reasons such as the inaccuracy of ellipsoidal approximations, redundancy problem or alignment of the external force with the shank. However, it is promising that the reported body link is the shank, close to the original thigh link.

A.3.3 Third scenario: dynamic motion

An alternative to the proposed method is to completely remove accelerations from the equation of motion, assuming that the robot is static. In this case, we can disable the first stage of the algorithm as it is not required anymore. This alternative is, of course, simpler and less demanding regarding online calculations. However, it only works when the motion is very slow. Remember that there are uncertainties in the model, state

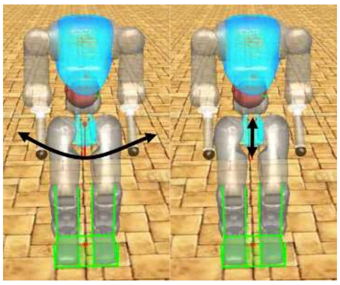


Figure A.6 – In the dynamic scenario, the robot is performing a combination of two different tasks at the same time: following a sinusoidal trajectory of 15cm up and down and turning around the yaw axis with the same regime for 50 degrees.

estimation, geometry approximation, etc. If these uncertainties are dominant, i.e., their magnitude is larger than the external force, the algorithm might jump between different body links and provide a less reliable estimation. Therefore, we are interested to know whether accelerations can help in dynamic motions or not. In other words, the first stage of the estimator is supposed to compensate dynamic effects so that the error vector δ_E merely depends on the external force and uncertainties.

To investigate the role of accelerations, we consider a second version of the estimator where accelerations are set to zero. Such static estimator is compared with the original one over some dynamic motions in Figure A.7. The robot is performing up and down motions with turning around the yaw axis, demonstrated in Figure A.6. Meanwhile, we apply different forces and compare the estimated values coming out of the two static and dynamic variations. As expected, it is obvious that the dynamic formulation performs better in estimating the force magnitude, even in the absence of external force where the magnitude is zero. It can also provide a more stable body link estimation, although it fails over the course of last perturbation where the ellipsoid is too narrow to find intersections. Overall, accelerations are helping to keep consistency and compensate the fast motions of the robot. Therefore, the proposed algorithm performs better than static formulations, with a minimal computational requirement.

In case of small force magnitudes or swift motions, another high-level algorithm is needed to threshold the estimated force and filter out jumps in the body index. Adding more damping to the controller will also reduce these jumps. The required precision, of course, depends on the final application of the proposed method. In case of human-robot interaction or model identification, indeed a statistically correct estimation over a certain period of time is expected, and occasional jumps are tolerable. In an online control, however, one should carefully filter these jumps to avoid instability.

A.4 Discussion

In this chapter, we proposed an estimation architecture of multiple stages to determine the global state of the robot, accelerations and external disturbances at the same time. We fuse available sensory data to calculate global states and from the mismatch observed

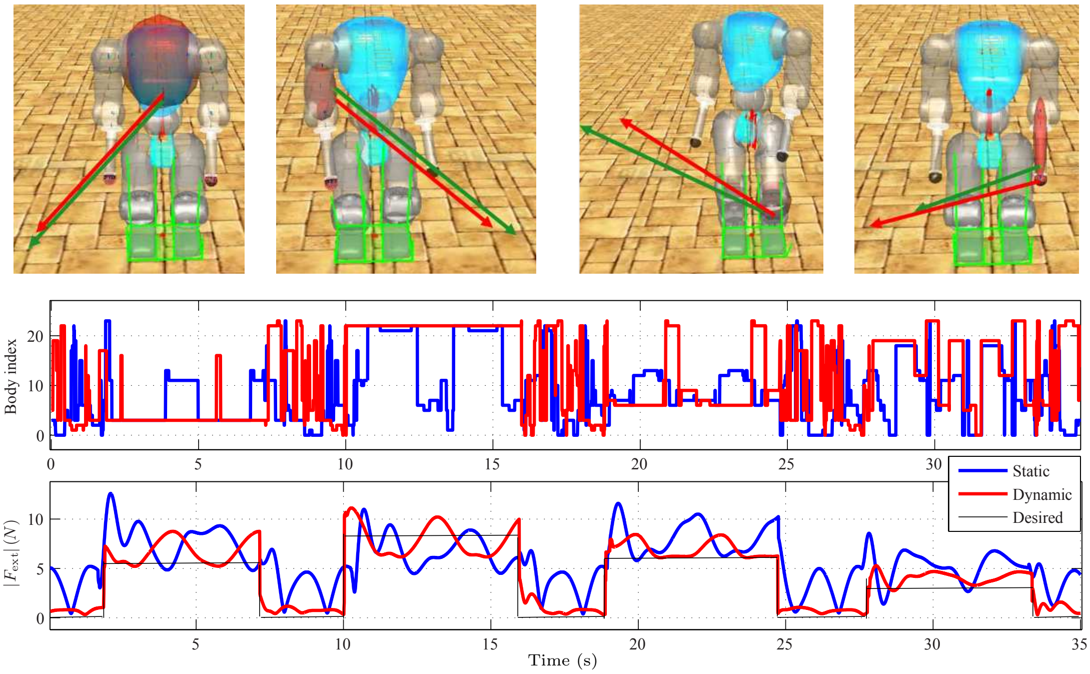


Figure A.7 – Comparing performance of the full algorithm in dynamic motions with the case where the first stage is turned off. In this case, accelerations are zero and the algorithm is static. Here, we apply four forces to different body links and plot the estimated force magnitude and the body index. The geometry of the force is also shown on top of the curves. It is obvious that the dynamic estimator performs better, at least over the first three perturbations. It stably reports the body index and rarely jumps to another value. The last pull is more challenging to detect however, as the ellipsoid is too narrow to find intersections. Note that the curves are not meaningful when no force is applied, though the dynamic estimator can estimate the force magnitude (which is zero) better than the static estimator. Full demonstrations are available in the multimedia attachment.

in the equation of motion, we estimate the external force vector. Then we search over all body links to find the point where this external force is applied. The procedure requires knowledge about link geometries which we approximate by ellipsoids to reduce the burden of calculations. Although the proposed method requires optimization for determining accelerations, in the next stages, the closed-form solutions help speeding up the search process.

We characterized the estimation precision for different body links, force directions and dynamic motions. The algorithm proves to provide acceptable results for rather small forces (equivalent to 0.5-1kg compared to the weight of robot which is about 30kg). This is promising for application on the real robot although a higher level filter is required. The process is, in fact, challenging due to model mismatches, state estimation inaccuracies, ellipsoid approximations, singularities, redundancies and the noise. In this chapter, we tried to implement the algorithm as realistic as possible, simulating conditions on the real robot. The proposed algorithm has certain advantages over other similar works:

- Estimating force magnitude and direction.
- Finding force application location.
- Working in dynamical motions as well.
- Computationally very fast.
- Handling floating based calculations.
- Handling many degrees of freedom.

However currently, the method is limited to detection of a single external force. In future, we would like to extend the method to detect external moments and possibly multiple forces. Detecting a single moment is rather straightforward since it can be simply detected from the contact wrenches measured. Considering exact link geometries as well as testing the method on the real robot are also part of our future work. Please refer to the multimedia attachment for full demonstrations of the scenarios discussed in this chapter.

B Inverse Kinematics

We introduce a novel inverse kinematics algorithm in this appendix which is robust to singularities. The motivation behind this study was to improve the performance of balancing and walking tasks by letting the robot stretch the legs completely and avoid high torques in crouched postures. These torques lead to high energy consumptions in the non-ideal actuators of the knees and harm the series elastic elements in the long term. Stretched legs, on the other hand, lead to singularities in the Jacobian matrices which mean controllability issues. They further require swift motions in the knee joints which are not possible on the real robot as well. Therefore, we introduce a nonlinear optimization method in this part to resolve these issues as well as respecting the joint-angle limits at the same time. The proposed method uses similar soft strategies of our inverse dynamics (in chapter 2) to impose priorities between Cartesian tasks, namely tracking the given CoM horizontal positions and compromising for the vertical positions in favor of constraints. The experiments of this part involve various symmetric and asymmetric squatting and balancing motions on the real hardware which show effectiveness of the proposed method. In fact, the safe operational regions used in this chapter were integrated into our inverse-dynamics method as well and validated in simulations. However, due to hardware issues on the torque sensors, they remained to be validated on the hardware in future and we did not report them in the thesis[1].

Publication Note: The material presented in this chapter is adopted from:

- Salman Faraji and Auke Jan Ijspeert. "Singularity-tolerant inverse kinematics for bipedal robots: An efficient use of computational power to reduce energy consumption." IEEE Robotics and Automation Letters, 2(2):1132–1139, 2017.

---

[1] All the videos of this chapter could be found at https://youtu.be/tFx-WaNbNOY

B.1 Background

Bipedal robots are often mechanically very complex, designed to perform various types of tasks apart from walking. The location and the number of degrees of freedom (DoF) are mainly inspired by the human skeletal system, comprising limbs with at least six joints. From a control perspective, it becomes difficult however to plan a trajectory for each joint individually to perform a desired motion at the end-effector level. Therefore, one prefers to transform complex joint-space formulations into Cartesian space (Khatib, 1987), making trajectory planning easier. Using such transformations, one can easily convert Cartesian trajectories to joint motions, required by individual joints in the robot. Cartesian control is popular in manipulation (Khatib, 1987), humanoid balance (Sentis et al., 2010) and locomotion (Moro et al., 2011). The output of this transformation can generate joint positions, velocities, or accelerations. The first two quantities are often used in position-controlled robots (Inverse Kinematics, IK), while accelerations are used more often with the full dynamics model of the robot (Mistry et al., 2010), resulting in desired joint torques (Inverse Dynamics, ID).

Although IK conversion is nonlinear, there are various methods to solve it either in a closed-form (Diankov, 2010) or iteratively (Goldenberg et al., 1985), both suitable for online control. The ID problem is more complex but linear with respect to accelerations, forces, and torques. One can solve this linear system in a closed-form (Mistry et al., 2010) or set up a quadratic optimization problem (QP) to consider boundary constraints (Righetti et al., 2013). These constraints can ensure satisfaction of physical limitations on joint torques, contact frictions, and joint positions. The latter constraint is typically realized by putting boundaries on accelerations based on Taylor series expansion of joint positions. In chapter 3, we successfully combined our QP-based ID algorithm with advanced state-estimation and torque tracking methods and demonstrated various compliant balancing behaviors on the COMAN robot. However, on one hand, the position limit constraint was not effective as it produced largely infeasible accelerations very close to boundaries. On the other hand, the algorithm was not numerically stable enough in singular positions. In practice, operating in crouched postures with bent knees caused higher torques and therefore, less precise tracking performance. Besides, the consequence over the long term was permanent spring deflections in series elastic elements, increased backlash, over-heating and large power consumption in addition to less human-like postures.

In the literature, there are plenty of algorithms proposed for stretched-knee walking. Apart from model-free walking approaches (Van der Noot et al., 2015), which deal with singularity problem differently, a large number of model-based algorithms are based on preplanned knee trajectories, where other joint angles are adjusted to realize desired swing or Center of Mass (CoM) motions (Ogura et al., 2006; Handharu et al., 2008). Likewise, the walking planner proposed in (Kurazume et al., 2005) modifies a parametrized 2D CoM trajectory to limit large knee velocities close to singular positions. For arbitrary balancing tasks, however, such periodic trajectories can not be found. The idea of limiting velocities is inspiring, though, the time linkage between positions and velocities is encoded in the CoM trajectory, not in the IK method. In other words, preplanned Cartesian CoM velocities are found in a way to satisfy limitations on the velocity of the knee joint. The same paradigm is proposed in (Naksuk and Lee, 2005) for balancing tasks, but again, the trajectory planner handles stretched knee postures.

Given desired Cartesian trajectories, the IK or ID method is supposed to find joint trajectories that follow the task as precise as possible while satisfying physical constraints automatically. In this article, instead of focusing on planning trajectories, we aim at studying physical limitations and propose algorithms that handle them in a unified online control setup. We limit our study to IK methods to explore important aspects of joint position boundaries and singularities exclusively. In future, however, force/torque constraints of our previously developed ID method (in chapter 3) will be combined with geometrical constraints studied in this article to address a larger number of hardware limitations.

Stretched-leg postures are more convenient for the mechanical hardware of humanoid robots, but they introduce two major difficulties to the control problem:

1. Singularities: which mainly refer to the alignment of the hip, knee and ankle joints and lead to ill-conditioned Jacobians. Such postures limit controllability and might produce large velocities in the knee joint.
2. Joint position limits: which should be respected together with velocity limitations to avoid impacts, especially in the knee.

Handling these problems requires a robust method that safely approaches the singularity, does not vibrate, and safely leaves the singularity again. Besides, in all these phases, the method should find the closest solution to the desired task. Singularities and joint limits can be handled in planning level or inside the low-level IK algorithm. For any specific task, one can adjust the planner to match better with the geometry of the robot. However, we focus on the baseline IK method to make it robust and general, without modifying Cartesian planners.

For manipulators, robust IK methods (Chiaverini et al., 1994; Buss, 2004) have been developed to deal with singularity conditions robustly. These methods either bring the target position closer to the manipulator (Buss, 2004) or find the closest solution (Chiaverini et al., 1994). The former falls into planner-level category while the latter is more interesting as it provides a generic IK method, normally in the form of least-squares error minimization. To handle stability of trajectories in singular postures, one can add a damping term which is widely used in literature (Chiaverini et al., 1994; Kryczka et al., 2011; Sugihara, 2011). Such damping can improve numerical stability in redundant robots as well (Kryczka et al., 2011). Although one can modify weightings in these unconstrained least-squares optimizations to avoid joint limits (Dariush et al., 2008), expressing them explicitly in constrained optimizations is also popular (Escande et al., 2010). A more general form of such optimization is formulated in (Suleiman, 2016) or (Vaillant et al., 2016) where velocities are strictly bounded or adaptively damped respectively.

All previously mentioned methods are based on time-integration, where the outcome of optimization is being integrated over time to obtain desired joint positions. Another class of IK methods solves the exact nonlinear constraint by performing many iterations in the same time-step (Wang and Chen, 1991). Each iteration here is similar to solving a quadratic program like before, aiming at getting closer to the exact solution. Adding position and velocity limitations as well as singularities are less studied in this class of optimizations, mainly because of computational cost.

Focusing on previously-mentioned geometrical constraints, we formulate a general IK module that handles inequality constraints in the form of nonlinear optimizations. We use generalized-coordinate models instead of per-limb models and go beyond walking, to target arbitrary whole body balancing tasks for our floating-based humanoid robot in 3D. By reimplementing popular IK methods in the literature, we show that getting to singular positions and coming out in a safe manner can most of the time be problematic for IK methods based on time-integration. Therefore, the novelty of this work lies in proper analysis and handling of singularities and joint limitations via the proposed nonlinear method which combines positions, velocities and inequality constraints in the same optimization. Our analysis covers multiple behavioral and computational aspects, proving applicability of the proposed method for the real robot. This is demonstrated for a couple of different symmetric and asymmetric balancing tasks. The structure of this chapter is as follows: in the next section, we formulate different IK optimization problems and present our proposed formulation. Next, we will demonstrate simulations and experiment results, characterizing the performance and handling physical limitations. Finally, we conclude the chapter by discussing possible future improvements in the last section.

B.2 Methodology

The problem of inverse kinematics refers to the conversion of a set of Cartesian trajectories to joint-space. These trajectories which are called tasks hereafter, describe the translational or rotational motion of interesting points of the robot, for example CoM, hands, or feet. Given that sometimes tasks cannot be realized exactly due to the singularities or joint position limits, one might compromise a few, depending on the application and precision requirements. Imagine we have $x \in \mathbb{R}^M$ tasks, where a subset of size $N \leq M$ can be compromised. If the robot has $q \in \mathbb{R}^K$ Degrees of Freedom (DoF), the goal is to find q that satisfies $f(q) = x + \begin{bmatrix}\delta & 0\end{bmatrix}^T$ where the slack variable $\delta \in \mathbb{R}^N$ is to be minimized. Alternatively, one can find $\dot{q}$ that satisfies $[\partial f(q)/\partial q]\dot{q} = \dot{x} + \begin{bmatrix}\delta & 0\end{bmatrix}^T$ and integrate to get q over time. In case of adding limitations, an inequality of the form $l_b \leq g(q, \dot{q}) \leq u_b$ should be satisfied as well, where $l_b, u_b \in \mathbb{R}^K$ represent limits and $g(q, \dot{q})$ could be a nonlinear function. Is it better to optimize positions and then differentiate to find velocities or alternatively, find velocities first and then integrate them to get positions? This question has many aspects, including computational cost, tracking precision, robustness in singularities, ability to approach joint limits and ability to escape from singularities safely. Here, we consider five different IK algorithms and compare them regarding the previously mentioned criteria. The first three (IK_1, IK_2 and IK_3) are common in robotics while the other two (IK_4 and IK_5) are new ones proposed in this chapter.

B.2.1 IK_1: Error integration

Similar to (Chiaverini et al., 1994), one can find joint delta angles based on the task error and then integrate over time. To handle singularities, we also introduce damping factors (diagonal positive definite matrix R), resulting in the following unconstrained quadratic optimization problem:

$$\begin{aligned} &\min_{\Delta q,\delta} \delta^T Q \delta + \Delta q^T R \Delta q \\ &f(q^-) + \frac{\partial f(q^-)}{\partial q^-}\Delta q = x + \begin{bmatrix}\delta \\ 0\end{bmatrix} \end{aligned} \tag{B.1}$$

where q^- is the previous desired trajectory and $\Delta q \in \mathbb{R}^K$ is motion adjustment to be found. The matrix $R \in \mathbb{R}^{KxK}$ is the well-known damping in least-square methods (Chiaverini et al., 1994). Imagine S_N is a selection matrix which takes the N compromised tasks out of the vector x if multiplied from left, i.e. $S_N x$. Similarly, S_{M-N} selects the rest of the tasks. Defining the Jacobian $J = \partial f(q^-)/\partial q^-$ and the error $E = x - f(q^-)$, the optimization of (B.1) has a closed-form solution, calculated by setting the derivative of the Lagrange equation to zero:

$$\begin{bmatrix}\Delta q \\ \lambda\end{bmatrix} = \begin{bmatrix} J^T S_N^T Q S_N J + R & J^T S_{M-N}^T \\ S_{M-N} J & 0 \end{bmatrix}^\dagger \begin{bmatrix} J^T S_N^T Q s_N \\ S_{M-N}\end{bmatrix} E \tag{B.2}$$

where † is Moore-Penrose pseudo-inverse and λ is Lagrange coefficient for $M - N$ exact constraints. The desired trajectory and its derivative are then found by:

$$q = q^- + \Delta q, \quad \dot{q} = \frac{\Delta q}{\Delta t} \tag{B.3}$$

where Δt is the time-step. This fast formulation is equivalent to integrating velocities, only requiring to solve a linear system of certain dimensions. Note that the error, however, converges to zero over time-steps with certain dynamics.

B.2.2 IK_2: Conjugate gradient method

In this method, we perform all iterations in a single optimization at each time-step:

$$\begin{aligned} &\min_{q,\delta} \delta^T Q \delta \\ &f(q) = x + \begin{bmatrix}\delta \\ 0\end{bmatrix} \end{aligned} \tag{B.4}$$

This optimization is solved via conjugate gradient method where each iteration is solved similar to (B.2) with same damping mechanism. The velocities are then found by solving the following optimization:

$$
\begin{aligned}
&\min_{\dot{q},\delta} \delta^T Q \delta + \Delta t^2 \dot{q}^T R \dot{q} \\
&\frac{\partial f(q)}{\partial q} \dot{q} \Delta t = \dot{x} \Delta t + \begin{bmatrix} \delta \\ 0 \end{bmatrix}
\end{aligned} \tag{B.5}
$$

where the time-step Δt is used to preserve consistency with (B.1). This optimization can be solved in closed-form, similar to (B.2). Here, we find the exact solution for both positions and velocities at each time-step, yet without inequality constraints.

B.2.3 IK_3: Integrating errors with inequality constraints

To handle position and velocity limits, one can introduce boundary conditions to (B.1), i.e. constraining the first stage of IK_1. An arbitrary safety criterion can also be defined as a function of q and $\dot{q}$. For example:

$$
g(q, \dot{q}) = \frac{(2q - (q^l + q^u))^2}{(q^u - q^l - 2q_s)^2} + \frac{\dot{q}^2}{\dot{q}^2_{\max}} - 1 \leq 0 \tag{B.6}
$$

which represents an ellipse spanning between minimum and maximum joint limits $q^l, q^u \in \mathbb{R}^K$, allowing for maximum velocity $\dot{q}_{\max}$ in the middle and zero velocity in boundaries (refer to Figure B.5A). Reducing velocity boundaries when approaching joint limits helps avoiding impacts and sudden stopping which is harmful for the mechanical hardware. The variable q_s is a safety margin for the joint limit. If position controllers of the real robot overshoot in certain trajectories, this variable helps avoiding reaching the limit and producing impacts. Equation (B.6) is element-wise, though we avoid indices for simplicity. One can define polygon-based safe regions as well, similar to (Suleiman, 2016). The quadratic optimization problem for this stage will be:

$$
\begin{aligned}
&\min_{\Delta q,\delta} \delta^T Q \delta + \Delta q^T R \Delta q \\
&f(q^-) + \frac{\partial f(q^-)}{\partial q^-} \Delta q = x + \begin{bmatrix} \delta \\ 0 \end{bmatrix} \\
&g(q^- + \Delta q, \Delta q / \Delta t) \leq 0
\end{aligned} \tag{B.7}
$$

Next, the unknown positions q and velocities $\dot{q}$ are calculated in a similar way to (B.3).

B.2.4 IK_4: Direct position optimization

The formulation of our proposed IK method is similar to (B.4), though with inequality constraints:

$$\begin{aligned}
&\min_{q,\delta} \delta^T Q \delta \\
&f(q) + \gamma[\frac{\partial f(q)}{\partial q}(q - q^-) - \dot{x}\Delta t] = x + \begin{bmatrix}\delta \\ 0\end{bmatrix} \\
&g(q, (q - q^-)/\Delta t) \leq 0
\end{aligned} \tag{B.8}$$

The regulator γ is introduced to help the joint getting out of singular positions faster, as explained in the next section. The novel formulation of IK_4 is similar to IK_2, although the optimization method is not conjugate gradient anymore and velocities result from differentiation, instead of being linked to the derivatives of the task $\dot{x}$ directly.

B.2.5 IK_5: Two-slack optimization

The flexible formulation of IK_4 allows for escaping the singularity by incorporating the knowledge of $\dot{x}$ into the optimization. A similar way is to define a new slack variable on velocities:

$$\begin{aligned}
&\min_{q,\delta,\epsilon} \delta^T Q \delta + \gamma \epsilon^T Q \epsilon \\
&S_N[\frac{\partial f(q)}{\partial q}(q - q^-) - \dot{x}\Delta t] = \epsilon \\
&f(q) = x + \begin{bmatrix}\delta \\ 0\end{bmatrix} \\
&g(q, (q - q^-)/\Delta t) \leq 0
\end{aligned} \tag{B.9}$$

which decouples velocity and position equations, resulting in slightly faster convergence shown later. The behavioral performance however remains the same as IK_4.

Note that IK_1 and IK_3 are similar in the sense that they both integrate velocities to find positions. The integration in these methods is over time, where trajectories reach the target with particular dynamics. On the other side, IK_2, IK_4 and IK_5 are similar because they perform all iterations at once to reach the desired reference trajectory faster. In IK_3, IK_4 and IK_5, we add inequality constraints to make sure the motion is feasible whereas in IK_1 and IK_2, there is not such guaranty.

B.3 Results

The five IK methods are evaluated in this section over different symmetric and asymmetric balancing tasks for our robot COMAN. This kid-size robot weighs about 30kg and is about 1m tall. Disabling the upper body of the robot, we have $K = 18$ DoF out of which six degrees are floating base variables for the pelvis of the robot. We consider six Cartesian tasks of three dimensions: orientations of the two feet and torso (three

Parameter	simulations	real robot
Q	diag([1 .. 1])	diag([1 .. 1])
R	diag([1e-2 .. 1e-2])	diag([1e-2 .. 1e-2])
γ	10	10
n	15 iterations	15 iterations
$\dot{q}_{\max}$	4 rad/s	2 rad/s
q_s	0 deg	5 deg
q^l and q^u	joint specific	joint specific

Table B.1 – All parameters used in our IK formulations. Except γ and R to be tuned, the rest of these parameters are robot specific or for safety reasons. The matrix R is manually tuned to ensure precise and robust tracking of our fast motions. The parameter γ depends on the speed of motion. Larger values make our fastest task numerically unstable. For slow tasks however, this value can be increased.

tasks), positions of the feet and position of CoM (three tasks). In this chapter, we do not explore redundant systems and limit our application to a fully actuated case ($M = 18$). We also compromise the CoM position task ($N = 3$) to handle singularities better. The position and orientation of both feet are fixed in our experiments. All closed-form solutions like (B.2) are calculated with LU factorization of Eigen library while inequality-constrained optimizations are solved with SNOPT. This package uses sequential quadratic programming and a similar factorization method. Besides, the damping factors R are already implemented in SNOPT which provide numerical stability of trajectories in singular conditions. The model of the robot is also calculated by SD-Fast and forward simulations are done in Webots. For this chapter, we consider a unit-diagonal Q matrix that gives equal weight to different CoM tasks. In optimization-based methods (IK_2, IK_3, IK_4, IK_5), we iterate n times and always use the solution of the previous time-step to make the optimization faster, i.e. warm starting. In general, the optimization is continuous but non-convex, because of nonlinear constraints. Previous solutions however help to find locally optimal continuous solutions. Simulations and control of the real robot are done on a Core-i5 1.7GHz CPU in C++ language, using no balancing feedback and purely sending position commands. All parameters are listed in Table B.1.

B.3.1 Symmetric squatting without singularity

The first trivial task is an up-down motion of CoM (at 1Hz) without reaching limits or singularities. This scenario merely compares baseline performance of the algorithms. Although iterating $n = 15$ times is enough, we explore fewer iterations as well to investigate the effectiveness of this extra computational cost. Note that our optimization package uses SQP method, and we only limit major iterations, not those performed at each QP stage. Figure B.1 demonstrates the tracking error of the five IK methods in logarithmic scale as well as the computation time. On average, IK_4 and IK_5 are better than others, even after reducing the number of iterations. IK_2 and IK_3 are more sensitive to the number of iterations, however. We can also notice that extra iterations of IK_4

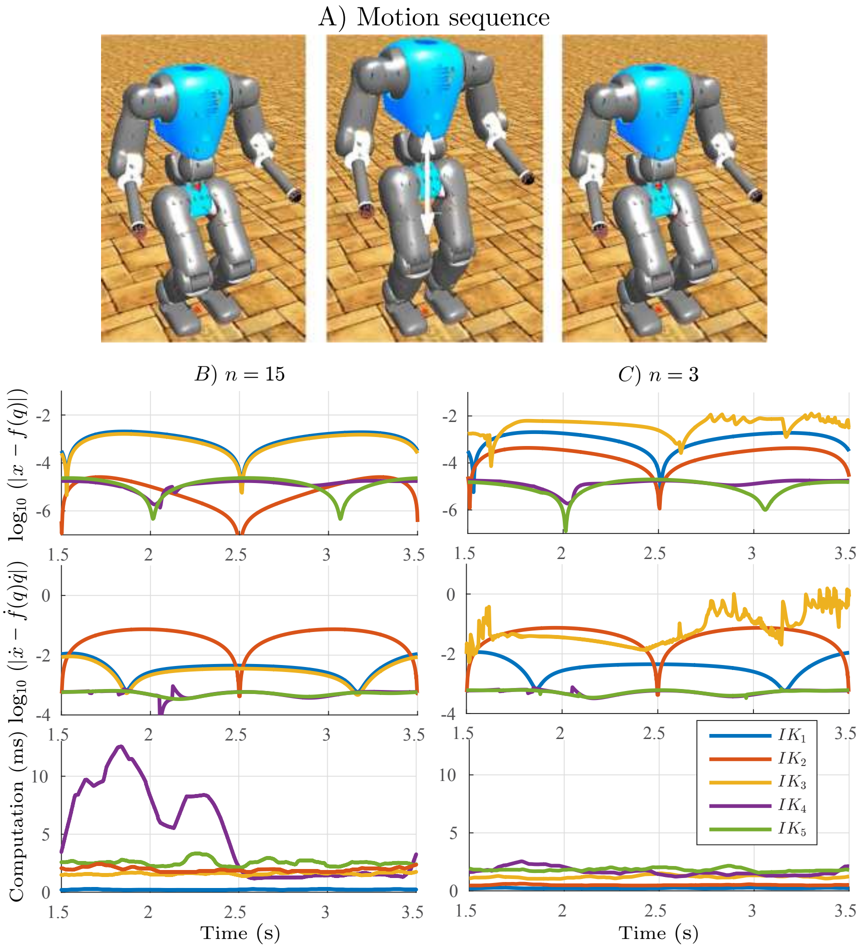


Figure B.1 – Simple symmetric squatting simulations are shown in A) without reaching singularities. We repeat the test for $n = 15$ in B) and $n = 3$ iterations in C). In each group, top: task error, middle: task derivative error and bottom: computation time (filtered). All IK methods provide convincing tracking down to few millimeters. IK_2, IK_4 and IK_5 perform roughly 2 orders of magnitude better, thanks to exact nonlinear equations. With fewer iterations, IK_2 and IK_3 perform worse while IK_4 slightly loses precision. This means that most of the work is done in minor iterations of the first QP step in the nonlinear optimization. The fastest algorithm is IK_1 in this simple test, but less precise.

and IK_5 are not improving the error considerably. For this simple yet fast scenario, IK_1 seems to have enough precision and light computation time which makes it attractive for many robotic applications. However, we will show that it is not well-suited for singular motions.

B.3.2 Symmetric squatting with singularity

One is now interested to know how these IK methods behave when going to singular postures. The results of this test are shown in a similar squatting task of a larger CoM excursion in Figure B.2. In this case, the reference trajectory goes beyond limitations, though all IK algorithms can comply with it, thanks to flexible formulations. Although they all approach the singularity without vibration, they leave this posture with different dynamics. Integration based methods (IK_1 and IK_3) are slower while direct position-based methods (IK_2, IK_4 and IK_5) escape faster. The motion of IK_1 is infeasible as it goes beyond $\dot{q}_{\max}$. It is expected, though, because there is no boundary imposed. Here, IK_4 and IK_5 are less sensitive to iteration numbers, and the extra iterations do not improve the precision considerably. Other algorithms (IK_2 and IK_3) are very sensitive, though, often introducing large delays.

B.3.3 Asymmetric squatting with singularity

So far we only explored symmetric tasks with few joints being active. One is also interested to know how these algorithms perform in asymmetric cases where only one leg goes to singularity at a time. Here, we simulate similar squatting with singularities and add a tilt and roll motion to the torso. Figure B.3 shows the resulting trajectories, where we observe that IK_1, IK_2 and IK_3 fail. IK_4 and IK_5 are both stable, though IK_5 freezes with $n = 3$ iterations. Here, IK_4 and IK_5 cost more computation times (with $n = 15$) but guarantee a feasible motion. Unlike symmetric cases, IK_1 and IK_2 produce infeasible motions here which are not desired. Even though IK_3 was able to handle symmetric singularities and satisfy safety criteria, it is not able to handle asymmetric motions due to its limited integrating nature. This test proves the capability of our proposed formulations (IK_4 and IK_5) to handle different arbitrary tasks, where IK_4 only needs few iterations which cost around 2ms of computation time.

B.3.4 Simulating other joint limits

Along with singularities, we mentioned that joint limits are also important for IK algorithms. Here, we simulate a large tilting motion of the torso where the CoM is also required to be high. Leaning forward is fine, but when leaning backward, the hip-pitch joints reach the limit, where the IK algorithm must leave the singular position in the knee, compromise the CoM task more, and eventually respect the limit in the hip. This task is not doable with IK_1 and IK_2 due to violating boundaries and with IK_3 due to the slow integrating nature. IK_4 is, however, able to handle the task demonstrated in Figure B.4. If we use IK_2, the robot ignores infeasible hip trajectories which might affect the actual CoM position and endanger the overall balance shown in the accompanying video.

B.3.5 Effect of boundary design

One might be interested in using polygon boundary models (Suleiman, 2016) instead of circular ones (B.6). The ultimate effect depends on the task velocity and the design of such polygon. As observed in Figure B.5, polygons can generally add a delay when coming out of the singularity, since the derivative of velocity with respect to position is bounded close to joint limits. With a circular shape, however, the joint can quickly

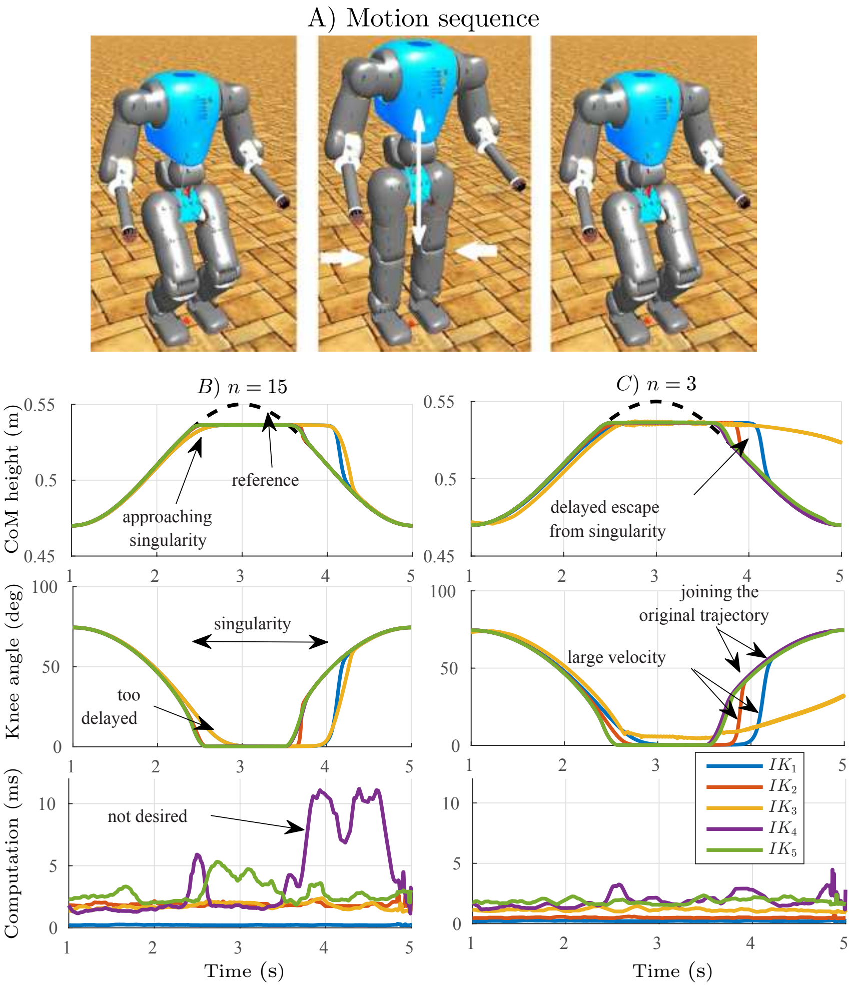


Figure B.2 – Large symmetric squatting simulations are shown in A), spending few moments in singularity. We repeat the test for $n = 15$ in B) and $n = 3$ iterations in C). In each group, top: CoM trajectory, middle: knee angle and bottom: computation time (filtered). Although the reference trajectory is infeasible, except IK_3, other algorithms can comply, i.e going to singular posture without vibration and coming out. IK_1 and IK_3 have a large delay when coming out of the singular posture. They result in large velocities (often infeasible) afterwards to catch up with the actual trajectory. IK_2, IK_4 and IK_5 come out faster however. When performing fewer iterations, IK_4 and IK_5 can still do the job while IK_2 and IK_3 perform worse. IK_4 requires more iterations than IK_5 on average. Considering the version with fewer iterations however, it turns out that the extra computation is not practically useful in this case.

accelerate and escape the singularity.

B.3.6 Effect of γ

As said before, the variable $\gamma \geq 0$ mainly influences dynamics of recovering from singularities in IK_4 and IK_5. Consider Figure B.6, where the real robot is performing

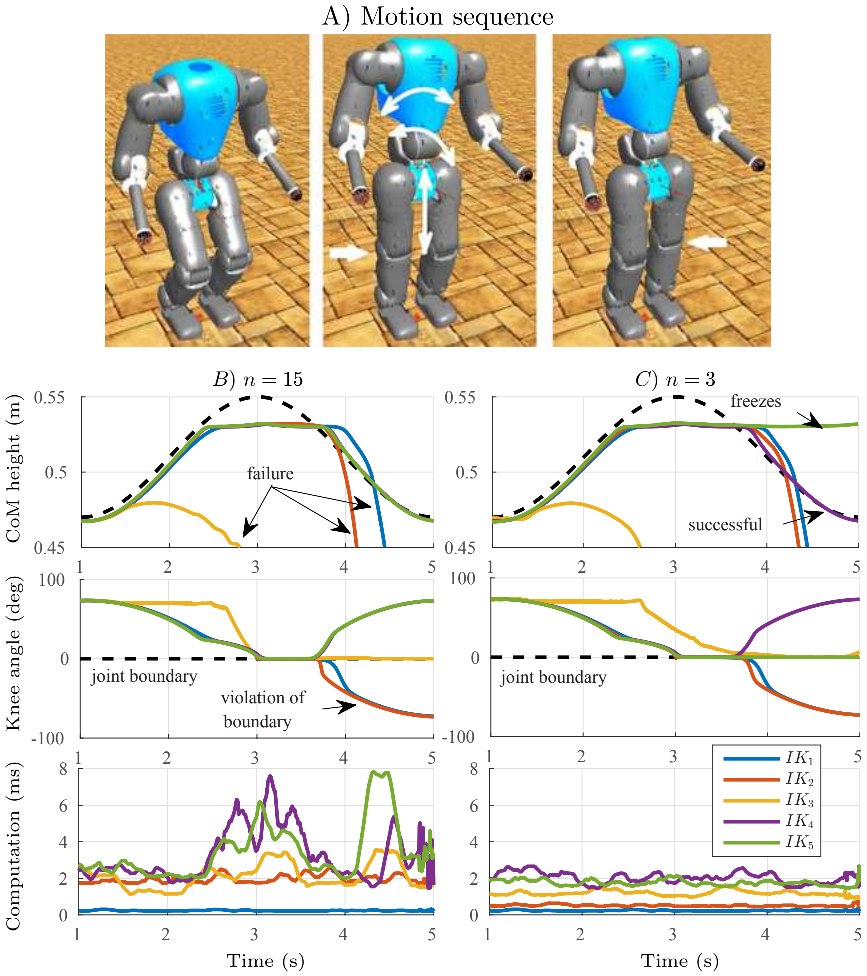
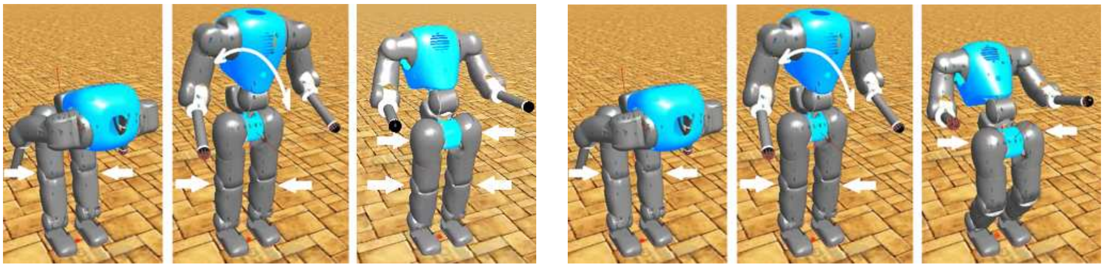


Figure B.3 – Large asymmetric squatting simulations are shown in A), where each leg spends few moments in singularity, but not at the same time. We repeat the test for $n = 15$ in B) and $n = 3$ iterations in C). In each group, top: CoM trajectory, middle: knee angle and bottom: computation time (filtered). Here, only IK_5 ($n = 15$) and IK_4 can survive the motion, while other versions either get frozen or lead to falling eventually. It is also observed that IK_1 and IK_2 can cause infeasible knee angles which is not desired. IK_5 requires more iterations to perform the task while IK_4 can still survive with few iterations.

symmetric squatting with singularity. In the case of $\gamma = 0$, although the joint can still leave the singularity (with more delay), when reaching the actual knee trajectory, it suddenly changes velocity which is not desired (demonstrated in Figure B.5C too). Large accelerations cause high currents which are harmful for electronics of the robot. Here, γ helps to get out of singularity faster, by using the knowledge of $\dot{x}$ which is negative (Figure B.5B). Besides, γ can smoothen the transition when joining the actual trajectory too. Although slower tasks might still be stable with larger γ, we found that $\gamma = 10$ is enough to cover all simulations mentioned in this chapter. Larger values make IK_4 and

A) IK_2: Motion sequence B) IK_4: Motion sequence



C) Knee angle D) Hip angle

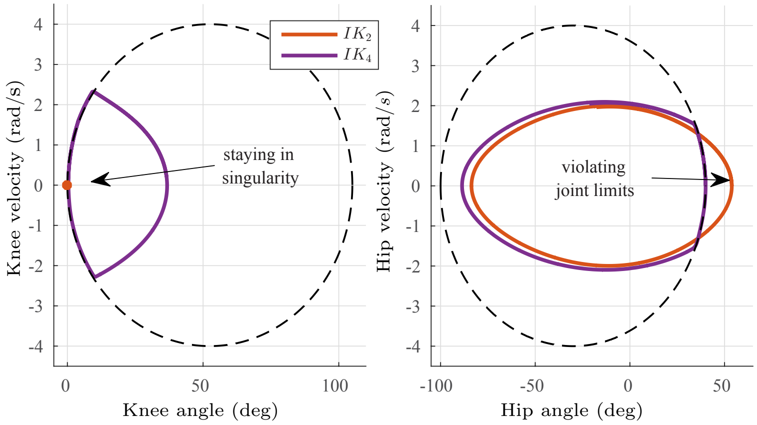


Figure B.4 – When leaning backward, the robot must leave the singular posture in the knee to respect the joint limit in the hip-pitch joint. IK_2 gives infeasible hip angles A) and keeps the knee in a straight posture while IK_4 can easily handle the task B). Trajectories of the knee and hip joints are shown in C) and D) respectively. Refer to the accompanying video for full demonstrations.

IK_5 unstable in very fast motions. One can also think of limiting accelerations by adding more inequalities, but this might add unwanted oscillations (Suleiman, 2016). Besides, such constraints do not help to get out of singularity faster, because the knowledge of $\dot{x}$ is missing.

B.3.7 How singularities improve power consumption

We tested a rather simple motion on the real robot both in a singular and multiple crouched postures to compare the power consumption. Demonstrated in Figure B.7, the robot is performing a lateral motion with torso twisting. This scenario is computationally hard due to asymmetry, but IK_4 can still handle it even with $n = 3$ iterations. We have plotted the average power consumption of the robot in all scenarios as well. It is known that humanoids consume more power in crouched postures. Here we fit a line to experimental measurements to quantify the steep rise of power consumption. In the most crouched scenario which is quite similar to many humanoid robot demonstrations, COMAN consumes over three times the power of stretched leg scenario (because of nearly zero knee torques in stretched-leg postures, similar to Figure B.6D). This test

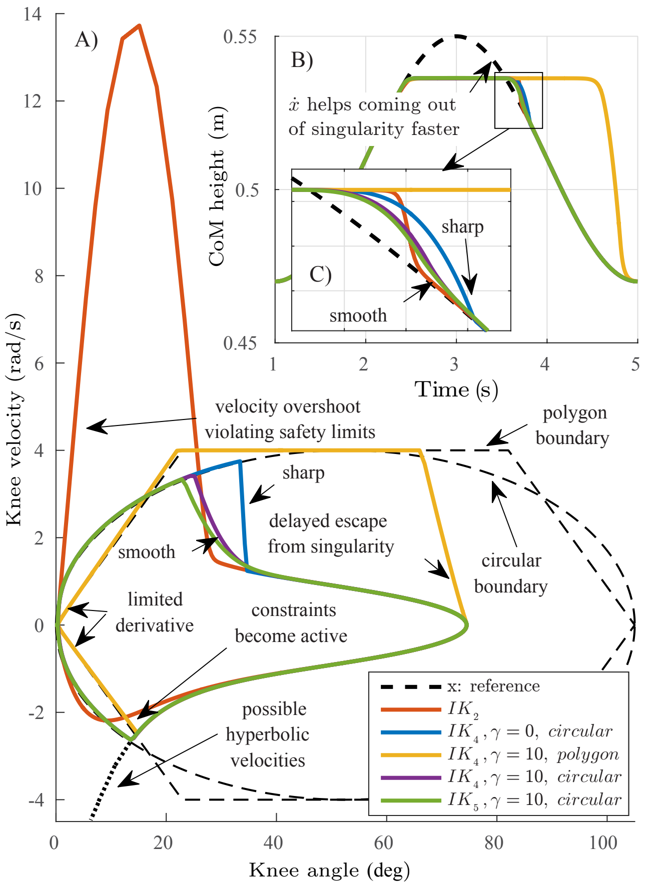


Figure B.5 – The role of joint boundaries and the parameter γ in our IK algorithms. A) and B) reference and actual CoM height trajectories, C) knee trajectories in position-velocity space. Starting from an arbitrary point within the feasible boundaries, the knee joint approaches the singularity in 0 degrees over the defined boundary. It spends few moments in the singular position without vibrations and then comes out with certain dynamics. The boundaries limit hyperbolic velocities of the joint when approaching singularity. When coming out, however, it takes time for the joint to catch the actual trajectory again. Different boundaries might introduce certain delays, depending on the task. Here, a too safe design of polygon boundary proposed by (Suleiman, 2016) leads to a high delay. Note also that removing γ results in a delayed escape from singularity and instantaneous jump in the velocity which is not desired (refer to Figure B.6 too).

motivates the benefit of our proposed IK method in many balancing tasks to reduce power consumption and provide more human-like postures.

B.4 Discussion

Balancing with crouched knees is very popular for humanoids as it provides full controllability properties. However, it brings many mechanical problems in the long term and requires more electrical power. Motivated by solving these practical issues, we formulated

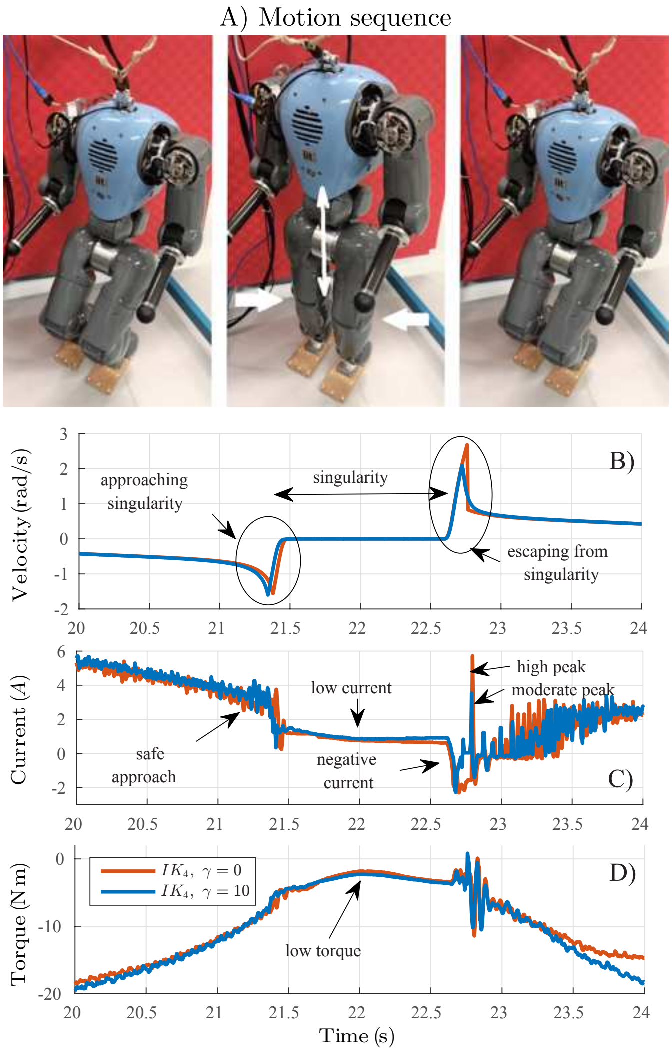


Figure B.6 – The real robot in A) is performing large and symmetric squatting motions (like Figure B.2) with IK_4 while reaching singularities. This task requires large velocities demonstrated in B) when approaching or coming out of singularity. One can clearly observe that the current C) and joint torque D) decrease drastically in singular position. Thanks to circular boundaries, the knee joint does not hit the limit with high current, i.e. safely approaches the joint limit and singularity. Coming out of singularity, however, the algorithm without γ results in a momentary peak in current while introducing γ can decrease this dangerous peak. Another remarkable point is that the joint produces negative current directly after leaving singularity. In these short moments, the robot transfers weight to the knee and loses potential energy.

and tested traditional IK algorithms, aiming at performing balance with stretched legs. Thanks to flexible damped least-squares formulations, these algorithms were numerically robust in singular positions and could handle to some extent, the safety criteria when reaching joint limits. However, due to the integrative nature, they largely fail in arbitrary

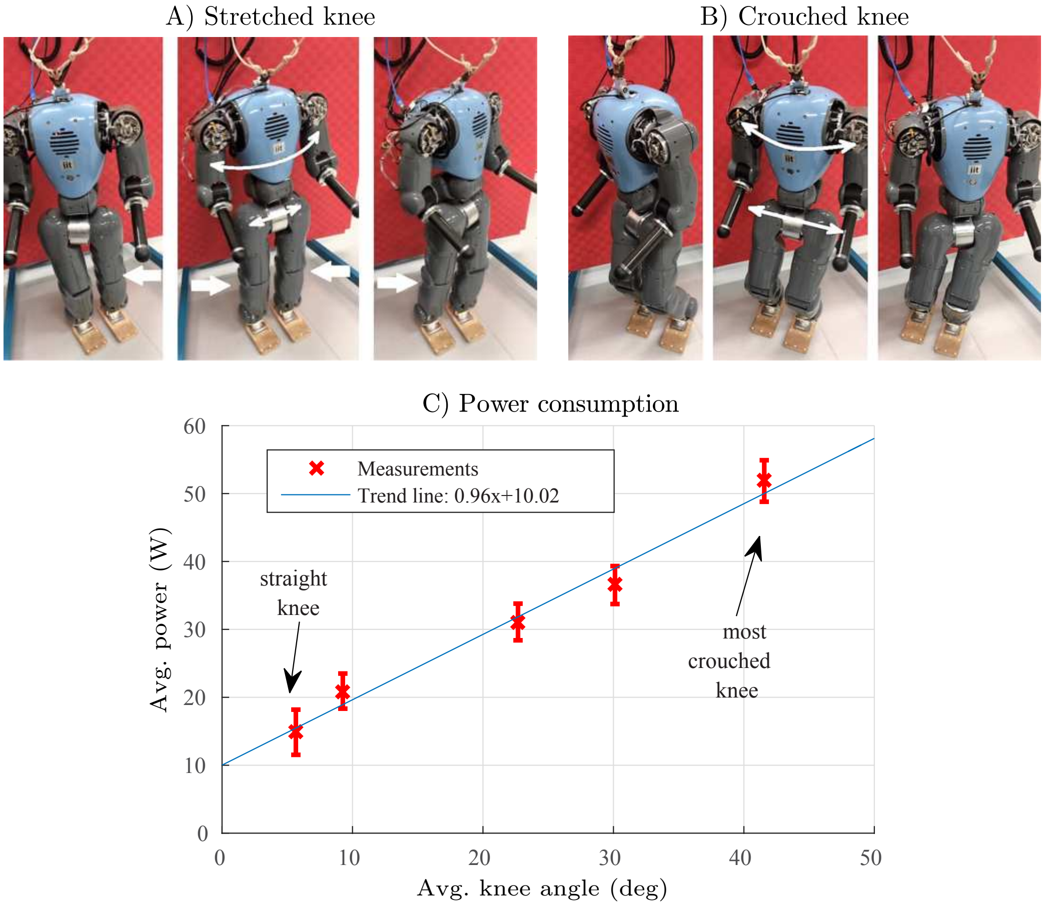


Figure B.7 – Demonstration of lateral and twisting motions on the real robot with A) stretched and B) crouched knees. C) One can observe that crouched postures consume much more power on average. Here, the consumptions of electronics (about 108W) and quiet stretched-leg stance (about 10W) are subtracted.

symmetric and asymmetric tasks where rapid escape from singularity plays an important role in tracking. These algorithms often freeze or even lead to falling in such scenarios. Besides, in the literature, most of the straight-knee walking algorithms use preplanned trajectories which limit the generality of the IK block for arbitrary balancing tasks.

We formulated nonlinear optimizations instead to remove the time integration and introduced constraints to handle safety criteria. The idea of using damped least-squares, boundaries, and nonlinear optimizations is adapted from literature. The novelty of this work, however, lies in combining them together and incorporating the knowledge of task derivatives to improve the behavior, especially when coming out of singularities. This is the key feature to handle arbitrary balancing tasks where each knee might go to singularity and come out at different times (refer to the accompanying video). To the extent of our knowledge, no other IK method combines position and velocity variables together and mix them with inequality constraints.

All IK methods were compared extensively over tasks of different nature. Control-related difficulties faced with singularities are not limited to numerical stability. We showed that the two proposed IK algorithms have superior performance in approaching singularities and coming out of them as well. IK_5 is generally faster than IK_4, but behaves similarly. Over asymmetric tasks however, it requires same computation times and shows more sensitivity to the number of iterations. Therefore, IK_4 outperforms all other algorithms, even with a fewer number of iterations. We also showed that for our nonlinear formulation, handling the typical 18 DoF of a balancing humanoid is computationally affordable (about 2 milliseconds). We consider implementation improvements and extension to more DoF for future works.

An interesting secondary result was also presented, comparing the power consumption of stretched and crouched balancing tasks. The steep rise of power motivates application of our proposed IK algorithm as a low-level block that robustly transforms arbitrary Cartesian tasks to joint-space motions. Such block can improve power consumption and mechanical durability of humanoid hardware. In future, we are going to use the ideas we developed here in our ID formulation of chapter 3 to provide a general framework that handles geometrical and dynamical limitations while tolerating difficulties of singular postures.

C 3LP Symbolic Equations

The parametric matrices C describing 3LP dynamics in single and double support phases are found by Maple. These matrices are sparse and have duplicate elements, since the sagittal and lateral dynamics are similar. In this appendix, we provide symbolic equations for the case where the limb inertias I_i and torso and slope angles (θ and ϕ) are zero. With those variables, 3LP equations are much more complicated. Define auxiliary parameters:

$$\begin{aligned}
u &= h_3 + h_1 \\
v &= h_2 - h_1 \\
c_1 &= m_1 u h_1 + m_2 v^2 \\
c_2 &= m_1 u h_1 + 2 m_2 v^2 \\
c_3 &= -m_1 h_1 + m_2 v - m_2 h_1 \\
c_4 &= (-1 + \frac{2t}{T_{\text{ds}}}) h_1 + v
\end{aligned} \tag{C.1}$$

The matrices in (6.10) take the form:

$$\begin{aligned}
C_X^{\text{ss}} &= \frac{g}{c_1} \begin{bmatrix} (-\frac{c_2}{h_2} + m_2 v) I & \frac{m_1 h_1 (v+u)}{h_2} I & \frac{v c_3}{h_2} I \\ m_2 h_1 I & (m_1 h_1 - m_2 v) I & c_3 I \\ 0 & 0 & 0 \end{bmatrix} \\
C_U^{\text{ss}} &= \frac{h_1}{c_1} \begin{bmatrix} \frac{c_2}{h_2^2 m_2} I & -\frac{v}{h_2} I \\ \frac{v}{h_2} I & -I \\ 0 & 0 \end{bmatrix} \\
C_V^{\text{ss}} &= \frac{t}{T_{\text{ss}}} C_U^{\text{ss}} \\
C_W^{\text{ss}} &= \frac{h_1}{c_1} \begin{bmatrix} \frac{vu}{h_2} I & -\frac{v}{h_2} I \\ uI & -I \\ 0 & 0 \end{bmatrix} \\
C_s^{\text{ss}} &= \frac{gw}{c_1} \begin{bmatrix} \frac{c_2}{h_2} J \\ m_2 v J \\ 0 \end{bmatrix}
\end{aligned} \tag{C.2}$$

The matrices in (6.22) are found as:

$$
\begin{aligned}
C_X^{\text{ds}} &= \tfrac{gh_1m_1}{c_2} \begin{bmatrix} 0 & 0 & 0 \\ (-1+\frac{t}{T_{\text{ds}}})I & I & -\frac{t}{T_{\text{ds}}}I \\ 0 & 0 & 0 \end{bmatrix} \\
&+ \tfrac{gm_2}{c_2} \begin{bmatrix} 0 & 0 & 0 \\ c_4 I & -2vI & c_4 I \\ 0 & 0 & 0 \end{bmatrix} \\
C_U^{\text{ds}} &= \tfrac{h_1}{c_2} \begin{bmatrix} 0 & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix} \\
C_V^{\text{ds}} &= (1-\tfrac{t}{T_{\text{ds}}}) \; C_U^{\text{ds}} \\
C_W^{\text{ds}} &= \tfrac{h_1}{c_2} \begin{bmatrix} 0 & 0 \\ uI & -I \\ 0 & 0 \end{bmatrix} \\
C_s^{\text{ds}} &= 0
\end{aligned}
\tag{C.3}
$$

where $I \in \mathbb{R}^{2\times 2}$ is identity matrix and $J = [0\ 1]^T$. The closed form transition matrices in (6.12) and (6.23) are then calculated symbolically based on these matrices. Altogether, offline calculations require about $3.7k$ FLOPS in a maple-optimized code which take few microseconds on an average core-i5 computer if implemented in C++. The results are constant matrices in (6.13) and (6.13) which can be used online with a much faster speed.

D DLQR for Constrained Systems

For a discrete constrained system of:

$$\begin{aligned} X[k+1] &= AX[k] + BU[k] \\ CX[k+1] &= 0 \end{aligned} \tag{D.1}$$

which has a set-point solution $\bar{X}[k]$ and $\bar{U}[k]$, the error system is defined as:

$$\begin{aligned} E[k+1] &= AE[k] + B\Delta U[k] \\ CE[k+1] &= 0 \end{aligned} \tag{D.2}$$

where $E[k] = X[k] - \bar{X}[k]$. Here, regardless of the control strategy, the constraint represented by C should always be satisfied. Consider $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times M}$ and $C \in \mathbb{R}^{P\times N}$. The DLQR optimization problem for this system is:

$$\begin{aligned} &\min_{E[k],\Delta U[k]} \textstyle\sum_{k=0}^{\infty} E[k]^T Q E[k] + \Delta U[k]^T R \Delta U[k] \\ &s.t. \begin{cases} E[k+1] = AE[k] + B\Delta U[k] \\ CE[k+1] = 0 \end{cases} \quad k \geq 0 \end{aligned} \tag{D.3}$$

Assume we find a matrix $\tilde{C} \in \mathbb{R}^{(N-P)\times N}$ that forms a complete basis with C. In other words, the matrix S defined by:

$$S = \begin{bmatrix} \tilde{C} \\ C \end{bmatrix} \tag{D.4}$$

has a full rank. Now, we define a new variable $Z[k] = SE[k]$ which will produce the following new DLQR problem:

$$\begin{aligned} &\min_{Z[k],\Delta U[k]} \textstyle\sum_{k=0}^{\infty} Z[k]^T \tilde{Q} Z[k] + \Delta U[k]^T \tilde{R} \Delta U[k] \\ &s.t. \begin{cases} Z[k+1] = \tilde{A}E[k] + \tilde{B}\Delta U[k] \\ \tilde{C}Z[k+1] = 0 \end{cases} \quad k \geq 0 \end{aligned} \tag{D.5}$$

where:

$$\begin{gathered}\tilde{Q} = S^{-T}QS^{-1}, \quad \tilde{R} = R \\ \tilde{A} = SAS^{-1}, \quad \tilde{B} = SB, \quad \tilde{C} = CS^{-1}\end{gathered} \tag{D.6}$$

Note that:

$$Z[k] = SE[k] = \begin{bmatrix}\tilde{C}E[k] \\ CE[k]\end{bmatrix} = \begin{bmatrix}Y[k] \\ 0\end{bmatrix} \tag{D.7}$$

where $Y[k] \in \mathbb{R}^{N-P}$ is reduced to exclude the constraint. Assuming $P < N$, a full rank C and $M >= P$, we can find P independent components of $\Delta U[k]$ and rewrite the constraint in terms of these components. Without loss of generality, assume these components are the last P components of $\Delta U[k]$, referred to as $\Delta W[k]$ hereafter:

$$\Delta U[k] = \begin{bmatrix}\Delta V[k] \\ \Delta W[k]\end{bmatrix} \tag{D.8}$$

where $\Delta V[k] \in \mathbb{R}^{M-P}$ and $\Delta W[k] \in \mathbb{R}^{P}$. Consider we decompose matrices $\tilde{Q}$ and $\tilde{R}$ as follows:

$$\tilde{Q} = \begin{bmatrix}\tilde{Q}^{vv} & \tilde{Q}^{vw} \\ \tilde{Q}^{wv} & \tilde{Q}^{ww}\end{bmatrix}, \quad \tilde{R} = \begin{bmatrix}\tilde{R}^{vv} & \tilde{R}^{vw} \\ \tilde{R}^{wv} & \tilde{R}^{ww}\end{bmatrix} \tag{D.9}$$

where the last lower right corner of size $P \times P$ is taken out with an index ^{ww}. Likewise, system matrices $\tilde{A}$ and $\tilde{B}$ can be decomposed to:

$$\begin{bmatrix}Y[k+1] \\ 0\end{bmatrix} = \begin{bmatrix}\tilde{A}^{vv} & \tilde{A}^{vw} \\ \tilde{A}^{wv} & \tilde{A}^{ww}\end{bmatrix}\begin{bmatrix}Y[k] \\ 0\end{bmatrix} + \begin{bmatrix}\tilde{B}^{vv} & \tilde{B}^{vw} \\ \tilde{B}^{wv} & \tilde{B}^{ww}\end{bmatrix}\begin{bmatrix}\Delta V[k] \\ \Delta W[k]\end{bmatrix} \tag{D.10}$$

With such decomposition, we can take $\Delta W[k]$ out of (D.10) from the last P rows:

$$\begin{aligned}\Delta W[k] &= \tilde{G}Y[k] + \tilde{H}\Delta V[k] \\ \tilde{G} &= -(\tilde{B}^{ww})^{-1}\tilde{A}^{wv} \\ \tilde{H} &= -(\tilde{B}^{ww})^{-1}\tilde{B}^{wv}\end{aligned} \tag{D.11}$$

and:

$$\begin{aligned}Y[k+1] &= \bar{A}Y[k] + \bar{B}\Delta V[k] \\ \bar{A} &= \tilde{A}^{vv} + \tilde{B}^{vw}\tilde{G} \\ \bar{B} &= \tilde{B}^{vv} + \tilde{B}^{vw}\tilde{H}\end{aligned} \tag{D.12}$$

Now, given that we have resolved the constraint, we can form an equivalent DLQR design in terms of $Y[k]$ and $\Delta V[k]$ by replacing $\Delta W[k]$ in all terms:

$$\begin{aligned} &\min_{Y[k],\Delta V[k]} \textstyle\sum_{k=0}^{\infty} \\ &Y[k]^T\bar{Q}Y[k] + \Delta V[k]^T\bar{R}\Delta V[k] + 2Y[k]^T\bar{N}\Delta V[k] \\ &s.t.\ \ Y[k+1] = \bar{A}Y[k] + \bar{B}\Delta V[k] \quad k \geq 0 \end{aligned} \tag{D.13}$$

which has a standard DLQR format without constraint. The new cost matrices in (D.13) are defined as:

$$\begin{aligned} \bar{Q} &= \tilde{Q}^{vv} + \tilde{G}^T\tilde{R}^{ww}\tilde{G} \\ \bar{R} &= \tilde{R}^{vv} + \tilde{H}^T\tilde{R}^{ww}\tilde{H} + \tilde{R}^{vw}\tilde{H} + \tilde{H}^T\tilde{R}^{wv} \\ \bar{N} &= \tilde{G}^T(\tilde{R}^{ww}\tilde{H} + \tilde{R}^{wwT}\tilde{H}^T + \tilde{R}^{vwT} + \tilde{R}^{wv}) \end{aligned} \tag{D.14}$$

We call the optimal DLQR gain matrix for (D.13) $\bar{K}$ which acts on $Y[k] = \tilde{C}E[k]$ and produces $\Delta V[k]$. The other part of system input $\Delta W[k]$ can be calculated by (D.11). Overall, the system input is:

$$\Delta U[k] = \begin{bmatrix} -\bar{K} \\ \tilde{G} - \tilde{H}\bar{K} \end{bmatrix} \tilde{C}E[k] \tag{D.15}$$

which optimally satisfies the constraint and initial DLQR problem in (D.3).

E Time-Projection for Constrained Systems

Time-projection controller for a constrained system is derived very similarly to a normal system. Consider all formulations and decompositions of Appendix A. The instantaneous error $e(t) = x(t) - \bar{x}(t)$ in a constrained system would evolve until the next control time by:

$$\hat{E}_t[k+1] = A(\tau)e(t) + B(\tau)\delta\hat{U}_t[k] \tag{E.1}$$

where $\tau = (k+1)T - t$ and the constraint applies $C\hat{E}_t[k+1] = 0$. Here we assume a constant input $\delta\hat{U}_t[k]$ applied to the system which yields to a predicted error $\hat{E}_t[k+1]$. Remember the hat notation is used to emphasize that these predicted quantities are just calculated at time t and they are not real system variables. Imagine we define $z(t) = Se(t)$, $\tilde{A}_t = SA(\tau)S^{-1}$, $\tilde{B}_t = SB(\tau)$ and perform a decomposition similar to (D.10):

$$\begin{aligned} \hat{Z}_t[k+1] &= \tilde{A}_t z(t) + \tilde{B}_t \delta\hat{U}_t[k] \\ \begin{bmatrix} \hat{Y}_t[k+1] \\ 0 \end{bmatrix} &= \begin{bmatrix} \tilde{A}_t^v \\ \tilde{A}_t^w \end{bmatrix} z(t) + \begin{bmatrix} \tilde{B}_t^{vv} & \tilde{B}_t^{vw} \\ \tilde{B}_t^{wv} & \tilde{B}_t^{ww} \end{bmatrix} \begin{bmatrix} \delta\hat{V}_t[k] \\ \delta\hat{W}_t[k] \end{bmatrix} \end{aligned} \tag{E.2}$$

which describes system evolution from time t to $(k+1)T$ whereas the constraint in (D.5) describes same evolution from kT to $(k+1)T$. Here the subscript t indicates dependency on t. Note that the last P elements of $z(t)$ might not be zero in $kT < t < (k+1)T$, but the constraint implies that these last P elements have to be zero at time instances kT and $(k+1)T$. Similar to (D.11), we can take $\delta\hat{W}[k]$ out of the last P equations:

$$\begin{aligned} \delta\hat{W}_t[k] &= \tilde{G}_t z(t) + \tilde{H}_t \delta\hat{V}_t[k] \\ \tilde{G}_t &= -(\tilde{B}_t^{ww})^{-1}\tilde{A}_t^w \\ \tilde{H}_t &= -(\tilde{B}_t^{ww})^{-1}\tilde{B}_t^{wv} \end{aligned} \tag{E.3}$$

and new system matrices would be defined as:

$$\begin{aligned}\hat{Y}_t[k+1] &= \bar{A}_t z(t) + \bar{B}_t \delta\hat{V}_t[k] \\ \bar{A}_t &= \tilde{A}^v_t + \tilde{B}^{vw}_t \tilde{G}_t \\ \bar{B}_t &= \tilde{B}^{vv}_t + \tilde{B}^{vw}_t \tilde{H}_t \end{aligned} \tag{E.4}$$

Note that in (D.12) and (E.4), the constraint is resolved. In other words, system matrices are adjusted such that they account for the effect of W inputs which aim at satisfying the constraint. Now, we can consider time-projection for this free system as follows. Imagine an initial reduced state $\hat{Y}_t[k]$ evolves in time by $\delta\hat{V}_t[k]$ according to (D.12) and yields to $\hat{Y}_t[k+1]$. Similarly, the current error $z(t)$ evolves in time by $\delta\hat{V}_t[k]$ and yields to the same $\hat{Y}_t[k+1]$ according to (E.4). Now, the system of equations in (7.10) can be applied here as well:

$$\begin{bmatrix} \bar{A} & \bar{B} - \bar{B}_t & \cdot \\ K & I & \cdot \\ \cdot & -\tilde{H}_t & I \end{bmatrix} \begin{bmatrix} \hat{Y}_t[k] \\ \delta\hat{V}_t[k] \\ \delta\hat{W}_t[k] \end{bmatrix} = \begin{bmatrix} \bar{A}_t \\ 0 \\ \tilde{G}_t \end{bmatrix} Se(t) \tag{E.5}$$

where the solution defines $\delta u(t) = \delta\hat{U}_t[k]$ to be applied at time instance t.

F Time-Projection Stability Proof

Consider a simple system of:

$$\dot{x}(t) = x(t) + u(t) + w(t) \tag{F.1}$$

where $u(t)$ is the control input and $w(t)$ is disturbance. Consider a control period T and steady-state solutions $\bar{x} = 0$ and $\bar{u} = 0$. The closed-form evolution of this system could be written as:

$$X[k+1] = e^{T} X[k] + (e^{T} - 1)U[k] \tag{F.2}$$

assuming a constant $U[k]$ applied to the system. Now, imagine we use a discrete controller at time instances kT to adjust the constant input with a law of:

$$U[k] = -\Gamma X[k] \tag{F.3}$$

The stability of closed-loop system suggests that:

$$|e^{T} - \Gamma(e^{T} - 1)| < 1 \tag{F.4}$$

which puts boundaries on Γ:

$$1 < \Gamma < \frac{e^{T} + 1}{e^{T} - 1} \tag{F.5}$$

For this system, the DLQR controller satisfies this criteria. The time-projection controller takes a state $x(t)$, maps it to the previous time-sample kT and finds the control input based on the discrete controller Γ. Without loss of generality, assume t is between the first two time-samples ($kT < t < (k+1)T$) and:

$$\begin{aligned} x(t) &= e^{t} \hat{X}_t[k] + (e^{t} - 1)\delta\hat{U}_t[k] \\ \delta\hat{U}_t[k] &= -\Gamma \hat{X}_t[k] \end{aligned} \tag{F.6}$$

where $\hat{X}_t[k]$ is a possible predicted initial state that can lead the current measurement $x(t)$. The two laws of (F.6) and system equations (F.1) result in the following closed-loop system:

$$\begin{aligned} x(t) &= e^t(-\frac{\delta\hat{U}_t[k]}{\Gamma}) + (e^t - 1)\delta\hat{U}_t[k] \\ \dot{x}(t) &= (1 + \frac{1}{-\frac{e^t}{\Gamma} + e^t - 1})x(t) \end{aligned} \tag{F.7}$$

which is found by resolving $\hat{X}_t[k]$ in (F.6) and plugging $u(t) = \delta\hat{U}_t[k]$ in (F.1). In order to produce finite feedbacks, the denominator in (F.7) should not have a zero in $0 < t < T$. In other words, the root of denominator t_0 should be outside $[0, T]$.

$$-\frac{e^{t_0}}{\Gamma} + e^{t_0} - 1 = 0 \quad \rightarrow \quad t_0 = \ln(\frac{1}{1 - \frac{1}{\Gamma}}) \tag{F.8}$$

This leads to the following two conditions:

$$\begin{cases} \ln(\frac{1}{1-\frac{1}{\Gamma}}) < 0 \rightarrow \frac{1}{1-\frac{1}{\Gamma}} < 1 \rightarrow \Gamma < 0 \\ \ln(\frac{1}{1-\frac{1}{\Gamma}}) > T \rightarrow \frac{1}{1-\frac{1}{\Gamma}} > e^T \rightarrow \Gamma < \frac{e^T}{e^T - 1} \end{cases} \tag{F.9}$$

This will further tighten the boundaries of (F.5) to:

$$1 < \Gamma < \frac{e^T}{e^T - 1} \tag{F.10}$$

Note that DLQR design does not necessarily satisfy this criteria, unless proper state and input cost matrices are chosen in the objective. Consider the lyapunov function $V(t) = \frac{1}{2}x(t)^2$ whose derivative is:

$$\dot{V}(t) = x(t)\dot{x}(t) = \epsilon(t)x(t)^2 \tag{F.11}$$

where:

$$\epsilon(t) = 1 + \frac{1}{-\frac{e^t}{\Gamma} + e^t - 1} \tag{F.12}$$

It is obvious that $\epsilon(t)$ is monotonically decreasing, because $\Gamma > 1$ and thus:

$$\dot{\epsilon}(t) = \frac{e^t(\frac{1}{\Gamma} - 1)}{(-\frac{e^t}{\Gamma} + e^t - 1)^2} < 0 \tag{F.13}$$

Since $\epsilon(0) = 1 - \Gamma < 0$, one can conclude that:

$$\dot{V}(t) = \epsilon(t)x(t)^2 < \epsilon(0)x(t)^2 \tag{F.14}$$

which proves that $V(t)$ is decreasing and the system is stable with any choice of Γ in (F.10). In order to compare the LQR and time-projection controllers with a simple

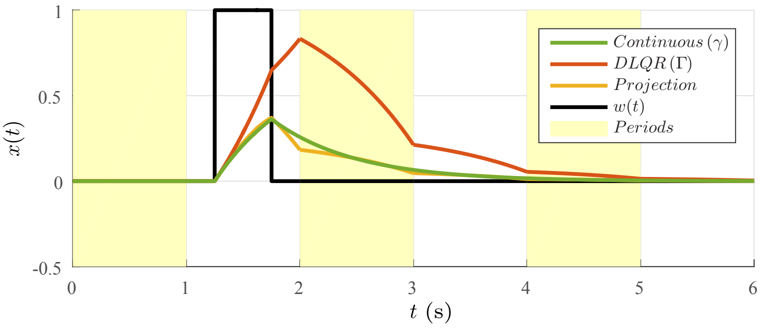


Figure F.1 – One degree-of-freedom system with continuous, DLQR and time-projection controllers.

continuous controller of $u(t) = -\gamma x(t)$, we find the discrete feedback gain Γ by converting closed-loop eigen-values. The discrete eigen-value (Λ) of (F.4) is converted to a continuous eigen-value (λ) by a logarithm operation:

$$\Lambda = e^T - \Gamma(e^T - 1) = e^{\lambda T} \tag{F.15}$$

Given that $\lambda = 1 - \gamma$, we can λ as:

$$\gamma = -\ln(e^T - \Gamma(e^T - 1))/T + 1 \tag{F.16}$$

This let us compare the three types of controller on a fair basis. Consider we select $T = 1$, $Q = 1$ and $R = 1$ for DLQR design which yields $\Gamma \simeq 1.43$ and it satisfies (F.10). The equivalent continuous gain for this system would be $\gamma \simeq 2.37$. Figure F.1 demonstrates how an inter-sample disturbance could be rejected by these controllers. The DLQR controller overshoots, simply because of a late response which only starts at time $t = 2$. The continuous controller γ smoothly damps the disturbance while the time-projection controller performs damping in multiple steps with the same rate as the continuous controller. One can imagine that with decreasing T, the time-projection controller will converge to the continuous controller.

Bibliography

Peter G Adamczyk and Arthur D Kuo. Mechanical and energetic consequences of rolling foot shape in human walking. *Journal of Experimental Biology*, 216(14):2722–2731, 2013.

Farhad Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *Robotics, IEEE Transactions on*, 21(5):834–849, 2005.

Mazen Al Borno, Martin De Lasa, and Aaron Hertzmann. Trajectory optimization for full-body movements with complex contacts. *Visualization and Computer Graphics, IEEE Transactions on*, 19(8):1405–1414, 2013.

R McN Alexander. Optimum muscle design for oscillatory movements. *Journal of theoretical Biology*, 184(3):253–259, 1997.

Michael P Allen et al. Introduction to molecular dynamics simulation. *Computational soft matter: from synthetic polymers to proteins*, 23:1–28, 2004.

Bassam Alrifaee, Masoumeh Ghanbarpour Mamaghani, and Dirk Abel. Centralized non-convex model predictive control for cooperative collision avoidance of networked vehicles. In *Intelligent Control (ISIC), IEEE International Symposium on*, pages 1583–1588, 2014.

Soran Aminiaghdam, Christian Rode, Roy Müller, and Reinhard Blickhan. Increasing trunk flexion transforms human leg function into that of birds despite different leg morphology. *Journal of Experimental Biology*, 220(3):478–486, 2017.

Frank C Anderson and Marcus G Pandy. Dynamic optimization of human walking. *Journal of biomechanical engineering*, 123(5):381–390, 2001.

Frank C Anderson and Marcus G Pandy. Individual muscle contributions to support in normal walking. *Gait & posture*, 17(2):159–169, 2003.

Joel Andersson, Johan Åkesson, and Moritz Diehl. Casadi: A symbolic package for automatic differentiation and optimal control. In *Recent advances in algorithmic differentiation*, pages 297–307. Springer, 2012.

Emanuel Andrada, Christian Rode, Yefta Sutedja, John A Nyakatura, and Reinhard Blickhan. Trunk orientation causes asymmetries in leg function in small bird terrestrial locomotion. *Proceedings of the Royal Society of London B: Biological Sciences*, 281 (1797):20141405, 2014.

Shinya Aoi and Kazuo Tsuchiya. Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback. *IEEE Transactions on robotics*, 22(2):391–397, 2006.

Jonathan Arreguit, Salman Faraji, and Auke J. Ijspeert. Fast multi-contact whole-body motion planning with limb dynamics. *In preparation*, 2018.

Fumihiko Asano, Masaki Yamakita, Norihiro Kamamichi, and Zhi-Wei Luo. A novel gait generation for biped walking robots based on mechanical energy constraint. *Robotics and Automation, IEEE Transactions on*, 20(3):565–573, 2004.

Barton J Bacon. Quaternion-Based Control Architecture for Determining Controllability/Maneuverability Limits. In *AIAA Guidance, Navigation, and Control Conference*, 2012.

Jon Baker. Techniques for geometry optimization: A comparison of cartesian and natural internal coordinates. *Journal of computational chemistry*, 14(9):1085–1100, 1993.

Catherine E. Bauby and Arthur D.‘ Kuo. Active control of lateral balance in human walking. *Journal of Biomechanics*, 33(11):1433–1440, November 2000.

Thomas Bedford and CG Warner. The energy expended while walking in stooping postures. *British journal of industrial medicine*, 12(4):290, 1955.

John E. A. Bertram. Constrained optimization in human walking: cost minimization and gait plasticity. *J. Exp. Biol.*, 208(Pt 6):979–991, March 2005.

John E. A. Bertram and Andy Ruina. Multiple walking speed–frequency relations are predicted by constrained optimization. *Journal of theoretical Biology*, 209(4):445–453, 2001.

Pranav A Bhounsule, Andy Ruina, and Gregg Stiesberg. Discrete-Decision Continuous-Actuation control: balance of an inverted pendulum and pumping a pendulum swing. *Journal of Dynamic Systems, Measurement, and Control*, 137(5):051012, 2015.

Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.

Michael Bloesch, Marco Hutter, Mark A Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, and Roland Siegwart. State estimation for legged robots-consistent fusion of leg kinematics and IMU. *Robotics*, page 17, 2013.

Thiago Boaventura, Claudio Semini, Jonas Buchli, Marco Frigerio, Michele Focchi, and Darwin G Caldwell. Dynamic torque control of a hydraulic quadruped robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1889–1894. IEEE, 2012.

Júlia Borràs, Christian Mandery, and Tamim Asfour. A whole-body support pose taxonomy for multi-contact humanoid robot motions. *Science Robotics*, 2(13):eaaq0560, 2017.

Ronan Boulic, Nadia Magnenat Thalmann, and Daniel Thalmann. A global human walking model with real-time kinematic personification. *The visual computer*, 6(6): 344–358, 1990.

Karim Bouyarmane and Abderrahmane Kheddar. Multi-contact stances planning for multiple agents. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5246–5253. IEEE, 2011.

J. M. Brockway. Derivation of formulae used to calculate energy expenditure in man. *Hum Nutr Clin Nutr*, 41(6):463–471, November 1987.

Stanislas Brossette, Joris Vaillant, François Keith, Adrien Escande, and Abderrahmane Kheddar. Point-cloud multi-contact planning for humanoids: Preliminary results. In *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pages 19–24. IEEE, 2013.

Michael Gordon Browne. *Neuromechanical adaptations to real-time biofeedback of the center of pressure during human walking.* PhD thesis, North Carolina State University Chapel Hill, 2016.

Raymond C. Browning and Rodger Kram. Energetic cost and preferred speed of walking in obese vs. normal weight women. *Obes. Res.*, 13(5):891–899, May 2005.

Raymond C. Browning and Rodger Kram. Effects of obesity on the biomechanics of walking at different speeds. *Med Sci Sports Exerc*, 39(9):1632–1641, September 2007.

Thomas Buschmann, Sebastian Lohmeier, Mathias Bachmayer, Heinz Ulbrich, and Friedrich Pfeiffer. A collocation method for real-time walking pattern generation. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 1–6. IEEE, 2007.

Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17 (1-19):16–16, 2004.

Katie Byl and Russ Tedrake. Approximate optimal control of the compass gait on rough terrain. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1258–1263. IEEE, 2008.

Germana Cappellini, Yuri P Ivanenko, Richard E Poppele, and Francesco Lacquaniti. Motor patterns in human walking and running. *Journal of neurophysiology*, 95(6): 3426–3437, 2006.

Robert R Caron, Robert C Wagenaar, Cara L Lewis, Elliot Saltzman, and Kenneth G Holt. Center of mass trajectory and orientation to ankle and knee in sagittal plane is maintained with forward lean when backpack load changes during treadmill walking. *Journal of biomechanics*, 46(1):70–76, 2013.

Giovanni A Cavagna, H Thys, and A Zamboni. The sources of external work in level walking and running. *The Journal of physiology*, 262(3):639–657, 1976.

Stephanie H Chanteau and James M Tour. Synthesis of anthropomorphic molecules: the nanoputians. *The Journal of organic chemistry*, 68(23):8750–8766, 2003.

Stefano Chiaverini, Bruno Siciliano, and Olav Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on control systems technology*, 2(2):123–134, 1994.

Min Gyu Choi, Jehee Lee, and Sung Yong Shin. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Transactions on Graphics (TOG)*, 22 (2):182–203, 2003.

Youngjin Choi, Bum-Jae You, and Sang-Rok Oh. On the stability of indirect ZMP controller for biped robot systems. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1966–1971. IEEE, IEEE, 2004.

Shu-Yun Chung and Oussama Khatib. Contact-consistent elastic strips for multi-contact locomotion planning of humanoid robots. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6289–6294. IEEE, 2015.

Luca Colasanto, Nikos G Tsagarakis, and Darwin G Caldwell. A compact model for the compliant humanoid robot COMAN. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 688–694. IEEE, IEEE, 2012.

Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

Steven H Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1983–1988. IEEE, 2005.

Steven H Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

Steven H Collins, Peter G Adamczyk, and Arthur D Kuo. Dynamic arm swinging in human walking. *Proceedings of the Royal Society of London B: Biological Sciences*, 276(1673):3679–3688, 2009.

Adria Colomé, Diego Pardo, Guillem Alenya, and Carme Torras. External force estimation during compliant robot manipulation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3535–3540. IEEE, 2013.

Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. Generalized biped walking control. *ACM Transactions on Graphics (TOG)*, 29(4):130, 2010.

Ronita L Cromwell. Movement strategies for head stabilization during incline walking. *Gait & posture*, 17(3):246–253, 2003.

Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 295–302. IEEE, 2014.

Behzad Dariush, Michael Gienger, Bing Jian, Christian Goerick, and Kikuo Fujimura. Whole body humanoid control from human motion descriptors. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2677–2684. IEEE, 2008.

P. de Leva. Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *J Biomech*, 29(9):1223–1230, September 1996.

Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 279–286. IEEE, 2014.

Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11):1940–1950, 2007.

Rosen Diankov. *Automated Construction of Robotic Manipulation Programs.* PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.

Nicolas Docquier, Antoine Poncelet, and Paul Fisette. ROBOTRAN: a powerful symbolic gnerator of multibody models. *Mechanical Sciences*, 4(1):199–219, 2013.

Jiro Doke, J. Maxwell Donelan, and Arthur D. Kuo. Mechanics and energetics of swinging the human leg. *J. Exp. Biol.*, 208(Pt 3):439–445, February 2005.

J. M. Donelan, R. Kram, and A. D. Kuo. Mechanical and metabolic determinants of the preferred step width in human walking. *Proc. Biol. Sci.*, 268(1480):1985–1992, October 2001.

J Maxwell Donelan, Rodger Kram, and Arthur D Kuo. Mechanical work for step-to-step transitions is a major determinant of the metabolic cost of human walking. *Journal of Experimental Biology*, 205(23):3717–3727, 2002.

Maureen Doyle. *A barrier algorithm for large nonlinear optimization problems.* PhD thesis, stanford university, 2003.

Ken Endo and Hugh Herr. A model of muscle–tendon function in human walking at self-selected speed. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(2):352–362, 2014.

Janice J Eng and David A Winter. Kinetic analysis of the lower limbs during walking: what information can be gained from a three-dimensional model? *Journal of biomechanics*, 28(6):753–758, 1995.

Johannes Englsberger, Christian Ott, Maximo A Roa, A Albu-Schaffer, and Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4420–4427. IEEE, IEEE, 2011.

Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *ICRA 2010-IEEE International Conference on Robotics and Automation*, pages 3733–3738, 2010.

Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

Felix Faber and Sven Behnke. Stochastic optimization of bipedal walking using gyro feedback and phase resetting. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 203–209. IEEE, 2007.

Maurice F Fallon, Matthew Antone, Nicholas Roy, and Seth Teller. Drift-Free Humanoid State Estimation fusing Kinematic, Inertial and LIDAR sensing. *Humanoid Robots (Humanoids)*, 2014.

Petros Faloutsos, Michiel Van de Panne, and Demetri Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 251–260. ACM, 2001.

Salman Faraji and Auke J. Ijspeert. Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5564–5571. IEEE, 2016.

Salman Faraji and Auke J. Ijspeert. 3lp: A linear 3d-walking model including torso and swing dynamics. *the international journal of robotics research*, 36(4):436–455, April 2017a.

Salman Faraji and Auke J. Ijspeert. Singularity-tolerant inverse kinematics for bipedal robots: An efficient use of computational power to reduce energy consumption. *IEEE Robotics and Automation Letters*, 2(2):1132–1139, 2017b.

Salman Faraji and Auke J. Ijspeert. Modeling Robot Geometries Like Molecules, Application to Fast Multicontact Posture Planning for Humanoids. *IEEE Robotics and Automation Letters*, 3(1):289–296, 2018a.

Salman Faraji and Auke J. Ijspeert. Scalable closed-form trajectories for periodic and non-periodic human-like walking. *arXiv preprint arXiv:1803.10048*, 2018b.

Salman Faraji, Soha Pouya, Rico Moeckel, and Auke J. Ijspeert. Compliant and adaptive control of a planar monopod hopper in rough terrain. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4818–4825. IEEE, 2013.

Salman Faraji, Soha Pouya, and Auke J. Ijspeert. Robust and agile 3d biped walking with steering capability using a footstep predictive approach. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014a.

Salman Faraji, Soha Pouya, and Auke J. Ijspeert. Versatile and Robust 3d Walking with a Simulated Humanoid Robot (Atlas): a Model Predictive Control Approach. In *Robotics and Automation (ICRA), IEEE International Conference on*, 2014b.

Salman Faraji, Luca Colasanto, and Auke J. Ijspeert. Practical considerations in using inverse dynamics on a humanoid robot: Torque tracking, sensor fusion and Cartesian control laws. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1619–1626. IEEE, 2015.

Salman Faraji, Philippe Muellhaupt, and Auke J. Ijspeert. Time-projection control to recover inter-sample disturbances, application to walking control. *arXiv preprint arXiv:1801.02150*, 2018a.

Salman Faraji, Hamed Razavi, and Auke J. Ijspeert. Push recovery with stepping strategy based on time-projection control. *arXiv preprint arXiv:1801.02151*, 2018b.

Salman Faraji, Amy R. Wu, and Auke J. Ijspeert. A simple model of mechanical effects to estimate metabolic cost of human walking. *Nature Scientific Reports*, 8(1):10998, July 2018c.

Claire T Farley and Thomas A McMahon. Energetics of walking and running: insights from simulated reduced-gravity experiments. *Journal of Applied Physiology*, 73(6): 2709–2712, 1992.

Siyuan Feng, X Xinjilefu, Weiwei Huang, and Christopher G Atkeson. 3d walking based on online optimization. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 21–27. IEEE, IEEE, 2013.

Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization-based Full Body Control for the DARPA Robotics Challenge. *Journal of Field Robotics*, 32 (2):293–312, 2015.

Siyuan Feng, X Xinjilefu, Christopher G Atkeson, and Joohyung Kim. Robust dynamic walking using online foot step optimization. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5373–5378. IEEE, 2016.

Daniel P. Ferris, Gregory S. Sawicki, and Monica A. Daley. A physiologist's perspective on robotic exoskeletons for human locomotion. *Int J HR*, 4(3):507–528, September 2007.

Carrie A Francis, Amy L Lenz, Rachel L Lenhart, and Darryl G Thelen. The modulation of forward propulsion, vertical support, and center of pressure by the plantarflexors during human walking. *Gait & posture*, 38(4):993–997, 2013.

Sean Gallagher, Jonisha Pollard, and William L Porter. Locomotion in restricted space: Kinematic and electromyographic analysis of stoopwalking and crawling. *Gait & posture*, 33(1):71–76, 2011.

Steven A Gard, Steve C Miff, and Arthur D Kuo. Comparison of kinematic and kinetic methods for computing the vertical motion of the body center of mass during walking. *Human movement science*, 22(6):597–610, 2004.

Peter Gawthrop, Ian Loram, Martin Lakie, and Henrik Gollee. Intermittent control: a computational theory of human control. *Biological cybernetics*, 104(1-2):31–51, 2011.

Peter J Gawthrop and Liuping Wang. Intermittent model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 221(7):1007–1018, 2007.

Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A Hoepflinger, and Roland Siegwart. Control of dynamic gaits for a quadrupedal robot. In *Robotics and automation (ICRA), 2013 IEEE international conference on*, pages 3287–3292. IEEE, 2013.

Thomas Geijtenbeek, Michiel van de Panne, and A Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):206, 2013.

Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(3):263–273, 2010.

Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

Joseph J Godges, Priscilla G MacRae, and Keith A Engelke. Effects of exercise on hip range of motion, trunk muscle performance, and gait economy. *Physical Therapy*, 73 (7):468–477, 1993.

Andrew Goldenberg, Beno Benhabib, and Robert Fenton. A complete generalized solution to the inverse kinematics of robots. *IEEE Journal on Robotics and Automation*, 1(1): 14–20, 1985.

Mario Gomes and Andy Ruina. Walking model with no energy cost. *Physical Review E*, 83(3):032901, 2011.

Keith E Gordon, Daniel P Ferris, and Arthur D Kuo. Metabolic and mechanical energy costs of reducing vertical center of mass movement during gait. *Archives of physical medicine and rehabilitation*, 90(1):136–144, 2009.

Jinger S. Gottschall and Rodger Kram. Energy cost and muscular activity required for leg swing during walking. *J. Appl. Physiol.*, 99(1):23–30, July 2005.

Alena Grabowski, Claire T. Farley, and Rodger Kram. Independent metabolic costs of supporting body weight and accelerating body mass during walking. *J. Appl. Physiol.*, 98(2):579–583, February 2005.

R Grasso, M Zago, and F Lacquaniti. Interactions between posture and locomotion: motor patterns in humans walking with bent posture versus erect posture. *Journal of Neurophysiology*, 83(1):288–300, 2000.

Robert D Gregg and Mark W Spong. Bringing the compass-gait bipedal walker to three dimensions. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4469–4474. IEEE, 2009.

Robert D Gregg, Adam K Tilton, Sal Candido, Timothy Bretl, and Mark W Spong. Control and planning of 3-D dynamic walking with asymptotically stable gait primitives. *Robotics, IEEE Transactions on*, 28(6):1415–1423, 2012.

DW Grieve and Ruth J Gear. The relationships between length of stride, step frequency, time of swing and speed of walking for children and adults. *Ergonomics*, 9(5):379–399, 1966.

Robert J Griffin, Georg Wiedebach, Sylvain Bertrand, Alexander Leonessa, and Jerry Pratt. Straight-Leg Walking Through Underconstrained Whole-Body Control. *arXiv preprint arXiv:1709.03660*, 2017.

Martin Grimmer, Matthew Holgate, Robert Holgate, Alexander Boehler, Jeffrey Ward, Kevin Hollander, Thomas Sugar, and André Seyfarth. A powered prosthetic ankle joint for walking and running. *Biomedical engineering online*, 15(3):141, 2016.

Gaël Guennebaud, Benoît Jacob, and others. *Eigen v3*. 2010.

Samuel R Hamner, Ajay Seth, Katherine M Steele, and Scott L Delp. A rolling constraint reproduces ground reaction forces and moments in dynamic simulations of walking, running, and crouch gait. *Journal of biomechanics*, 46(10):1772–1776, 2013.

Nandha Handharu, Jungwon Yoon, and Gabsoon Kim. Gait pattern generation with knee stretch motion for biped robot using toe and heel joints. In *Humanoid Robots, 2008. 8th IEEE-RAS International Conference on*, pages 265–270. IEEE, 2008.

S Javad Hasaneini, CJB Macnab, John EA Bertram, and Henry Leung. The dynamic optimization approach to locomotion dynamics: human-like gaits from a minimally-constrained biped model. *Advanced Robotics*, 27(11):845–859, 2013.

JM Hausdorff, L Zemany, C-K Peng, and AL Goldberger. Maturation of gait dynamics: stride-to-stride variability and its temporal organization in children. *Journal of Applied Physiology*, 86(3):1040–1047, 1999.

Kris Hauser, Timothy Bretl, and J-C Latombe. Non-gaited humanoid locomotion planning. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 7–12. IEEE, 2005.

Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, Martin Riedmiller, and others. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

N. C. Heglund and G. A. Cavagna. Mechanical work, oxygen consumption, and efficiency in isolated frog and rat muscle. *Am. J. Physiol.*, 253(1 Pt 1):C22–29, July 1987.

Warren J Hehre. *A guide to molecular mechanics and quantum chemical calculations*, volume 2. Wavefunction Irvine, CA, 2003.

H Hemami and CL Golliday. The inverted pendulum and biped stability. *Mathematical Biosciences*, 34(1):95–110, 1977.

Hooshang Hemami, F Weimer, and S Koozekanani. Some aspects of the inverted pendulum problem for modeling of locomotion systems. *IEEE Transactions on Automatic Control*, 18(6):658–661, 1973.

Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737, 2010a.

Andrei Herdt, Nicolas Perrin, and P-B Wieber. Walking without thinking about it. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 190–195. IEEE, 2010b.

Hugh Herr and Marko Popovic. Angular momentum in human walking. *Journal of Experimental Biology*, 211(4):467–481, 2008.

Alexander Herzog, Ludovic Righetti, Felix Grimminger, Peter Pastor, and Stefan Schaal. Momentum-based Balance Control for Torque-controlled Humanoids. *arXiv preprint arXiv:1305.2042*, 2013.

Alexander Herzog, Ludovic Righetti, Felix Grimminger, Peter Pastor, and Stefan Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IEEE International Conference on Intelligent Robotics Systems*, 2014.

Alexander Herzog, Stefan Schaal, and Ludovic Righetti. Structured contact force optimization for kino-dynamic motion generation. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2703–2710. IEEE, 2016.

Jessica K Hodgins and Nancy S Pollard. Adapting simulated behaviors for new characters. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 153–162. ACM Press/Addison-Wesley Publishing Co., 1997.

Michael G Hollars, Dan E Rosenthal, and Michael A Sherman. Sd/fast user's manual. *Symbolic Dynamics Inc*, 1991.

Bradley Howard, James Yang, and Burak Ozsoy. Optimal posture and supporting hand force prediction for common automotive assembly one-handed tasks. *Journal of Mechanisms and Robotics*, 6(2):021009, 2014.

Weiwei Huang, Junggon Kim, and Christopher G Atkeson. Energy-based optimal step planning for humanoids. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3124–3129. IEEE, 2013.

Sung-moon Hur, Sung-Kyun Kim, Yonghwan Oh, and Sang-Rok Oh. Joint torque servo of a high friction robot manipulator based on time-delay control with feed-forward friction compensation. In *RO-MAN, IEEE*, pages 37–42, 2012.

M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer. ANYmal - toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931, 2017.

Marco Hutter, C David Remy, Mark A Hoepflinger, and Roland Siegwart. High compliant series elastic actuation for the robotic leg ScarlETH. In *Proc. of the International Conference on Climbing and Walking Robots (CLAWAR)*, 2011.

Fumiya Iida, Yohei Minekawa, Jürgen Rummel, and André Seyfarth. Toward a human-like biped robot with compliant legs. *Robotics and Autonomous Systems*, 57(2):139–144, 2009.

Auke J. Ijspeert. Biorobotics: using robots to emulate and investigate agile locomotion. *Science*, 346(6206):196–203, October 2014.

Verne T Inman, Howard D Eberhart, and others. The major determinants in normal and pathological gait. *J Bone Joint Surg Am*, 35(3):543–558, 1953.

Yuri P Ivanenko, Renato Grasso, Velio Macellari, and Francesco Lacquaniti. Control of foot trajectory in human locomotion: role of ground contact forces in simulated reduced gravity. *Journal of neurophysiology*, 87(6):3070–3089, June 2002.

Craig T Johnson and Robert D Lorenz. Experimental identification of friction and its compensation in precise, position controlled mechanisms. *Industry Applications, IEEE Transactions on*, 28(6):1392–1398, 1992.

Peter Kaiser, Eren E Aksoy, Markus Grotz, Dimitrios Kanoulas, Nikos G Tsagarakis, and Tamim Asfour. Experimental evaluation of a perceptual pipeline for hierarchical affordance extraction. In *International Symposium on Experimental Robotics*, pages 136–146. Springer, 2016.

Shuuji Kajita and Kazuo Tani. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Robotics and Automation, 1991. Proceedings., IEEE International Conference on*, pages 1405–1411. IEEE, IEEE, 1991.

Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246. IEEE, 2001a.

Shuuji Kajita, Osamu Matsumoto, and Muneharu Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2299–2306. IEEE, 2001b.

Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. IEEE, 2003.

Thomas R Kane and David A Levinson. Multibody dynamics. *Journal of applied Mechanics*, 50(4b):1071–1078, 1983.

Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Eiichi Yoshida, and Jean-Paul Laumond. Disturbance observer that estimates external force acting on humanoid robots. In *Advanced Motion Control (AMC), 2012 12th IEEE International Workshop on*, pages 1–6. IEEE, 2012.

Kenji Kawamura, Akihiro Tokuhiro, and Hideo Takechi. Gait analysis of slope walking: a study on step length, stride width, time factors and deviation in the center of pressure. *Acta Medica Okayama*, 45(3):179–184, 1991.

Matthew Kelly and Andy Ruina. Non-linear robust control for inverted-pendulum 2d walking. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4353–4358. IEEE, 2015.

Matthew Kelly, Matthew Sheen, and Andy Ruina. Off-line controller design for reliable walking of ranger. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1567–1572. IEEE, IEEE, 2016.

Rafael Kelly, Jesús Llamas, and Ricardo Campa. A measurement procedure for viscous and coulomb friction. *Instrumentation and Measurement, IEEE Transactions on*, 49 (4):857–861, 2000.

D Casey Kerrigan, Mary K Todd, Ugo Della Croce, Lewis A Lipsitz, and James J Collins. Biomechanical gait alterations independent of speed in the healthy elderly: evidence for specific limiting impairments. *Archives of physical medicine and rehabilitation*, 79 (3):317–322, 1998.

O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *The I. J. of Robotics Research*, 3(1):43–53, 1987.

Donghyun Kim, Ye Zhao, Gray Thomas, Benito R Fernandez, and Luis Sentis. Stabilizing series-elastic point-foot bipeds using whole-body operational space control. *IEEE Transactions on Robotics*, 32(6):1362–1379, 2016.

David Kluger, Matthew J Major, Stefania Fatone, and Steven A Gard. The effect of trunk flexion on lower-limb kinetics of able-bodied gait. *Human movement science*, 33: 395–403, 2014.

Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, 2012.

Mayuresh V Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32 (10):1361–1379, 1996.

Elaine E Kozma, Nicole M Webb, William EH Harcourt-Smith, David A Raichlen, Kristiaan D'Août, Mary H Brown, Emma M Finestone, Stephen R Ross, Peter Aerts, and Herman Pontzer. Hip extensor mechanics and the evolution of walking and climbing capabilities in humans, apes, and fossil hominins. *Proceedings of the National Academy of Sciences*, page 201715120, 2018.

Przemyslaw Kryczka, Kenji Hashimoto, Hideki Kondo, Aiman Omer, Hun-ok Lim, and Atsuo Takanishi. Stretched knee walking with novel inverse kinematics for humanoid robots. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3221–3226. IEEE, 2011.

Przemyslaw Kryczka, Petar Kormushev, Nikos G Tsagarakis, and Darwin G Caldwell. Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3352–3357. IEEE, 2015.

Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2589–2594. IEEE, 2014.

A. D. Kuo. A simple model of bipedal walking predicts the preferred speed-step length relationship. *J Biomech Eng*, 123(3):264–269, June 2001.

Arthur D Kuo. Stabilization of lateral motion in passive dynamic walking. *The International journal of robotics research*, 18(9):917–930, September 1999.

Arthur D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *J Biomech Eng*, 124(1):113–120, February 2002.

Arthur D Kuo, J Maxwell Donelan, and Andy Ruina. Energetic consequences of walking like an inverted pendulum: step-to-step transitions. *Exercise and sport sciences reviews*, 33(2):88–97, 2005.

Ryo Kurazume, Shuntaro Tanaka, Masahiro Yamashita, Tsutomu Hasegawa, and Kyushu Yoneda. Straight legged walking of a biped robot. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 337–343. IEEE, 2005.

Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 155–162. ACM, 1996.

Dinh Phong Le, Junho Choi, and Sungchul Kang. External force estimation using joint torque sensors and its application to impedance control of a robot manipulator. In *Control, Automation and Systems (ICCAS), 2013 13th International Conference on*, pages 1794–1798. IEEE, 2013.

Luc Le Tien, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. Friction observer and compensation for control of robots with joint torque measurement. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3789–3795. IEEE, 2008.

Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 491–500. ACM, 2002.

Kwan S Lee, Don B Chaffin, GD Herrin, and AM Waikar. Effect of handle height on lower-back loading in cart pushing and pulling. *Applied Ergonomics*, 22(2):117–123, 1991.

Mark Leeney. Fast quaternion slerp. *International Journal of Computer Mathematics*, 86(1):79–84, 2009.

Sébastien Lengagne, Joris Vaillant, Eiichi Yoshida, and Abderrahmane Kheddar. Generation of whole-body optimal dynamic multi-contact motions. *The International Journal of Robotics Research*, 32(9-10):1104–1119, 2013.

Alain Leroux, Joyce Fung, and Hugues Barbeau. Postural adaptation to walking on inclined surfaces: I. Normal strategies. *Gait & Posture*, 15(1):64–74, February 2002.

Ling Li and Xiaoyan Liu. Simulating human walking on special terrain: up and down slopes. *Computers & Graphics*, 24(3):453–463, 2000.

Nejc Likar and Leon Zlajpah. External joint torque-based estimation of contact information. *International Journal of Advanced Robotic Systems*, 11, 2014.

Hun-ok Lim, Samuel Agus Setiawan, and Atsuo Takanishi. Balance and impedance control for biped humanoid robot locomotion. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 494–499. IEEE, 2001.

May Q. Liu, Frank C. Anderson, Michael H. Schwartz, and Scott L. Delp. Muscle contributions to support and progression over a range of walking speeds. *J Biomech*, 41(15):3243–3252, November 2008.

Vipul Lugade, Victor Lin, and Li-Shan Chou. Center of mass and base of support interaction during gait. *Gait & posture*, 33(3):406–411, 2011.

Ian R Manchester and Jack Umenberger. Real-time planning with primitives for dynamic walking over uneven terrain. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4639–4646. IEEE, 2014.

Rodolfo Margaria. Positive and negative work performances and their efficiencies in human locomotion. *Int. Z. Angew. Physiol. Einschl. Arbeitsphysiol.*, 25(4):339–351, December 1968.

Tim Marler, Lindsey Knake, and Ross Johnson. Optimization-based posture prediction for analysis of box lifting tasks. In *International Conference on Digital Human Modeling*, pages 151–160. Springer, 2011.

Firas Massaad, Thierry M Lejeune, and Christine Detrembleur. The up and down bobbing of human walking: a compromise between muscle work and efficiency. *The Journal of physiology*, 582(2):789–799, 2007.

Jacob Mattingley and Stephen Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

Christophe Maufroy, H Moritz Maus, and André Seyfarth. Simplified control of upright walking by exploring asymmetric gaits induced by leg damping. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 491–496. IEEE, 2011.

H-M Maus, SW Lipfert, M Gross, J Rummel, and A Seyfarth. Upright human gait did not provide a major mechanical challenge for our ancestors. *Nature communications*, 1:70, 2010.

T. McGeer. Passive dynamic walking. *IJRR*, 9(2):62–82, 1990.

Andrew Stuart McIntosh, Karen T Beatty, Leanne N Dwan, and Deborah R Vickers. Gait dynamics on an inclined walkway. *Journal of biomechanics*, 39(13):2491–2502, 2006.

Olivier Michel. Cyberbotics ltd. webots$^{\text{TM}}$: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5, 2004.

Michael Mistry and Ludovic Righetti. Operational space control of constrained and underactuated systems. *Robotics: Science and systems VII*, pages 225–232, 2012.

Michael Mistry, Jonas Buchli, and Stefan Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3406–3412. IEEE, 2010.

S. Mochon and T. A. McMahon. Ballistic walking. *J Biomech*, 13(1):49–57, 1980.

Michael B. Monagan, Keith O. Geddes, K. Michael Heal, George Labahn, Stefan M. Vorkoetter, James McCarron, and Paul DeMarco. *Maple 10 Programming Guide*. Maplesoft, Waterloo ON, Canada, 2005.

LA Month and R Ho Rand. An application of the Poincaré map to the stability of nonlinear normal modes. *Journal of Applied Mechanics*, 47(3):645–651, 1980.

Igor Mordatch, Martin De Lasa, and Aaron Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics (TOG)*, 29(4):71, 2010.

Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31 (4):43, 2012.

Federico L Moro, Nikolaos 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 *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 364–370. IEEE, IEEE, 2011.

Mohamad Mosadeghzad, Gustavo A Medrano-Cerda, Nikos Tsagarakis, and Darwin G Caldwell. Impedance control with inner PI torque loop: Disturbance attenuation and impedance emulation. In *Robotics and Biomimetics (ROBIO), International Conference on*, pages 1497–1502. IEEE, 2013.

Roy Müller, Christian Rode, Soran Aminiaghdam, Johanna Vielemeyer, and Reinhard Blickhan. Force direction patterns promote whole body stability even in hip-flexed walking, but not upper body stability in human upright walking. *Proceedings. Mathematical, Physical, and Engineering Sciences*, 473(2207):20170404, November 2017.

Nirut Naksuk and CS George Lee. Utilization of movement prioritization for whole-body humanoid robot trajectory generation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1079–1084. IEEE, 2005.

Richard R Neptune and Craig P McGowan. Muscle contributions to whole-body sagittal plane angular momentum during walking. *Journal of biomechanics*, 44(1):6–12, 2011.

Tomoyuki Noda, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Super-flexible skin sensors embedded on the whole body, self-organizing based on haptic interactions. *Human-Robot Interaction in Social Robotics*, page 183, 2012.

Katsuhiko Ogata. *Discrete-time control systems*, volume 2. Prentice Hall Englewood Cliffs, NJ, 1995.

Yu Ogura, Hun-ok Lim, and Atsuo Takanishi. Stretch walking pattern generation for a biped humanoid robot. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 352–357. IEEE, 2003.

Yu Ogura, Kazushi Shimomura, Hideki Kondo, Akitoshi Morishima, Tatsu Okubo, Shimpei Momoki, Hun-ok Lim, and Atsuo Takanishi. Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3976–3981. IEEE, 2006.

Masafumi Okada and Tetsuro Miyazaki. Motion design of multi degrees of freedom robot with dynamical consistency using motion reduction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3102–3107. IEEE, 2013.

Henrik Olsson, Karl J \AAström, Carlos Canudas de Wit, Magnus Gäfvert, and Pablo Lischinsky. Friction models and friction compensation. *European journal of control*, 4 (3):176–195, 1998.

Aiman Musa M Omer, Yu Ogura, Hideki Kondo, Akitoshi Morishima, Giuseppe Carbone, Marco Ceccarelli, Hun-ok Lim, and Atsuo Takanishi. Development of a humanoid robot having 2-DOF waist and 2-DOF trunk. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 333–338. IEEE, IEEE, 2005.

David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, 2013.

Justus D. Ortega and Claire T. Farley. Minimizing center of mass vertical movement increases metabolic cost in walking. *J. Appl. Physiol.*, 99(6):2099–2107, December 2005.

Jong H Park and Kyong D Kim. Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3528–3533. IEEE, 1998.

Jong Hyeon Park and Hoam Chung. Impedance control and modulation for stable footing in locomotion of biped robots. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 3, pages 1786–1791. IEEE, 1999.

Pragasen Pillay and Ramu Krishnan. Modeling, simulation, and analysis of permanent-magnet motor drives. II. The brushless DC motor drive. *Industry Applications, IEEE Transactions on*, 25(2):274–279, 1989.

Brahayam Ponton, Alexander Herzog, Stefan Schaal, and Ludovic Righetti. A convex model of humanoid momentum dynamics for multi-contact motion generation. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 842–849. IEEE, 2016.

Michael Posa, Cecilia Cantu, and Russ tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 399–406. IEEE, 1995.

Jerry Pratt and Gill Pratt. Intuitive control of a planar bipedal walking robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2014–2021. IEEE, 1998.

Jerry Pratt, Chee-Meng Chew, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.

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(10):1117–1133, 2012.

Marc H Raibert. *Legged robots that balance.* MIT press, 1986.

Marc H Raibert, H Benjamin Brown Jr, and Michael Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75–92, 1984.

John R Rebula, Peter D Neuhaus, Brian V Bonnlander, Matthew J Johnson, and Jerry E Pratt. A controller for the littledog quadruped walking on rough terrain. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1467–1473. IEEE, 2007.

Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan, and Stefan Schaal. Optimal distribution of contact forces with inverse-dynamics control. *The International Journal of Robotics Research*, 32(3):280–298, 2013.

Dustyn Roberts, Howard Hillstrom, and Joo H. Kim. Instantaneous Metabolic Cost of Walking: Joint-Space Dynamic Model with Subject-Specific Heat Rate. *PLOS ONE*, 11(12):e0168070, December 2016.

Rodrigo Gomes da Rosa, Natalia Andrea Gomeñuka, Henrique Bianchi de Oliveira, and Leonardo Alexandre Peyré-Tartaruga. Inclined Weight-Loaded Walking at Different Speeds: Pelvis-Shoulder Coordination, Trunk Movements and Cost of Transport. *Journal of Motor Behavior*, 50(1):73–79, February 2018.

Nicholas Rotella, Michael Bloesch, Ludovic Righetti, and Stefan Schaal. State estimation for a humanoid robot. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 952–958. IEEE, 2014.

Juergen Rummel, Yvonne Blum, H Moritz Maus, Christian Rode, and Andre Seyfarth. Stable and robust walking with compliant legs. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5250–5255. IEEE, 2010.

Devjani Saha, Steven Gard, and Stefania Fatone. The effect of trunk flexion on able-bodied gait. *Gait & posture*, 27(4):653–660, 2008.

Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

Gregory S Sawicki and Daniel P Ferris. Mechanics and energetics of incline walking with robotic ankle exoskeletons. *Journal of Experimental Biology*, 212(1):32–41, 2009.

H Bernhard Schlegel. Geometry optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):790–809, 2011.

Luis Sentis, Jaeheung Park, and Oussama Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on robotics*, 26 (3):483–501, 2010.

Maziar Sharbafi and Andre Seyfarth. FMCH: a new model for human-like postural control in walking. In *Intelligent Robots and Systems, 2015.(IROS 2015). Proceedings. 2015 IEEE/RSJ International Conference on*, volume 1, pages 5742–5747. IEEE, 2015.

Maziar Sharbafi, Christophe Maufroy, H Moritz Maus, Andre Seyfarth, Majid Nili Ahmadabadi, and Mohammad Javad Yazdanpanah. Controllers for robust hopping with upright trunk based on the virtual pendulum concept. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2222–2227. IEEE, 2012.

Maziar Sharbafi, Andre Seyfarth, and Guoping Zhao. Locomotor sub-functions for control of assistive wearable robots. *Frontiers in neurorobotics*, 11:44, 2017.

Mark Snaterse, Robert Ton, Arthur D Kuo, and J Maxwell Donelan. Distinct fast and slow processes contribute to the selection of preferred step frequency during human walking. *Journal of Applied Physiology*, 110(6):1682–1690, 2011.

Seungmoon Song and Hartmut Geyer. Regulating speed and generating large speed transitions in a neuromuscular human walking model. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 511–516. IEEE, IEEE, 2012.

Emmanouil Spyrakos-Papastavridis, Gustavo A Medrano-Cerda, Nikos G Tsagarakis, Jian S Dai, and Darwin G Caldwell. A push recovery strategy for a passively compliant humanoid robot using decentralized lqr controllers. In *Mechatronics (ICM), 2013 IEEE International Conference on*, pages 464–470. IEEE, 2013.

Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

Manoj Srinivasan. Fifteen observations on the structure of energy-minimizing gaits in many simple biped models. *J R Soc Interface*, 8(54):74–98, January 2011.

Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72, 2006.

Benjamin Stephens. Humanoid push recovery. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 589–595. IEEE, 2007.

Benjamin J Stephens and Christopher G Atkeson. Dynamic balance force control for compliant humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1248–1255. IEEE, 2010a.

Benjamin J Stephens and Christopher G Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 52–59. IEEE, 2010b.

MW Strohmayr and David Schneider. The dlr artificial skin step ii: Scalability as a prerequisite for whole-body covers. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4721–4728. IEEE, 2013.

Tomomichi Sugihara. Solvability-unconcerned inverse kinematics by the levenberg–marquardt method. *IEEE Transactions on Robotics*, 27(5):984–991, 2011.

Wael Suleiman. On inverse kinematics with inequality constraints: new insights into minimum jerk trajectory generation. *Advanced Robotics*, pages 1–9, 2016.

Toru Takenaka, Takashi Matsumoto, and Takahide Yoshiike. Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1084–1091. IEEE, 2009.

Yaolong Tan and Ioannis Kanellakopoulos. Adaptive nonlinear friction compensation with parametric uncertainties. In *American Control Conference, 1999. Proceedings of the 1999*, volume 4, pages 2511–2515. IEEE, 1999.

Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

Silvio Traversaro, Andrea Del Prete, Riccardo Muradore, Lorenzo Natale, and Francesco Nori. Inertial parameter identification including friction and motor dynamics. *arXiv preprint arXiv:1410.4410*, 2014.

N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settimi, A. Rocchi, and A. Ajoudani. Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259, 2017.

Nikos G Tsagarakis, Stephen Morfey, Gustavo Medrano Cerda, Li Zhibin, and Darwin G Caldwell. Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 673–678. IEEE, 2013.

Yao-Yang Tsai, Wen-Chieh Lin, Kuangyou B Cheng, Jehee Lee, and Tong-Yee Lee. Real-time physics-based 3d biped character animation using an inverted pendulum model. *IEEE transactions on visualization and computer graphics*, 16(2):325–337, 2010.

P Huang Tzu-wei and Arthur D Kuo. Mechanics and energetics of load carriage during human walking. *Journal of Experimental Biology*, 217(4):605–613, 2014.

Brian R. Umberger. Stance and swing phase costs in human walking. *J R Soc Interface*, 7(50):1329–1340, September 2010.

Brian R Umberger, Karin GM Gerritsen, and Philip E Martin. A model of human muscle energy expenditure. *Computer methods in biomechanics and biomedical engineering*, 6 (2):99–111, 2003.

Joris Vaillant, Abderrahmane Kheddar, Hervé Audren, François Keith, Stanislas Brossette, Adrien Escande, Karim Bouyarmane, Kenji Kaneko, Mitsuharu Morisawa, Pierre Gergondet, et al. Multi-contact vertical ladder climbing with an hrp-2 humanoid. *Autonomous Robots*, 40(3):561–580, 2016.

Joris Vaillant, Karim Bouyarmane, and Abderrahmane Kheddar. Multi-character physical and behavioral interactions controller. *IEEE transactions on visualization and computer graphics*, 23(6):1650–1662, 2017.

Nicolas Van der Noot, Auke J. Ijspeert, and Renaud Ronsse. Biped gait controller for large speed variations, combining reflexes and a central pattern generator in a neuromuscular model. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6267–6274. IEEE, IEEE, 2015.

Kirill Van Heerden. Planning COM trajectory with variable height and foot position with reactive stepping for humanoid robots. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6275–6280. IEEE, IEEE, 2015.

Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated Annealing: Theory and Applications*, pages 7–15. Springer, 1987.

Mark Vlutters, Edwin HF Van Asseldonk, and Herman Van der Kooij. Center of mass velocity-based predictions in balance recovery following pelvis perturbations during human walking. *Journal of experimental biology*, 219(10):1514–1523, 2016.

L Vogt and W Banzer. Measurement of lumbar spine kinematics in incline treadmill walking. *Gait & Posture*, 9(1):18–23, 1999.

Alexandra S Voloshina, Arthur D Kuo, Monica A Daley, and Daniel P Ferris. Biomechanics and energetics of walking on uneven terrain. *Journal of Experimental Biology*, 216(21): 3963–3970, 2013.

Miomir Vukobratović and Branislav Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

Jack M Wang, David J Fleet, and Aaron Hertzmann. Optimizing walking controllers. *ACM Transactions on Graphics (TOG)*, 28(5):168, 2009.

Jack M Wang, Samuel R Hamner, Scott L Delp, and Vladlen Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. Graph.*, 31(4):25, 2012.

L-CT Wang and Chih-Cheng Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.

Wei Wang, Robert NK Loh, and Edward Y Gu. Passive compliance versus active compliance in robot-based automated assembly systems. *Industrial Robot: An International Journal*, 25(1):48–57, 1998.

Alexander Werner, Bernd Henze, Diego A Rodriguez, Jonathan Gabaret, Oliver Porges, and Máximo A Roa. Multi-contact planning and control for a torque-controlled humanoid robot. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5708–5715. IEEE, 2016.

Eric R Westervelt, Jessy W Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback control of dynamic bipedal robot locomotion*, volume 28. CRC press, 2007.

Eric C Whitman. *Coordination of Multiple Dynamic Programming Policies for Control of Bipedal Walking.* PhD thesis, Carnegie Mellon University, 2013.

Eric C Whitman and Christopher G Atkeson. Control of instantaneously coupled systems applied to humanoid walking. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 210–217. IEEE, 2010.

Keith R Williams. The relationship between mechanical and physiological energy estimates. *Medicine and Science in Sports and Exercise*, 17(3):317–325, 1985.

Alexander W Winkler, C Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018.

David A Winter. Foot trajectory in human gait: a precise and multifactorial motor control task. *Physical therapy*, 72(1):45–53, 1992.

J. M. Workman and B. W. Armstrong. Metabolic cost of walking: equation and model. *J. Appl. Physiol.*, 61(4):1369–1374, October 1986.

Amy R Wu and Arthur D Kuo. Determinants of preferred ground clearance during swing phase of human walking. *Journal of Experimental Biology*, 219(19):3106–3113, October 2016.

Amy Ronning Wu, Salman Faraji, and Auke J. Ijspeert. Effects of trunk lean and ground slope on leg angle asymmetry during gait. *In preparation*, 2018.

X Xinjilefu, Siyuan Feng, and Christopher G Atkeson. Dynamic state estimation using Quadratic Programming. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 989–994. IEEE, 2014a.

X Xinjilefu, Siyuan Feng, Weiwei Huang, and C Atkeson. Decoupled State Estimation for Humanoids Using Full-body Dynamics. In *IEEE Intl. Conf. on Robotics and Automation (ICRA), Hong Kong, China*, 2014b.

KangKang Yin, Kevin Loken, and Michiel Van de Panne. Simbicon: Simple biped locomotion control. In *ACM Transactions on Graphics (TOG)*, volume 26, page 105. ACM, 2007.

Yangwei You, Songyan Xin, Chengxu Zhou, and Nikos Tsagarakis. Straight leg walking strategy for torque-controlled humanoid robots. In *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on*, pages 2014–2019. IEEE, 2016.

M. Y. Zarrugh and C. W. Radcliffe. Predicting metabolic cost of level walking. *Eur J Appl Physiol Occup Physiol*, 38(3):215–223, April 1978.

MY Zarrugh, FN Todd, and HJ Ralston. Optimization of energy expenditure during level walking. *European journal of applied physiology and occupational physiology*, 33 (4):293–306, 1974.

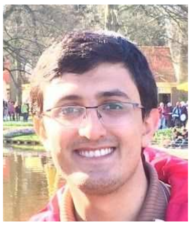
P. Zaytsev, S.J. Hasaneini, and A. Ruina. Two steps is enough: No need to plan far ahead for walking balance. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6295–6300, May 2015.

Karl E Zelik and Arthur D Kuo. Human walking isn't all hard work: evidence of soft tissue contributions to energy dissipation and return. *Journal of Experimental Biology*, 213(24):4257–4264, 2010.

Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.

Mingguo Zhao, Ji Zhang, Hao Dong, Yu Liu, Liguo Li, and Xuemin Su. Humanoid Robot Gait Generation Based on Limit Cycle Stability. In *RoboCup*, pages 403–413. Springer, 2008.

Salman Faraji



EPFL STI IBI-STI BIOROB
ME-D 1 1215
Station 9
CH-1015 Lausanne
Switzerland

Date of Birth: 24 Oct. 1988
Email: salman.faraji@epfl.ch
Website: http://biorob.epfl.ch/people/faraji
Linkedin: https://ch.linkedin.com/in/salman-faraji-9b841964

Education

PhD **École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.** 2013 - 2018

Doctoral program in Robotics and Control, Biorob laboratory.

M.Sc. **École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.** 2011 - 2013

Micro-Engineering, Robotics and Autonomous Systems.

Master thesis: Model-based walking controller for humanoid robots.

B.Sc. **Sharif University of Technology, Iran**. 2007 - 2011

Electrical Engineering, Digital Systems and Control.

Bachelor thesis: A new structure for mobile modular robots.

Research Interests

Control, Optimization, Robot and Human Locomotion, Learning

Awards & Honors

2007: Ranked 3rd out of 400,000**, nationwide university entrance exam.**

2007: Member of the National Iranian **Elites Foundation**.

2011: Ranked 2nd, Robocup small-size soccer league, Istanbul, Turkey.

2011: Ranked 1st out of 190, GPA: 19.36/20**,** Sharif University of Technology, Iran.

2011: Best Bachelor Project Award, Sharif University of Technology, Iran.

2013: Ranked 4th out of 61, GPA: 5.66/6**,** Micro-Engineering Master's program, EPFL, Switzerland.

2013: Best Master Project Award, Commune d'Ecublens, EPFL, Switzerland.

Languages skills

English (Fluent), German (B1-B2), French (A2), Arabic (A2), Persian (Native)

Sports - Activities

Volleyball, Swimming, table tennis.

Lifesaving diploma from Iranian national lifesaving and diving federation.

Publications

JOURNAL ARTICLES

S. Faraji, A. R. Wu and A. Ijspeert. A simple model of mechanical effects to estimate metabolic cost of human walking, in Nature Scientific Reports, 8(1):10998, 2018.

S. Faraji and A. Ijspeert. 3LP: a linear 3D-walking model including torso and swing dynamics, in The International Journal of Robotics Research (IJRR), vol. 36, num. 4, p. 436 - 455, 2017.

S. Faraji and A. Ijspeert. Modeling robot geometries like molecules, application to fast multi-contact posture planning for humanoids, in IEEE Robotics and Automation Letters (RA-L), 2017.

S. Faraji and A. Ijspeert. Singularity-tolerant inverse kinematics for bipedal robots: An efficient use of computational power to reduce energy consumption, in IEEE Robotics and Automation Letters (RA-L), vol. 2, num. 2, p. 1132 - 1139, 2017.

S. Faraji and M. S. Tavazoei. The effect of fractionality nature in differences between computer simulation and experimental results of a chaotic circuit, in Central European Journal Of Physics, vol. 11, num. 6, p. 836-844, 2013.

S. Haghzad, **S. Faraji** and S. Bagheri. Finding Proper Configurations for Modular Robots by Using Genetic Algorithm on Different Terrains, in International Journal of Materials, Mechanics and Manufacturing, vol. 1, num. 4, p. 360-365, 2013.

CONFERENCE ARTICLES

S. Faraji and A. Ijspeert. Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics. Intelligent Robots and Systems (IROS), Daejeon, South Korea, 2016.

S. Faraji, L. Colasanto and A. Ijspeert. Practical considerations in using inverse dynamics on a humanoid robot: torque tracking, sensor fusion and Cartesian control laws. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015.

S. Faraji, S. Pouya and A. Ijspeert. Robust and Agile 3D Biped Walking With Steering Capability Using a Footstep Predictive Approach. Robotics Science and Systems (RSS), Berkeley, CA, USA, 2014.

S. Faraji, S. Pouya, C. G. Atkeson and A. J. Ijspeert. Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach. 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

S. Faraji, S. Pouya, R. Moeckel and A. J. Ijspeert. Compliant and adaptive control of a planar monopod hopper in rough terrain. 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013.

JOURNAL ARTICLES IN PROCESS

S. Faraji, P. Müllhaupt and A. Ijspeert. Time-projection control to recover inter-sample disturbances, application to walking control, under review in the International Journal of Robotics Research (IJRR).

S. Faraji, H. Razavi and A. Ijspeert. Push recovery with stepping strategy based on time-projection control, under review in the International Journal of Robotics Research (IJRR).

S. Faraji and A. Ijspeert. Scalable closed-form trajectories for periodic and non-periodic human-like walking, submitted to ACM Transaction on Graphics (TOG).

J. Arreguit, **S. Faraji** and A. Ijspeert. Fast multi-contact motion planning with a 5-link model and molecule-inspired formulations for humanoids, submitted.

A. R. Wu, **S. Faraji** and A. Ijspeert. Modeling asymmetries of torso bending in inclined walking over different speeds, in preparation.

POSTERS

A. R. Wu, , **S. Faraji**, A. Ijspeert. Simple combinations of major energy components to predict cost of human walking. IROS workshop, Vancouver, Canada, 2017.

J. P. Arreguit, **S. Faraji**, A. Ijspeert. Multi-contact motion planning with a five-mass template model and vector-based equations. IROS workshop, Vancouver, Canada, 2017.

S. Faraji and A. Ijspeert. A singularity-tolerant inverse kinematics including joint position and velocity limitations. Dynamic Walking, Mariehamn, Finland, 2017.

S. Faraji and A. Ijspeert. A virtual tactile sensing suit for humanoids based on dynamic equations and internal sensors. Dynamic Walking, Holly, Michigan, USA, 2016.

S. Faraji and A. Ijspeert. Time-Projection control on 3LP, a simple idea to deal with intermittent pushes online. Dynamic Walking, Holly, Michigan, USA, 2016.

S. Faraji and A. Ijspeert. 3LP: A linear model of locomotion including falling, swing and torso dynamics. Dynamic Walking, Holly, Michigan, USA, 2016.

S. Faraji, S. Pouya and A. Ijspeert. Robust 3D Walking Using Inverse Dynamics And Footstep Planning With Model Predictive Control. 9th Dynamic Walking Conference (DWC 2014), ETH Zurich, Switzerland, 2014.

S. Pouya, **S. Faraji**, R. Möckel and A. Ijspeert. Dynamics Modeling and Control Architecture for Efficient, Manoeuvrable and Robust Monoped Hopping over Rough Terrain. 8th Dynamic Walking Conference (DWC 2013), Pittsburgh, USA, 2013.

TALKS

S. Faraji, Model based control approaches for multi-modal locomotion of a humanoid robot, Automatic Control Laboratory, EPFL, 2014.

S. Faraji, Multi-Modal locomotion, Movement generation and control group, Max Planck Institute, 2015

S. Faraji, 3LP walking model: a Fresh Breath for optimizations, Workshop on Robust Optimization-Based Control and Planning for Legged Robots, ICRA conference, 2016.

S. Faraji, Walking Control of Bipedal Robots with Heavy Legs Using Foot Stepping Strategy, Robotic Systems Laboratory, ETH Zurich, 2018.