

# Optimization-based motion generation for multiped robots in contact scenarios

**Doctoral Thesis****Author(s):**

Herzog, Alexander

**Publication date:**

2017-10-02

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000199099>

**Rights / license:**

[In Copyright - Non-Commercial Use Permitted](#)

DISS. ETH NO. 24533

# **Optimization-based motion generation for multiped robots in contact scenarios**

A thesis submitted to attain the degree of  
DOCTOR OF SCIENCES of ETH ZURICH  
(Dr. sc. ETH Zurich)

presented by  
**ALEXANDER HERZOG**

Diploma Computer Science, Karlsruhe Institute of Technology

born on May 23, 1986  
citizen of Germany

accepted on the recommendation of

Prof. Dr. Jonas Buchli (ETH Zurich), Examiner  
Prof. Dr. Ludovic Righetti (New York University, USA | Max Planck Intelligent Systems, Tübingen), Co-examiner  
Prof. Dr. Stefan Schaal (Max Planck Intelligent Systems Tübingen), Co-examiner  
Prof. Dr. Roland Siegwart (ETH Zurich), Co-examiner

2017

*dedicated to Ella and Liam Herzog*

Movement Generation and Control Group  
Autonomous Motion Department  
Max Planck Institute for Intelligent Systems, Tübingen  
Germany

Agile & Dexterous Robotics Lab  
Department of Mechanical and Process Engineering  
ETH Zurich  
Switzerland

© 2017 Alexander Herzog. All rights reserved.

# Abstract

---

The introduction of legged robots into society has the potential to impact many aspects of our lives. The versatile morphology of robots with arms and legs, or *multipeds*, allows them to operate in a broad spectrum of environments. For example, legged robots could assist at construction sites or locomote in rocky terrain under persistent interaction with the environment through contact. These behaviors require algorithms that translate high level operator commands into movement and contact interaction with the environment, steering robots towards successful task completion.

Traditional planning and control methods are limited by a core property of legged robots; *they can fall*. Control algorithms must respond quickly in order to keep a multiped in balance, and planners are required to respond to changes in the environment. This thesis presents a movement control framework designed to control complex contact interactions between a multiped robot and its environment. An optimal control based architecture is proposed that treats planning and control as one coherent optimization program. The lookahead capabilities of planners are combined with the reactivity of feedback controllers in a multi-layer optimization architecture. An optimal control problem plans whole-body motion and contact forces over a lookahead while a fast feedback loop computes motor commands consistent with the current higher layer plan. At the core of our analysis we exploit the mathematical structure relating the robot's center of mass to the contact forces between the robot and the environment, leading to a specialized algorithm combining planning and control.

A trajectory optimization method is proposed to compute whole-body motion and corresponding contact force plans. It is decomposed into two sub-problems with reduced complexity and solved with a composition of specialized solvers. A reactive feedback controller tracks the high level plan using hierarchical inverse dynamics. Our control architecture successfully plans and controls a stepping task over complex terrain.

Optimal control depends on an accurate robot model and often ignores real-world sensor noise and delay. Further, the short reaction time required to prevent legged robots from falling limits the amount of computation time permitted. Therefore, we demonstrate the validity of our approach and modeling assumptions through experimental validation on a torque controlled humanoid robot.

An efficient implementation of our hierarchical inverse dynamics solver in a 1kHz control loop exhibits robust push rejection in a series of balancing experiments. The proposed controller successfully balances in a wide range of scenarios, including on a moving see-saw, on a sliding platform, and being pushed while standing on one foot and tracking a desired trajectory with the other.



## Zusammenfassung

---

Die Integration von Robotern in unsere Gesellschaft hat das Potenzial viele Aspekte unseres Lebens zu beeinflussen. Aufgrund ihrer vielseitigen Morphologie können Roboter mit Armen und Beinen, oder *multiped* Roboter, in einem breiten Spektrum unterschiedlicher Umgebungen agieren. Laufroboter könnten zum Beispiel auf Baustellen assistieren oder sich durch steiniges Gelände fortbewegen oder andere Aufgaben erledigen, die ähnlich eine permanente Wechselwirkung zwischen dem Roboter und seiner Umgebung durch ihren mechanischen Kontakt erfordern. Um derartige Bewegungsabläufe zu realisieren, werden Algorithmen benötigt, die High-Level-Befehle in Ganzkörper-Bewegung und Kontaktkräfte zwischen Roboter und der Umgebung transferieren und sie somit aufgabenorientiert steuern können. Allerdings sind Konventionelle Planungs- und Regelungsalgorithmen aufgrund einer Kerneigenschaft von mehrbeinigen Robotern nur bedingt einsetzbar: *sie können fallen*. Regler müssen schnell reagieren, um Roboter auf Beinen in Balance zu halten und Planungsalgorithmen müssen zeitnah auf Änderungen in der Umgebung antworten. In der vorliegenden Arbeit wird eine Methode zur Regelung komplexer Bewegungsabläufe von mehrbeinigen Robotern und ihrer Interaktion mit der Umgebung durch Kontakt vorgestellt. Eine optimierungsbasierter Architektur, die Planung und Regelung in Form eines kohärenten Optimierungsprogramms auffasst, wird diskutiert. Diese mehrschichtige Optimierungsarchitektur kombiniert das Vorausblicken von Planern und die Reaktionsfähigkeit von Regelungsschleifen. Ein optimales Steuerungsproblem plant Ganzkörper-Bewegungen und Kontaktkräfte über einen Zeithorizont, während eine schnelle Rückkopplungsschleife Aktorbefehle in Kohärenz mit dem aktuellen Plan der höheren Ebene berechnet. Im Kern unserer Analyse nutzen wir die mathematische Struktur, die aus der Relation zwischen dem Massenschwerpunkt und den Kontaktkräften eines Roboters hervorgeht und die zu spezialisierten Algorithmen führen, die Planung und Regelung kombinieren.

Es wird eine Methode zur Trajektorienoptimierung vorgestellt, mit der Ganzkörper-Bewegungen und entsprechende Kontaktkräfte berechnet werden können. Das Optimierungsprogramm wird in zwei Teilprobleme verringerter Komplexität heruntergebrochen und mit einer Komposition spezialisierter Solver gelöst. Der Plan von höherer Ebene wird mit einem reaktiven Regler unter Verwendung von Hierarchical Inverse Dynamics getrackt. Unsere Regelungsarchitektur ist in der Lage das Laufen in einem komplexen Gelände zu planen und zu regeln.

Optimale Steuerung basiert auf akkurate Roboter Modellen und ignoriert häufig Sensorrauschen und Laufzeitverzögerungen, welche in der Praxis oft üblich sind. Des Weiteren ist die, den Regelungsalgorithmen zur Verfügung stehende Rechenzeit,

## ***Zusammenfassung***

---

durch die kurze Reaktionszeit, die benötigt wird um Roboter am Fallen zu hindern, begrenzt. In dieser Arbeit wird daher die Gültigkeit unseres Ansatzes und getroffener Modellannahmen anhand von Experimenten mit einem kraftgeregelten Humanoiden Roboter demonstriert. In einer Reihe von Balancierexperimenten, weist eine effiziente Implementierung unseres Hierarchical Inverse Dynamics Reglers in einer 1 kHz Schleife robuste Stoßabsorption aus. Die vorgestellte Regelungsarchitektur balanciert unseren Roboter erfolgreich in einem breiten Spektrum von Szenarien, unter anderem auf einem Schaukelbett, einer Rollplattform und unter Stoßeinwirkung, wenn der Roboter auf einem Bein steht, während er mit dem anderen eine Tracking Aufgabe ausführt.

## Acknowledgements

---

The research work leading to this thesis has been a fantastic journey with wonderful people whose support has been essential for me throughout these past years. I would like to express my gratitude to all of them, especially my colleagues from the CLMC laboratory at USC in Los Angeles and the Movement Generation and Control Group at the Max Planck Institute for Intelligent Systems, Tübingen.

Many thanks go to my PhD committee for reviewing this thesis and for coming together for my doctoral defense. Further, I would like to thank Jonas, who advised me from the ETH side and helped a lot in shaping this thesis. I would like to thank my colleagues from the lab, who provided me with their feedback and helped improving this thesis.

Ludovic has been an especially remarkable person, who has influenced my science and personal life in so many positive ways. He impressed me with his technical brilliance during every single discussion we had. I was proud of being part of the fruitful research environment he created for his students. Ludovic has not only been a great advisor, but also one of my best friends. I will miss all the white-board sessions, all the experiments with our robots and all the after hours discussions and adventures we have experienced together throughout these past years.

I am very thankful to Stefan for giving me the opportunity to do my research at the CLMC laboratory and Autonomous Motion Department. His research visions always kept me curious and goal oriented. I am thankful for the great opportunity of working with high performance robots and getting the chance to collaborate with exceptional researchers in Tübingen and Los Angeles. I would also like to thank Stefan and Andrea for their advice on private issues, especially in these last months.

I have always enjoyed working with Peter and Mrinal. Their passion for robotics research has motivated me to pursue my PhD in this area and was also a key aspect for my decision to join X Inc. after finishing my PhD. Jeannette has been an amazing advisor, friend and collaborator. I will always remember our first few months at the Autonomous Motion Department when we were two of the first employees buying carpets and furniture and moving robots around. I would like to thank her for managing so many things besides doing great science and still be such a fantastic person to work with. My *deepest* gratitude goes to Daniel for all the coffee breaks, for answering all my coding questions and for all the discussions on neural networks. I would like to thank Manuel for our collaboration on reinforcement learning and for the poster we did together when we were master students and for all the adventures we experienced together. I owe huge gratitude to Franziska, who has always been there for heated discussions on machine learning and control

## *Acknowledgements*

---

topics.

We had a great time working with our humanoid robots. I would like to thank Sean, Nick, John and Maximilien for all the fun we had working together. I am thankful for having Felix help us with our robots and experimental setup; none of our experimental results would have been possible without him. Many thanks also go to my roommates from Belmont: Daniel, Yevgen and Artem. They were not only great colleagues, but also good friends.

I would like to thank Vincent for organizing our software, Bharath for being my roommate in Iceland, Bilal and the rest of his team for bringing the MPI soccer tournament cup to the lab, Alonso for his acting skills at Tübingens theaters and Jim and Nathan for the discussions on Riemannian Geometry. My greatest gratitude also goes to Brahayam for our discussions on the centroidal dynamics, Julian for our collaboration on the hopper robot, Akshara for the frozen yoghurt and discussions on humanoid locomotion, Sebastian on all the advice related to ETH and control theory, Vince for the fun trips to San Francisco and others and Alina and Christina for sharing an office with me during the time I wrote my thesis.

My advisors, colleagues and friends contributed in one way or another to this thesis. However, without the support of my family, I would not have been able to do the work I love so much. I would like to express my deepest gratitude to them.

Ohne meine Familie wäre diese Arbeit nicht möglich gewesen. Sie hat mir schon mein ganzes Leben lang, und besonders in den letzten Jahren, Halt und Kraft gegeben. Ich möchte allen voran meiner Frau Eleonora danken, weil Sie mich trotz langen Abenden im Labor und weiten Distanzen stets unterstützt hat und an mich geglaubt hat. Meiner lieben Mutter danke ich für Ihre aufopferungsvolle Liebe und Erziehung und ebenso meinem Bruder und meiner Schwester, meinen Neffen und meiner Nichte und meiner Schwägerin und Schwager.

August 31, 2017

*Alexander Herzog*

## **Financial Support**

This research was mainly supported by the Max-Planck-Society, the Max Planck ETH Center for Learning Systems and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 637935). It was also supported by by National Science Foundation grants IIS- 1205249, IIS-1017134, EECs-0926052, the Office of Naval Research and the Okawa Foundation.

# Contents

---

<b>Abstract</b>	i
<b>Zusammenfassung</b>	iii
<b>Acknowledgements</b>	v
<b>Preface</b>	1
<b>1. Introduction</b>	3
1.1. Scope . . . . .	4
1.2. Motion planning for floating-base robots . . . . .	6
1.3. Whole-body control . . . . .	11
1.4. Background . . . . .	12
<b>2. Contributions</b>	19
2.1. An optimization-based control architecture . . . . .	21
2.2. Contribution: structural analysis, dedicated solvers and evaluations	23
2.3. Publications . . . . .	24
2.4. List of contributions . . . . .	28
2.5. Complete List of Publications . . . . .	29
<b>3. Optimization-based motion generation for multiped robots in contact scenarios</b>	31
Paper I: “Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid” . . . . .	33
I.1. Introduction . . . . .	34
I.2. Hierarchical Inverse Dynamics . . . . .	36
I.3. Linear and angular momentum regulation . . . . .	41
I.4. Experimental Setup . . . . .	44
I.5. Experiments . . . . .	47
I.6. Discussion . . . . .	61
I.7. Conclusion . . . . .	65
Paper II: “Trajectory generation for multi-contact momentum control” .	67
II.1. Introduction . . . . .	68
II.2. Momentum trajectory optimization . . . . .	69
II.3. Momentum LQR . . . . .	74

II.4. Whole-body control . . . . .	75
II.5. Simulation Results . . . . .	76
II.6. Discussion . . . . .	79
II.7. Conclusion . . . . .	82
Paper III: “Structured contact force optimization for kino-dynamic motion generation” . . . . .	85
III.1. Introduction . . . . .	86
III.2. Kinodynamic Motion Generation . . . . .	87
III.3. Reaction Force Representations . . . . .	89
III.4. Optimal Control Formulations . . . . .	95
III.5. Experiments . . . . .	100
III.6. Conclusion . . . . .	102
<b>4. Conclusion and future directions</b>	<b>105</b>
4.1. Summary . . . . .	105
4.2. Future directions . . . . .	106
<b>Appendices</b>	<b>109</b>
<b>A. TV-LQR for hierarchical task-space policies</b>	<b>111</b>
<b>Bibliography</b>	<b>113</b>
<b>Curriculum Vitae</b>	<b>131</b>

# Preface

---

This thesis discusses an optimization-based control approach for legged robots. Chapter 1 introduces the problem of motion generation for legged robots and embeds the thesis into context of related approaches. At the end of the chapter a brief description of technical background is provided. Chapter 2 formalizes the control architecture and briefly summarizes the contribution of this thesis. In Chapter 3, three self-contained publications are presented that constitute the contribution of this thesis. Finally, Chapter 4 concludes the thesis and provides pointers to future work.



# 1

## Chapter

### Introduction

---

Robots have the potential to match or even surpass mechanical properties of their counterparts from nature, whether humans or other legged animals. The design of algorithms for full body robotic locomotion and manipulation in unstructured environments is an exciting research problem leading to a broad range of societal applications. For example, robots could assist with labor intensive tasks such as helping human operators working in reconfigurable manufacturing pipelines [1–4, 136] or construction sites [50]. Guided by task descriptions of a supervisor, robots could locomote through the environment, operate tools and manipulate objects in their surroundings. Similarly, they could enter areas, too dangerous for humans, for example, rescue or disaster relief scenarios [6, 70, 87, 93]. With demographic shifts in many countries across the world, elderly care could require the assistance of service robots to cope with increasing demanding and decreasing numbers of personnel. In order to advance the state of the art to a point where robots can safely fill these roles in society, we need to understand how task-oriented motion generation in legged robots can be achieved. Robots have so far mainly been used in industrial settings in well-controlled environments for the purpose of executing precisely pre-programmed, repetitive movements. Autonomy is largely achieved with robots that have a fixed base and are inherently stable, meaning they cannot fall over when they move or touch the environment. In contrast, machines that move freely and interact with an uncertain environment, such as an excavator for example, are still operated by a user and lack autonomy. Similarly, a freely moving legged robot must coordinate its movement with forces it exerts on the environment to both perform a given task and to avoid tipping over. This thesis addresses autonomous motion generation for robots with arms and legs performing tasks in constant physical interaction with their environment. Advantages of legged robots over classical industrial manipulators is a substantially extended workspace. A legged robot can use stairs to access different working areas or step on a ladder to

reach for an object. As many potential applications for robots are in environments normally inhabited by humans or other legged animals, designing a robot with arms and legs may suit them to operate in these environments. Disaster relief in a human engineered environment [6, 70, 87, 93] such as a factory with ladders and doors is simpler for a humanoid robot to move through, whereas quadrupeds excel in environments such as pipelines or shafts. Independent of morphology, robots with arms and legs can potentially use each endeffector for both locomotion and manipulation, making them suitable for a wide range of tasks. For example, they can climb a ladder, screw in a light bulb, or operate a foot pedal, making them perhaps the most versatile robots in existence.

## 1.1. Scope

A core difficulty in generating motion for robots with arms and legs is avoiding to fall. It is necessary to balance the overall body at all time and react to disturbances quickly. Therefore, the physics of multipeds poses hard bounds on computation time allocated to control algorithms. They need to be quick enough in deciding where to step, hold, or push against in order to prevent the robot from falling and achieve a task successfully. In general, tasks that involve contact with the environment require algorithms that reason over limb motion and contact forces at the same time, in order to ensure consistency with the dynamics of the robot. For example, a robot walking up a steep ramp is expected to keep its center of mass above the feet due to friction force constraints. Footholds must be chosen to prevent slippage, and contact with hand-rails needs to be exploited to avoid falling over.

Traditional planning approaches are not applicable to dynamic legged robots, as they compute joint motion first and find controls after and they may find joint trajectories inconsistent with contact force constraints. Further, many control methods do not scale due to the high dimensionality of the systems we are interested in. For example, the robot in Figure 1.1 has on the order of 40 joints (dimension of their configuration space) while the state space is at least 80 dimensional. Moreover, while it is relatively easy to sample in configuration space, it becomes very hard to sample in state space, as configurations are also subject to nonlinear dynamic constraints. This makes scalability of standard motion generation algorithms a major concern.

In addition to questions of scaling and dynamic feasibility, motion planning for legged robots must deal with the specific challenges of underactuation and contact switching. Legged robots are inherently underactuated, meaning we do not have direct control over the robot's pose in space. *Robots can fall*. Therefore, contact with the environment is essential not only for task achievement but also for plain balancing. In addition, contacts between the robot and its environment pose several issues for planning, estimation and control. Due to instantaneous contact events, for example at heel strike and toe-off during walking, legged systems exhibit hybrid dynamics, with both continuous and instantaneous components,

which must be addressed during planning and control [153]. Reliable control of legged systems has been limited to specialized tasks due to the high complexity of multiped robots. Walking on flat ground, for example, has been demonstrated to work robustly [10, 66, 94, 117, 126, 145] on quadrupedal and even bipedal robots. However, these approaches cannot be applied to a broader range of tasks where contact with non-coplanar surfaces and several endeffectors becomes necessary.

The goal of this thesis is to design a motion generation algorithm for robots with arms and legs to accomplish tasks under contact interaction with the environment. For example, we envision robots, operated by high level commands, to move in obstructed construction sites, assist in assembling heavy objects, or go on rescue missions in rocky terrain. We will use optimal control techniques to express task descriptions in the form of performance measures and task constraints. We focus on locally optimal solutions and trade-off modeling complexity and control bandwidth by solving several optimization problems at various time-scales. A floating-base robot can be described as a tree of rigid bodies connected through joints starting from a base that is not fixed, but can move freely in space. Examples are wheeled robots [141], snake-like robots [28] and *multipeds*, i.e. robots with arms and legs [27, 76, 117]. They all move their floating-base by pushing against the ground under different types of contact. The focus in this thesis will be rigid and static contacts typically exploited by robots with arms and legs. All of the algorithms developed in this thesis are tested with humanoid robots. However, it is important to note that these algorithms are generic and can be used for robots with very different morphologies and with an arbitrary number of legs and arms.

Motion generation requires processing sensors, deciding on a control policy, and executing that policy. A robot perceives its state (for instance joint and base configuration) and the state of the environment (such as location of stepping stones) through sensor readings processed with estimation algorithms [14, 123]. A control policy then continuously maps estimated states to control actions (for example joint torques) to be executed by the robot. Classical design choices [5] for policies separate a planner from a tracking controller. The controller then rejects disturbances in order to realize the plan on the robot. However, large disturbances that require re-planning become problematic, for example when parts of the floor become inaccessible. This distinguishes mobile robotics from other application area of control, where such large disturbances can be avoided by design. Instead, we present a motion architecture that provides an adjustable plan in the form of an optimal control problem. A pre-computed plan can be re-computed quickly with updated constraints in the environment or changes in the robot state.

As a robot operator it is desirable to express a task in terms of features of the robot state and its environment, such as the location of objects or the robot itself after successful task completion. Ideally, the operator should not be expected to explicitly account for the dynamics or contact properties of the robot. It is then up to motion generation algorithms to reason about task description, system dynamics and contact geometry in order to compute suitable policies automatically. Conceptually, a task description should not have to be re-designed as the robot state or environment constraints change. The general class of model-based optimal

control techniques suites this requirement and is the core of the control architecture presented in this thesis. Our method does not separate planning from control, but treats both in the form of a consistent optimization problem solved at different time-scales.

Optimal control approaches rely on models of the robot and are limited by their quality. For example, perfect knowledge of mass distribution is assumed, and sensors are considered free of noise and delay. Finding solutions to optimal control problems is often computationally expensive, but at the same time hard real-time computational requirements have to be met for execution on real systems. We will demonstrate the validity of our approach and modeling assumptions through experimental validation on a torque controlled humanoid robot.

## Outline

This thesis proposes a motion generation approach formulated as an optimal control problem for multiped robots. In the remainder of this chapter, we will introduce the difficulties of the addressed problem, discuss common approaches and embed our contribution into context.

Section 1.2 discusses the problem of whole-body motion planning for legged robots. Whole-body feedback control is discussed in Section 1.3. Finally, we briefly discuss technical background in Section 1.4 to motivate our control architecture design in the following chapters.

## 1.2. Motion planning for floating-base robots

In the following, we discuss traditional approaches to motion planning and control. We will briefly point out their benefits and open problems that this thesis will address. This section is split into three parts. We discuss traditional path-planning approaches in Section 1.2.1 and extend the discussion to more recent trajectory optimization methods in Section 1.2.2. In Section 1.2.3 we explain planning approaches based on the centroidal dynamics.

### 1.2.1. Path-planning in configuration space

Classical motion planning [88] attempts to find configuration sequences, *paths*, of a robot that lead to a collision free movement. The main focus of such approaches has been to compute paths under complex configuration space constraints. Sampling-based motion planning methods like RRT [89] or PRM [64] have focused on computation of paths in configuration space, where a path consists of simply connected configurations not necessarily parameterized by time. These randomized algorithms have shown to efficiently find feasible paths for complex configuration-space constraints. However, post-processing is required to find smooth trajectories and solutions mostly focus on feasibility of the path, but neglect optimality. RRT\* [79] in addition allows optimization over plans from a distance function expressed

in configuration space. However, the resulting state-space trajectories are not necessarily realizable by legged robots due to contact constraints and underactuation of their floating-base.

Traditional path planning in configuration space [88] is not directly applicable to legged robots, because contact force constraints are not taken into account. The general path planning problem [64, 88, 89] has been extended to the problem of acyclic motion planning [22] to address multi-contact scenarios. They compute quasi-static postures that guarantee existence of contact forces keeping the robot in balance. Usually, contact locations and quasi static configurations are found consecutively. Sampling-based methods are often applied [22, 54, 138] to search for footfalls and quasi-static robot configurations. However, the geometry of the space of stable postures makes sampling difficult. Lower dimensional manifolds intersect in a combinatorial manner [134]. Sampling-based acyclic planners[22, 54, 138] are often better suited to search contact configurations in environments with complex obstacle geometry where local solutions are insufficient.

Some two step methods [20, 40, 143] split potential foothold selection from pose selection. They first search for a sequence of surfaces in the environment, then locate exact contact points using an optimization-based posture refinement algorithm under kinematic constraints. The aforementioned approaches compute a sequence of quasi-static postures and require additional post-processing [17, 111] to connect paths. They have been shown to work on a robot performing contact switching tasks [20, 40], but due to quasi-static assumptions, these behaviors lead to rather slow movements and need additional trajectory stabilization mechanisms.

Sampling methods are favorable for finding feasible solutions when locally optimal solutions are not sufficient. However, the vast majority of tasks for legged robots do not contain overly complex configuration-space constraints, therefore suggesting that local solutions are sufficient in most cases.

### 1.2.2. Whole-body trajectory optimization.

Optimal control techniques have been used for trajectory generation of whole-body plans. In the following, we introduce transcription methods and discuss more recent developments in the field of legged robots.

Optimal control theory is mostly influenced by Pontryagin's [114] and Bellman's [11] principles. The key observation of Bellman was that the optimal action taken at any state is independent of how one arrived at that state. The recursive nature of this statement leads to the Dynamic Programming (DP) methodology [12, 88] which has been very successful for solving discrete problems. The DP algorithm recursively updates an optimal policy for trajectories of fixed length to the new optimal policy for trajectories prolonged by one timestep. Without additional knowledge of the problem structure, DP quickly becomes intractable due to the curse of dimensionality [12], which is especially problematic in robotics applications due to their high dimensional and continuous state spaces. Simplification of the problem can be achieved by either approximating the problem until it be-

## *1. Introduction*

---

comes tractable or falling back to locally optimal solutions. The latter leads to approaches such as differential dynamic programming (DDP) [68] that use second order approximations of dynamics and cost functions to converge to a locally optimal solution. However while, resulting policies can be used in fast control-cycles, extensions to more general state and control constraints require additional approximations [140, 142].

The Pontryagin maximum principle [114] states necessary (but in general not sufficient) conditions on the optimality of a state and control trajectory. This extends naturally to trajectory optimization methods that attempt to find an optimal state-control trajectory instead of a control policy. These transcription methods [13] can be extended to model predictive control methods [33, 47], that solve a trajectory optimization problem at each control cycle. Therefore the trajectory optimization program itself is a control policy. These methods can be naturally extended to include additional constraints over states and controls, such as contact force constraints. However, optimization problems can be computationally demanding requiring more careful design for application in fast control-loops.

Since constraints are essential to express contact properties and task-relevant state-space limits, we are going to use direct transcription techniques to allow natural incorporation of state and control constraints. In order to use the trajectory optimization problem as a policy, we solve simpler optimal control problems on a faster time-scale.

Transcription methods have been proposed to optimize over full state and control trajectories for dynamic locomotion tasks [45, 100, 110, 129]. They define a trajectory optimization problem over the full dynamics and pre-specify when and where endeffectors are in contact with the floor. This approach has been extended to include more complex robot models and contact geometry [92]. A sequence of contact events is pre-computed, assigning endeffectors to planes in the environment. Whole-body joint and torque trajectories are then optimized with a transcription method. The separation of whole-body motion generation into a) contact sequence planning [20, 40, 143] and b) whole-body trajectory optimization is a viable control approach in robot experiments in multi-contact interaction tasks [26, 92, 146]. However, planning of whole-body motion has been computationally demanding and required offline planning. Nevertheless, experimental validation of resulting trajectories [146] motivates a similar separation of the problem. As such, the motion generation algorithm proposed in this thesis optimizes over contact locations on pre-defined [32, 71, 112] surfaces in the environment. In our work we target near real-time whole-body trajectory planning with additional decomposition of the optimization problem to exploit the centroidal dynamics.

Some local whole-body optimization methods [30, 39, 101, 106, 115, 139] do not require pre-planning of contact sequences. Instead, contact events are included explicitly in the form of decision variables into their optimization. However, unlike acyclic planning, they rely on appropriate initial conditions in order to converge to a good local optimum. Approaches including contact events into their optimization [30, 39, 101, 106, 115, 139] mainly differ in the way contact constraints are embedded into their formulation. A key difficulty for gradient-based methods

is the discrete nature of inelastic contacts. In some methods [30, 45, 110, 115] complementarity constraints are used, which mathematically require that non-zero endeffector forces are only possible if the distance to a surface is zero (and vice-versa). Other optimizers [39, 101, 106] relax this physical constraint to smoothen contact dynamics [133] or simplify the mapping between whole-body motion and contact forces [142]. Smoothing of contact dynamics may lead to unrealistic solutions to favor convergence of the optimizer. On the other hand, pre-defining a direct mapping between motion and force [142] may ignore solutions that exploit force redundancy. Further, introducing contact events in the form of decision variables adds nonlinearities into the optimization problem that might require better initial guesses for local optimizers. Results on robots [84, 102, 106] are promising, but still rare.

The trajectory optimization problem has been further simplified in planning techniques for manipulators [75, 118], where the dynamics of the manipulator were reduced to a simple integrator. Without having to consider dynamics constraints, these methods can efficiently plan motion and avoid obstacles assuming that resulting trajectories can be tracked sufficiently well. This approach does not directly apply to legged robots that cannot track arbitrary state trajectories due to contact force constraints. Extensions to humanoids [36, 128] typically assume quasi-static motions, where the robot is expected to move slowly. However, in cases where contact forces can be safely disregarded, state-trajectory optimization becomes an efficient alternative.

In this thesis, we follow the approach of local trajectory optimization with Newton-methods. We solve trajectory optimization problems at varying time-scales in order to use it in a fast feedback loop and at the same time plan over a horizon. Our solver is tailored to the dynamics of legged robots decoupling the trajectory optimization problem over the full equations of motion into a) a local state-trajectory optimization problem and b) an optimal control problem over the center of mass dynamics of the system.

### 1.2.3. Stepping pattern generation with centroidal dynamics

Floating-base robots that locomote and interact with the environment through contact forces have an inherent coupling between kinematic motion and admissible contact forces. Since such robots are not fully actuated, restrictions on contact forces constrain the space of possible motion trajectories. Thus, a common approach in motion generation for legged robots is to solve two problems in sequence. First design admissible force profiles together with center of mass and momentum trajectories, and then, in a second step, compute the whole-body motion, for example using inverse kinematics.

Floating-base robots mainly differ from a manipulator in that they have to apply forces on the environment in order to move their center of mass. Thus, momentum conservation becomes a core consideration in motion generation for legged robots [154]. Any motion of the robot implies a linear acceleration of the CoM together with angular momenta of all links summed up at the CoM (the overall momentum).

As in any mechanical system the change of momentum has to be consistent with the external forces acting on the robot. This relation is described in the *centroidal dynamics* of the robot (see Section 1.4.3). Legged robots, unlike fixed-base manipulators, cannot apply arbitrary forces to their base. Thus, they can only steer their momentum through admissible contact forces with the environment.

Some approaches exploit centroidal dynamics with simplifying assumptions [29, 38, 85, 112, 132, 146, 157]. Restrictions of the centroidal dynamics typically sacrifice either a) full controllability of CoM and momentum b) automatic computation of foot step locations, CoM and momentum c) full constraint satisfaction on contact wrenches or d) generality of environment geometry. These trade-offs may work for some tasks, but may limit performance in others. For example, expressing a tracking cost on the angular momentum may be important for planning motion that induces a specific momentum profile into the system.

Consider, for example, a swing leg motion that requires a torque around the hip. While the effect of limb motion could be ignored for flat-ground walking, it may become crucial for example for swinging up a leg to a stepping stone located high up. More generally speaking, state space restrictions resulting from the full kinematics are not trivially included when the centroidal dynamics is simplified or considered independent of the whole-body.

The full kinematics and centroidal dynamics are required to describe dynamic limb motion and contact in 3d environments. Therefore, the approach in this thesis reasons over the full dynamics of the robot including whole-body motion. We exploit geometrical properties of the centroidal dynamics to design specialized numerical solvers for our problem.

Unfortunately, the relation between contact forces and the change of momentum they incur is not linear (see Equation (1.5)) and thus complicates control and planning techniques. This leads to methods which make simplifying assumptions to render the dynamics linear while still capturing the important aspects of the full model. A successful approach for walking on mostly flat ground has been the simplification of the centroidal dynamics to the linear inverted pendulum model (LIPM) [53, 72, 147]. The simplifying assumptions are a) that the Com height remains constant, b) the angular momentum remains constant and c) all contact points are located on the floor. It became clear over the last decades that these assumptions were sufficient to demonstrate robust walking performance on flat ground [66, 94, 117, 126]. Planning approaches based on simplified models [38, 49, 55, 151, 157] allow generation of dynamic motions and often lead to good performance on the robot.

However, the type of motion that can be generated with simplified models cannot generally handle non-coplanar contacts or non-constant angular momentum. Therefore, approaches for trajectory optimization over the full centroidal dynamics have been presented [26, 30, 45]. Often a hierarchical planning approach is taken, where a momentum plan is computed first and resolved into a whole-body trajectory after. Designing a desired momentum with such approaches is difficult for tasks with motion-induced momentum profiles, such as jumping [151]. A recent method [30] optimizes over the centroidal dynamics and whole-body motion com-

bined. However, off-the-shelf numerical optimizers are used, potentially ignoring structure of the centroidal dynamics. Instead, our method decouples the whole-body motion and force optimization problem into a) a state-trajectory optimization and b) an optimal control problem over centroidal dynamics and solver for motion and force profiles iteratively.

This line of research [26, 30, 60, 112] makes it more and more evident that including optimization over the centroidal dynamics benefits motion generation for multi-peds in complex contact scenarios. Thus, understanding the geometrical structure of centroidal dynamics [8, 25, 154] and its implication to optimal control design [34, 62] is a promising direction for specialized optimizer design. Therefore, we analyze geometrical properties of control formulations resulting from centroidal dynamics in Paper III leading to specialized solver design.

### 1.3. Whole-body control

Planning algorithms often operate on simplified dynamics models with relatively large time discretization. Control architectures typically obtain trajectories from planners and enhance them with feedback policies to be realized in a fast control loop on the whole-body. Position control has been widely used to track CoM plans generated from pattern generators as discussed in Section 1.2.3. This leads to stiff behavior on the robot which is effective for well estimated floor geometry, but lacks robustness in case of premature or delayed contact with the ground. Recently more and more torque controlled robots have become available [27, 37, 46, 66, 130] with force control capabilities. They gave rise to control architectures for compliant behavior and more robustness when the robot interacts with an uncertain environment [74]. With the seminal work of Hogan [63], impedance control found broad application in manipulation where the closed-loop behavior of the robot is designed to appear like an impedance to the environment. An impedance defines how a robot reacts to disturbances from the environment, but it does not actively control interaction forces. Force control [156] on the other hand directly controls forces actively applied by a robot. Force and position tasks can be combined together by strict separation of the two using the full model of the system [83]. More generally, tasks (position, force, etc.) can be composed from strictly prioritized sub-tasks often applied in whole-body control architectures [65, 120, 131]. It allows realization of closed-loop behavior of high priority tasks and redundancy resolution with lower prioritized tasks. Traditionally, bounds such as actuator limits or friction force constraints were not handled inside such hierarchies, because they require solving optimization programs that can handle inequality constraints, which lead to a more complex class of convex optimization programs. Quadratic programming approaches [122, 137] solve importance weighted equality and inequality tasks and constraints including actuation and contact force constraints. Lexicographical programming [41, 59, 125] opened the way to more general task composition in the form of strictly prioritized equality and inequality constraints. Validation of these methods in real-time feedback loops on torque controlled robots, however, was missing.

In this thesis, we demonstrate that hierarchical inverse dynamics indeed can be implemented inside of a fast 1 kHz control loop on the robot. We design a QP cascade variant to solve lexicographical optimization programs dedicated for fast control cycles and hierarchical inverse dynamics problems. We will demonstrate robust balancing on the real robot despite model inaccuracies and sensor noise (see Paper I).

## 1.4. Background

With the previous sections we positioned the approach in this thesis into the context of related work. We will make use of model-based optimal control approaches throughout the description of our control architecture. In the following section, we would like to introduce optimization techniques and modeling assumptions relevant for the rest of the thesis.

### 1.4.1. Numerical Optimization

With the continuous increase in computational efficiency of modern computers and availability of optimization toolboxes [51, 98, 148], numerical optimization has found widespread application in the robotics community. Optimization is a powerful tool interfacing problem formulation and numeric algorithms that find solutions to those problems. Many control problems can be formulated in the form of a) an objective function describing the control performance as a function of states and controls and b) constraints describing the set of feasible solutions. Optimization algorithms find a solution that maximizes the control performance over a constrained set. Optimization problems are often classified according to their difficulty. One of the most important divisions is the one between convex [21] and non-convex problems [107]. Convex problems have a relatively easily accessible global optimum, whereas non-convex problems need to fall back to locally optimal solutions, because in most cases the global solution cannot be found in polynomial time. Among both classes we can again find more fine-grained classifications of optimization programs.

The algorithms presented in this thesis are mainly in the form of quadratic programs (QPs) and quadratically constrained quadratic programs (QCQPs). QPs and QPQPs both consist of quadratic objective functions. QCQPs can have equality and inequality constraints described by quadratic functions. QPs are more restrictive as they allow only linear constraints and thus are a special case of QCQPs. In general, both classes belong to the set of non-convex functions. However, if in addition we assume that all quadratic functions are positive semi definite and equality constraints remain linear, we restrict the two problem classes to convex QPs and convex QCQPs.

For example, optimal control problems for linear system dynamics often result in convex QPs [34, 55]. They can minimize a tracking cost over state and control variables restricted by linear dynamics constraints and potentially control limits

in the form of inequality constraints. If we consider the more general class of dynamical systems that are either linear in states or linear in controls, but not linear in both, we need to express dynamics constraints in the form of quadratic equalities, resulting in non-convex QCQPs in general. This is for example the case for the centroidal dynamics explained in Section 1.4.3 and utilized in the control architecture proposed in this thesis.

### 1.4.2. Complexity of numerical optimal control problems

In Section 1.2.2, we briefly summarized common approaches to the general optimal control problem initiated by Bellman [11] and Pontryagin [114]. Here, we narrow the discussion to direct transcription methods and outline common trade-offs in optimizer design motivating our motion generation architecture presented in Chapter 2.

Numeric optimal-control methods optimize a performance cost defined over state and control trajectories under constraints in the form of the system dynamics equations. Performance costs may for example penalize tracking cost of some desired state trajectory, like the endeffector motion of a robot, value distance to an obstacle or favor small control actions to reduce energy injected into the system. Additional constraints may express restrictions on the state or control space, for example actuator saturation or kinematic limits. Although, state and control trajectories may be infinite dimensional and the time horizon unbounded in general, for the scope of this thesis we restrict ourselves to finite horizon problems with finite dimensional state and control trajectory representations. Trajectories can be represented in the form of weighted basis functions (for instance polynomials) or discretized in time. Design of cost functions, specifically in robotics, can become a difficult problem by itself. Describing the performance quality of a general task directly in joint space of a robot is tedious and lacks intuition. Instead, a common approach is to break down the overall performance into a composition of performance sub-costs expressed over lower dimensional *features* such as cartesian motions of single robot links or the center of mass or similarly cartesian forces acting between robot and environment. A common approach to compose an overall cost, is to rank sub-costs against each other according to their importance either with a strict prioritization or quantitative importance weighting. For example, the performance of a walking motion may be specified from cartesian swing foot trajectories that dominate the overall behavior and a nominal posture to be achieved at lower priority. Quantitative importance weighting computes an overall cost from a weighted sum of sub-costs. Strict prioritization is achieved when a higher priority sub-cost is optimized independent of how sub-optimal lower priority costs are. Then, lower priority terms can be optimized with the additional constraint that optimality of higher priority terms must not be violated. Formulations of this kind result in lexicographical quadratic programming [31, 41, 58].

Strictly prioritized tasks are more expressive, because they contain importance weighted tasks, but have to be solved with an optimization program of higher complexity. The control architecture proposed in this thesis and outlined in Sec-

tion 2.1 optimizes a hierarchy of prioritized tasks in a 1 kHz loop.

An optimal control problem typically takes the current state of a plant as input and returns the (approximately) optimal control command. Such a mapping from state to control is a control policy. Depending on the mathematical form of objective and dynamics equations, a control policy can be pre-computed independent of a state-input. In the more general case, we need to solve an optimization program at each control cycle leading to model predictive control methods [47]. For example, if the optimal control problem is given by linear dynamics and a quadratic cost, the global optimal policy can be expressed as a linear feedback gain on the states which is independent of the current state of the system. On the other hand, if we use dynamics that are linear in controls or states but not both, for example the centroidal dynamics (see Section 1.4.3), evaluating a control policy requires solving a non-convex QCQP.

Oftentimes, using a simplified or reduced form of the dynamics breaks down the complexity class of resulting optimization programs leading to faster control cycles and potentially fewer local optima. For example, the state space of the centroidal dynamics is often restricted to the Linearized Inverted Pendulum Model (LIPM) leading to a dynamics in the form of a linear system. Thus, instead of a non-convex QCQP, the optimal control problem reduces to a convex QP. On the one hand, a global solution can be found efficiently with a convex QP, but on the other hand the space of state trajectories and controls is limited.

Another way of reducing the problem complexity is to chose shorter control horizons. Where reducing the control horizon can improve computational speed of a solver, trivializing it to 1 reduces the complexity class of the optimal control problem in many cases. This way, control-affine dynamics (for example a floating-base robot) become linear with respect to the decision variables of an optimal control problem. Therefore, accurate plant modeling can be used with low computational complexity at the sacrifice of a preview horizon.

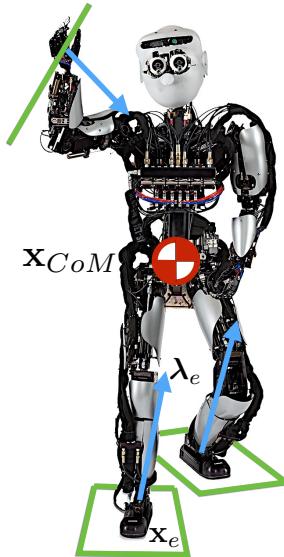
The trade-off between 1) accuracy of dynamics models, 2) length of preview horizon and 3) computational complexity motivates a control architecture, where differently complex plant dynamics are used on different time scales. Therefore, the proposed control architecture in this thesis solves optimal control problems on different time-scales with varying model complexity and horizon length (see Figure 2.1).

### **1.4.3. Dynamics of floating-base robots.**

The control approach presented in this thesis makes use of the robot dynamics at several levels of the architecture. In the following, we discuss the equations of motion of humanoid robots and highlight how interaction with the environment through contact enters into the model.

The rigid-body dynamics model of a legged robot is usually written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}^T \boldsymbol{\lambda} \quad (1.1)$$



**Figure 1.1.:** An illustration of the torque controlled humanoid robot used for the experiments presented in this thesis. We visualize a hypothetical contact scenario where the robot is standing on two **surfaces** and holds on to a **bar**. The humanoid exerts **contact forces**  $\lambda$  on the environment at the contact locations  $\mathbf{x}_e$ . The **center of mass**  $\mathbf{x}_{CoM}$  is located above the hip when the robot is standing upright.

where  $\mathbf{q} \in \mathbb{SE}(3) \times \mathbb{R}^n$  is a vector of position and orientation of the robot in space and its joint configuration,  $\mathbf{M}(\mathbf{q})$  is the inertia matrix,  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$  is the vector of generalized forces including Coriolis, centrifugal and gravitational forces,  $\boldsymbol{\tau}$  is the vector of actuation torques,  $\boldsymbol{\lambda}$  the contact forces,  $\mathbf{S}^T = [\mathbf{0}_{n \times 6} \ \mathbf{I}_{n \times n}]^T$  is the actuation matrix reflecting the unactuated pose of the robot in space and  $\mathbf{J}$  is the contact Jacobian. General optimal control formulations exist [97, 115, 129] that operate directly on the dynamics in Equation (1.1). However, making the motion generation problem for legged robots tractable for real-time applications is challenging. Algorithms exploiting the explicit structure of multiped robots are necessary. The goal of this thesis is to develop numerical methods tailored to robots that can be explained by the dynamics equation above. We can separate the robot dynamics into actuated and unactuated parts

$$\mathbf{M}_u \ddot{\mathbf{q}} + \mathbf{h}_u = \mathbf{J}_u^T \boldsymbol{\lambda} \quad (1.2)$$

$$\mathbf{M}_a \ddot{\mathbf{q}} + \mathbf{h}_a = \boldsymbol{\tau} + \mathbf{J}_a^T \boldsymbol{\lambda} \quad (1.3)$$

where  $\mathbf{M}_u, \mathbf{h}_u, \mathbf{J}_u$  are the 6 rows of Equation (1.1) describing the unactuated part and  $\mathbf{M}_a, \mathbf{h}_a, \mathbf{J}_a$  describe the actuated part. We note that Equation (1.3) can be trivially resolved for joint torques  $\boldsymbol{\tau}$  implying that once  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$  are specified, there is a unique torque vector  $\boldsymbol{\tau}$  that is consistent with the robot dynamics. On the other hand, Equation (1.2) is independent of joint torques and thus is not affected by internal forces directly. These unidirectional relations between change of motion  $\ddot{\mathbf{q}}$  and contact forces  $\boldsymbol{\lambda}$  and internal forces  $\boldsymbol{\tau}$  motivate a hierarchical approach for motion generation. Algorithms presented in this thesis, first compute motion and contact force variables  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$  that satisfy Equation (1.2) and then in a second step substitute  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$  into Equation (1.3) in order to obtain internal forces  $\boldsymbol{\tau}$ .

On the other hand, separating computation of motion trajectories  $\mathbf{q}$  and contact forces  $\boldsymbol{\lambda}$  is not straight forward. They have to be consistent with Equation (1.2), but also result in admissible contact forces. For example, if the robot is pushing against a heavy block, the contact forces on the ground need to be computed in order to prevent the feet from slipping. Thus, the contact forces need to remain inside of friction cones

$$|f_x^2 + f_y^2| < \mu f_z \quad (1.4)$$

where  $f_z$  is the linear component of  $\boldsymbol{\lambda}$  perpendicular to a contact surface,  $f_x, f_y$  are the tangential components and  $\mu$  is the friction coefficient of the surface material. The coupling between motion and force in Equation (1.2) in combination with inequality constraints over contact forces in Equation (1.4) implies complex constraints on the domain of  $\mathbf{q}, \dot{\mathbf{q}}$ . Planning states  $\mathbf{q}, \dot{\mathbf{q}}$  directly in state-space [36, 75, 118, 128], therefore, becomes limiting. The hybrid nature of contact-interaction tasks poses an additional problem. Forces  $\boldsymbol{\lambda}$  can only be applied to the environment when contact is created between the robot and a surface in the environment. For example, when a robot pushes against a stepping stone, contact forces are subject to friction constraints in Equation (1.4). At the moment contact between foot and ground is released the corresponding ground reaction forces in Equation (1.2) must be  $\mathbf{0}$ . Thus, contact sequencing pose an problem for planning and control of legged robots.

In our approach, we treat the problem of contact scheduling decoupled from the rest of the optimal control problem in order to reduce complexity of the planning problem.

The unactuated part of the floating-base dynamics in Equation (1.2) corresponds to the Newton-Euler Equations of the system. It can equivalently be written in a form [53, 73, 154] that makes the coupling between motion of the robot and contact forces explicit

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{H}}(\mathbf{q})\dot{\mathbf{q}} = \left[ \begin{array}{c} M\mathbf{g} + \sum_e \mathbf{f}_e \\ \sum_e (\kappa_e + (\mathbf{x}_e(\mathbf{q}) - \mathbf{x}_{CoM}(\mathbf{q})) \times \mathbf{f}_e) \end{array} \right] \quad (1.5)$$

$$\mathbf{M}_a(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}_a(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}_a(\mathbf{q})^T \boldsymbol{\lambda} \quad (1.6)$$

where  $\mathbf{H} \in \mathbb{R}^{6 \times (n+6)}$  maps joint velocities  $\dot{\mathbf{q}}$  into the 6-dimensional overall momentum of the robot [108, 152],  $M$  is the robot mass,  $\mathbf{g}$  the gravity vector,  $\mathbf{x}_{CoM}(\mathbf{q})$  the center of mass of the robot,  $\mathbf{f}_e, \boldsymbol{\kappa}_e$  contact forces and moments acting at points  $\mathbf{x}_e(\mathbf{q})$  on the robot (see Figure 1.1). The 6 rows on the left-hand side of Equation (1.5) express the change of momentum purely from motion variables  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$  independent from contact forces. The right-hand side of Equation (1.5) expresses the change of momentum as a function of generalized contact forces  $\mathbf{f}_e, \boldsymbol{\kappa}_e$ , contact locations  $\mathbf{x}_e(\mathbf{q})$  and the center of mass of the robot  $\mathbf{x}_{CoM}(\mathbf{q})$ . The *centroidal dynamics* in Equation (1.5) state that the motion of a robot is constrained by the momentum created by contact forces and vice-versa. This relation is exploited in the control framework discussed in this thesis to design algorithms specifically for multiped robots and contact-interaction tasks.

The centroidal dynamics is a reduced dynamical system inside of the full robot dynamics (see Equation (1.1)). The centroidal state  $\mathbf{h}$  consisting of CoM  $\mathbf{x}_{CoM}$  and momentum  $\mathbf{H}\dot{\mathbf{q}}$  is steered by contact forces  $\mathbf{f}_e, \boldsymbol{\kappa}_e$  and contact locations  $\mathbf{p}_e := \mathbf{x}_e(\mathbf{q})$ . It is a better structured dynamical system compared to the full robot dynamics. Still, the cross-product between states and controls in Equation (1.2) introduces a nonlinearity. Nevertheless, we will show in Paper III that despite the complex nature of the centroidal dynamics, it has structure that can be exploited to benefit optimal control algorithms.

A common approach to motion generation for floating-base robots has been to simplify the centroidal dynamics. Resulting models are often easier to handle and can be used to plan CoM, momentum and contact force plans for a more restricted set of tasks. The Linearized Inverted Pendulum Model (LIPM) [53, 72, 147] has been probably the most famous simplification of the centroidal dynamics. Restricting the CoM height and momentum to remain constant and have the contact locations  $\mathbf{p}_e$  be limited to the ground, the cross product in Equation (1.5) is simplified to a linear relation between states and controls.

This chapter introduced the topic and scope of this thesis. We motivated the importance of legged robots and discussed difficulties in autonomous motion generation for such systems. Our research was embedded in the context of related work and methods were introduced that are used throughout the remainder of the thesis. We revisited the equations of motion of legged robots and explained the role of centroidal dynamics. In the following chapter, we will discuss the overall control architecture and highlight the contribution of this thesis.



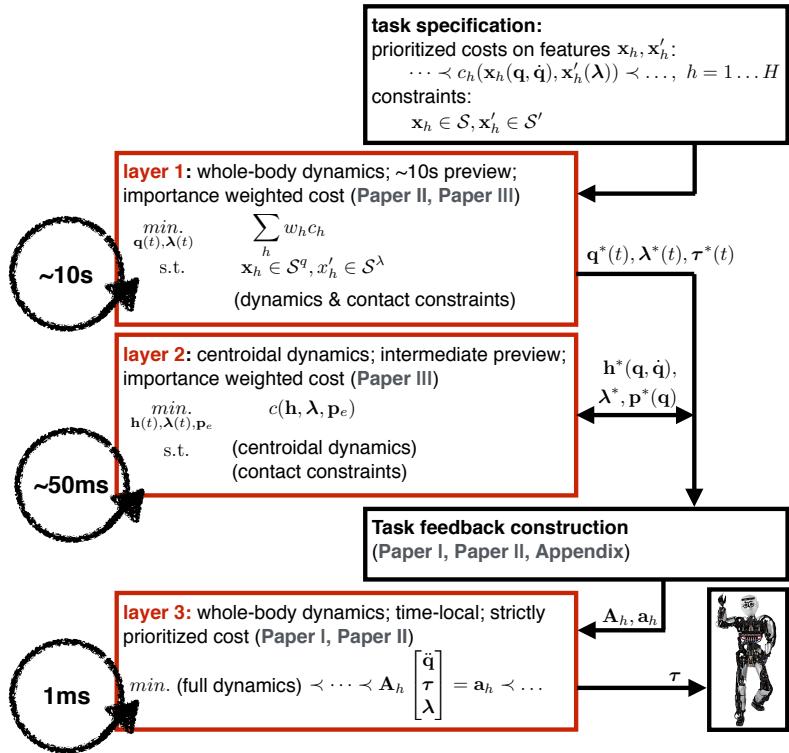
# 2

## Chapter

## Contributions

---

This chapter states the contribution of the thesis. In Section 2.1 we formulate our control approach in order to give technical background to strengthen the claims of the contribution. We embed our contribution into the context of the overall formulation in Section 2.2. In Section 2.3 each publication is introduced with a summary and specific contribution and put into context with the overall thesis. Finally, we list the specific contributions in Section 2.4.



**Figure 2.1.:** A visualization of our motion generation architecture. We solve optimization algorithms from problem formulations expressed inside the red boxes. The problem in *layer 1* is solved with the longest preview and complete whole-body dynamics. At the next fastest time-scale in *layer 2* an optimal control problem over a reduced dynamics is solved. Then at the fastest time-scale in *layer 3* we solve a hierarchical optimization program. Algorithm design, structural analysis and experimental evaluation of the architecture is presented in Paper I, Paper II, Paper III.

## 2.1. An optimization-based control architecture

In this section, we describe our optimization-based motion generation and control framework implemented and analyzed throughout Paper I, Paper II and Paper III.

Ideally, a motion generation framework accepts an intuitive task description and resolves it into actuator commands to be executed on a robot. Optimal control is well-suited to express such problems, but without dedicated solvers, computational complexity of the problem quickly becomes intractable. We describe a motion planning and control framework that is based on optimal control formulations. Structural properties of floating-base robots as discussed in Section 1.4.3 are considered to allow for dedicated algorithm design. A visualization of our framework is provided in Figure 2.1. This thesis made contributions in the analysis, implementation and evaluation of the proposed motion generation architecture as stated in Section 2.2.

### Optimal control formulation

This thesis envisions a motion generation framework consisting of three optimization-based controllers solved at different time-scales. We formulate an overall optimal control problem solved through layer 1 to layer 3 with different levels of modeling granularity. The overall problem is expressed as a hierarchy of  $H$  optimization programs

$$\min_{\mathbf{q}, \boldsymbol{\tau}, \boldsymbol{\lambda}} \quad \int_0^T c_h(\mathbf{x}_h^q(\mathbf{q}, \dot{\mathbf{q}}), \mathbf{x}_h^\lambda(\boldsymbol{\lambda}), \mathbf{x}_h^\tau(\boldsymbol{\tau})) \quad (2.1)$$

$$\text{Priority}_{h-1} \prec \text{ s.t. } \mathbf{M}\dot{\mathbf{q}} + \mathbf{h} = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}^T \boldsymbol{\lambda} \prec \text{ Priority}_{h+1} \quad (2.2)$$

$$\begin{aligned} \mathbf{x}_h^q &\in \mathcal{S}_h^q, \mathbf{x}_h^\lambda \in \mathcal{S}_h^\lambda, \mathbf{x}_h^\tau \in \mathcal{S}_h^\tau \\ h &= 1, \dots, H \end{aligned} \quad (2.3)$$

where  $(\cdot)_h$  indexes priority level  $h$ ,  $c_h$  is a cost of state and control *features*  $\mathbf{x}_h^q, \mathbf{x}_h^\lambda, \mathbf{x}_h^\tau$  over a time-horizon  $T$ . We optimize over robot configurations  $\mathbf{q}$ , velocities  $\dot{\mathbf{q}}$ , joint torques  $\boldsymbol{\tau}$  and contact forces  $\boldsymbol{\lambda}$  over the constraint sets  $\mathcal{S}_h^q, \mathcal{S}_h^\lambda, \mathcal{S}_h^\tau$ . Independent of the priority level, we constrain the optimization to satisfy the equations of motion of the robot in Equation (2.2). We consider a hierarchy of costs and constraints, where  $(\cdot)_h \prec (\cdot)_{h+1}$  defines the ordering.

The overall optimal control problem in Equation (2.1) cannot be solved with conventional numerical methods for integration in a real-time control architecture. Instead, we propose to solve the problem at different time-scales, trading off complexity of the formulation against bounds on computation. At the slowest time scale (layer 1 in Figure 2.1) we compute a state and control trajectory over a relatively-long time horizon using the full robot dynamics. At layer 2 a faster optimal control problem is solved using a dynamical system of reduced complexity

## 2. Contributions

---

(see Equation (1.5)). At the fastest time-scale in layer 3 we shorten the time horizon to a single timestep and solve the full hierarchical problem. In the following we describe our formulation and proposed optimization architecture in more detail.

### Cost and constraints

The inputs to our motion generator are task performance descriptions and constraints. Quality of task achievement is measured in the form of  $H$  prioritized sub-costs  $c_h$  measuring task performance as a function of state and force *features*  $\mathbf{x}_h^q(q), \mathbf{x}_h^\lambda(\lambda), \mathbf{x}_h^\tau(\tau)$ . The terms  $\mathbf{x}_h^q(q), \mathbf{x}_h^\lambda(\lambda), \mathbf{x}_h^\tau(\tau)$  are often called *tasks*, but we will refer to them as *features* here to disambiguate the terminology. For example,  $\mathbf{x}_h^q$  could be the endeffector position and  $c_h$  could penalize the proximity to an obstacle. Task constraints  $\mathcal{S}_h^q$  could for example describe regions of the floor accessible by the endeffector, or actuation limits can be expressed in  $\mathcal{S}_h^\tau$ .

Our approach allows cost terms  $c_h$  to be prioritized against each other. Prioritization may be expressed strictly as in Equation (2.1), meaning  $c_h$  cannot improve by lowering performance of  $c_{h+1}$ ; for example, a higher priority cost may value obstacle avoidance and a lower priority term may encourage a favorable posture. However, importance of cost-terms can also be expressed in weighted form. In order to keep the overall motion generation problem tractable, we employ strict prioritization only at the fastest time-scale.

The choice of the time-horizon,  $T$ , impacts computational complexity of the overall problem. We use longer horizons in layer 1 and reduce the preview in layer 2 to favor computational speed. In layer 3 we optimize over a single timestep,  $T = 1$ , in favor of a strictly hierarchical task over the full dynamics.

Contacts with the environment are expected to be rigid and static and are expressed in the form of constraints  $\mathcal{S}_h^q, \mathcal{S}_h^\lambda$ . With this assumption, our approach is general across a broad range of tasks that involve multi-contact planning in three dimensions, and at the same time contains rich structure to be exploited by numerical optimization techniques.

### Time-scales

At the first level of our control architecture (see Figure 2.1) we solve an optimal control problem over the full robot dynamics. In order to keep the problem tractable, we approximate strictly prioritized cost terms  $c_h, c_{h+1}, \dots$  with importance weighting. In our contributions, we experiment with time horizons in the order of 10 seconds, however the horizon vary depending on the task and robot. Further, we assume that the robot’s actuators can reliably generate sufficient joint torques so that we can neglect  $\mathcal{S}_h^\tau$  and torque features  $\mathbf{x}_h^\tau$ . Thus, at this optimization layer we ignore actuation limits and re-introduce them at layer 3 to guarantee admissible torque commands.

In the third and fastest control layer, we solve an optimal control problem over

a single timestep. The full robot dynamics is used and task descriptions and constraints are resolved in a strict hierarchy. Resulting optimal joint torques are assumed to be feasibly tracked by the robot's low-level joint torque controllers. The fastest and slowest time-scales (layer 3 and layer 1), respectively trade-off fast feedback with strict prioritization with weighted prioritization over a preview window of intermediate length. However, it is seen to be beneficial to add an intermediate layer that can react quicker to unpredicted disturbances using an intermediate preview  $T$ . We suggest solving an optimal control problem at intermediate frequency with the centroidal dynamics (see Equation (1.5)) and weighted prioritization of the cost. Although at this level whole-body trajectories cannot be re-planned, we can still update the control policy for momentum and contact forces as joint-level control is recomputed at the more fine-grained model in layer 1.

### Task feedback

The input into layer 1 is a task description in the form of a prioritized cost  $c_h$  and constraints  $\mathcal{S}_h^q, \mathcal{S}_h^\tau, \mathcal{S}_h^\lambda$  (see Equation (2.1)). The resulting optimal joint trajectory  $\mathbf{q}(t)$  and contact force profiles  $\boldsymbol{\lambda}(t)$  are used to compute the input for the remaining two layers. The intermediate level receives contact force profiles  $\boldsymbol{\lambda}$ , contact locations  $\mathbf{x}_e(q)$  and momentum trajectories  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) := \mathbf{H}\dot{\mathbf{q}}$  that can be directly computed from Equation (1.5) and forward kinematics. The fastest level takes as input task feedback policies on features  $\mathbf{x}_h^q, \mathbf{x}_h^\lambda, \mathbf{x}_h^\tau$ . A naive approach is to compute optimal feature trajectories from the solution of the first layer and hand-design PD gains for tracking. However, here we can exploit optimization techniques to compute feedback policies from the overall optimal control formulation. in Appendix A, we propose using time-varying LQR to construct feedback policies to be used by the fastest time-scale of our motion generator.

## 2.2. Contribution: structural analysis, dedicated solvers and evaluations

This section states the contributions of this thesis towards a robust implementation of the motion generation architecture visualized in Figure 2.1 on legged robots executing contact-interaction tasks.

We formulate and implement a whole-body motion planner and a hierarchical inverse dynamics controller to operate in layer 1 and layer 3 (*Contr. A1*). The whole-body motion planning problem constitutes an iterative algorithm composed of two sub-problems (*Contr. A2*) for which specialized solvers are proposed (*Contr. A7*).

At the core of our control architecture we employ optimal control methods using the centroidal dynamics (*Contr. A2*, *Contr. A5*, *Contr. A6*). We analyze geometrical properties of the centroidal dynamics (*Contr. A3*) in order to develop specialized, computationally-efficient solvers (*Contr. A7*). Further, our analysis

## 2. Contributions

---

leads to a sparse formulation for optimal control problems that can be exploited in layer 2 (*Contr. A4*). In our simulation experiments, we can compute ten second long trajectories in the order of seconds and track them with our hierarchical inverse dynamics controller in a simulated rough terrain stepping task. In order to solve the fastest time-scale control problem (in layer 3) inside of a 1 kHz control loop, we propose a QP cascade variant with partially-analytic solutions (*Contr. A6*). The reduction of computational complexity allows for an implementation of a hierarchical inverse dynamics controller in real-time on our humanoid robot (*Contr. B2*) in a number of varied balancing tasks (*Contr. B5*). We use operational space control (*Contr. B3*) to balance the robot on one and two feet using hierarchical feedback policies (*Contr. A5, Contr. B1*). This robust behavior is achieved using a momentum regulator, which to the best of our knowledge, has not been demonstrated successfully on a torque controlled humanoid before (*Contr. B4*).

Additionally, our theoretical contributions and algorithms found application outside of this thesis (*Contr. B2*). We have recently developed a reactive stepping controller for a humanoid with passive ankles [80] and controlled resulting foot trajectories with our hierarchical inverse dynamics controller. In another work, we exploit the geometrical properties of the centroidal dynamics in order to design a convex approximation of these dynamics [112] leading to efficient whole-body motion generation.

In summary, the class of problems addressed by the motion generation architecture proposed in Section 2.1 contains mathematical structure that was exploited in the contributions of this thesis, enabling robust whole-body control behaviors on our hardware.

## 2.3. Publications

In the following, we summarize the contributions of the papers presented in this cumulative thesis and put them in context of one another.

### Paper I

Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, Ludovic Righetti, “Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid”. In *Autonomous Robots*, 2016.

#### Background

Inverse dynamics is interesting for whole-body control because it allows for the composition of complex behaviors from a set of lower-dimensional tasks. Using the full dynamics model allows for composition of both position as well as force tasks in consistency with the full dynamics. Conflicting tasks can be weighed

against each other to express, for example, the importance of CoM tracking over swing leg tracking in the case of a walking motion. On the other hand, strict priority levels become important if a quantitative importance weighting is not meaningful. For example, actuator limits, friction constraints and motion tracking could be prioritized at three separate levels. The expressive power of hierarchical inverse dynamics, however requires fast optimizers that exploit problem structure. Further, only evaluation on real hardware can answer the question if computational speed, modeling accuracies and sensor quality is sufficiently addressed in realistic settings.

### Contribution

We demonstrate that hierarchical inverse dynamics can be implemented on a torque controlled humanoid robot. Our robot showed robust behavior in a set of balancing tasks despite model inaccuracies and sensor noise (*Contr. B5*). To the best of our knowledge this was the first demonstration of a) hierarchical inverse dynamics with inequality constraints running in a feedback loop on a torque controlled humanoid (*Contr. B3*) and b) experiments with feedback control on the overall (linear and angular) momentum of the system (*Contr. B4*). An LQR-based feedback policy on the system momentum is evaluated against a common PD design and showed to work more robustly than the naive approach (*Contr. B1*). Our hierarchical tasks are solved inside of a 1kHz control loop at every control cycle (*Contr. B2*). To achieve this high control bandwidth, we introduce a QP cascade variant (*Contr. A6*) that exploits structure of hierarchical inverse dynamics tasks to reduce computational complexity. In our experiments, we use pure model-based control without joint stabilization.

### Context

The input to the whole-body controller presented in this paper can be seen as a hierarchy of desired task feedback policies and constraints of lower dimensional quantities expressed from joint states and contact forces. Our controller takes into account the full equations of motion to accomplish these tasks in consistency to the best of its capabilities. As such, our hierarchical inverse dynamics controller can be used to track motion plans and realize feedback policies generated from slower time-scale planners (as explained in Section 2.1), specifically those presented in Paper II and Paper III. Our controller could realize closed-loop dynamics of the overall momentum as generated by the planner presented in Paper II. This allowed our humanoid to step over rough terrain in simulation. Further, we were able to use a Linearized Inverted Pendulum Model (LIPM) based stepping controller [80] for a humanoid with passive ankles, embedded in our whole-body controller. We split CoM height control and swing leg motion tracking into separate priority levels to guarantee LIPM assumptions to hold when planning for foot locations.

## Paper II

Alexander Herzog, Nicholas Rotella, Stefan Schaal, Ludovic Righetti, “Trajectory generation for multi-contact momentum control”. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2015.

### Background

Floating-base robots must apply forces to the environment in order to move their center of mass or change the overall angular momentum of their body. Contact forces are required to be admissible and in dynamic consistency with the change of momentum induced by the motion of the body. The coupling of motion and force is a major obstacle in motion generation for humanoid robots. Methods that generate whole-body motion on legged robots typically design contact force profiles at a first stage using simplifications of the centroidal dynamics. Then, whole body motion is planned around resulting center of mass trajectories for example through inverse kinematics. Simplifications of the centroidal dynamics are often limiting (see for instance Section 1.2.3) and whole-body motion becomes sub-optimal because it is constrained by pre-planned CoM or momentum profiles. Momentum-control has been demonstrated to work well for balancing on humanoid robots [61], but feedback design especially for planned momentum trajectories is rarely addressed.

### Contribution

This paper presents an optimization-based control architecture that plans trajectories with the full centroidal dynamics (*Contr. A1*) and tracks resulting force and momentum profiles with our hierarchical inverse dynamics controller presented in Paper I. We generate an optimal closed-loop behavior using time-varying LQR (*Contr. A5*) and split it into two priority levels: a) optimal closed-loop dynamics resulting from our LQR policy and b) regularization on forces. This allows us to exploit redundancy resolution capabilities of our hierarchical whole-body controller to track momentum with an optimal closed-loop behavior and at the same time keep degrees of freedom open for other high priority tasks. Simulation experiments are presented on a stepping task over difficult terrain.

### Context

The planning and control architecture presented in this paper builds up on the hierarchical inverse dynamics controller from Paper I. Our momentum tracking policy is designed to make use of the hierarchical task resolution of our whole-body controller. It has been evaluated in the context of momentum regulation in Paper I and is motivated from the robust behavior on the real robot. The whole-body trajectories computed in this paper turned out to converge after a few iterations, therefore motivating the generalization to a consistent optimal control and a dedicated solver in Paper III. The structure of centroidal dynamics is further analyzed in Paper III leading to better informed optimal control solvers. This

paper demonstrates the combined behavior of layer 1 and layer 3 of the proposed motion generation architecture (see Section 2.1) in a realistic rough terrain stepping task.

## Paper III

Alexander Herzog, Stefan Schaal, Ludovic Righetti, “Structured contact force optimization for kino-dynamic motion generation”. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

### Background

Recently, whole-body motion generation has been addressed with optimal control techniques. Approaches have been presented that simplify the contact dynamics and decouple contact force computation from whole-body trajectory generation potentially limiting their application. Others attempt to solve the full whole-body planning problem with off-the-shelf solvers potentially missing out on problem structure that could lead to more efficient solvers. As we are interested in realizing whole-body motion generation in fast real-time loops, it becomes necessary to analyze structure of problem specific optimal control formulations. The centroidal dynamics is being used for trajectory planning algorithms in recent publications. Understanding the structure of the problem and its implication to optimization algorithms can lead to more efficient optimization algorithms and efficient approximations.

### Contribution

We formulate the whole-body motion planning problem as an optimal control problem with a cost over joint and contact force trajectories over a finite horizon. Understanding that contact forces and the whole-body motion are only coupled through the momentum generated by either of the two, we suggest a decomposition of the optimal control problem. We solve the overall problem by computing iteratively solutions to a) a trajectory optimization problem over the whole body and b) an optimal control problem over the centroidal dynamics (*Contr. A2*). This leads to a decomposition into two better structured optimization programs compared to the original problem and suggest a computationally efficient alternative (*Contr. A7*) to off-the-shelf solvers. An advantage of this decoupling is that existing trajectory optimization algorithms [75, 118] can be applied to the trajectory optimization problem. Further, we analyze the centroidal dynamics and reveal structure beneficial for dedicated optimal control solvers. We show that the non-convex dynamics can be constructed as a difference of convex functions (*Contr. A3*) leading us to better informed optimization algorithms [62, 112]. In addition, it turns out that the centroidal dynamics can be written such that computational benefits from both sequential and simultaneous optimal control formulations are inherited (*Contr. A4*), which is not the case for general optimal control problems.

### Context

Our consistent optimal control formulation and decoupling is motivated from experiments with the control architecture presented in Paper II. It is an optimal control formulation of the previously proposed algorithm for whole-body motion generation. The structure identified in the centroidal dynamics can be exploited to write new receding horizon controllers that would operate at an intermediate time-scale (layer 2 in Figure 2.1). For example, we were able to convexify the centroidal dynamics in [112] allowing us to compute motion profiles for the whole-body in less than a second and regularize the change of momentum without losing the convex structure. Plans resulting from our algorithm can then be used to generate feedback policies similar to those in Paper II and tracked with the whole-body controller from Paper I.

## 2.4. List of contributions

The contributions of this thesis are grouped into theoretical and algorithmic findings (*Contr. A1 - Contr. A7*) and experimental evaluation (*Contr. B1 - Contr. B5*) as follows.

**Contr. A1** We propose a computationally efficient whole-body motion generation and control framework for multi-contact behaviors of robots with arms and legs. The approach exploits the structure of robot dynamics through centroidal momentum to create a numerical framework.

**Contr. A2** We formulate a whole-body optimal control problem and decompose it into two problems for which dedicated solvers exist: a) trajectory optimization over the whole-body and b) optimal control with centroidal dynamics.

**Contr. A3** We describe the geometric structure of the centroidal dynamics. In particular, we show that the non-convex part can be written as a difference of convex functions that can be computed by construction without falling back to rank-revealing operations.

**Contr. A4** We show that the optimal control problem over centroidal dynamics can be written in sequential form while preserving the problem sparsity. It allows to preserve linear time complexity while significantly decreasing the number of optimization variables and constraints.

**Contr. A5** A locally optimal feedback law for momentum tracking is derived using time-varying LQR and embedded into two priority levels: a) desired closed-loop momentum dynamics and b) optimal contact force regularization.

**Contr. A6** Exploiting the structure of robot dynamics, we propose a QP cascade variant using a minimal number of dynamic constraints and show that the

first level of the hierarchy can be solved analytically, therefore significantly improving computational complexity.

**Contr. A7** Specialized optimization algorithms are implemented to exploit the structure of the centroidal dynamics.

**Contr. B1** We present experimental validation of the capabilities of our LQR momentum control design. We compare our feedback policy to standard PD design for balancing on one and two legs.

**Contr. B2** We show experiments with a numerical implementation of the QP cascade in a realtime 1KHz control loop. This C++ library is used in all humanoid experiments and by the whole "movement generation and control group" for motion control of legged robots.

**Contr. B3** We demonstrate hierarchical inverse dynamics with operational space feedback control in a 1 kHz control loop on our humanoid.

**Contr. B4** To the best of our knowledge we are the first to demonstrate (linear and angular) momentum regulation on a torque controlled humanoid.

**Contr. B5** Systematic balancing experiments with hierarchical inverse dynamics and (linear+angular) momentum regulation demonstrate state of the art disturbance rejection performance. The systematic tests can also serve as a benchmark for other algorithms.

## 2.5. Complete List of Publications

In the following, we list all publications published through out the course of the PhD studies.

### Book chapters and journal articles

- L. Righetti and A. Herzog. *Momentum-Centered Control of Contact Interactions*, pages 339–359. Springer International Publishing, Cham, 2017. ISBN 978-3-319-51547-2. doi: 10.1007/978-3-319-51547-2\_14"
- A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti. Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid. *Autonomous Robots*, 40(3):473–491, 2016. ISSN 1573-7527. doi: 10.1007/s10514-015-9476-6
- A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal. Learning of grasp selection based on shape-templates. *Autonomous Robots*, 36(1):51–65, 2014. ISSN 1573-7527. doi: 10.1007/s10514-013-9366-8

## 2. Contributions

---

### Conference articles

- A. Herzog, S. Schaal, and L. Righetti. Structured contact force optimization for kino-dynamic motion generation. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 2703–2710, Oct 2016. doi: 10.1109/IROS.2016.7759420
- A. Herzog, N. Rotella, S. Schaal, and L. Righetti. Trajectory generation for multi-contact momentum-control. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2015
- A. Herzog, L. Righetti, F. Grumminger, P. Pastor, and S. Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2014
- A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-Based Learning of Grasp Selection. In *IEEE Intl Conf on Robotics and Automation*, 2012
- M. Khadiv, A. Herzog, S. A. Moosavian, and L. Righetti. Step timing adjustment: A step toward generating robust gaits. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2016
- M. Khadiv, S. Kleff, A. Herzog, S. A. Moosavian, and L. Righetti. Stepping stabilization using a combination of dcm tracking and step adjustment. In *RSI International Conference on Robotics and Mechatronics*, 2016
- B. Ponton, A. Herzog, S. Schaal, and L. Righetti. A convex model of momentum dynamics for multi-contact motion generation. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2016
- N. Rotella, A. Herzog, S. Schaal, and L. Righetti. Humanoid momentum estimation using sensed contact wrenches. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2015
- J. Bohg, J. Romero, A. Herzog, and S. Schaal. Robot arm pose estimation through pixel-wise part classification. In *IEEE Intl Conf on Robotics and Automation*, 2014

### In preparation

- M. Khadiv, A. Herzog, S. A. Moosavian, and L. Righetti. A robust walking controller based on online step location and duration optimization for bipedal locomotion. *ArXiv e-prints*, Apr 2017

# 3

Chapter

## **Optimization-based motion generation for multiped robots in contact scenarios**

---



---

## Paper I: “Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid”

### Authors

Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, Ludovic Righetti

### Abstract

Hierarchical inverse dynamics based on cascades of quadratic programs have been proposed for the control of legged robots. They have important benefits but to the best of our knowledge have never been implemented on a torque controlled humanoid where model inaccuracies, sensor noise and real-time computation requirements can be problematic. Using a reformulation of existing algorithms, we propose a simplification of the problem that allows to achieve real-time control. Momentum-based control is integrated in the task hierarchy and a LQR design approach is used to compute the desired associated closed-loop behavior and improve performance. Extensive experiments on various balancing and tracking tasks show very robust performance in the face of unknown disturbances, even when the humanoid is standing on one foot. Our results demonstrate that hierarchical inverse dynamics together with momentum control can be efficiently used for feedback control under real robot conditions.

Published in:  
*Autonomous Robots*, pp. 473–491, Volume 40, Issue 3, 2016

Award:  
Finalist for the Best WBC-Paper Award 2016, IEEE-RAS Technical Committee on Whole-Body Control

Notes:  
This journal article is an extended version of a conference paper. [59]

## I.1. Introduction

<sup>1</sup>We expect autonomous legged robots to perform complex tasks in persistent interaction with an uncertain and changing environment (e.g. in a disaster relief scenario). Therefore, we need to design algorithms that can generate precise but compliant motions while optimizing the interactions with the environment. In this context, the choice of a control strategy for legged robots is of primary importance as it can drastically improve performance in the face of unexpected disturbances and therefore open the way for agile robots, whether they are locomoting or performing manipulation tasks.

Robots with torque control capabilities [16, 65], including humanoids [27, 103, 109], are becoming increasingly available and torque control algorithms are therefore necessary to fully exploit their capabilities. Indeed, such algorithms often offer high performance for motion control while guaranteeing a certain level of compliance [16, 74, 125, 127]. In addition, they also allow for the direct control of contact interactions with the environment [65, 121, 122], which is required during operation in dynamic and uncertain environments. Recent contributions have also demonstrated the relevance of torque control approaches for humanoid robots [67, 109, 137]. We can essentially distinguish two control approaches.

*Passivity-based* approaches on humanoids were originally proposed in [67] and recently extended in [109]. They compute admissible contact forces and control commands under quasi-static assumptions. The great advantage of such approaches is that they do not require a precise dynamic model of the robot. Moreover, robustness is generically guaranteed due to the passivity property of the controllers. However, the quasi-static assumption can be a limitation for dynamic motions.

On the other hand, controllers based on the *full dynamic model* of the robot have also been successfully implemented on legged robots [65, 96, 122, 137, 146]. These methods essentially perform a form of inverse dynamics. The advantage of such approaches is that they are in theory well suited for very dynamic motions. However, sensor noise (particularly in the velocities), limited torque bandwidth and the need for a precise dynamic model make them more challenging to implement. Moreover, it is generally required to simplify the optimization process to meet time requirements of fast control loops (typically 1 kHz on modern torque controlled robots). Although there are many contributions showing the potential of such approaches in simulation [42, 86, 127, 135], evaluations on real robots are still rare due to the lack of torque controlled humanoid platforms and the complexity in conducting such experiments.

Extensions of the inverse dynamics approach have been proposed where it is also possible to control hierarchies of tasks using the full dynamics of the robot. The main advantage is that it is possible to express complicated behaviors directly at the task level with a strict enforcement of hierarchies between tasks. It is, for example, useful to ensure that a balancing task will take precedence over a task of lower importance in case of conflicting goals. Early hierarchical approaches are

---

<sup>1</sup>Part of the material presented in this paper has been presented at the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

based on pseudo-inverse techniques [131] and take inspiration directly from techniques used for manipulators [104]. However, pseudo-inverse-based controllers are limited as they cannot properly handle inequality constraints such as torque limits or friction cone constraints. More recently, generalizations have been proposed [31, 41, 95, 125] that naturally allow the inclusion of arbitrary types of tasks including inequalities. The resulting optimization problems are phrased as cascades of quadratic programs (QPs). Evaluation of their applicability was done in simulation and it has been shown that these algorithms are fast enough to be implemented in a real-time fast control loop for inverse kinematics. It has also been argued that they can be implemented fast enough for the use with inverse dynamics and can work on robots with model-uncertainty, sensor noise and limited torque bandwidth. But to the best of our knowledge, these controllers have never been used as feedback-controllers on real torque-controlled humanoids.

In [125], the trajectories computed in simulation are replayed on a real robot using joint space position control, but the method is not used for feedback control in task-space on the robot. This work is very interesting because it demonstrates that trajectories generated by a hierarchical inverse dynamics are such that they can be used on a real system. However, it is important to note that this does not show that feedback control can be done using these controllers. Indeed, when replaying trajectories, feedback is reduced to joint level tracking. Therefore it is not possible to directly control interaction forces during multi-contact tasks or to close a feedback loop directly around the tasks of interests, for example the center of gravity (CoG), that respects the desired hierarchies. It is worth mentioning that [65] recently successfully implemented a controller using the full dynamics of the robot and task hierarchies on a torque controlled quadruped robot. The approach is based on pseudo-inverses and not QPs which makes it potentially inefficient to handle inequalities (e.g. friction cone constraints, torque saturation, center of pressure constraints, etc...).

During balancing and walking tasks, an appropriate control of the CoG is of major importance. Recently, it has been realized that the control of both the linear (i.e. the CoG) and angular momentum of the robot could be very beneficial for balancing and walking tasks. The control of overall momentum was originally proposed in [73] using a resolved rate control framework and it was recently extended in [91] where it was integrated with an inverse dynamics controller. It has been shown in several contributions [91, 150] that the regulation of momentum could be very powerful for control on humanoids. Despite the growing popularity for momentum-based control approaches, there have been very few evaluations of such techniques on a real humanoid robot [137]. In [137] the momentum-based control is computed using a simplification of the optimization problem and does not necessarily generate the optimal command. Moreover, the control command generated from inverse dynamics is used in conjunction with a joint PD controller and not used as the sole feedback controller of the system (i.e. there are two distinct feedback pathways, one coming from the momentum control through inverse dynamics and the other coming from desired joint positions at the joint level). To the best of our knowledge a momentum-based controller has never been evaluated

either in a complete hierarchical inverse dynamics framework or without additional joint PD stabilization.

As advanced torque control techniques are developed there is a need to evaluate these techniques on torque-controlled platform to assess their capabilities and also their drawbacks. Such an evaluation is the main goal of the paper. In a recent contribution [61], we have demonstrated that hierarchical inverse dynamics controllers could be efficiently used on a torque-controlled humanoid robot. In particular, we demonstrated robust performance during balancing and tracking tasks when using a momentum-based balance control approach. We also proposed a method to simplify the optimization problem by factoring the dynamics equations of the robot such that we could significantly reduce computational time and achieve a 1kHz control-loop.

**Contribution** In this contribution, we extend our preliminary work and present extensive experimental evaluations. First, we show modifications we applied to the algorithm, originally proposed by [31, 77], that were necessary to execute it in a real-time feedback control setting for inverse dynamics tasks (Section I.2). We also propose a method to systematically compute the feedback gains for the linear and angular momentum control task by using a linear optimal control design approach (Section I.3). This leads us to the main contribution of this paper, where we show experiments with extensive quantitative analysis for various tasks (Sections I.4 and III.5). We show that the momentum-based controller with optimal feedback gains can improve robot performance. Balancing experiments in various conditions demonstrate performances that are comparable to, if not better than, current state of the art balancing algorithms, even when the robot is balancing on one foot. Tracking and contact switching experiments also show the versatility of the approach. It is, to the best of our knowledge, the first demonstration of the applicability of the methods proposed in [31] or [125] as feedback controllers on torque controlled humanoids (i.e. without joint space PD control) with the use of a momentum-based control approach. In the last section, we discuss the experimental results as compared to the state of the art.

## I.2. Hierarchical Inverse Dynamics

In this section, we detail our modeling assumptions, give a short summary on how tasks can be formulated as desired closed-loop behaviors and revisit the original solver formulation [31]. In Section I.2.3 we then propose a simplification to reduce the complexity of the original formulation. The simplification is also applicable to any other inverse dynamics formulation.

### I.2.1. Modelling Assumptions and Problem Formulation

In the following, we describe the constraints and tasks that are considered by the hierarchical inverse dynamics. They will all be written as affine functions of joint

and body accelerations, joint torques and contact forces in order to formulate the control problem as a series of quadratic programs. They constitute the variables that will be optimized by the controller.

**Rigid Body Dynamics** Assuming rigid-body dynamics, we can write the equations of motion of a robot as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}_c^T \boldsymbol{\lambda} \quad (3.1)$$

where  $\mathbf{q} = [\mathbf{q}_j^T \ \mathbf{x}^T]^T$  denotes the configuration of the robot.  $\mathbf{q}_j \in \mathbb{R}^n$  is the vector of joint positions and  $\mathbf{x} \in \text{SE}(3)$  denotes the position and orientation of a frame fixed to the robot with respect to an inertial frame (the floating base).  $\mathbf{M}(\mathbf{q})$  is the inertia matrix,  $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$  is the vector of all non-contact forces (Coriolis, centrifugal, gravity, friction, etc.),  $\mathbf{S} = [\mathbf{I}_{n \times n} \ \mathbf{0}]$  represents the underactuation,  $\boldsymbol{\tau}$  is the vector of commanded joint torques,  $\mathbf{J}_c$  is the Jacobian of the contact constraints and  $\boldsymbol{\lambda}$  are the generalized contact forces.

**Contact constraints** End effectors are constrained to remain stationary. We express the constraint that the feet (or hands) in contact with the environment do not move ( $\dot{\mathbf{x}}_c = \text{const}$ ) by differentiating it twice and using the fact that  $\ddot{\mathbf{x}}_c = \mathbf{J}_c \ddot{\mathbf{q}}$ . We get the following equality constraint

$$\mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0}. \quad (3.2)$$

**Center of pressure** To ensure stationary contacts, the center of pressure (CoP) at each end effector needs to reside in the interior of the end effector's support polygon. This can be expressed as a linear inequality by expressing the ground reaction force at the zero moment point.

**Friction cone** For the feet not to slip we constraint the ground reaction forces (GRFs) to stay inside the friction cones. In our case, we approximate the cones by pyramids to have linear inequality constraints in the contact forces.

**Torque and joint limits** Especially important for generating control commands that are valid on a robot is to take into account actuation limits  $\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{max}$ . The same is true for joint limits, which can be written as  $\ddot{\mathbf{q}}_{min} \leq \ddot{\mathbf{q}}_j \leq \ddot{\mathbf{q}}_{max}$ , where the bounds are computed in the form  $\ddot{\mathbf{q}}_{min/max} \propto \tanh(\mathbf{q} - \ddot{\mathbf{q}}_{min/max})$ .

**Motion and force control tasks** Motion controllers can be phrased as  $\ddot{\mathbf{x}}_{ref} = \mathbf{J}_x \ddot{\mathbf{q}} + \dot{\mathbf{J}}_x \dot{\mathbf{q}}$ , where  $\mathbf{J}_x$  is the task Jacobian and  $\ddot{\mathbf{x}}_{ref}$  is a reference task acceleration that will correspond to a desired closed-loop behavior (e.g. obtained from a PD-controller). Desired contact forces can be directly expressed as equalities on the generalized forces  $\boldsymbol{\lambda}$ . In general, we assume that each control objective can be expressed as a linear combination of  $\ddot{\mathbf{q}}$ ,  $\boldsymbol{\lambda}$  and  $\boldsymbol{\tau}$ , which are the optimization

variables of our problem.

At every control cycle, the equations of motion (Equation (3.1)), the constraints for physical consistency (torque saturation, CoP constraints, etc.) and our control objectives are all expressed as affine equations of the variables  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\tau}$ . Tasks of the same priority can then be stacked vertically into the form

$$\mathbf{A}\mathbf{y} + \mathbf{a} \leq 0, \quad (3.3)$$

$$\mathbf{B}\mathbf{y} + \mathbf{b} = 0, \quad (3.4)$$

where  $\mathbf{y} = [\ddot{\mathbf{q}}^T \ \boldsymbol{\lambda}^T \ \boldsymbol{\tau}^T]^T$ ,  $\mathbf{A} \in \mathbb{R}^{m \times (2n+6+6c)}$ ,  $\mathbf{a} \in \mathbb{R}^m$ ,  $\mathbf{B} \in \mathbb{R}^{k \times (2n+6+6c)}$ ,  $\mathbf{b} \in \mathbb{R}^k$  and  $m, k \in \mathbb{N}$  the overall task dimensions and  $n \in \mathbb{N}$  the number of robot DoFs.  $c \in \mathbb{N}$  is the number of constrained end effectors.

The goal of the controller is to find  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$  and  $\boldsymbol{\tau}$  (and therefore a control command) that satisfies these objectives as well as possible. Objectives will be stacked into different priorities, with the highest priority in the hierarchy given to physical consistency. In a lower priority, we will express balancing and motion tracking tasks and we will put tasks for redundancy resolution in the lowest priorities.

### I.2.2. Hierarchical Tasks & Constraints Solver

The control objectives and constraints in Equations (3.3) and (3.4) might not have a common solution, but need to be traded off against each other. In case of a push, for instance, the objective to decelerate the CoG might conflict with a swing foot task. A tradeoff can be expressed in form of slacks on the expressions in Equations (3.3),(3.4). The slacks are then minimized in a quadratic program. We propose an algorithm that is a combination of the methods originally proposed in [31, 77].

$$\min_{\mathbf{y}, \mathbf{v}, \mathbf{w}} \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 + \epsilon \|\mathbf{y}\| \quad (3.5)$$

$$\text{s.t. } \mathbf{V}(\mathbf{A}\mathbf{y} + \mathbf{a}) \leq \mathbf{v}, \quad (3.6)$$

$$\mathbf{W}(\mathbf{B}\mathbf{y} + \mathbf{b}) = \mathbf{w}, \quad (3.7)$$

where matrices  $\mathbf{V} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{W} \in \mathbb{R}^{k \times k}$  weigh the cost of constraints against each other and  $\mathbf{v} \in \mathbb{R}^m$ ,  $\mathbf{w} \in \mathbb{R}^k$  are slack variables. Note that  $\mathbf{v}, \mathbf{w}$  are not predefined, but part of the optimization variables. The objective is regularized by a small value  $\epsilon$  (typically  $10^{-4}$ ), which ensures positive definiteness of the objective hessian. In the remainder, we write the weighted tasks using  $\bar{\mathbf{A}} = \mathbf{V}\mathbf{A}$ ,  $\bar{\mathbf{a}} = \mathbf{V}\mathbf{a}$ ,  $\bar{\mathbf{B}} = \mathbf{W}\mathbf{B}$ ,  $\bar{\mathbf{b}} = \mathbf{W}\mathbf{b}$ .

Although  $\mathbf{W}, \mathbf{V}$  allow us to trade-off control objectives against each other, strict prioritization cannot be guaranteed with the formulation in Equation (3.5). For instance, we might want to trade off tracking performance of tasks against each other, but we do not want to sacrifice physical consistency of a solution at any

cost. In order to guarantee prioritization, we solve a sequence of QPs, in which a QP with constraints imposed by lower priority tasks is optimized over the set of optimal solutions of higher priority tasks as proposed by [77]. Given one solution  $(\mathbf{y}_r^*, \mathbf{v}_r^*)$  for the QP of priority  $r$ , all remaining optimal solutions  $\mathbf{y}$  in that QP are expressed by the equations

$$\mathbf{y} = \mathbf{y}_r^* + \mathbf{Z}_r \mathbf{u}_{r+1}, \quad (3.8)$$

$$\bar{\mathbf{A}}_r \mathbf{y} + \bar{\mathbf{a}}_r \leq \mathbf{v}_r^*, \quad (3.9)$$

...

$$\bar{\mathbf{A}}_1 \mathbf{y} + \bar{\mathbf{a}}_1 \leq \mathbf{v}_1^*,$$

where  $\mathbf{Z}_r \in \mathbb{R}^{(2n+6+6c) \times z_r}$  represents a surjective mapping into the nullspace of all previous equalities  $\bar{\mathbf{B}}_r, \dots, \bar{\mathbf{B}}_1$  and  $\mathbf{u}_r \in \mathbb{R}^{z_r}$  is a variable that parameterizes that nullspace. We compute  $\mathbf{Z}_r$  from a Singular Value Decomposition (SVD). With this nullspace mapping we reduce the number of variables from one hierarchy level to the next by the number of locked degrees of freedom. In our implementation the SVD is computed in parallel with the QP at priority level  $r - 1$  and rarely finishes after the QP, i.e. it adds only a negligible overhead.

Now, we can express a QP of the next lower priority level  $r + 1$  and additionally impose the constraints in Equations (3.8), (3.9) in order to optimize over  $\mathbf{y}$  without violating optimality of higher priority QPs:

$$\min_{\mathbf{u}_{r+1}, \mathbf{v}_{r+1}} \|\bar{\mathbf{B}}_{r+1}(\mathbf{y}_r^* + \mathbf{Z}_r \mathbf{u}_{r+1}) + \bar{\mathbf{b}}_{r+1}\| + \quad (3.10)$$

$$\|\mathbf{v}_{r+1}\| + \epsilon \|\mathbf{y}\|$$

$$\text{s.t.} \quad \bar{\mathbf{A}}_{r+1}(\mathbf{y}_r^* + \mathbf{Z}_r \mathbf{u}_{r+1},) + \bar{\mathbf{a}}_{r+1} \leq \mathbf{v}_{r+1}, \\ \bar{\mathbf{A}}_r(\mathbf{y}_r^* + \mathbf{Z}_r \mathbf{u}_{r+1},) + \bar{\mathbf{a}}_r \leq \mathbf{v}_r^*, \quad (3.11)$$

...

$$\bar{\mathbf{A}}_1(\mathbf{y}_r^* + \mathbf{Z}_r \mathbf{u}_{r+1},) + \bar{\mathbf{a}}_1 \leq \mathbf{v}_1^*,$$

where we wrote the QP as in Equation (3.5) and substituted  $\mathbf{w}$  into the objective function. In order to ensure that we optimize over the optimal solutions of higher priority tasks, we added Equation (3.9) as an additional constraint and substituted Equation (3.8) into Equations (3.10)-(3.11). This allows us to solve a stack of hierarchical tasks recursively as originally proposed by [77]. Right-multiplying  $\mathbf{Z}_r$  to inequality matrices  $\bar{\mathbf{A}}_r$  creates zero rows for constraints that do not have degrees of freedom left. For example, after the GRFs are decided, CoP and friction constraints become obsolete. This way the number of inequalities reduces potentially from one QP to the other. Note that this optimization algorithm is guaranteed to find the optimal solution in a least-squares sense while satisfying priorities.

With this formulation we combine the two benefits of having inequalities in all hierarchical levels [77] and reducing the number of variables from one QP to the other [31].

### I.2.3. Decomposition of Equations of Motion

Hierarchical inverse dynamics approaches usually have in common that consistency of the variables with physics, i.e. the equations of motion, need to be ensured. In [31] these constraints are expressed as equality constraints (with slacks) resulting in an optimization problem over all variables  $\ddot{\mathbf{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}$ . In [95] a mapping into the nullspace of Equation (3.1) is obtained from a SVD on Equation (3.1). In both cases, complexity can be reduced as we will show in the following. We decompose the equations of motion as

$$\mathbf{M}_u(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_u(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}_{c,u}^T \boldsymbol{\lambda}, \quad (3.12)$$

$$\mathbf{M}_l(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_l(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_{c,l}^T \boldsymbol{\lambda} \quad (3.13)$$

where Equation (3.12) is just the first  $n$  equations of Eq. (3.1) and Equation (3.13) is the last 6 equations related to the floating base. The latter equation can then be interpreted as the Newton-Euler equations of the whole system [154]. They express the change of momentum of the robot as a function of external forces. A remarkable feature of the decomposition in Equations (3.12), (3.13) is that the torques  $\boldsymbol{\tau}$  only occur in Equation (3.12) and are exactly determined by  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$  in the form

$$\boldsymbol{\tau} = \mathbf{M}_u(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}_u(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}_{c,u}^T \boldsymbol{\lambda} \quad (3.14)$$

Since  $\boldsymbol{\tau}$  is linearly dependent on  $\ddot{\mathbf{q}}, \boldsymbol{\lambda}$ , for any combination of accelerations and contact forces there always exists a solution for  $\boldsymbol{\tau}$ . It is given by Equation (3.14). Therefore, it is only necessary to use Equation (3.13) as a constraint for the equations of motion during the optimization (i.e. the evolution of momentum is the only constraint).

Because of the linear dependence, all occurrences of  $\boldsymbol{\tau}$  in the problem formulation (i.e. in Equations (3.3)-(3.4)) can be replaced with Equation (3.14). This reduces the number of variables in the optimization from  $(2n+6+6c)$  to  $(n+6+6c)$ . This decomposition thus eliminates as many variables as there are DoFs on the robot. This simplification is crucial to reduce the time taken by the optimizer and allowed us to implement the controller in a 1 kHz feedback control loop.

**Remark** The simplification that we propose<sup>2</sup> can appear trivial at first sight. However, it is worth mentioning that such a decomposition is always ignored in related work despite the need for computationally fast algorithms [31], [95], [137].

---

<sup>2</sup>We originally proposed the simplification in a technical note [57].

### I.2.4. Solution to the first priority

Since we are interested in writing inverse dynamics controllers, we set the highest priority tasks to always be the Newton-Euler Equations (Equation (3.13)) together with torque saturation constraints. We then need to find the space of solutions for equations

$$\mathbf{B}_1 \mathbf{y} + \mathbf{b}_1 = 0 \quad (3.15)$$

$$-\tau_{sat} \leq \tau(\mathbf{y}) \leq \tau_{sat} \quad (3.16)$$

with  $\tau(\mathbf{y})$  given by Equation (3.14),  $\mathbf{B}_1 = [\mathbf{M}_1 - \mathbf{J}_{c,l}^T]$  and  $\mathbf{b}_1 = \mathbf{N}_1$ . In this case, we can obtain the space of solutions (cf. Equation (3.8)) without having to solve a QP. A trivial solution can be readily obtained, thus reducing computation time. Indeed, it is always possible to satisfy the equations of motion together with the torque saturation constraints exactly by choosing  $\boldsymbol{\tau} = \boldsymbol{\lambda} = \mathbf{0}$  and resolving for  $\ddot{\mathbf{q}} = -\mathbf{M}^{-1}\mathbf{N}$  using Equation (3.1). The resulting solution will be in the set of minimizers, i.e.

$$\exists \mathbf{u}_1 : \quad \mathbf{y} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{0} \end{bmatrix} = -\mathbf{B}_1^\dagger \mathbf{b}_1 + \mathbf{Z}_1 \mathbf{u}_1, \quad (3.17)$$

$$\wedge \quad \boldsymbol{\tau}(\mathbf{y}) = \mathbf{0} \quad (3.18)$$

with  $\mathbf{Z}_1$  computed as described in Section I.2.2. We can then obtain  $\mathbf{y}_1^* = -\mathbf{B}_1^\dagger \mathbf{b}_1, \mathbf{v}_1^* = \mathbf{0}$ , which is required to construct the QP for priority  $r = 2$ . Although  $\mathbf{y}_1^*$  may violate torque saturation constraints, Equations (3.17), (3.18) guarantee that an admissible  $\mathbf{y}$  can always be found and will be found in the following QPs. With this choice of  $\mathbf{y}_1^*$  there is no need to invert  $\mathbf{M}$ . Note that  $\mathbf{B}_1$  represents the Newton-Euler equations of the system and is always of full row rank<sup>3</sup> and thus computing  $\mathbf{B}_1^\dagger$  requires only inverting the  $6 \times 6$  sized positive definite matrix  $\mathbf{B}_1 \mathbf{B}_1^T$ . By designing the first hierarchy level in this way, we can improve computation time by avoiding to solve the first QP while already reducing the size of the problem by 6 variables for the next priority.

## I.3. Linear and angular momentum regulation

As we mentioned in the introduction, we are interested in writing desired feedback behaviors using hierarchical inverse dynamics and more specifically, we are interested in controlling the linear and angular momentum of the robot. The feedback controller that regulates momentum is often written as a PID controller with hand-tuned gains. Such control design does not take into account the coupling between linear and angular momentum during a multi-contact task and can potentially lead to a controller which is sub-optimal and difficult to tune.

---

<sup>3</sup>The part of  $\mathbf{M}_1$  multiplying the base acceleration is always full rank.

In this section, we write the momentum regulation problem as a force control task and then use a simple LQR design to compute a linear optimal feedback control law. This feedback law is then used to compute a desired closed-loop behavior in the hierarchical inverse dynamics controller. The advantage of such design is that it fully exploits multi-contacts and momentum coupling while significantly simplifying the design of the controller by reducing the number of open parameters.

### I.3.1. Linear and angular momentum models

The control of momentum and CoG is inherently both a kinematic and a force task. Indeed, using the centroidal momentum matrix [108], one can find a linear mapping between the overall robot momentum and the robot joint and pose velocities

$$\mathbf{h} = \mathbf{H}_G(\mathbf{q})\dot{\mathbf{q}} \quad (3.19)$$

where  $\mathbf{h} = [\mathbf{h}_{lin}^T \ \mathbf{h}_{ang}^T]^T$  is the system linear and angular momentum expressed at the CoG. The matrix  $\mathbf{H}_G$  is called the centroidal momentum matrix. The derivative of Equation (3.19) allows us to express the rate of change of the momentum and the CoG

$$\begin{aligned} \dot{\mathbf{x}}_{cog} &= \frac{1}{m} \mathbf{h}_{lin} \\ \dot{\mathbf{h}} &= \mathbf{H}_G \ddot{\mathbf{q}} + \dot{\mathbf{H}}_G \dot{\mathbf{q}} \end{aligned} \quad (3.20)$$

This formulation has been often used in a resolved acceleration scheme where the centroidal momentum matrix is viewed as the task Jacobian (e.g. in [91]).

Using the Newton-Euler equations, the total change of momentum can also be written in terms of the external forces

$$\begin{aligned} \dot{\mathbf{x}}_{cog} &= \frac{1}{m} \mathbf{h}_{lin} \\ \dot{\mathbf{h}} &= \left[ \begin{matrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ [\mathbf{x}_i - \mathbf{x}_{cog}] \times & \mathbf{I}_{3 \times 3} & \dots \end{matrix} \right] \boldsymbol{\lambda} + \begin{bmatrix} mg \\ \mathbf{0} \end{bmatrix}, \end{aligned} \quad (3.21)$$

where  $mg$  is the gravitational force,  $\boldsymbol{\lambda}$  the vector of generalized external forces,  $[\cdot] \times$  maps a vector to a skew symmetric matrix, s.t.  $[\mathbf{x}] \times \boldsymbol{\lambda} = \mathbf{x} \times \boldsymbol{\lambda}$  and  $\mathbf{x}_i$  is the position of the  $i^{th}$  contact point.

We see that the rate of momentum change can equivalently be written either as a kinematic task (i.e. a function of  $\ddot{\mathbf{q}}$  as in Equation (3.20)) or a force task (i.e. a function of  $\boldsymbol{\lambda}$  as in Equation (3.48)). The matrix in front of  $\ddot{\mathbf{q}}$  or  $\boldsymbol{\lambda}$  is viewed as the Jacobian of the task.

In general, deriving a momentum control law with Equation (3.48) might be better because we do not have to compute  $\dot{\mathbf{H}}_G$ , which usually is acquired through numerical derivation and might suffer from magnified noise. In addition, in Equation (3.48) external forces can be interpreted as the control inputs of the system, which is a useful interpretation for control design, as we explain below.

### I.3.2. LQR design for momentum control

A desired momentum behavior is typically achieved using a PD control law, for example

$$\dot{\mathbf{h}}_{des} = \mathbf{P} \begin{bmatrix} m(\mathbf{x}_{ref} - \mathbf{x}_{cog}) \\ \mathbf{0} \end{bmatrix} + \mathbf{D}(\mathbf{h}_{ref} - \mathbf{h}) + \dot{\mathbf{h}}_{ref}$$

where  $\mathbf{h}_{ref}$  and  $\mathbf{x}_{ref}$  are reference momentum and CoG trajectories. Using Equation (3.20), a desired closed-loop behavior is then added in the hierarchical inverse dynamics as

$$\begin{aligned} \mathbf{H}_G \ddot{\mathbf{q}} + \dot{\mathbf{H}}_G \dot{\mathbf{q}} \\ = \mathbf{P} \begin{bmatrix} m(\mathbf{x}_{ref} - \mathbf{x}_{cog}) \\ \mathbf{0} \end{bmatrix} + \mathbf{D}(\mathbf{h}_{ref} - \mathbf{h}) + \dot{\mathbf{h}}_{ref} \end{aligned} \quad (3.22)$$

There are, however, several issues with such an approach. First, the tuning of the PD controller can be problematic. In our experience, on the real robot it is necessary to have different gains for different contact configurations to ensure proper tracking which leads to a time consuming process with many open parameters. Second, such a controller does not exploit the coupling between linear and angular momentum rate of change that is expressed in Equation (3.48).

We propose to use the model of Equation (3.48) to compute optimal feedback gains. We linearize the dynamics and compute a LQR controller by selecting a desired performance cost. We find a control law of the form

$$\lambda = -\mathbf{K} \begin{bmatrix} \mathbf{x}_{cog} \\ \mathbf{h} \end{bmatrix} + \mathbf{k}(\mathbf{x}_{ref}, \mathbf{h}_{ref}) \quad (3.23)$$

that contains both feedback and feedforward terms. A desired closed-loop behavior for the momentum that appropriately takes into account the momentum coupling is then computed. The desired task used in the hierarchical inverse dynamics controller is then written as

$$\begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ [\mathbf{x}_i - \mathbf{x}_{cog}] \times & \mathbf{I}_{3 \times 3} & \dots \end{bmatrix} \left( \lambda + \mathbf{K} \begin{bmatrix} \mathbf{x}_{cog} \\ \mathbf{h} \end{bmatrix} - \mathbf{k} \right) = 0 \quad (3.24)$$

We project the control  $\lambda$  into the momentum space such that we can use the available redundancy during multi-contact tasks to optimize the internal forces further. It would not be possible if we used directly Equation (3.23).

The proposed approach takes into account the coupling between linear and angular momentum, which will prove beneficial in the experimental section. Moreover, we specify the performance cost once and for all and the feedback gains are computed optimally for every contact and pose configuration of the robot at a low computational cost. In our experience, it drastically simplified the application on the real robot.

**Remark** In our experiments, we use an infinite horizon LQR design and compute gains for key poses of the robot, one for each contact configurations. During a contact transition we interpolate between the old and new set of gains to ensure continuous control commands. This solution is not ideal from a theoretical point of view as the interpolation does not guarantee stable behavior, but it works well in practice. Indeed, the contact transitions are very fast and all the trajectories were planned in advance. It would also be straightforward to linearize the dynamics at every control sequence and use a receding horizon controller with time-varying gains to allow online replanning of desired trajectories.

## I.4. Experimental Setup

In this section, we detail the experimental setup, the low-level feedback torque control, the state estimation algorithm and the limitations of the hardware. These details are important in order to understand the strengths and limitations of the presented experiments. They should also ease the reproduction of the experimental results on other platforms.

### I.4.1. Sarcos Humanoid Robot

The experiments were done on the lower part of the Sarcos Humanoid Robot [27], shown in Figure 3.1. It consists of two legs and a torso. The legs have 7 DoFs each and the torso has 3 DoFs. Given that the torso supports a negligible mass, because it is not connected to the upper body of the robot and its motion does not significantly influence the dynamics, we froze these DoFs during the experiments. The legs of the robot are 0.82m high. Each foot is 0.09m wide and 0.25m long. Note also that the front of the foot is made of a passive joint that is rather flexible, located 10cm before the tip of the foot. Moving the CoP across this link makes the foot bend and causes the robot to fall. This makes the effectively used part of the sole rather small for a biped. The total robot mass is 51kg.

The robot is actuated with hydraulics and each joint consists of a Moog Series 30 flow control servo valve that moves a piston. Attached to the piston is a load cell to measure the force at the piston. A position sensor is also located at each joint. Each foot has a 6-axis force sensor and we mounted an IMU on the pelvis of the robot from which we measure angular velocities and linear accelerations of the robot in an inertial frame. An offboard computer sends control commands to the robot and receives sensor information in real-time at 1 kHz. The control commands consist of the desired current applied to each valve. We used a computer running a linux kernel patched with Xenomai 2.6.3 for real-time capabilities.

### I.4.2. Low-level torque control

For each actuator, we implemented a torque feedback controller that ensures that each joint produces the desired force generated by the hierarchical inverse dynamics controller. The controller essentially computes desired flow directly in terms of



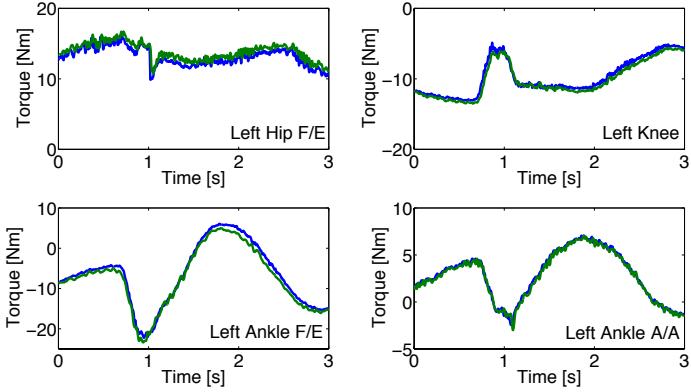
**Figure 3.1.:** The lower part of the Sarcos Humanoid. (Credit:Luke Fisher Photography)

valve current. The controller we implemented is very much inspired from the work in [15, 16], with the difference that we implemented a simpler version where piston velocity feedback has a constant gain. The constant gain allows us to avoid the computation of the piston chamber sizes and the measurement of the pressure inside. The control law is

$$v = PID(F_{des}, F) + K\dot{x}_{piston} + d \quad (3.25)$$

where  $v$  is the valve command,  $PID$  is a PID controller according to desired force command and force measured from the load cells,  $K$  is a positive gain,  $\dot{x}_{piston}$  is the piston velocity (computed from the joint velocity and the kinematic model) and  $d$  is a constant bias.

This controller design allowed us to achieve good torque tracking performance. It is important to note that such performance was necessary to achieve good performance in the hierarchical inverse dynamics controller. Figure 3.2 illustrates the torque tracking performance during a balancing experiment.



**Figure 3.2.:** Example of torque tracking performance during a balancing experiment. The left hip flexion/extension, left knee and left ankle flexion/extension and adduction/abduction joints are shown. Both desired (blue) and actual (green) torques are shown.

### I.4.3. State estimation

An accurate estimation of the floating base pose and twist is important for a good performance of the inverse dynamics controller. We used a recently developed approach [123] based from the ideas in [14]. The estimation uses an extended Kalman filter that fuses information from both the IMU and leg kinematics. The filter handles contact switching and makes no assumption about the gait or contact location in the world but only uses the knowledge that a leg is in contact. It also has favorable observability characteristics which make it particularly convenient for our experiments. More details on the filter can be found in [123].

### I.4.4. Dynamic model

Our dynamic model is based on the CAD model of the robot. This means that it is not very accurate as it does not take into account the contribution of the hydraulic hoses, the electronics or any type of friction in the model. We expect to have even better performance once we perform a good identification of the dynamics [9, 99] but it is interesting to note that good results with hierarchical inverse dynamics can be obtained without a perfect dynamic model. It demonstrates that these methods are robust to model uncertainty in a compilation of balancing and tracking tasks.

### I.4.5. Experimental tools

For our experiments we use different tools to generate and measure disturbances on the robot. We built a push stick that has a FTN-Mini 45 force sensor attached. It measures the applied force over time when we push the robot. We conduct experiments where the robot is standing on a rolling platform or a tilting platform that is put on top of a beam. In both scenarios we attached a Microstrain 3DM-GX3-25 IMU (See Figure 3.3) to the plate that the robot is standing on in order to measure linear acceleration and angular velocities of the platform when a disturbance is applied. These sensors are connected to the controlling computer together with the robot sensors, which allows for easy synchronization of the readings. We use real-time ethernet for the force sensor and real-time USB for the IMU.

## I.5. Experiments

We formulated balancing and motion tracking tasks using the algorithm discussed in Section I.2 together with the momentum controller discussed in Section I.3 and evaluated them on the Sarcos Humanoid described in Section I.4. The performance of the controller was evaluated in different scenarios: balancing experiments and a tracking task in single and double support. A summary of the experiments is shown in the attached movie <sup>4</sup>.

For all the experiments, we run the hierarchical inverse dynamics controller as a feedback controller. The desired torque commands computed by the controller are directly sent to the robot. We do not use any joint PD controller for stabilization (i.e. feedback control is only done in task space). A diagram visualizing the flow of control variables is presented in Figure 3.4.

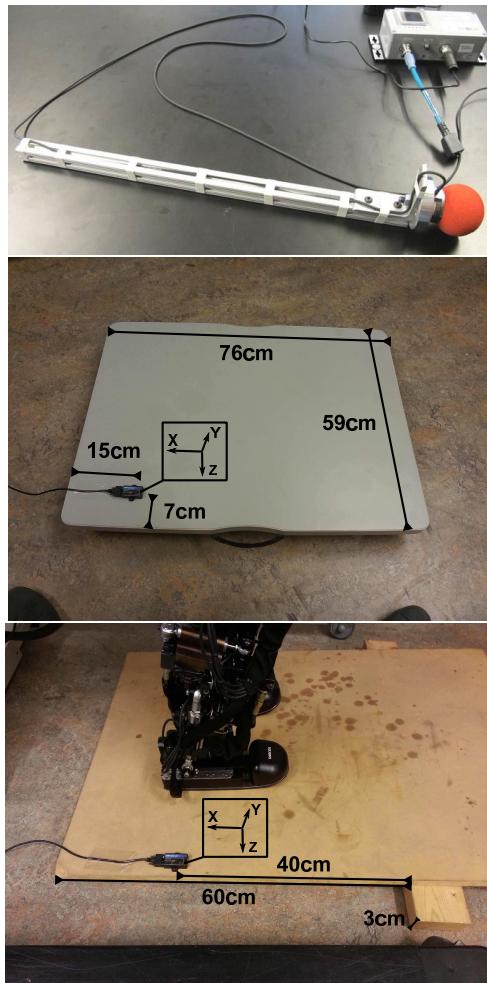
### I.5.1. Processing Time

The computation time of the solver mainly depends on a) the number of DoFs of the robot b) the number of contact constraints and c) the composed tasks. All experiments were performed on an Intel Core i7-2600 CPU with a 3.40GHz processor. Subsequent QPs (cf. Section I.2.2) were solved with an implementation of the Goldfarb-Idnani dual-method [52] using the Eigen matrix library. In the real robot experiments we use the 14 DoF lower part of a humanoid to perform several tasks in a 1kHz control loop. In the following, however, we construct a more complex stepping task in simulation for the full 25 DoF robot. The goal is to a) evaluate the speedup from the simplification proposed in Section I.2.3 and b) give an intuition on how the method scales with the complexity of the robot.

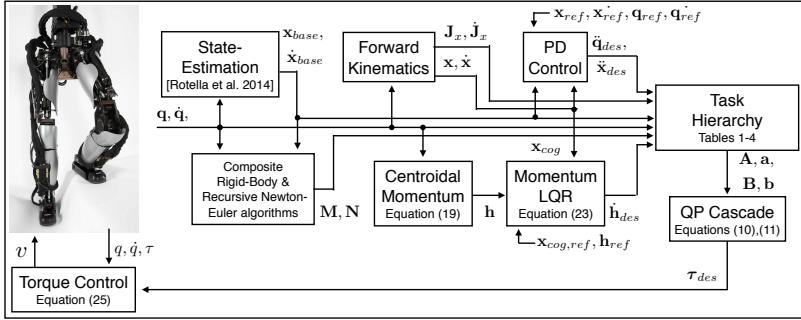
We summarized in Table 3.1 the hierarchy that is used in simulation. The highest two priorities satisfy hardware limitations and dynamic constraints, the third priority task tracks a predefined center of gravity and swing foot motion and the remaining priorities resolve redundancies on motion and forces. The problem

---

<sup>4</sup>The movie is also available on [www.youtube.com/watch?v=jMj3Uv2Q8Xg](http://www.youtube.com/watch?v=jMj3Uv2Q8Xg)



**Figure 3.3.:** We attach a FTN-Mini 45 force sensor to a stick (top) to measure forces during pushes. In some of our experiments the robot stands on a rolling platform (middle) or a balancing board (bottom). In both scenarios an IMU is attached to the platform, which allows measuring linear accelerations and angular velocities, when disturbances are applied. The black box shows the internal frame of the IMU.



**Figure 3.4.:** An overview of the control structure used in the presented experiments.

size changes depending on the number of contacts  $c$  ( $c = 2$  in double support and  $c = 1$  in single support). The proposed decomposition removed 25 equality constraints and 25 optimization variables. We measured the computation time of both versions of the hierarchical solver, one with the full EoM and one with the proposed reduction as plotted in Figure 3.21. Looking at the worst case (as this is significant for execution in a time critical control loop) we can reduce computation time by 40%. In our experiments with a 14 DoF robot, this speedup allows us to run a 1 kHz control-loop as we will demonstrate in the following sections. It would not have been possible by using this algorithm without the simplification. Going from a 14 DoF robot to a 25 DoF robot with similar task setup makes the peak computation time rise from 1ms to 3ms. In our speed comparison in Figure 3.21 one can see that computation time varies with the number of constrained end effectors, which can be problematic if the number of contacts increases too much (e.g. when using both hands and feet).

### I.5.2. Balance Control Experiments

In the first set of experiments on the robot, we were interested in systematically evaluating the balance capabilities of the momentum-based controller with hierarchical inverse dynamics. First, we compare the performance of the balance control when using the LQR design and the PD controller described in Section I.3 and then test the performance of the robot when balancing on a rolling platform and a balancing board.

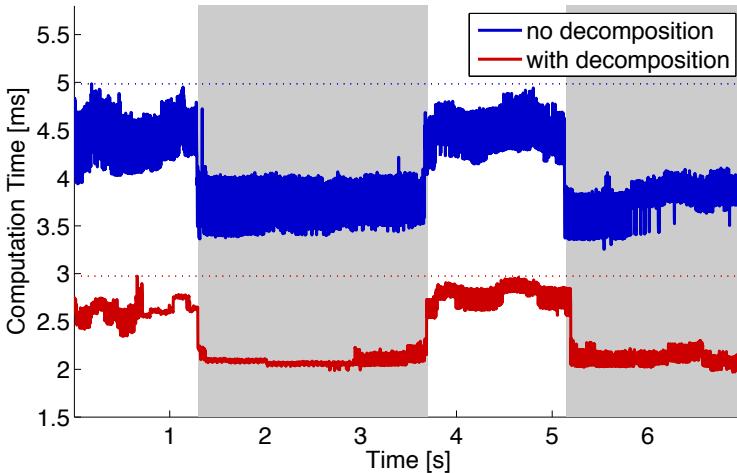
Rank	Nr. of eq/ineq constraints	Constraint/Task
1	25 eq 6 eq $2 \times 25$ ineq	Equation (3.12) (not required for simplified problem) Newton Euler Equation (3.13) torque limits
2	$c \times 6$ eq $c \times 4$ ineq $c \times 4$ ineq $2 \times 25$ ineq	Contact constraints, Eq. (3.2) Center of Pressure, Sec. I.2.1 Friction cone, Sec. I.2.1 joint acceleration limits, Sec. I.2.1
3	3 eq $(2 - c) \times 6$	PD control on CoG PD control on swing foot
4	$25 + 6$ eq	PD control on posture
5	$c \times 6$ eq	regularizer on GRFs
<b>DoFs:</b> 25		<b>max. time:</b> 5 ms / 3 ms

**Table 3.1.:** Full Humanoid Stepping Task for Speed Comparison. The maximum computation time was observed in double support ( $c = 2$ ).

### Specification of the tasks

The specification of the task is summarized in Table 3.2 together with the maximum running time for one control cycle. The physical constraints are put in the highest priority. In the second priority, we put kinematic contact constraints, acceleration limits and constraints on reaction forces, i.e. CoP boundaries and friction cones with a higher weight on CoPs. In the third hierarchy level, we express our desired closed loop-dynamics on the momentum together with a PD controller on the posture and ground reaction force regularization. Here, we prefer to have the momentum control together with the posture control on the same level, since the kinematic contact constraints (2nd priority) lock 12 DoFs and the momentum control another 6 DoFs. Given that we only have 20 DoFs (including 6 for the floating base), we are left with too few DoFs to keep a good posture. In our experience this allowed the robot to keep a better looking posture given the limited redundancy available.

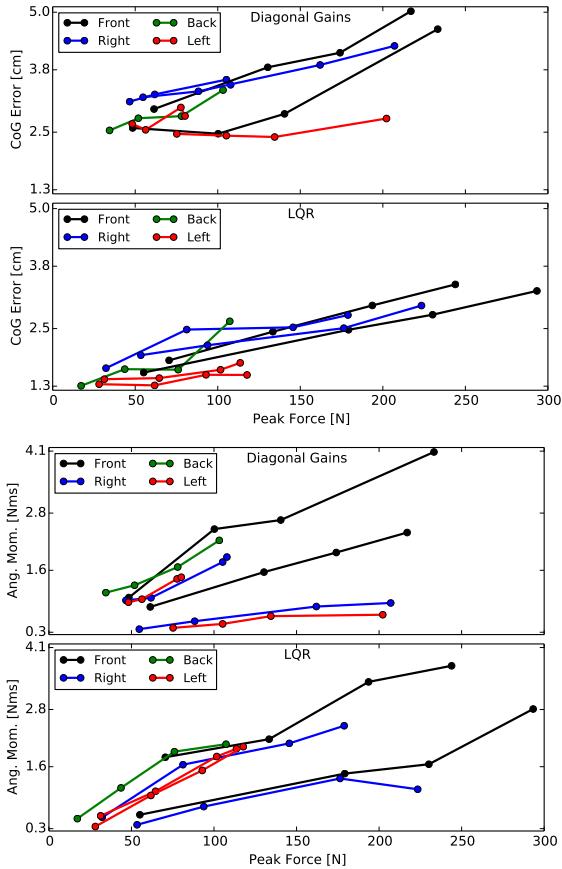
We did not carefully tune the weights between the tasks in the same priority level but merely selected an order of magnitude by choosing among 4 different weights for all tasks,  $10^4, 1, 10^{-1}$  or  $10^{-4}$ . If not stated otherwise, we chose  $\mathbf{P}, \mathbf{D}$  gains to be diagonal matrices in all of our tasks except the momentum control. We put a higher weight on the momentum control since balancing is our main objective and gave less weight to posture control and regularization of ground reaction torques.



**Figure 3.5.:** Processing time of a stepping task (see Table 3.1) using the decomposition proposed in Section I.2.3 (red) and the same task performed without the decomposition (blue). The dotted line represents the maximum computation per control cycle respectively. Intervals shaded in gray show the robot in single support phase. In the remaining time the robot is in double support. With the proposed decomposition we decreased the computation time by approximately 40%.

#### Comparison of momentum controllers

In this experiment we pushed the robot impulsively with our push-stick at various contact points with different force directions. The robot was pushed at 4 points on the torso above the hip (from the front, right, back and left) and at 3 points at hip level (from the front, right and left) as can be seen in the attached video. The electronics of the robot are attached at the back part of the hip which is why we did not push it at that point. At each of the 7 points we applied 4 pushes of increasing impact up to peak forces of 290 N and impulses of 9.5 Ns, which is on the upper scale of pushes in related work [96, 109, 116]. This episode of experiments is executed with both variations of momentum control discussed in Section I.3: PD control with diagonal gain matrices and with optimal gains from LQR design. We put a reasonable amount of effort into finding parameters for both controllers in order to be able to compare them and in both cases we tried to find



**Figure 3.6.:** The robot was pushed from 4 sides at the block above the hip and at the hip. At each point of attack it was pushed 4 times. This figure plots the peak forces of the pushes against the maximum CoG displacement (top two figures) and against the maximum angular momentum (bottom two figures). The 1st and 3rd figures from the top show experiments performed with diagonal gain matrices. In the 2nd and 4th plot experiments were conducted with the LQR momentum controller. The impulses of the pushes were increasing roughly linearly with the peak forces. A list of peak impulses is shown in Table 3.3. It can be seen that overall the CoG error remains lower with the LQR controller, while the angular momentum behaves similarly.

Rank	Nr. of eq/ineq constraints	Constraint/Task
1	6 eq $2 \times 14$ ineq	Newton Euler Equation (3.13) torque limits
2	$2 \times 6$ eq $2 \times 4$ ineq $2 \times 4$ ineq $2 \times 14$ ineq	Contact constraints, Eq. (3.2) Center of Presure, Sec. I.2.1 Friction cone, Sec. I.2.1 joint acceleration limits, Sec. I.2.1
3	6 eq $14 + 6$ eq $2 \times 6$ eq	momentum control, Eqs. (3.24) or (3.23) PD control on posture regularizer on GRFs
<b>DoFs:</b> 14		<b>max. time:</b> 0.9 ms

**Table 3.2.:** Hierarchy for experiments in Double Support

	PD Control							
	Above Hip Joint				At Hip Joint			
	F	R	B	L	F	R	B	L
Peak Force [N]	233	108	103	80	217	207	202	
Impulse [Ns]	7.9	4.7	5.1	3.9	9.1	9.0	6.9	
max. CoG Error [cm]	4.6	3.5	3.4	2.8	5.0	4.3	2.8	
max. Lin. Mom. [Nm]	22.6	10.2	15.8	6.9	13.1	7.9	9.3	
max. Ang. Mom. [Nms]	4.1	1.9	2.2	1.5	2.4	0.9	0.7	
LQR								
	Above Hip Joint				At Hip Joint			
	F	R	B	L	F	R	B	L
	244	179	107	114	293	223	118	
Peak Force [N]	6.9	6.8	6.2	4.7	9.5	8.6	4.3	
Impulse [Ns]	3.4	2.8	2.7	1.8	3.3	3.0	1.5	
max. CoG Error [cm]	19.8	13.6	16.3	9.1	14.4	9.2	9.5	
max. Ang. Mom. [Nms]	3.7	2.5	2.1	2.0	2.8	1.1	2.0	

**Table 3.3.:** Here we list the maximum pushes applied to the robot, where each column in top and bottom tables shows the properties of one push. When the first 7 pushes (top table) were applied, the momentum was controlled with PD control using diagonal gain matrices. For the last seven pushes (bottom table), LQR gains were used. The robot was pushed from the (F)ront, (R)ight (B)ack and (L)eft either above the hip joint or at the height of the hip joint. The first two rows in each table describe the peak force and impulse of each push. The last 3 rows in each table show the maximum deviation of the CoG and the linear and angular momentum of the robot after an impact.

parameters that would lead to fast damping of momentum, with a slight preference for damping linear momentum to ensure that the CoG was tracked properly. Our resulting LQR performance cost was

$$\sum_t^{\infty} \begin{bmatrix} \mathbf{x}_{cog} \\ \mathbf{h} \end{bmatrix}^T \mathbf{Q} \begin{bmatrix} \mathbf{x}_{cog} \\ \mathbf{h} \end{bmatrix} + \boldsymbol{\lambda}^T \mathbf{R} \boldsymbol{\lambda}, \quad (3.26)$$

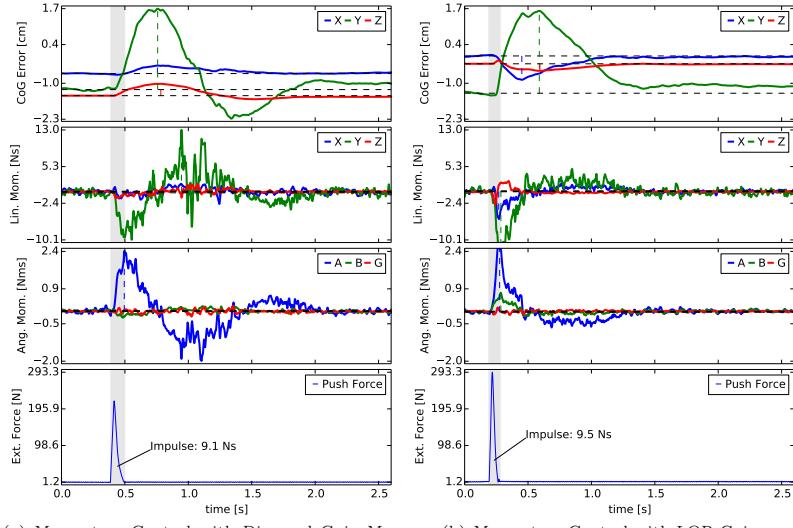
with  $\mathbf{Q} = diag([30, 30, 50, .5, .5, .5, .1, .1, .1])$ ,  $\mathbf{R} = diag([0.1, 0.1, 0.01, 2, 2, 2])$ . For both momentum control tasks, the robot was able to withstand impacts with high peak forces and strong impulses without falling. For every push, the change in momentum was damped out quickly and the CoG was tracked after an initial disturbance. While it is difficult to compare the performance of the controllers with other existing state-of-the-art algorithms because very little quantitative data is available and performance can drastically change across robots, it seems that both controllers perform at least as well as, if not better than, other approaches for which data is available [96, 109, 116]. Indeed, the robot was able to absorb impact up to peak forces of 290 N and impulses of 9.5 Ns. We summarized the information for the strongest pushes in each direction in Table 3.3 as a reference.

In Figure 3.6 we systematically plotted the measured peak forces against the maximum deviation of the CoG and angular momentum for both controllers in order to see the typical behavior of the robot. The impulses are not plotted as they were proportional to the peak forces in all our experiments. The maximum error for both angular momentum and CoG tended to be proportional to the peak force for all experiments. We notice from the figure that for both momentum controllers we get similar maximum deviations in angular momentum. However, with the LQR gains we see a significant improvement in recovering the CoG. From Figure 3.7 we also see how the LQR controller recovers quicker although the robot was pushed harder than with the controller using diagonal gain matrices.

Figure 3.7 shows a typical response for both controllers where we plotted the impact force together with the CoG tracking error and the momentum. We notice that in both cases the disturbance is damped quickly. We notice that although the peak force is higher for the LQR controller, the response is better behaved than for the PD controller and the momentum is damped faster.

While it is always difficult to ensure that a better set of parameters couldn't be found for the PD controller, this result suggests that the LQR design performs better than the PD controller. Moreover, the LQR design is much simpler to tune because the design of a performance cost has a more intuitive meaning than PD gains and it can capture the coupling between linear and angular momentum. The other advantage of the LQR design is that once the cost function is fixed, new gains can be computed for various poses and desired momentum behaviors automatically without manual re-tuning. This aspect was very helpful for the contact switching and single support task that we present in the following section.

The balance controller that is implemented does not rely on co-planar feet and is able to produce very compliant behaviors. When we pushed the robot with a



(a) Momentum Control with Diagonal Gain Matrices

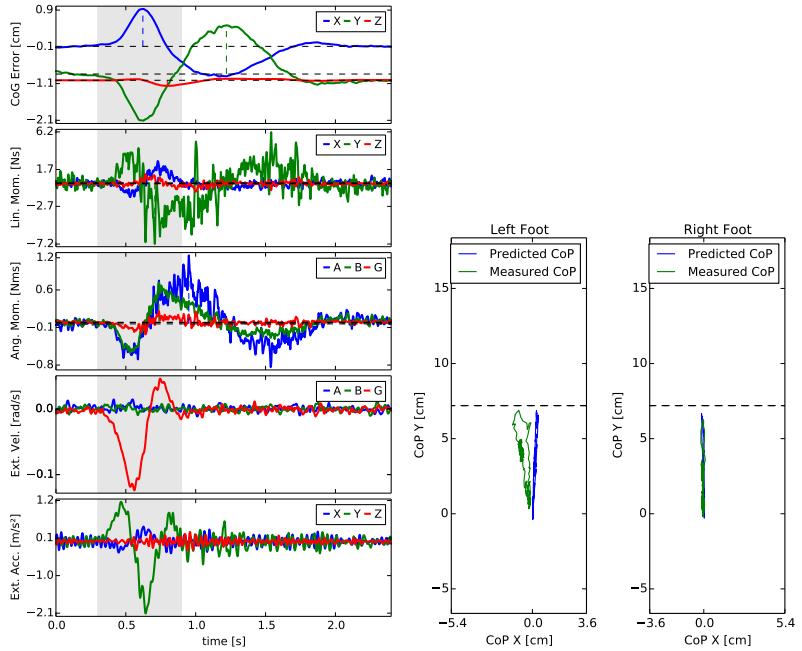
(b) Momentum Control with LQR Gains

**Figure 3.7.:** In this figure we compare typical push recoveries when we run momentum control with diagonal PD gain matrices (left) and with LQR gains (right). Although the push is stronger for the LQR controller (bottom plots), the CoG error (top plots) does not deviate from its stationary position more than with the PD controller. Both the linear and angular momentum of the robot (middle two plots) are damped out quickly by the LQR controller and the CoG comes to rest faster as well.

constant force at various parts, it stayed in balance and adapted its posture in a compliant manner. We also tested the controller when the feet were not co-planar, but one foot was put on top of a block. These behaviors can be seen in the attached movie.

### Balancing on moving platforms

For our next experiment, we put the biped on a rolling platform and rotated and moved it with a rather fast change of directions. A typical behavior of the robot is plotted in Figure 3.8. The angular velocity and linear acceleration measured from the IMU that we attached to the rolling platform are plotted together with the CoG tracking error and momentum. Although the CoG is moving away from its desired



**Figure 3.8.:** The top three plots on the left-hand side show the CoG error and momentum when the robot is balancing on the rolling platform. The next two figures on the left-hand side plot the angular velocity and linear acceleration of the platform. The platform is displaced in Y direction with the robot facing the direction of the disturbance. The figures on the right-hand side show predicted and measured CoPs. Please refer to the text for a discussion of the results.

position when the platform is moving around, it remained bounded, the momentum was damped out and the robot kept standing and recovered CoG tracking. The stationary feet indicated that forces were applied that were consistent with our CoP boundaries. We notice from the figure that the CoPs, as they were predicted from the dynamics model, are approximately correct. However, one can expect that a higher precision might be needed to achieve dynamic motions which could be achieved with an inertial parameters estimation procedure [99].

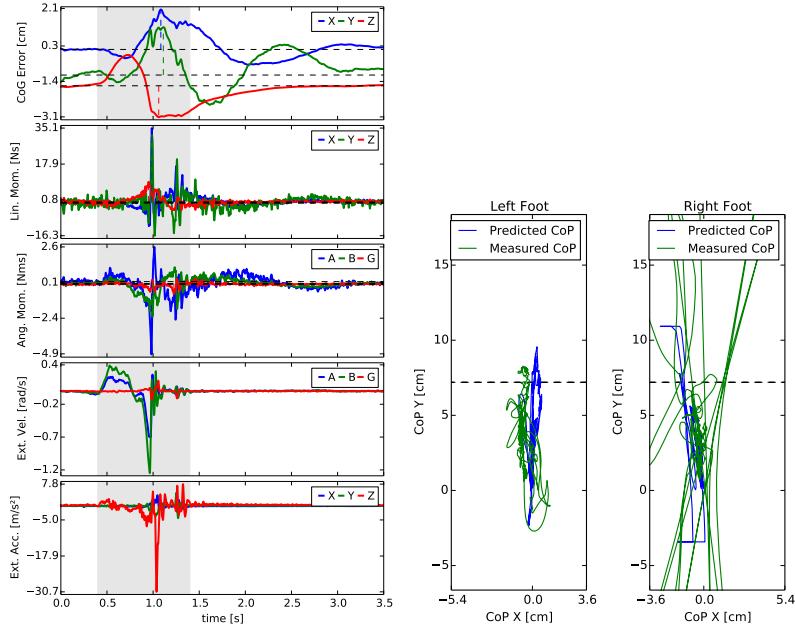
In an additional scenario, the biped was standing on a balancing board. We ran the experiment with two configurations for the robot: in one case the robot is standing such that the board motion happens in the sagittal plane and in the other case the motion happens in the lateral plane. In this case, the slope was varied in a range of  $[-2.8^\circ; 5.6^\circ]$  elevating the robot up to 6.9 cm. Even for quite rapid changes in the slope, the feet remained flat on the ground. Compared to the push experiment the CoPs were moving in a wider range, but still remained in the interior of the foot soles with a margin. In this case, we notice a discrepancy in the predicted contact forces and the real ones, making the case for the need for a better dynamics model. Eventually, we dropped the robot from the maximum height (Figure 3.9), such that the feet bounced off the ground at impact and tilted for a moment as can be seen in the video. Yet, the robot recovered and was still able to balance. We notice in the figure that in this case, the measured CoP of the foot that was lifted drastically differs from the predicted ones. We also note that the predicted CoPs reach their admissible boundary as is seen from the flat horizontal lines.

When we increased the pushes on the robot, eventually the momentum could not be damped out fast enough anymore and the robot reached a situation where the optimization could not find solutions that would balance the robot anymore (i.e. the slack variable associated to the desired momentum becomes very high) and the biped fell. In these cases the constraint that the feet have to be stationary was too restrictive. A higher level controller that takes into account stepping (e.g. [116, 145]) becomes necessary to increase the stability margin.

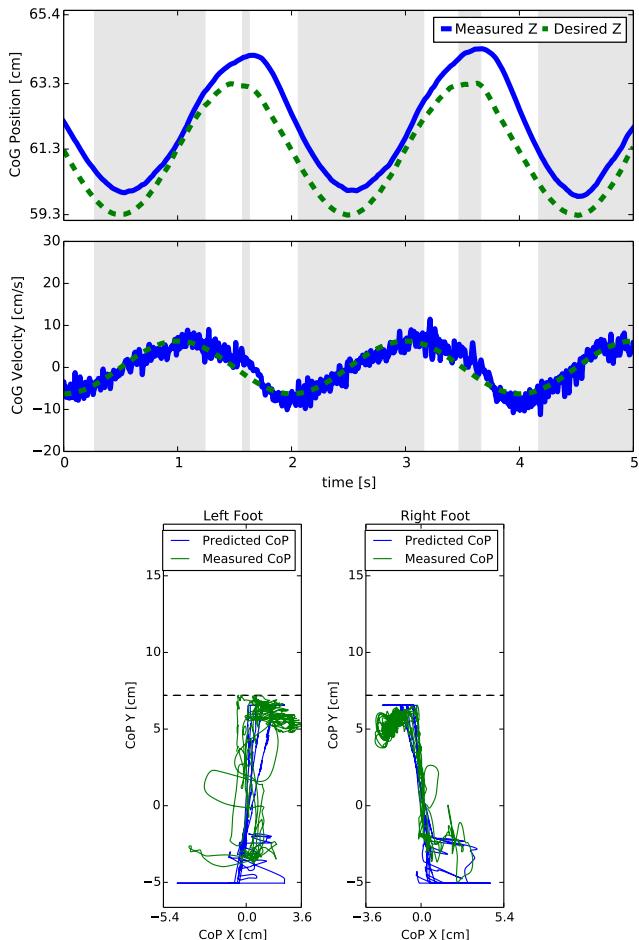
### I.5.3. Tracking Experiments in Double Support

In the next experiment the robot is performing a squatting like motion. We keep the same task hierarchy as in the balance experiments (see Table 3.2) and make the CoG track sine curves of 0.3 Hz and 0.5 Hz. The CoG is moving with an amplitude of 2 cm in the forward direction and an amplitude of 3 cm in the vertical direction. The results can be seen in the attached video.

In order to demonstrate the utility of the hierarchy, we setup a squatting experiment such that CoP constraints would be activated during the motion. We show the result of this experiment in Figure 3.10. We notice that the CoP constraints are active in most parts of the experiments. This constraint prevents the feet from tilting and the robot stays stable. We also notice that the real CoPs follow the predicted ones relatively well and stay inside the support polygon. As a consequence, the tracking of the CoG, which is in a lower priority, is not ideal but still achieves



**Figure 3.9.:** The top figure plots the robot CoG error and momentum when it was dropped from the highest point on the balancing board. In the bottom two plots one can see the angular velocity and linear acceleration of the balancing board, where we can identify the moment of impact with the ground at 1s. At that moment the right foot bounced off the ground and lost contact for an instance of time as can be seen in the measured CoP in the bottom plots and in the video. The admissible CoP hit the boundary and saturated in Y direction. Still, the robot was able to stabilize its feet and CoG and damp out the momentum.



**Figure 3.10.:** Tracking of the CoG in vertical direction during the squatting task when CoP constraint is active. The top two plots show the CoG desired and actual vertical positions and velocities. The grayed area corresponds to periods during which the CoP constraint is active. The bottom plot shows the evolution of the CoPs for both feet. The horizontal blue lines of the desired CoP correspond to an active constraint.

a reasonable performance. CoG velocity tracking is still achieved reasonably well. We note that the discrete activation/deactivation of the constraint is not directly visible on the CoG motion behavior.

This experiment illustrates the importance of hierarchies. By solving a QP without a hierarchy, there would be two possibilities, either the CoP constraint is set as a hard constraint of the optimization problem and there is no guarantees that a solution to the problem exists or it is put in the cost function with the CoG task and the solution is necessarily a trade-off between contact constraints and motion tasks. Exploiting a hierarchical setup, we are guaranteed to find a solution to the optimization problem and at the same time we are guaranteed that the CoG tracking task does not interfere with the CoP constraint.

#### I.5.4. Single Support Experiments

The experiments in the previous sections were done while the robot remained in double support. The goal of this experiment is to show that the controller can handle more complicated tasks involving contact switching and that the robot is able to balance on a single foot in face of disturbances. Further, we evaluate all capabilities in a single task: contact switching, push rejection in single support and tracking a leg motion.

First the robot moves from a double support position to a single support position where the swing foot is lifted 10 cm above the ground while balancing. This motion consists of 3 phases. First, the robot moves its CoG towards the center of the stance foot. Then an unloading phase occurs during which the contact force regularization enforces a zero contact force to guarantee a continuous transition when the double support constraint is removed. In the final phase, a task controlling the motion of the swing foot is added to the hierarchy. Our contact switching strategy is simple but guarantees that continuous control commands were sent to the robot. For more complicated tasks, such guarantees can always be met by using automatic task transitions such as in [69]. The composition of hierarchies is summarized in Table 3.4. Concerning computation time, the controller computes a solution in average well below 1ms but a maximum at 1.05ms is reached a few times during the unloading phase due to many constraints becoming active at the same time.

Once in single support, we pushed the robot to verify that it is balancing. Impacts were applied with peak forces up to 150 N and impulses between 4.5 Ns and 5.8 Ns. We saw a quick recovery of the CoG while the CoPs stayed bounded. In order to control the foot we used Cartesian position control (i.e. the swing foot task consists of a PD controller for the foot position in Cartesian space).

We repeat the experiment where the robot, once it is standing on one leg, swings its leg up and down tracking 0.25 Hz sine curves on the hip and knee flexion/extension joints (as can be seen in Figure 3.11). In this case, we swap the Cartesian foot control for a desired trajectory of the hip and knee in joint space (i.e. a task that consists of time-varying positions for both joints). Indeed, while in simulation Cartesian tracking is perfect, on the real robot the tracking performance of the Cartesian task of the swing foot is not satisfactory when moving at

Rank	Nr. of eq/ineq constraints	Constraint/Task
1	6 eq	Newton Euler Equation (3.13)
	$2 \times 14$ ineq	torque limits
2	$2 \times 4$ ineq	Center of Pressure, Sec. I.2.1
	$2 \times 4$ ineq	Friction cone, Sec. I.2.1
3	$2 \times 14$ ineq	joint acceleration limits, Sec. I.2.1
	6 eq.	Linear and angular momentum control
4	12/6 eq.	Contact constraints, Eq. (3.2)
	0/6 eq.	Foot motion (swing)
	14 eq.	PD control on posture
4	$2 \times 6/1 \times 6$ eq.	regularizer on GRFs
<b>DoFs:</b> 14		<b>max. time:</b> 1.05 ms

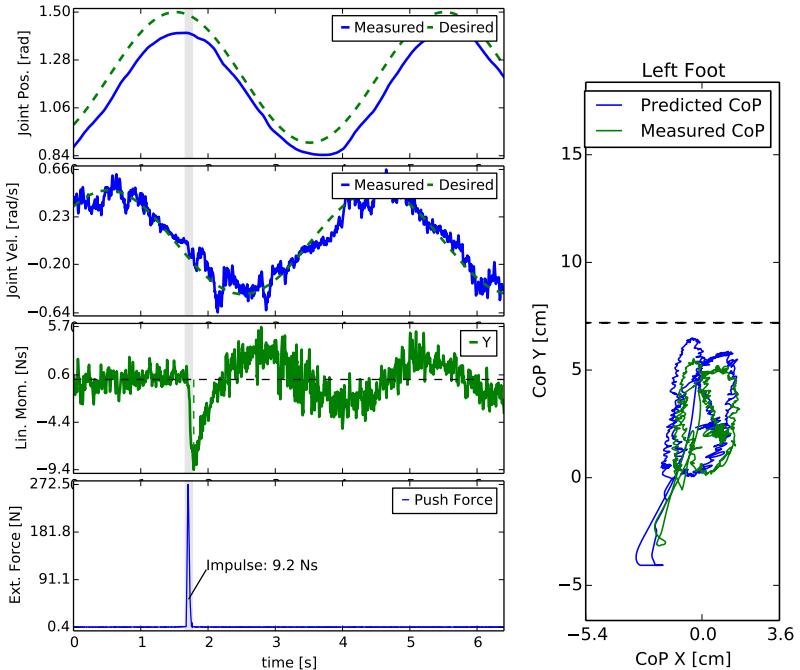
**Table 3.4.:** Hierarchy for Single Support experiments

higher speeds and amplitudes. We suspect that several reasons can explain the problem. One of the reasons seems to be due to the amount of noise present in the position sensors such that it is difficult to increase the position gains while still being able to damp the system. Since the feedback loop is closed around the foot position, which is estimated through forward kinematics, its velocity estimation seems to suffer from the cumulative noise of all the sensors. In this case, a direct joint control suffers less from that problem. Another problem could come from an insufficiently accurate model of the swing leg dynamics where unmodeled dynamics could become more dominant.

While the robot is performing the task, it is pushed strongly at the hip from the front as can be seen in the video. The joint tracking together with the linear momentum and the push force are shown in Figure 3.11. A spike in momentum can be seen at the moment of impact, but is damped and remains bounded. During the push, the CoP constraint is activated when the CoP comes close to the heel. Thanks to the inequality constraint, the foot does not start tilting and the robot can recover from the push. What is remarkable is that the swing leg tracking is barely affected although the push is comparable to the strongest impacts we applied in double support. It is worth mentioning again that the foot size of the robot is rather small compared to other humanoids.

## I.6. Discussion

In the following we discuss the results we presented and how they relate to other approaches.



**Figure 3.11.:** Tracking error on the position and velocity of the knee joint (left-hand side, top two plots) when the robot is standing on one leg and pushed with a peak force of 270 N (left-hand side, bottom plot). Although the linear momentum of the system (left-hand side, 3rd plot from top) was perturbed for a moment, joint tracking was barely affected. The bottom plot shows the CoP of the stance foot, which saturates close to the heel during the push, such that the foot does not start tilting.

### I.6.1. Task design and hierarchies

In our experiments, we exploit hierarchical separation of physical constraints and tasks, such as the robot dynamics and reaction force constraints and balancing tasks. This is guaranteed to always find a feasible solution while generating physically consistent solutions and generating admissible task dynamics as close as possible to the desired ones. As we have shown in the experiments, this is important to keep balance in cases when reaction force constraints conflict with lower priority tasks. Here, hierarchies can guarantee that performance in balancing is traded off, but the physical consistency of the solution is always guaranteed. Note that with a QP formulation this would not be possible.

From our experience, we prefer to keep posture tracking task and momentum control in the same priority and adapt importance by weights. It seems that the 14 DoFs of our robot are too limiting and do not leave enough freedom when posture control is put in a separate rank. Using a full humanoid with arms will increase the flexibility of the robot and allow for more hierarchy levels. We expect to have, for example, manipulation tasks in a higher priority than balancing tasks.

As mentioned in the experiment description we used only a small set of weights for our tasks. This already gives us a balancing performance that is at least as good as related work if not better. Better performance can be achieved by adjusting parameters and hierarchical setup more specific to tasks of interest [59]. It is important to note that in both contributions we do not use joint PD stabilization, but verify that the performance is solely the effect of the hierarchical inverse dynamics controller.

### I.6.2. Relation to other balancing approaches

The balance control strategy used in this paper is similar to the formulation of the momentum-based controller presented by Lee and Goswami [90, 91]. Our formulation has the great advantage of solving a single optimization problem instead of several ones and can therefore guarantee that the control law will be consistent with all the constraints (joint limits, accelerations, torque saturation, CoP limits and contact force limitations). As we have seen in the experiments, consistency with inequality constraints can be very important to improve robustness. Furthermore, we search over the full set of possible solutions and thus we are guaranteed to find the optimum, where [90, 91] optimize over sub-parts of the variables sequentially. It is also different from the work of [73] because we explicitly take into account contact forces in the optimization and not purely kinematics, which allows us to optimize the internal forces created by the contacts.

The balance controller is related to the work of [137]. In [137], the authors write the whole optimization procedure using Equation (3.1) with constraints similar to the ones we use. However, with the optimization problem being complicated, they actually solve a simpler problem where the contact forces are first determined and then desired accelerations and torques are computed through a least-square solution. From that point of view, torque saturation and limits on accelerations are not

accounted for. In our experiments, no tradeoff is necessary and we solve for all the constraints exactly. Further, the capability of strict task prioritization makes the design of more complicated tasks like balancing on one foot easier. Also, separating the EoM from kinematic contact constraints allows to keep solutions consistent with the dynamics even in postures where the feet cannot be kept stationary.

We have also shown in the paper that the use of a LQR design for the momentum task can simplify the controller design and improve robot performance by taking into account the coupling between linear and angular momentum. This design was particularly useful for the contact switching and single support task. Indeed, using the PD control approach, it was not possible to use the same gains in double or single support to regulate the CoG. With the LQR design, the gains for the momentum control were automatically computed using the same cost function and therefore no specific gain tuning was needed.

### I.6.3. Relations to other hierarchical inverse dynamics solvers

The implemented QP cascade is a combination of the two algorithms [31, 77]. We use a surjective Nullspace map  $\mathbf{Z}_r$  (cf. Equation 3.8), similar to [31]. However, in [31] inequality constraints are included only in the first priority, i.e.  $\mathbf{A}_1 = \dots = \mathbf{A}_r$ . Instead, the proposed method allows for prioritization among inequality constraints as it was done in [77]. This variant of QP cascades combines the benefits of both algorithms. On the one hand variables are eliminated from one QP to the other, resulting in faster solvable QPs. On the other hand, it allows for prioritization of inequality constraints, which we exploit e.g. to give more importance to hardware limitations than to contact constraints.

Although fast enough for our experiments on the lower part of our humanoid robot, the speed comparison in Section I.5.1 shows that a more efficient algorithm is required when we run feedback control on the full 25 DoF robot. A method that is tailored to solve inequality constrained hierarchical tasks was derived in [41]. With an active set method dedicated to solve prioritized inequality constraints, it can outperform QP cascades in terms of speed. The QP cascade used in this algorithm trades off computational efficiency to a simpler implementation, where the handling of inequality constraints is passed on to an off-the shelf QP solver. As the focus of this paper was the experimental evaluation of the problem formulation, a QP cascade with the employed modifications was sufficiently fast and relatively easy to implement. Another practical advantage of QP cascades is the easily implemented regularization term in Equation (3.5), which increases numerical robustness in face of task singularities as discussed in [77]. The approach of [41] might also be interesting because it would directly allow the use of warm-start techniques to speed up the computations. Warm starting the optimizer should significantly improve computation time since during most tasks the active set does not change much from one control step to the other. Any inverse dynamics approach using either QP cascades [95],[125] or Hierarchical QP algorithm might profit from the decomposition proposed in Section I.2.3, as it is not required to compute an SVD of the full equations of motion, but only of the last six rows.

## I.7. Conclusion

In this paper, we have presented experimental results using a hierarchical inverse dynamics controller. A variant of cascades of QPs was presented and implemented in a 1 kHz feedback-control loop. We used LQR to formulate momentum controllers for balancing and tracking tasks. Our main focus then was the experimental evaluation of the control framework on a torque controlled humanoid robot. In a series of experiments, we evaluated systematically the balancing and tracking capabilities of the robot. The humanoid showed a robust performance in single and double support and was able to recover from pushes and other disturbances. Our results suggest that the use of complete dynamic models and hierarchies of tasks for feedback control is a feasible approach, despite the model inaccuracies and computational complexity. For future work, we would like to integrate higher level planners to compose more complex tasks such as walking.

## Acknowledgements

We would like to thank Ambarish Goswami and Seungkook Yun for hosting us at the Honda Research Institute for one week and for their precious help in understanding the original momentum-based controller. We would also like to thank Ambarish Goswami and Sung-Hee Lee for giving us an early access to their publication. We are also grateful to Daniel Kappler for helping us with the videos. Last, but not least, we would like to thank the anonymous reviewers for their very valuable comments that helped improve the final version of the paper.

This research was supported in part by National Science Foundation grants IIS-1205249, IIS-1017134, CNS-0960061, EEECS-0926052, the DARPA program on Autonomous Robotic Manipulation, the Office of Naval Research, the Okawa Foundation, and the Max-Planck-Society. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding organizations.



## Paper II: “Trajectory generation for multi-contact momentum control”

### Authors

Alexander Herzog, Nicholas Rotella, Stefan Schaal, Ludovic Righetti

### Abstract

Simplified models of the dynamics such as the Linearized Inverted Pendulum Model (LIPM) have proven to perform well for biped walking on flat ground. However, for more complex tasks the assumptions of these models can become limiting. For example, the LIPM does not allow for the control of contact forces independently, is limited to co-planar contacts and assumes that the angular momentum is zero. In this paper, we propose to use the full momentum equations of a humanoid robot in a trajectory optimization framework to plan its center of mass, linear and angular momentum trajectories. The model also allows for planning desired contact forces for each end-effector in arbitrary contact locations. We extend our previous results on linear quadratic regulator (LQR) design for momentum control by computing the (linearized) optimal momentum feedback law in a receding horizon fashion. The resulting desired momentum and the associated feedback law are then used in a hierarchical whole body control approach. Simulation experiments show that the approach is computationally fast and is able to generate plans for locomotion on complex terrains while demonstrating good tracking performance for the full humanoid control.

Published in:

*Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pp. 874 - 880, 2015

Notes:

Presented at the IEEE-RAS International Conference on Humanoid Robots (2015)

## II.1. Introduction

Humanoid robots locomoting and performing manipulation tasks on uneven ground are required to actively apply forces on objects and the floor in order to achieve their task successfully. A direct effect of contact forces is a change of momentum in the robot, which on the one hand is necessary to move the center of mass, but on the other hand restricts the type of limb motion that the robot can perform. The nonlinear nature of the angular momentum dynamics makes preview-based control computationally hard in general and even with an admissible angular momentum trajectory open control parameters like the joint motion and feedback control remain problematic for use in a whole body control framework.

Successful applications of simplified momentum models have been shown on robots walking quite robustly over flat ground. A common approach is to use the linear inverted pendulum model (LIPM), which has been exploited for preview control since it was introduced by Kajita et. al [72]. In [55], a model predictive control (MPC) approach is formulated that finds foot steps on a flat ground together with a compatible CoM location on a horizontal plane. CoM profiles can then be realized on the full body together with other limb motion (e.g. swing leg) controllers [132], [135] [43], [59]. The assumption of a horizontal CoM motion and flat ground can be relaxed to a pre-designed CoM height profile [38], [7]. Although, less restrictive than the original LIPM, these approaches are either built for point feet or leave parts of the dynamics uncontrolled. Depending on the terrain, pre-defining the CoM along a fixed direction may result in suboptimal or infeasible reaction forces. Approaches that leave the regime of linear models have been proposed, for instance for long jumps [151] leading to more complex task behavior. However, they often require task specific models that for example take into account the swing leg dynamics. Going even further, we have seen work that optimizes over the whole joint trajectories and the full momentum together [30]. The impressive near-to physical motions, however, come at a high computational cost. Time-local controllers, have shown that balancing performance of robots is improved, when overall angular momentum is damped out directly [58, 91]. Nevertheless, it has remained unclear how angular momentum profiles should be chosen in more dynamic motions.

In this paper, we consider the full momentum dynamics of the robot for generation of CoM and momentum trajectories and the according reaction force profiles at each contact. We formulate the problem as a continuous-time optimal control problem in a sequential form. A mode schedule is predefined together with end effector trajectories. A desired angular momentum trajectory, which is required for the limb motion, is generated from a simple inverse kinematics forward integration and realized with admissible contact forces in a least squares optimal sense. Since the kinematics information is processed separate from the optimization over the dynamics, our optimization procedure is relatively fast (for example compared to [30]). In our previous work [58] we have proposed a method to compute control gains for the momentum task using a LQR design approach. We showed that it was able to significantly improve performance for balancing tasks on a real hu-

manoid. However, this approach was only used for stabilization. In this paper, we extend the approach to receding horizon tracking control. We use the planned trajectories to generate feedback gains using a LQR design where we linearize the non-linear momentum dynamics around the optimized trajectory. In a simulated stepping task, our humanoid traverses a terrain with changing height and angled stepping stones successfully. A whole-body controller computes joint torques that realize feedback loops on momentum as well as the swing leg motion and respect additional balancing and hardware constraints and generates physically realizable contact forces. A good tracking of overall momentum is achieved using our receding horizon LQR design.

This paper is organized as follows. In Sec II.2 we formulate an optimal control problem to obtain momentum trajectories of the robot. Resulting trajectories are controlled with a LQR feedback design as discussed in Sec II.3 which is then incorporated into a whole-body control approach as explained in Sec II.4. In Sec II.5 we demonstrate our control framework on a simulated stepping task on rough terrain. We discuss our results in Sec II.6 and finish with a conclusion in Sec II.7.

## II.2. Momentum trajectory optimization

In this section, we describe how the robot momentum is planned together with admissible contact forces. We will first discuss the reduced dynamics model which will then be used to phrase an optimal control problem to generate CoM and momentum profiles. At the end of the section we describe how a desired angular momentum is chosen for optimization.

### II.2.1. Dynamic model

We reduce the state of a robot to its center of mass (CoM)  $\mathbf{r}$  and overall momentum

$$\mathbf{h} = \begin{bmatrix} \mathbf{l} \\ \boldsymbol{\kappa} \end{bmatrix}, \quad (3.27)$$

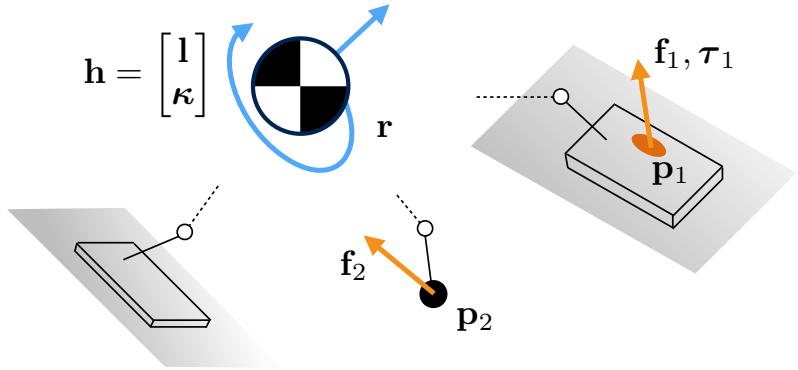
with linear  $\mathbf{l}$  and angular  $\boldsymbol{\kappa}$  momenta. The state of the system is controlled by forces  $\mathbf{f}_i$  and torques  $\boldsymbol{\tau}_i$  acting at points  $\mathbf{p}_i$  on the mechanical structure. The momentum dynamics as illustrated in Fig 3.12 can be written

$$M\dot{\mathbf{r}} = \mathbf{l} \quad (3.28)$$

$$\dot{\mathbf{l}} = M\mathbf{g} + \sum \mathbf{f}_i \quad (3.29)$$

$$\dot{\boldsymbol{\kappa}} = \sum \boldsymbol{\tau}_i + \sum (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i, \quad (3.30)$$

where  $M$  is the mass of the robot. The contact points  $\mathbf{p}_i$  can for instance be point foot locations that are touching the ground or they can represent the position of a handle that the robot holds on to. Flat feet can be modeled by several contact



**Figure 3.12.:** A sketch of the reduced model used through out this paper. It shows the center of mass (CoM)  $\mathbf{r}$  of the system and contacts with the environment. States and controls are color-labeled blue and orange respectively. Flat contacts that consist of a force, center of pressure and normal torque (here  $\mathbf{f}_i, \boldsymbol{\tau}_i, \mathbf{p}_i$ ) can be modeled (e.g. a hand touching a wall) as well as point contacts like  $\mathbf{p}_2$ . There is no restriction on the pose of contacts nor the CoM. Momenta generated around the CoM are not necessarily zero.

points on the foot surface. Equivalently, we can represent the effective force at the center of pressure (CoP). Depending on the type of contact, it will be necessary to express constraints on the forces. In the case of point feet, no torques  $\boldsymbol{\tau}_i$  can be generated. Torques that are generated at the CoP of a flat foot are required to be normal to the foot surface. Further, if we assume stationary contact points, we need to restrict the wrench to remain in a friction cone.

### II.2.2. Optimal control formulation

In the following we use the momentum dynamics of the robot to plan for admissible contact forces that generate desired (linear and angular) momentum trajectories. Our goal is to compute force and contact point trajectories  $\mathbf{f}_i(t), \boldsymbol{\tau}_i(t), \mathbf{p}_i(t)$  that satisfy the momentum dynamics in Eqs (3.28-3.30) and contact constraints at all time. These trajectories are also required to steer the momentum through desired states over a time horizon  $T$ . Given the initial state of the robot CoM and

momentum  $\mathbf{r}(0), \mathbf{l}(0), \boldsymbol{\kappa}(0)$ , we can integrate Eqs (3.28-3.30) to obtain

$$\mathbf{l}(t) = \int_0^t (\sum \mathbf{f}_i(\delta) + M\mathbf{g})d\delta, \quad (3.31)$$

$$\mathbf{r}(t) = \mathbf{r}_0 + \frac{1}{M} \int_0^t \mathbf{l}(\delta)d\delta, \quad (3.32)$$

$$\begin{aligned} \boldsymbol{\kappa}(t) &= \int_0^t (\sum \boldsymbol{\tau}_i(\delta) + \\ &\quad \sum (\mathbf{p}_i(\delta) - \mathbf{r}(\delta)) \times \mathbf{f}_i(\delta))d\delta, \end{aligned} \quad (3.33)$$

Note that the states are expressed as (nested) functions of the reaction forces. They are constructed by integrating, summing and applying the cross-product on  $\mathbf{f}_i(t), \boldsymbol{\tau}_i(t), \mathbf{p}_i(t)$ . Given a naive idea of what the desired CoM  $\mathbf{r}_{des}(t)$  and momentum  $\mathbf{h}_{des}(t)$  should be (e.g. coming from an initial kinematic plan), we want to find contact forces that minimize the error

$$\begin{aligned} J = \sum_{t_0}^T ( & \| \dot{\mathbf{l}}(t_i) \|_{W_1}^2 + \| \mathbf{l}(t_i) - \mathbf{l}_{des}(t_i) \|_{W_2}^2 + \\ & \| \mathbf{r}(t_i) - \mathbf{r}_{des}(t_i) \|_{W_3}^2 + \| \boldsymbol{\kappa}(t_i) \|_{W_4}^2 + \\ & \| \boldsymbol{\kappa}(t_i) - \boldsymbol{\kappa}_{des}(t_i) \|_{W_5}^2 ), \end{aligned} \quad (3.34)$$

where we compute the errors at  $t_i \in (0; T]$  and  $W_i$  are diagonal weighting matrices that allow trade-offs between the cost terms. Penalizing rates of momenta reduces undesired oscillations in the motion.

### Adaptive end effector location

Throughout the discussion in this paper we assume that the end effector moves between a series of predefined locations. However, this assumption can be relaxed without making the optimization problem harder. We simply substitute  $\mathbf{p}_i = \bar{\mathbf{p}}_i + \tilde{\mathbf{p}}_i$ , where  $\bar{\mathbf{p}}_i$  is the foot sole location that remains stationary throughout a contact phase and  $\tilde{\mathbf{p}}_i$  is the (time-varying) center of pressure inside of the foot sole. In this reformulation both,  $\bar{\mathbf{p}}_i$  and  $\tilde{\mathbf{p}}_i$ , will be optimization variables and are chosen automatically. The support planes still have to be decided before-hand, e.g. using a dedicated acyclic contact planer [23, 143].

### II.2.3. Optimization procedure

We will now formulate the described problem into an optimization problem. First, the reaction forces are formulated as weighted basis functions, more specifically as polynomials of the form

$$\mathbf{f}(t; \mathbf{w}) = \alpha(t) \sum_{k=0}^{N-1} w_k t^k = \Phi^T(t) \mathbf{w}, \quad (3.35)$$

$$\alpha(t) = \begin{cases} 1 & \text{if in contact at } t \\ 0 & \text{else} \end{cases}, \quad (3.36)$$

with basis functions  $t^k$  and weights  $w_i$  summarized in vectors  $\Phi(t)$ ,  $\mathbf{w}$ . We define the mode-scheduling variable  $\alpha(t)$  which specifies contact activation and deactivation and is set before-hand, for example it could be obtained by a higher level planner. Polynomials have the advantage of generating smooth force trajectories by construction. The forces can then be written as

$$f_i^j(t; \mathbf{w}_i^j) = \Phi^T(t) \mathbf{w}_i^j, \quad j \in x, y, z \quad (3.37)$$

$$\tau_i(t; \mathbf{v}_i) = \mathbf{n}_i \Phi^T(t) \mathbf{v}_i \quad (3.38)$$

$$p_i^j(t; \mathbf{u}_i^j) = \Phi^T(t) \mathbf{u}_i^j, \quad j \in x, y \quad (3.39)$$

where the subscript identifies the end effector, the superscript represents the coordinates and  $\mathbf{n}$  is the normal vector of the foot sole. Contact forces, torques and CoPs thus become linear functions of polynomial coefficients  $\mathbf{w}_i^j$ ,  $\mathbf{v}_i$ ,  $\mathbf{u}_i^j$ . Expressing the states in Eqs (3.31-3.34) with Eqs (3.37-3.39) substituted, gives us the states as functions of time and polynomial coefficients. At the core of operations required to carry out the result are multiplication, summation and integration of polynomials, which can be computed analytically. As a result, we phrased our optimal control problem in sequential form, where our controls evaluate directly to states and the dynamics equations (cf. Eqs (3.28-3.29)) are implicitly incorporated. This allows us to phrase our optimization problem

$$\min_{\mathbf{x}} J(\mathbf{x}) \quad (3.40)$$

$$\text{s.t. } |p^j(t_i; \mathbf{x})| \leq \hat{p}, \quad j = x, y \quad (3.41)$$

$$|\tau(t_i; \mathbf{x})| \leq \hat{\tau}, \quad (3.42)$$

$$0 \leq -|f^j(t_i; \mathbf{x})| + \mu f^z(t_i; \mathbf{x}) \leq \hat{f}_z, \quad j = x, y \quad (3.43)$$

$$t_i = t_0 \dots T$$

where  $\mathbf{x}$  is a concatenation of variables  $\mathbf{w}_i^j, \mathbf{v}_i, \mathbf{u}_i^j$ . We find polynomial coefficients  $\mathbf{x}$  that minimize the error cost  $J$  while satisfying friction and support bound constraints on forces. Eqs (3.41-3.42) bound the CoPs and torques. In Eq (3.43) we impose a friction cone constraint approximated as pyramid and upper bound it by a sufficiently large value  $\hat{f}_z$  to keep the polynomials from penetrating the

lower bound and escaping above. In our constraints in Eqs (3.41-3.43), we wish to have bounds that hold at all  $t \in [0, T]$ ; in practice however, we express them at a finite number of time steps. Given the limited flexibility of polynomials, we get minor penetration of those constraints in intervals  $(t_i; t_{i+1})$ . Note that all our constraints are linear, whereas the objective function has quadratic terms as well as higher order (non-convex) terms due to the costs on angular momentum. Since the dynamics are not further simplified, we have the benefit that all states and controls are included in the cost  $J$  and none of them are left uncontrolled.

### Receding Horizon

Given that the problem in Eq (3.40) is non-convex, we cannot expect to find a global optimum in general but need to find a local optimum starting from an initial guess  $\mathbf{x}_0$ . In our approach we use a receding horizon technique. We start out by solving our problem for a horizon  $\tilde{T} < T$  and obtain an optimal solution  $\mathbf{x}_{[0, \tilde{T}]}^*$ . Typically, we start our desired motion such that it is easy to solve in the interval  $[0, \tilde{T}]$ , i.e. the robot is standing still and applying gravity compensation is already optimal. Then, we formulate our problem for the interval  $[\Delta, \tilde{T} + \Delta]$  where we initialize our optimizer with the previous solution. This has the advantage that an initial solution can be bootstrapped from an initial easier-to-solve configuration and then moved over the horizon to the difficult parts of the motion (with non-zero momenta and contact switches). Further, as we push our implementation to run in real-time, we seek to use it in a receding horizon control setting.

### Desired Angular Momentum

Most simplified momentum models assume that angular momentum is desired to remain at zero. This, however, ignores the fact that momentum might be required to move the robot limbs. For example in a stepping task we need to swing a leg, which in turn generates a momentum around the hip. Thus, it is not trivial to decide what the desired angular momentum should look like. On the other hand, if we have given joint and base trajectories  $\mathbf{q}(t)$  of the full robot, we can compute the centroidal momentum  $\mathbf{h}(t) = \mathbf{H}_Q \dot{\mathbf{q}}(t)$  using the centroidal momentum matrix  $\mathbf{H}_Q$  [108]. The resulting  $\mathbf{h}(t)$  is required to realize the whole-body motion  $\mathbf{q}(t)$ . Thus, in order to obtain a desired (centroidal) momentum for our trajectory optimization, we generate whole-body trajectories from a simple inverse kinematics integration and phrase the resulting  $\mathbf{l}_{des}(t), \boldsymbol{\kappa}_{des}(t)$  into our optimization cost in Eq (3.34). For our inverse kinematics integration, we define two tasks:

- The robot is to track a CoM motion  $\mathbf{r}(t)$  obtained from the result of Eq (3.40) and
- endeffectors are to track trajectories in form of 3d splines between contact locations.

Although dynamically not feasible, this method generates angular momentum profiles required to perform the task-imposed motion, for example the momentum induced by swinging a leg.

The final planning procedure is an iteration between a) generating a whole-body motion with inverse kinematics and b) realizing the resulting momentum profiles with admissible forces by solving Eq (3.40). In our experiments two passes on this kino-dynamic planning procedure already lead to convergence, i.e. forward integrated and optimized trajectories do not differ significantly. With this iteration we not only generate admissible force profiles that obey the momentum dynamics equations but also we can bootstrap angular momentum trajectories which are not obvious to design otherwise. Note that neither CoM nor momentum are predefined in advance. Instead an initial guess is given and the final trajectories are found automatically.

### II.3. Momentum LQR

In the previous section we described how reference trajectories in accordance with the momentum dynamics are obtained. In order to track those trajectories on the full robot, we propose a feedback law using a LQR design, which has shown superior performance compared to a naive PD gain approach in experiments on the real robot [58]. From our trajectory optimizer we obtain admissible states and controls

$$\mathbf{y}^* = \begin{bmatrix} \mathbf{r} \\ \mathbf{l} \\ \boldsymbol{\kappa} \end{bmatrix}, \boldsymbol{\lambda}^* = \begin{bmatrix} \vdots \\ \mathbf{f}_i \\ \boldsymbol{\tau}_i \\ \vdots \end{bmatrix}, \quad (3.44)$$

$$\dot{\mathbf{y}}^* = f(\mathbf{y}^*, \boldsymbol{\lambda}^*) = \begin{bmatrix} \frac{1}{M}\mathbf{l} \\ M\mathbf{g} + \sum \mathbf{f}_i \\ \sum \boldsymbol{\tau}_i + \sum (\bar{\mathbf{p}}_i - \mathbf{r}) \times \mathbf{f}_i \end{bmatrix}, \quad (3.45)$$

where we transform wrenches  $\mathbf{f}_i, \boldsymbol{\tau}_i$  to the stationary poles  $\bar{\mathbf{p}}_i$ .

The dynamics function in Eq (3.45) is discretized and linearized around the desired trajectories  $\mathbf{y}^*, \boldsymbol{\lambda}^*$ . The resulting time-varying linear dynamics are then used to formalize a finite horizon LQR problem. This yields a control policy

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^* - \mathbf{K}_t(\mathbf{y} - \mathbf{y}^*) \quad (3.46)$$

with time-varying feedforward and feedback terms that map errors in states into contact wrenches. Controlling the momentum with this feedback law requires 6 DoF for wrenches at each contact. Since our whole-body controller incorporates additional control objectives and force constraints, we compute directly the momentum rate that the wrenches generate

$$\dot{\mathbf{h}}_{ref} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ [\bar{\mathbf{p}}_i - \mathbf{r}^*] \times & \mathbf{I}_{3 \times 3} & \dots \end{bmatrix} \mathbf{K}_t (\mathbf{y}^* - \mathbf{y}) + \dot{\mathbf{h}}^*, \quad (3.47)$$

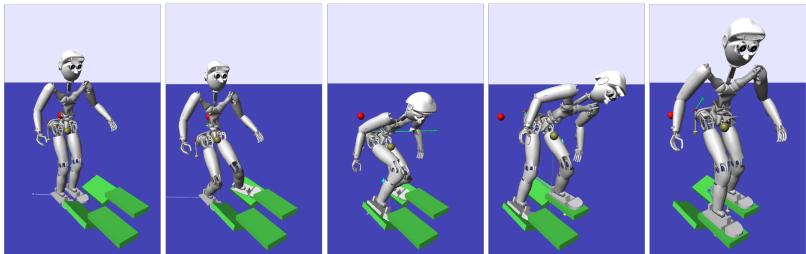
where  $[.] \times$  turns a cross-product into a matrix multiplication. The resolution of momentum rate to contact wrenches is then left to our whole-body controller as described in the next section. In this LQR design a quadratic performance cost is set once and then optimal gains are computed at each time step for the corresponding contact configuration. As we discussed in our previous work, in order to achieve compatible results with diagonal PD gain matrices, we had to design gains for different contact configurations, whereas the LQR design requires one performance cost and generated suitable gains automatically. In contrast to our previous work we linearize around desired trajectories, whereas in our robot experiments we used only two key configurations; this may become limiting in more versatile tasks such as walking over a rough terrain. In this paper, we focus on momentum trajectory generation and control. However, as we move towards robot experiments, we also address the issue of momentum estimation [124] from real robot sensors.

## II.4. Whole-body control

The trajectories generated with the model in Eqs (3.28-3.30) define the CoM, momentum and end effector forces of our humanoid. In order to track these trajectories on the full robot, we need to generate joint torques accordingly and at the same time control the limb motion and guarantee that other constraints are obeyed, e.g. joint limits. For this time-local control problem, we use inverse dynamics in QP Cascades, which we applied successfully on the real robot in previous work [59]. It allows us to phrase feedback controllers and constraints as functions of joint accelerations  $\ddot{\mathbf{q}}$ , external generalized forces  $\lambda$  and joint torques  $\tau$ . For instance, we can express a cartesian controller on the swing foot as an affine function of joint accelerations or a momentum controller as a function of reaction forces. The latter, for example, would be

$$\begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots \\ [\bar{\mathbf{p}}_i - \mathbf{r}] \times & \mathbf{I}_{3 \times 3} & \dots \end{bmatrix} \lambda + \begin{bmatrix} M\mathbf{g} \\ \mathbf{0} \end{bmatrix} = \dot{\mathbf{h}}_{ref}. \quad (3.48)$$

A reference momentum rate  $\dot{\mathbf{h}}_{ref}$  can be chosen from a LQR design as demonstrated in our previous work [58] and extended in Sec II.3. Given a set of controllers and tasks, we find torques that satisfy the dynamics equations of the full robot and at the same time generate the desired task feedback as good as possible. Tasks, however, may conflict, for instance moving the CoM may require moving the swing leg and vice versa. In these cases, QP cascades allow for two types of trade-offs; we can either weigh tasks against each other or we can prioritize them strictly.



**Figure 3.13.:** The humanoid robot traversing a terrain with stepping stones of different height and orientation.

Especially when it comes to trade-offs between constraints, tasks of interest and redundancy resolution, prioritization can facilitate successful control design.

## II.5. Simulation Results

This section describes our simulation results of the proposed control framework. We use a model of our Sarcos humanoid in the SL simulation environment. Contacts are simulated with a penalty method and stiff springs. All experiments are performed on a 2.7 GHz intel i7 processor with 16gb ram. A task is generated where the robot is to walk on stepping stones that increase in height from one step to the other as visualized in Fig 3.13 and shown in the attached video<sup>5</sup>. The z-axis of the inertial frame points up and the robot walks along the y-axis to the front. Two out of the four steps are tilted by 25°. Both the change in CoM height as well as the angled supports break the assumptions made in LIPM models and require the consideration of two separate contact wrenches. We generate swing leg trajectories using cubic splines to parameterize the pose of the foot. The humanoid starts out in double support at rest. After the first 3 seconds the left foot breaks contact and moves to an angled support surface located to the front left-hand side of the robot at an increased height (as shown in Fig 3.13). Then a contact switch occurs at every second, changing from single support to double support or vice versa. The second step is again a support surface angled inwards to the robot and located to the front right-hand side. Finally, the robot takes two steps onto a horizontal plateau located slightly below knee height.

The planner is initialized with a naive idea of the robot motion where we simply keep the base at a certain height above the feet. After integrating the desired swing foot positions together with the base height using inverse kinematics (as

---

<sup>5</sup>The video is also available on [www-amd.is.tuebingen.mpg.de/~herzog/15\\_07-Humanoids.mp4](http://www-amd.is.tuebingen.mpg.de/~herzog/15_07-Humanoids.mp4)

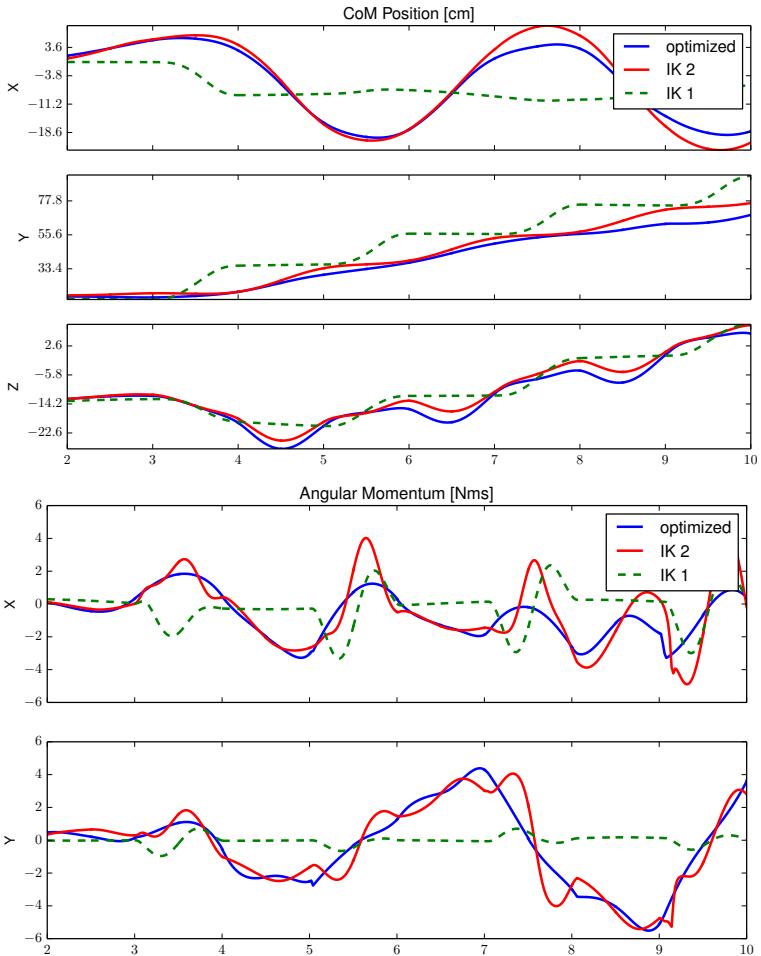
Rank	Nr. of eq/ineq constraints	Constraint/Task
1	6 eq	Newton Euler Equation
2	$2 \times 6$ eq	Contact constraints
	$2 \times 4$ ineq	Center of Presure
	$2 \times 4$ ineq	Friction cone
	$2 \times 25$ ineq	joint acceleration limits
3	6 eq	LQR momentum control
	6 eq	Cartesian swing foot control
	$25 + 6$ eq	PD control on posture
4	$2 \times 6$ eq	Contact forces control
5	3 eq	Base link orientation

**Table 3.5.:** Hierarchy of tasks in the stepping experiment

described in Sec II.2.3), we obtain resulting  $\mathbf{r}_{des}(t), \kappa_{des}(t)$  trajectories that take into account the angular momentum required to swing the legs from one stepping stone to the other. The inverse kinematics solution, which is physically not consistent, is then adjusted in the trajectory optimization step to be admissible with respect to constrained forces and CoPs. None of the force constraints were notably violated. As can be seen in Fig 3.14 the inverse kinematics-generated CoM trajectory is modified significantly in the lateral direction and as a consequence the angular momentum in the y-direction is modified as well. After a second iteration of inverse kinematics integration and trajectory optimization, the results have sufficiently converged and we stop. Polynomials of order 3 are chosen and initialized with zero. The planning process took 4 min and converged after only two iterations. This allows us to generate a complex motion rather quickly compared to motion planners that are based on more extensive models of the robot. The numerical optimization problem in Eq 3.40 is solved with SNOPT [51], a Sequential Quadratic Programming method.

Next, we construct a hierarchy of feedback controllers and constraints in order to realize the momentum profile on the full humanoid. At the highest priority we express the physical model of the full robot to obtain physically-consistent torques. This is followed by force and joint limits together with contact constraints. In the third priority we control the swing leg motion and the momentum and add a posture PD control with a relatively-low weight. In the priorities below, we regularize end effector forces and stabilize the base orientation. A summary of the task setup is given in Tab 3.5. The momentum is controlled with feedback gains designed as described in Sec II.3, where our performance cost is fixed for the whole run. We penalize state errors by 10, forces by 0.1 and torques by 0.5. Gains are generated over a 2sec horizon with a granularity of 200 time steps. A sequence of gains is recomputed every 10ms.

The robot was able to traverse the terrain successfully. The momentum trajec-



**Figure 3.14.:** Plans for the CoM and horizontal angular momentum. *IK 1*, *IK 2* show the profiles obtained from the inverse kinematics passes, whereas the blue graph is the final optimized trajectory. The resulting lateral CoM (top plot) was adapted quite significantly over the planning process to make it physically compatible with the reaction forces. Angular momenta are found that allow for stepping motions required to traverse the terrain.

ries, which were both dynamically consistent as well as compatible with the robots limb motions, could be tracked well as shown in Figs (3.17-3.16). Torques were generated that are in the bounds of the physical robot's torque limits. In the beginning of the task the planned CoM height forces the knees to stretch, which prevents the robot from using its knees to lift the CoM further, but instead it lifts the arms up rapidly. This can be avoided by adding a box constraint on the CoM in the optimization problem in order to consider kinematic limitations.

The LQR gain matrices have non-zero off-diagonal values as expected (cf. Fig 3.15). For instance, we can see that angular momentum is generated in order to correct for errors in CoM and linear Momentum, which would not be possible with diagonal PD gains. In fact, we tried to track the planned motion using diagonal PD gains. We started with values similar to the diagonal of the LQR gain, but we could not find parameters that were stable throughout several contact situations as was the case for the LQR gains. Increasing the terrain difficulty was mainly problematic due to kinematic limits because our naive swing leg trajectories required to keep the heel on the ground during the whole support phase, thus limiting the stepping height. Further, we noted problems with the simulator's contact model, which caused sporadic spikes in the force profile when we applied strong forces on the ground.

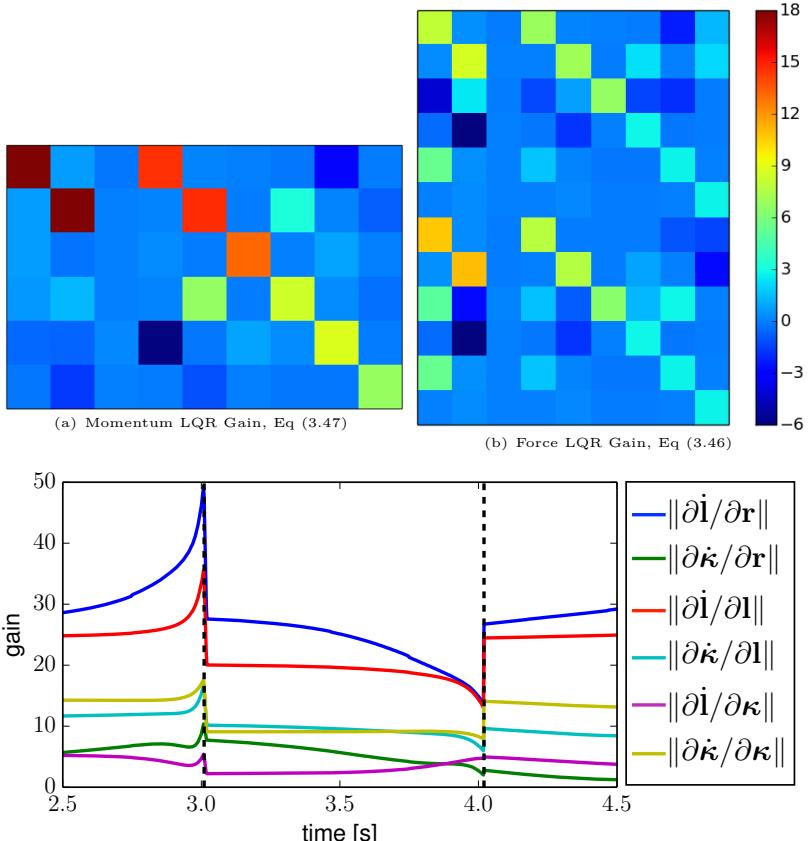
Overall, a complex task could be planned quickly including a non-trivial angular momentum profile that respected the end effector motion. The proposed feedback control law on the momentum showed good performance when it was embedded in an inverse dynamics task hierarchy.

## II.6. Discussion

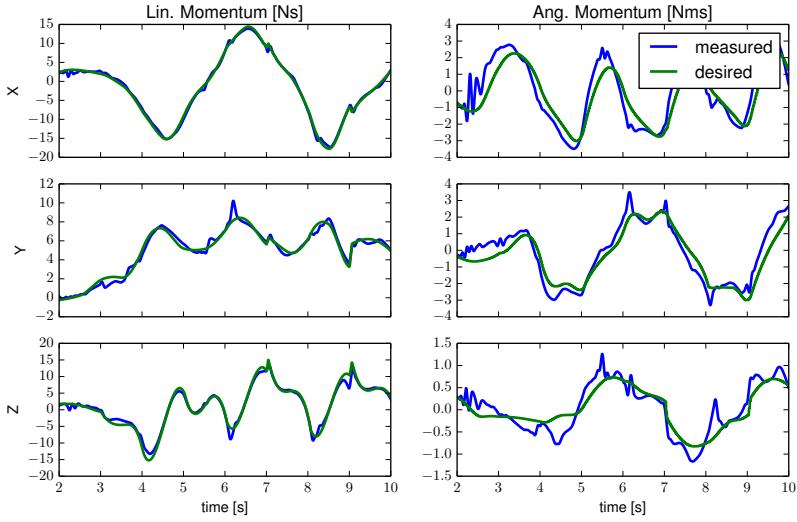
### Relation to simplified Momentum Dynamics

The momentum model in Eqs (3.28-3.30) is often simplified further in order to obtain linear dynamics, which then leads to computationally more efficient algorithms. However, turning multiplicative terms between variables in Eq (3.30) into linear terms requires potentially restrictive assumptions. E.g. in the LIPM the CoM height  $r_z$  is assumed to be constant, there is one effective  $\mathbf{p}_i$ , which lies in one horizontal plane, and the force is required to act along  $\mathbf{r} - \mathbf{p}$ .

Since a constant CoM height may be limiting for tasks that require vertical movement, the authors in [7] allow a predefined (not constant)  $r_z(t)$ . Substituting this assumption into Eq (3.30), and assuming constant  $\mathbf{p}_i$ , turns the horizontal angular momentum into a linear function of forces and torques. Assuming in addition that the nonlinear vertical angular momentum can be neglected, the authors end up again with linear dynamics. If we restrict our dynamics model further into a LIPM and use discretized dynamics (instead of polynomial trajectories) we can recover the approach of [55]. However, this can be restrictive if we want to step on various slopes or when a non-zero angular momentum is required, e.g. to consider limb motion.



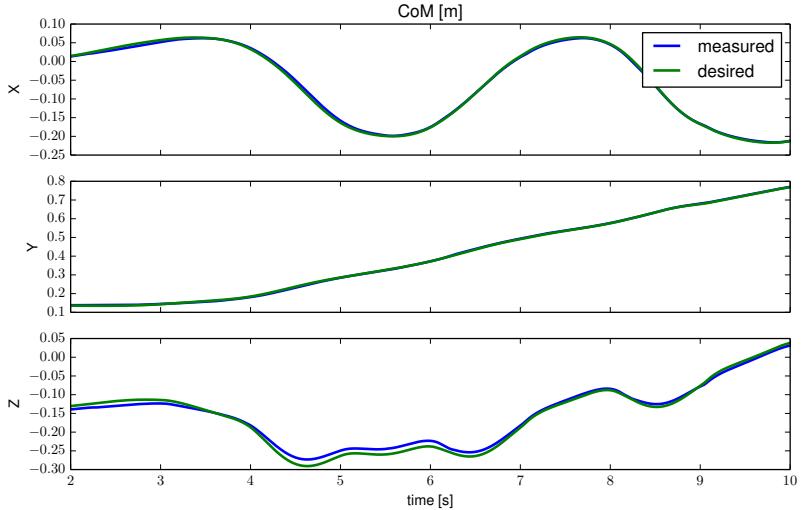
**Figure 3.15.:** Here we show a heat map visualization of (a) a momentum gain and (b) a force gain. The top 6 rows in (b) correspond to the wrench of the right foot, whereas the bottom rows correspond to the left foot. As we can see, the gains contain off-diagonal terms leading to coupling terms between linear and angular momentum. These terms are ignored in a naive diagonal PD gain design. The bottom plot shows the norm of 3x3 sub-blocks of the momentum gain plotted over time. The vertical dashed lines at  $t = 3$  and  $t = 4$  indicate contact switches. Gain profiles change significantly over time and contact configurations. The gains and momentum dynamics are discontinuous at contact switches. Nevertheless, discontinuities in joint torques were negligible.



**Figure 3.16.:** Linear and angular momentum are tracked well as the robot walks over the terrain. The oscillations in the angular momentum at  $t=2$  come from rapid arm motions when the robot was trying to move the CoM up and the knees were stretched. This can be avoided, e.g. by adding box constraints on the CoM in the trajectory optimization step to account for kinematic limits.

### Computation time

In our current work, we focused on the capabilities of the multi-contact dynamics model to generate dynamic motions on uneven terrain and we showed that these behaviors can be controlled on a full humanoid robot. However, the goal of separating the control process into a predictive control generation on a lower dimensional model and time-local control on the full dynamics has the potential for a fast implementation in a MPC fashion. In our experiments we saw potential drawbacks in our numerical optimization in Eq (3.40). Increasing the order of polynomials leads to slower convergence rates, whereas reducing the polynomial dimensions too far may lead to poor flexibility of the trajectory representation. This may be explained by the discrepancy of basis functions evaluated close to 0 and close to T leading to a poorly conditioned objective function. Piecewise constant trajectories may overcome this problem. Another point for improvement of the numerical procedure is the formulation of the optimal control problem. In the



**Figure 3.17.:** Measured and desired CoM of the robot when it was traversing the terrain. As the plots show, good tracking performance can be achieved with the proposed LQR design on the momentum.

current formulation, the objective function in Eq (3.34) has terms up to 4th order (squared norm of cross products). Our numerical solver is based on approximations up to second order, which may limit the region in which approximations are valid. It is possible (but out of the scope of this paper) to rewrite the problem into a (non-convex) Quadratically Constrained Quadratic Program where the objective function as well as the constraints are quadratic, leading to a better approximation in second order methods and potentially improving convergence. Pushing the implementation towards an online control algorithm is part of our future work.

## II.7. Conclusion

We presented an approach to control contact forces and momentum for humanoid robots. CoM and momentum profiles were obtained in an optimal control framework together with admissible contact forces. Feedback gains are generated from a LQR design, which generates time and contact-configuration dependent gains from a single performance cost. The resulting controller is embedded in an inverse dynamics-based whole-body controller together with other limb controllers

and constraints. We demonstrated the control framework on a simulation of the Sarcos humanoid traversing rough terrain. Physically admissible momentum and force trajectories could be found relatively quickly and were tracked well during the task execution. In future work, we will implement the discussed speed-up and push the optimal controller to run online.



## **Paper III: “Structured contact force optimization for kino-dynamic motion generation”**

### **Authors**

Alexander Herzog, Stefan Schaal, Ludovic Righetti

### **Abstract**

Optimal control approaches in combination with trajectory optimization have recently proven to be a promising control strategy for legged robots. Computationally efficient and robust algorithms were derived using simplified models of the contact interaction between robot and environment such as the linear inverted pendulum model (LIPM). However, as humanoid robots enter more complex environments, less restrictive models become increasingly important. As we leave the regime of linear models, we need to build dedicated solvers that can compute interaction forces together with consistent kinematic plans for the whole-body. In this paper, we address the problem of planning robot motion and interaction forces for legged robots given predefined contact surfaces. The motion generation process is decomposed into two alternating parts computing force and motion plans in coherence. We focus on the properties of the momentum computation leading to sparse optimal control formulations to be exploited by a dedicated solver. In our experiments, we demonstrate that our motion generation algorithm computes consistent contact forces and joint trajectories for our humanoid robot. We also demonstrate the favorable time complexity due to our formulation and composition of the momentum equations.

Published in:

*Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2703-2710, 2016

Award:

Finalist for the Best WBC-Paper Award 2016, IEEE-RAS Technical Committee on Whole-Body Control

Notes:

Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (2016)

### III.1. Introduction

Trajectory optimization and optimal control approaches have recently been very successful to control the locomotion of legged robots. Simplified linear models of the dynamics, usually variations of the linear inverted pendulum model [72],[7],[38],[55] have been widely used to compute the center of mass motion of legged robots, in particular biped robots. The linearity can be exploited to formulate optimal control problems that can be solved with quadratic programs, therefore allowing for fast computations of solutions in a model predictive control manner.

While these simplified models are useful, generalization to more complex situations where multiple non co-planar contact points might be desirable is limited. Moreover, they do not take into account how the generation of angular momentum affects the behavior of the robot, nor how individual contact points should be controlled to create a desired motion.

On the other hand, several contributions have shown that using the full dynamics model of the robot could be beneficial to generate even more complex behaviors [30],[92]. They have the advantage of explicitly taking into account the contact interactions with the environment and the dynamics of the robot. The problem with such approaches is that they usually require to solve non convex optimization problems which are computationally demanding. Several approaches have addressed the problem using variations of differential dynamic programming to compute optimal trajectories. The drawback of such algorithms is that they do not allow to easily add constraints on the state and controls. Other approaches formulate the problem as non-convex optimization problems, however in these cases the solvers used are generally off the shelf nonlinear solvers that do not exploit the structure of problem.

This paper addresses the problem of planning robot motion and interaction forces of a legged robot given a set of predefined contact points. In [60], the authors showed that such an approach could be taken to compute with a mild computational complexity trajectories that could be then directly included in whole-body control approaches and executed on a simulated humanoid. With a similar approach [26], interesting behaviors were demonstrated on a robot, confirming the interest in resolving such mathematical problems more efficiently.

In this paper, we analyze the structure of the problem induced by the physical model inherent of legged robots modeled through rigid body dynamics. In particular, we decompose the planning problem into two alternating optimization phases: first the contact forces and overall robot momentum are computed to satisfy dynamic constraints and in a second step the kinematic plan is resolved to satisfy the dynamic plan. Alternating these two phases lead to a locally optimal solution for the full kinodynamic plan.

We concentrate our efforts on the momentum optimal control problem. First, we show that by choosing the right representation of contact forces and momentum we can rewrite the problem as a Quadratically Constrained Quadratic Program for which a convex approximation of the constraints can be explicitly (and trivially) constructed. We then present two formulations, using either a simultaneous or

sequential optimal control formulation. We show how the sequential formulation of the problem can reduce the size of the problem while preserving its sparse structure for more efficient computation. Finally, preliminary numerical experiments demonstrate that exploiting this structure in a dedicated solver can significantly improve computational efficiency and quality of the solutions when compared to the same problem solved with a state of the art general purpose nonlinear solver.

The remainder of this paper is structured as follows. In Sec. III.2 we propose a motion generation algorithm that is based on optimization of the momentum equations. We write out different representations of contact forces in Sec. III.3 and discuss their algebraic properties. These properties are then exploited in Sec. III.4 to construct different variations of optimal control problems on the momentum equations of a floating-base robot. Then, in Sec. III.5 we show experiments with the proposed motion generation algorithm and conclude the paper.

## III.2. Kinodynamic Motion Generation

We are interested in the problem of motion generation for a robot in the presence of contact forces. In this section, we are going to write out an algorithm based on an optimal control formulation that is suited for motion generation in floating-base robots. Our underlying dynamics model is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau}_q + \mathbf{J}_e^T \boldsymbol{\lambda}, \quad (3.49)$$

where  $\mathbf{q}$  is the joint configuration,  $\mathbf{M}, \mathbf{N}$  the inertia matrix and nonlinear terms,  $\boldsymbol{\tau}_q$  are the actuation torques,  $\mathbf{S}^T$  the selection matrix,  $\mathbf{J}_e$  the end effector Jacobian and  $\boldsymbol{\lambda} = [\dots \mathbf{f}_e^T \ \boldsymbol{\tau}_e^T \ \dots]^T$  is the vector of forces  $\mathbf{f}_e$  and torques  $\boldsymbol{\tau}_e$  acting at end effector  $e$ . The equations of motion of a floating-base robot consist of two parts, the manipulator dynamics describing the torque at each joint, and the 6 rows of Newton-Euler equations that describe the change of the overall momentum in the system. The dynamics can be separated into the 6 rows of Newton-Euler equations and the actuated part as follows

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{H}}(\mathbf{q})\dot{\mathbf{q}} = \quad (3.50)$$

$$\begin{aligned} & \left[ \sum \boldsymbol{\tau}_e + \sum (\mathbf{x}_e(\mathbf{q}) - \mathbf{x}_{CoM}(\mathbf{q})) \times \mathbf{f}_e \right] \\ & \bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_q + \bar{\mathbf{J}}_e^T \boldsymbol{\lambda}, \end{aligned} \quad (3.51)$$

with the contact location  $\mathbf{x}_e(\mathbf{q})$  and center of mass (CoM)  $\mathbf{x}_{CoM}(\mathbf{q})$  computed through forward kinematics.  $\mathbf{H}(\mathbf{q})$  is the centroidal momentum matrix that maps  $\dot{\mathbf{q}}$  onto the overall linear and angular momentum of the system [108].  $\mathbf{M}, \bar{\mathbf{N}}, \bar{\mathbf{J}}_e^T$  represent quantities from Eq. (3.49) corresponding to the actuated joints.

From Eqs. (3.50-3.51), we observe that the planning process can be decomposed in

two steps. First, a vector of joint trajectories  $\mathbf{q}(t)$  is found together with contact force profiles  $\boldsymbol{\lambda}(t)$  that satisfy Equation (3.50). Then, suitable torques can be computed readily from Equation (3.51) assuming that sufficient joint torques can be provided by the actuators. In the following, we will discuss an optimal control approach that finds joint trajectories  $\mathbf{q}(t)$  together with force profiles  $\boldsymbol{\lambda}(t)$  that satisfy Eqs. (3.50-3.51). Our motion generation algorithm will be phrased as an optimal control problem of the form

$$\min_{\mathbf{q}, \mathbf{h}, \boldsymbol{\lambda}, \mathbf{c}} \quad J(\mathbf{q}, \mathbf{h}, \boldsymbol{\lambda}, \mathbf{c}) = \sum_t J_t(\mathbf{q}_t) + J'_t(\mathbf{h}_t, \boldsymbol{\lambda}_t, \mathbf{c}) \quad (3.52)$$

$$\text{s.t.} \quad \mathbf{c}_{e,t} = \mathbf{x}_e(\mathbf{q}_t) \quad (3.53)$$

$$\mathbf{h}_t = \begin{bmatrix} \mathbf{x}_{CoM}(\mathbf{q}_t) \\ \mathbf{H}(\mathbf{q}_t)\dot{\mathbf{q}}_t \end{bmatrix} \quad (3.54)$$

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \Delta f_t(\mathbf{h}_t, \boldsymbol{\lambda}_t) \quad (3.55)$$

$$\boldsymbol{\lambda}_t, \mathbf{c}_t \in \mathcal{S}_t \quad (3.56)$$

where we express objective functions on kinematic quantities  $J_t(\mathbf{q})$  as well as on momentum and contact locations  $J'_t(\mathbf{h}_t, \boldsymbol{\lambda}_t, \mathbf{c})$ . For instance could  $J_t(\mathbf{q})$  enforce an end effector motion from one contact location to another. Although contact locations  $\mathbf{c}_{e,t}$  and momentum  $\mathbf{h}_t$  can be written as functions of  $\mathbf{q}$  explicitly, we introduce redundant variables in order to simplify the optimization process as will become clear shortly. Eqs. (3.53-3.54) state explicitly how contact locations and momentum relate through forward kinematics. The momentum equations of our robot model (3.50) are then expressed as constraint in Eq. (3.55) with discretization parameter  $\Delta$ .  $\mathcal{S}$  is the set of feasible contact forces as will be discussed in more details later. The use of redundant variables allows us to decompose the overall problem into two better structured mathematical programs that allow the application of better informed solvers. The two sub-problems will be solved iteratively until convergence resulting in a solution for the original optimization problem. The first sub-problem is defined as

$$\min_{\mathbf{h}, \boldsymbol{\lambda}, \mathbf{c}} \quad \sum_{t,e} ||\mathbf{h}_t - \bar{\mathbf{h}}_t||^2 + ||\boldsymbol{\lambda}_t - \bar{\boldsymbol{\lambda}}_t||^2 + ||\mathbf{c}_t - \bar{\mathbf{c}}_t||^2 \quad (3.57)$$

$$\text{s.t.} \quad \mathbf{h}_{t+1} = \mathbf{h}_t + \Delta f_t(\mathbf{h}_t, \boldsymbol{\lambda}_t) \quad (3.58)$$

$$\boldsymbol{\lambda}_t, \mathbf{c}_t \in \mathcal{S}_t$$

This problem is optimized over the momentum and contact forces only. The resulting force and momentum profiles are dynamically consistent (i.e. they satisfy dynamics constraints (3.58)) and are as close as possible to reference profiles  $\mathbf{h}, \boldsymbol{\lambda}, \mathbf{c}$ . However, the joint state  $\mathbf{q}$  is ignored in Problem (3.57). Objectives on

kinematic quantities are considered in the kinematic sub-problem

$$\begin{aligned} \min_{\mathbf{q}} \sum_t J_t(\mathbf{q}_t) + & \left\| \begin{bmatrix} \mathbf{x}_{CoM}(\mathbf{q}_t) \\ \mathbf{H}(\mathbf{q}_t)\dot{\mathbf{q}}_t \end{bmatrix} - \bar{\mathbf{h}}_t \right\|^2 \\ & + \left\| \mathbf{x}_e(\mathbf{q}_t) - \bar{\mathbf{c}}_{e,t} \right\|^2 \end{aligned} \quad (3.59)$$

where additionally to optimizing  $J_t(\mathbf{q}_t)$ , we put constraints on momentum and contact location (cf. Eqs. (3.53-3.54)) from the original problem into the cost of our kinematic sub-problem in form of soft constraints. We will solve the original Problem (3.52) by iteratively solving problems (3.57), (3.59) and use the reference trajectories to enforce consistency between the two independent sub-problems.

---

**Algorithm 1** Momentum-centric motion generation

---

```

initialize  $\bar{\mathbf{h}}, \bar{\lambda}, \bar{\mathbf{c}}$ 
repeat
    solve Problem (3.59)
     $\bar{\mathbf{h}}, \bar{\mathbf{c}}_e := \begin{bmatrix} \mathbf{x}_{CoM}(\mathbf{q}_t) \\ \mathbf{H}(\mathbf{q}_t)\dot{\mathbf{q}}_t \end{bmatrix}, \mathbf{x}_e(\mathbf{q}_t)$ 
    solve Problem (3.57)
     $\bar{\mathbf{h}}, \bar{\mathbf{c}}_e := \mathbf{h}, \mathbf{c}_e$ 
until Solution of sub problems (3.59),(3.57) do not change

```

---

Each of the sub-problems minimizes parts of the original objective function and satisfies a subset of the original constraints. Additionally, the resulting momentum profiles from one sub-problem are optimized for consistency inside of the other up until convergence. With the proposed decomposition we have to solve two easier problems, where Problem (3.59) is an unconstrained optimization that has no notion of contact forces. Similar formulations, usually without costs on momentum, are often addressed in trajectory optimization for manipulators [75]. Problem (3.57) is a non-convex, but a well structured optimal control problem as we will discuss in the remainder of this paper.

### III.3. Reaction Force Representations

It is well known that the (linear and angular) momentum of a system can only be changed by external forces. In this section, we will discuss forces acting at contact points on a robot structure and we will show how the choice of representation of contact forces results in different mathematical programs with beneficial properties to be exploited by optimization algorithms.

### III.3.1. Contact Forces

A force  $\mathbf{f}_e$  and torque  $\boldsymbol{\tau}_e$  acting at contact point  $\mathbf{p}_e$  can be represented equivalently as a force and torque acting at the CoM  $\mathbf{r}$  where the torque acting at  $\mathbf{r}$  transforms to

$$\boldsymbol{\kappa}_e = \boldsymbol{\tau}_e + (\mathbf{p}_e - \mathbf{r}) \times \mathbf{f}_e \quad (3.60)$$

and  $\mathbf{f}_e$  remains the same. The change of momentum can then be expressed equivalently using either of the two representations as

$$\begin{aligned} \dot{\mathbf{h}} &= \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\ell} \\ \dot{\mathbf{k}} \end{bmatrix} = f(\mathbf{h}, \mathbf{f}, \boldsymbol{\kappa}) = \begin{bmatrix} M\mathbf{g} + \sum_e^{\frac{1}{M}\ell} \alpha_e \mathbf{f}_e \\ \sum_e \alpha_e \boldsymbol{\kappa}_e \end{bmatrix} \\ &= \begin{bmatrix} M\mathbf{g} + \sum_e^{\frac{1}{M}\ell} \alpha_e \mathbf{f}_e \\ \sum_e \alpha_e \boldsymbol{\tau}_e + \sum_e \alpha_e (\mathbf{p}_e - \mathbf{r}) \times \mathbf{f}_e \end{bmatrix} \end{aligned} \quad (3.61)$$

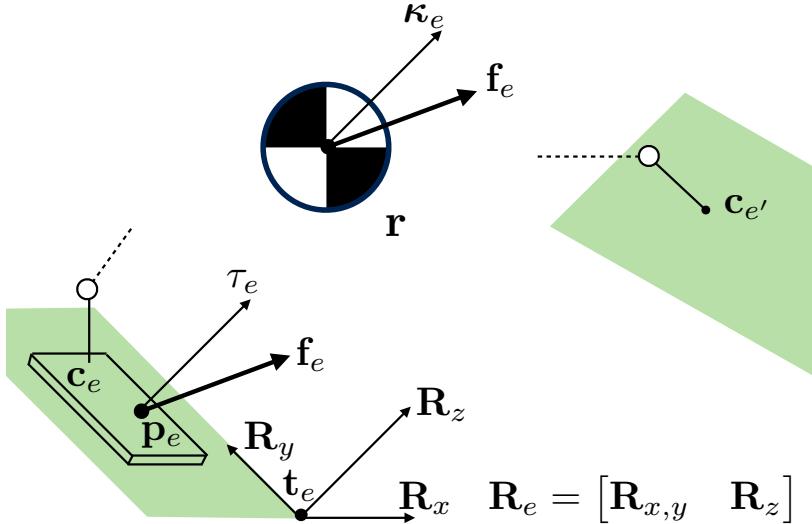
$$\alpha_e = \begin{cases} 1, & \text{if } e \text{ is in contact} \\ 0, & \text{else} \end{cases} \quad (3.62)$$

where  $\ell, \mathbf{k}$  are the linear and angular momentum. From Eq. (3.61) we can see that depending on how we chose the point of action the dynamics are either linear or not. One might be tempted to say that the linear dynamics lead to easier mathematical programs, which however is not the case when contact constraints have to be considered. Unilateral contacts, e.g. a hand pushing against a wall require that  $\mathbf{f}_e$  remains in a friction cone. On the other hand, point contacts cannot generate torques, i.e.  $\boldsymbol{\tau}_e = \mathbf{0}$ . In both cases, the center of pressure  $\mathbf{p}_e$  at which  $\mathbf{f}_e$  is acting (effectively) needs to reside inside of a support polygon or at the point of contact  $\mathbf{c}$ . We will now derive constraints on contact forces that generalize across various types of contacts, such as contact between surfaces (for instance a flat foot on the ground) or point contacts (for instance an elbow pushing on the table). For the following derivation, we define a contact surface for each contact  $e$  as shown in Fig. 3.18. Each contact is located on a surface described by a rotation matrix  $\mathbf{R}_e = [\mathbf{R}_{e,x,y} \quad \mathbf{R}_{e,z}]$  and translation  $\mathbf{t}_e$ . We can then represent forces, torques and locations in local coordinates as follows where we drop the index  $e$  for better readability

$$\mathbf{f} = \mathbf{R}\hat{\mathbf{f}}, \quad \boldsymbol{\kappa} = \mathbf{R}\hat{\boldsymbol{\kappa}}, \quad \boldsymbol{\tau} = \mathbf{R}_z\hat{\boldsymbol{\tau}}, \quad (3.63)$$

$$\mathbf{p} = \mathbf{R}_{x,y}\hat{\mathbf{p}} + \mathbf{t}, \quad \mathbf{c} = \mathbf{R}_{x,y}\hat{\mathbf{c}} + \mathbf{t}, \quad (3.64)$$

$$\hat{\boldsymbol{\tau}} \in \mathbb{R}, \quad \hat{\mathbf{p}}, \hat{\mathbf{c}} \in \mathbb{R}^2, \quad \hat{\mathbf{f}}, \hat{\boldsymbol{\kappa}} \in \mathbb{R}^3 \quad (3.65)$$



**Figure 3.18.:** A summary of notation used throughout the discussion of the momentum equations.

where the hat identifies local representations. We can now write contact constraints as

$$-\tau_{max} \leq \hat{\tau} \leq \tau_{max} \quad (3.66)$$

$$-\mathbf{p}_{max} \leq \hat{\mathbf{p}} - \hat{\mathbf{c}} \leq \mathbf{p}_{max} \quad (3.67)$$

$$-\mu \hat{f}_z \leq \hat{f}_x, \hat{f}_y \leq \mu \hat{f}_z, \quad (3.68)$$

where  $\mu$  is a friction parameter of the surface and  $\mathbf{p}_{max}$  is a vector of support limits and  $\tau_{max}$  is a bound on torques that are normal to the contact surface. If we want to describe a point foot, we set  $\tau_{max} = 0$ ,  $\mathbf{p}_{max} = \mathbf{0}$ . A contact between surfaces, e.g. floor and foot sole, allow for a wider range of  $\hat{\tau}, \hat{\mathbf{p}}$ . In Eq. (3.68) we restrict the force to remain in a pyramid which is used as an approximation for a friction cone. Note that in Eq. (3.67) we consider the support polygon to be rectangular for a clear notation, however, our derivation holds for more general polyhedra as well. Our contact model can be refined further [24] without loss of the structure as derived in the remainder of the paper. As expressed in Eqs. (3.66-3.68) contact

constraints are linear if we express them as functions of  $\mathbf{f}_e, \boldsymbol{\tau}_e$  acting at  $\mathbf{p}_e$ . On the other hand, if we chose to represent forces and torques as  $\mathbf{f}_e, \boldsymbol{\kappa}_e$  acting at the CoM  $\mathbf{r}$ , we will end up with non-linear contact constraints. In both cases, either the constraints (3.67) become nonlinear or the momentum equations (3.61) do. In the following, we will investigate the type of non-linearity that we introduce in Eqs. (3.61), (3.67) and how we can exploit it for development of optimization algorithms.

### III.3.2. Algebraic categorization

Depending on the choice of representation of the momentum dynamics (force acting at  $\hat{\mathbf{p}}$  or at  $\mathbf{r}$ ) the resulting optimization problem will be constructed from different sets of linear or non-linear functions. We are interested in tracking the algebraic properties of these functions as we construct mathematical programs using the dynamics and constraints from Eqs. (3.61), (3.67). Resulting objective functions and constraints will be classified as elements from the following sets of functions.

**Definition 1.** *We define the set of affine functions*

$$\begin{aligned} \mathcal{A} = \{ \mathbf{s} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \mid \\ \exists \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{a} \in \mathbb{R}^m : \\ \mathbf{s}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{a} \} \end{aligned}$$

**Definition 2.** *We define the set of positive semi definite quadratic functions*

$$\begin{aligned} \mathcal{Q}_+ = \{ \mathbf{s} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \mid \\ \exists \mathbf{Q}_i \in \mathbb{R}^{n \times n} \text{ p.s.d, } \mathbf{q}_i \in \mathbb{R}^n, c_i \in \mathbb{R} : \\ s_i(\mathbf{x}) = \mathbf{x}^T \mathbf{Q}_i \mathbf{x} + \mathbf{q}_i \mathbf{x} + c_i, i = 1 \dots m \} \end{aligned}$$

**Definition 3.** *We define the set of differences of positive semi definite functions*

$$\begin{aligned} \mathcal{Q}_{\pm} = \{ \mathbf{s} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \mid \\ \exists \mathbf{s}' \in \mathcal{Q}_+, \mathbf{P}_i \in \mathbb{R}^{n \times n} \text{ p.s.d:} \\ s_i(\mathbf{x}) = \mathbf{s}'(\mathbf{x}) - \mathbf{x}^T \mathbf{P}_i \mathbf{x}, i = 1 \dots m \} \end{aligned}$$

Throughout this paper we consider functions to be elements of  $\mathcal{Q}_{\pm}$ , only if the quadratic parameter matrices  $\mathbf{Q}_i, \mathbf{P}_i$  can be accessed separately (instead of only having access to the indefinite matrix  $\mathbf{Q}_i - \mathbf{P}_i$ ).

It is worth noting that  $\mathcal{Q}_{\pm}$  is closed under addition, scalar multiplication and composition with functions from  $\mathcal{A}$ , i.e.

$$\begin{aligned} \mathbf{s}, \mathbf{u} \in \mathcal{A}; \mathbf{v}, \mathbf{w} \in \mathcal{Q}_{\pm}; \beta \in \mathbb{R} \\ \Rightarrow (\beta(\mathbf{v} \circ \mathbf{s}) + (\mathbf{u} \circ \mathbf{w})) \in \mathcal{Q}_{\pm} \end{aligned} \quad (3.69)$$

Specifically, if we have quadratic parameter matrices  $\mathbf{Q}_i, \mathbf{P}_i$  constructed separately then the operations in Eq. (3.69) preserve this separation.

As can be noticed from Eqs. (3.61), (3.67) the nonlinearity introduced in the equations stems from cross products which can be classified as follows

**Theorem 1.** *The function  $\times(\mathbf{a}, \mathbf{b}) : \mathbb{R}^6 \rightarrow \mathbb{R}^3$ ,  $\times(\mathbf{a}, \mathbf{b}) = \mathbf{a} \times \mathbf{b}$  is an element of  $\mathcal{Q}_{\pm}$  and we can write out the corresponding matrices  $\mathbf{Q}_i, \mathbf{P}_i$  explicitly.*

*Proof.* It is straightforward to write a general cross product in a quadratic form with  $[\mathbf{a} \times \mathbf{b}]_i = [\mathbf{a}^T \quad \mathbf{b}^T] \mathbf{H}_i [\mathbf{a}^T \quad \mathbf{b}^T]^T, i = 1, 2, 3$ . Using an eigenvalue decomposition (for symmetric real matrices), we can then obtain  $\mathbf{H}_i = \mathbf{V}_i (\mathbf{D}_{+,i} - \mathbf{D}_{-,i}) \mathbf{V}_i^T$ , with  $\mathbf{D}_{+,i}, \mathbf{D}_{-,i}$  diagonal matrices with *only non-negative* values. Clearly, then  $\mathbf{a} \times \mathbf{b} \in \mathcal{Q}_{\pm}$  with  $\mathbf{Q}_i = \mathbf{V}_i \mathbf{D}_{+,i} \mathbf{V}_i^T, \mathbf{P}_i = \mathbf{V}_i \mathbf{D}_{-,i} \mathbf{V}_i^T$ .  $\square$

Note that constructing the quadratic parameter matrices  $\mathbf{Q}_i, \mathbf{P}_i$  for a general cross product can be carried out offline as it does not require any problem data. Through out the construction in the remainder of this paper, we will not have to perform any eigenvalue decomposition as this is typically costly. Instead, we exploit Eq. (3.69) in order to construct functions from  $\mathcal{Q}_{\pm}$ .

It is straightforward to show that Eqs. (3.66-3.68) can be expressed as  $\hat{\mathbf{s}}(\hat{\mathbf{f}}, \hat{\mathbf{p}}, \hat{\tau}) \leq 0$ , with  $\hat{\mathbf{s}} \in \mathcal{A}$ , i.e. if we choose to represent contact forces at  $\hat{\mathbf{p}}$  then the contact constraints are affine. On the other hand, the angular momentum (3.61) will result in a non-linear function. In fact, we will show that  $\hat{\mathbf{k}}(\hat{\mathbf{f}}, \hat{\mathbf{p}}, \hat{\tau}) \in \mathcal{Q}_{\pm}$  and we can construct the parameters  $\mathbf{Q}_i, \mathbf{P}_i$  separately inexpensively. On the other hand, if we chose to represent the force torque at the CoM, we have the reverse effect, i.e. the angular momentum becomes an affine function of the force-torques and as we will show the contact constraints will be expressed as  $\mathbf{s}(\mathbf{f}, \boldsymbol{\kappa}) \leq 0$ ,  $\mathbf{s} \in \mathcal{Q}_{\pm}$ .

### III.3.3. Decomposition of centroidal momentum dynamics

In this section, we will categorize the nonlinear functions in Eqs. (3.61), (3.66-3.68) into one of the categories defined in Sec. III.3.2. Categorizing the CoM torque  $\boldsymbol{\kappa}(\hat{\mathbf{f}}, \hat{\tau}, \hat{\mathbf{p}})$  (cf. Eq. (3.60)) requires consequent usage of Eq. (3.69) and Theorem (1). As a result one can show that  $\boldsymbol{\kappa}(\hat{\mathbf{f}}, \hat{\tau}, \hat{\mathbf{p}}) \in \mathcal{Q}_{\pm}$  and thus  $\hat{\mathbf{k}}(\hat{\mathbf{f}}, \hat{\tau}, \hat{\mathbf{p}}) \in \mathcal{Q}_{\pm}$ . Further, we will show that choosing to express external forces at the CoM will result in contact constraints that are expressed as inequalities on functions from  $\mathcal{Q}_{\pm}$ . Assuming a unilateral contact, we can write the torque acting at the center

of pressure as

$$\begin{aligned}\hat{\tau} &= \begin{bmatrix} 0 \\ 0 \\ \hat{\tau} \end{bmatrix} = \hat{\kappa} + (\hat{\mathbf{r}} - \begin{bmatrix} \hat{\mathbf{p}} \\ 0 \end{bmatrix}) \times \hat{\mathbf{f}} \\ &= \mathbf{R}^T \boldsymbol{\kappa} + (\mathbf{R}^T(\mathbf{r} - \mathbf{t}) - \begin{bmatrix} \hat{\mathbf{p}} \\ 0 \end{bmatrix}) \times (\mathbf{R}^T \mathbf{f})\end{aligned}$$

From the definition of the CoP we know that  $\hat{\tau}$  vanishes when projected on the contact surface, i.e.

$$\begin{aligned}\mathbf{S}^T &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \mathbf{0} &= \mathbf{S}^T \hat{\tau} = \mathbf{S}^T ((\mathbf{R}^T(\mathbf{r} - \mathbf{t})) \times (\mathbf{R}^T \mathbf{f})) - \\ &\quad \mathbf{S}^T ((\mathbf{S} \hat{\mathbf{p}}) \times (\mathbf{R}^T \mathbf{f})) + \mathbf{S}^T \mathbf{R}^T \boldsymbol{\kappa} \quad (3.70)\end{aligned}$$

$$= \mathbf{R}_{x,y}^T \boldsymbol{\kappa} + \mathbf{R}_{x,y}^T ((\mathbf{r} - \mathbf{t}) \times \mathbf{f}) + \mathbf{S}^T [\mathbf{R}^T \mathbf{f}]_{\times} \mathbf{S} \hat{\mathbf{p}} \quad (3.71)$$

In order to resolve for  $\hat{\mathbf{p}}$ , we first invert the premultiplied projector

$$\mathbf{S}^T [\mathbf{R}^T \mathbf{f}]_{\times} \mathbf{S} = \mathbf{R}_z^T \mathbf{f} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3.72)$$

$$(\mathbf{S}^T [\mathbf{R}^T \mathbf{f}]_{\times} \mathbf{S})^{-1} = \frac{1}{\mathbf{R}_z^T \mathbf{f}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (3.73)$$

We note that from the friction constraint (cf. Eq. (3.68)) we have  $\mathbf{R}_z^T \mathbf{f} > 0$ . Next, we resolve Eq. (3.71) for  $\hat{\mathbf{p}}$  and substitute it into the CoP constraints (cf. Eq. (3.67)). For better readability we will only show the result for the right-hand side of Eq. (3.67)

$$\mathbf{p}_{max} \geq \hat{\mathbf{p}} - \mathbf{c} \quad (3.74)$$

$$\Leftrightarrow \mathbf{p}_{max} \geq -(\mathbf{S}^T [\mathbf{R}^T \mathbf{f}]_{\times} \mathbf{S})^{-1} [\mathbf{R}_{x,y}^T \boldsymbol{\kappa} + \\ \mathbf{R}_{x,y}^T ((\mathbf{r} - \mathbf{t}) \times \mathbf{f})] - \mathbf{c} \quad (3.75)$$

$$\Leftrightarrow (\mathbf{p}_{max} - \mathbf{c}) \mathbf{R}_z^T \mathbf{f} \geq - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\mathbf{R}_{x,y}^T \boldsymbol{\kappa} + \\ \mathbf{R}_{x,y}^T ((\mathbf{r} - \mathbf{t}) \times \mathbf{f})), \quad (3.76)$$

where we used Eq. (3.73). Applying Eq. (3.69) and Theorem (1), one can transform Eq. (3.76) into a function inequality  $\mathbf{s}(\mathbf{f}, \boldsymbol{\kappa}) \leq \mathbf{0}$ ,  $\mathbf{s} \in \mathcal{Q}_{\pm}$  and again the quadratic parameter matrices can be constructed inexpensively.

## III.4. Optimal Control Formulations

Through out the following discussion, we will restrict ourselves to a predefined contact activation pattern (cf.  $\alpha_e$  in Eq. (3.61)). That means we predefine which end effector is in contact with the environment at which time. This could for instance be computed with an acyclic contact planner [144]. The location of the contact and the force-torque are expressed as decision variables and will be computed by the optimizer. The goal of this section is to define an optimal control problem that finds sequences of states and controls under the dynamics constraint in Eq. (3.61) and contact constraints Eq. (3.66-3.68) thus solving problem (3.57).

### III.4.1. Simultaneous Formulation

We start out with a simultaneous formulation, i.e. we optimize over both states and controls and express the dynamics constraint explicitly in the mathematical program. Here we represent contact forces at the center of pressure. We summarize the optimization variables into  $\mathbf{x}_{0:T} = [\dots \hat{\mathbf{f}}_{t,1:E}^T \hat{\mathbf{p}}_{t,1:E}^T \hat{\boldsymbol{\tau}}_{t,1:E}^T \mathbf{h}_{t+1}^T \dots]^T, t = 0 \dots T$  and  $\mathbf{c}_{1:E}$  with  $1 : E$  indexing all  $E$  contacts, and we write our optimal control problem as

$$\min_{\mathbf{x}_{0:T}, \mathbf{c}_{1:E}} \sum_t^T J_t(\mathbf{x}_t, \mathbf{c}_{1:E}) \in \mathcal{Q}_+ \quad (3.77)$$

$$\text{s.t. } 1) \mathbf{h}_{t+1} = \mathbf{h}_t + \Delta f_t(\hat{\mathbf{f}}_{t,1:E}, \hat{\mathbf{p}}_{t,1:E}, \hat{\boldsymbol{\tau}}_{t,1:E}) \quad (3.78)$$

$$\Leftrightarrow \mathbf{0} = \mathbf{x}_{t+1} + \mathbf{g}_t(\mathbf{x}_t) \in \mathcal{Q}_\pm \quad (3.79)$$

$$2) \text{ Eqs. (3.66 – 3.68)} \quad (3.80)$$

$$\Leftrightarrow \mathbf{0} \leq \mathbf{A}_t \mathbf{x}_t + \mathbf{A}'_t \mathbf{c}_{1:E} \in \mathcal{A} \quad (3.81)$$

This formulation is a Quadratically constrained Quadratic Program (QCQP), with a positive definite objective, linear inequality constraints and nonlinear equality constraints on functions from  $\mathcal{Q}_\pm$ . As one can note from (3.77-3.80), summands of the objective as well as constraints all depend only on time-local variables  $\mathbf{x}_{t:t+1}$  and on  $\mathbf{c}_{1:E}$ . This property has the advantage that optimization algorithms can be applied that exploit the sparsity inside of objective Hessian and constraint Jacobian. The constraint Jacobian has block form, where we have a block tridiagonal matrix on the left and a dense matrix on the right. The number of columns in the dense part scale with  $O(E)$ , i.e. they do not grow as  $T$  increases. The Lagrangian  $\mathcal{L}$  [107] of the optimization program in Eq. (3.77) has a Hessian that can be decomposed into 4 blocks as follows

$$\left[ \begin{array}{c|c} \nabla_{\mathbf{x}_0, \mathbf{x}_0}^2 \mathcal{L} & \\ \ddots & \\ & \nabla_{\mathbf{x}_T, \mathbf{x}_T}^2 \mathcal{L} \\ \hline \nabla_{\mathbf{x}, \mathbf{c}}^2 \mathcal{L}^T & \nabla_{\mathbf{c}, \mathbf{c}}^2 \mathcal{L} \end{array} \right]$$

We note that again the dense blocks on the left and bottom do not depend on the time discretization  $T$ . The top-left block on the other hand is block-diagonal. This sparsity is favorable in optimization algorithms, because typically an essential step in these solvers is to solve a linear system that inherits the sparsity properties from the constraint Jacobian and Lagrange Hessian. Especially, interior-point methods are known to preserve the band-diagonal sparsity structure [34].

### III.4.2. Sparse Contact Force Representation

In the following we will show that the sequential optimal control formulations for our problem can be transformed to an equivalent problem with a sparse structure similar to the simultaneous formulation. We will assume that a contact-phase  $e$  is defined as the time interval  $t \in [\sigma_e, \varepsilon_e)$  at which an end effector is touching the environment. Thus the contact activation from Eq. (3.61) can be refined to

$$\alpha_{e,t} = \begin{cases} 1, & \text{if } \sigma_e \leq t < \varepsilon_e \\ 0, & \text{else} \end{cases} \quad (3.82)$$

where contact-phase  $e$  starts at  $t = \sigma_e$  and ends at  $t = \varepsilon_e$ . For instance, a contact-phase could describe a foot step  $e$  where the heel hits the ground at  $t = \sigma_e$  and the last time step at which the toe is touching the ground is  $t = \varepsilon_e - 1$ . Two heel strikes with the same foot would correspond to two different contact phases  $e, e'$ . Further, we define forces and CoM torques as linear functions of a new set of variables  $\varphi_t, \psi_t$  as

$$\mathbf{f}_{e,t} = \varphi_{e,t} - 2\varphi_{e,t-1} + \varphi_{e,t-2} \quad (3.83)$$

$$\dot{\mathbf{r}}_{e,t} = \psi_{e,t} - 2\psi_{e,t-1} + \psi_{e,t-2} \quad (3.84)$$

$$\forall t < \sigma_e : \varphi_t = \psi_t = 0 \quad (3.85)$$

The variables  $\varphi_t, \psi_t$  correspond to the twice integrated contact forces and torques. General optimal control problems that are expressed in sequential form, usually lose their sparsity structure because the state variables need to be integrated out. In the following, we will use the change of variables in Eqs. (3.83-3.85) to show that this is not the case for our momentum optimization problem. We will derive expressions for the state variables  $\mathbf{r}, \mathbf{h}, \dot{\mathbf{h}}$  as functions of  $\varphi_t, \psi_t$ . The goal here is to show that the state variables at time step  $t$  do *not* depend on all previous

time steps  $t' = 0 \dots t$ , but only on a few ones. This has the advantage that a sequential formulation expressed with  $\varphi_t, \psi_t$  will maintain its sparse structure and at the same time reduce the number of variables and constraints compared to the formulation in Sec. III.4.1.

We compose the linear momentum rate as

$$\dot{\ell}_t(\varphi_{e,t-2:t}) = Mg + \sum_e \dot{\ell}_{e,t}(\varphi_{e,t-2:t})$$

$$\dot{\ell}_{e,t} = \alpha_{e,t} \mathbf{f}_{e,t} = \begin{cases} \varphi_{e,t} - 2\varphi_{e,t-1} + \varphi_{e,t-2}, & \text{if } t < \varepsilon_e \\ 0, & \text{else} \end{cases}$$

In order to construct expressions for the linear momentum and center of mass, we first define the sums of forces during a contact-phase over time

$$\ell_{e,t} = \sum_{i=0}^{t-1} \dot{\ell}_{e,i} = \begin{cases} \varphi_{e,t-1} - \varphi_{e,t-2}, & \text{if } t < \varepsilon_e \\ \varphi_{e,\varepsilon_e-1} - \varphi_{e,\varepsilon_e-2}, & \text{else} \end{cases}$$

$$M\mathbf{r}_{e,t} = \sum_{i=0}^{t-1} \ell_{e,i} = \begin{cases} \varphi_{e,t-2}, & \text{if } t < \varepsilon_e \\ (\varphi_{e,\varepsilon_e-2} + (t - \varepsilon_e) \times (\varphi_{e,\varepsilon_e-1} - \varphi_{e,\varepsilon_e-2})) & , \text{ else} \end{cases}$$

With the derived expressions we can now construct the linear state terms as follows

$$\dot{\ell}_t = Mg + \sum_e \dot{\ell}_{e,t}(\varphi_{e,t-2:t})$$

$$\ell_t = \ell_0 + \Delta \sum_{i=0}^{t-1} \dot{\ell}_i$$

$$= \ell_0 + \Delta t Mg + \Delta^2 \sum_e \ell_{e,t}(\varphi_{e,t-2:t}, \varphi_{e,\varepsilon_e-2:\varepsilon_e-1})$$

$$M\mathbf{r}_t = M\mathbf{r}_0 + \Delta \sum_{i=0}^{t-1} \ell_i = M\mathbf{r}_0 + \Delta t \ell_0 + \Delta^2 \frac{t(t-1)}{2} Mg +$$

$$+ \Delta^2 \sum_e M\mathbf{r}_{e,t}(\varphi_{e,t-2}, \varphi_{e,\varepsilon_e-2:\varepsilon_e-1}),$$

where  $\Delta$  is a discretization time. It is interesting to see, that the state variables at time  $t$  only depend on control variables at  $t' \in \{t-2, t-1, \varepsilon_e-1, \varepsilon_e-2 | e = 1 \dots E\}$ . This has the advantage that derivatives of the objective function and constraints remain sparse and as such result in more efficient optimization algorithms similar to

the simultaneous formulation. In the previous derivation we obtained expressions for the linear momentum, but expressions for the angular momentum can be readily derived with the same sparsity properties.

### III.4.3. Sparse Sequential Optimal Control

We will now express a sequential optimal control problem with respect to integrals of forces as they were defined in Sec. III.4.2. We concatenate decision variables into vectors  $\mathbf{x}_{0:T} = [\dots \varphi_{t,1:E}^T \psi_{t,1:E}^T \dots]^T$ ,  $\mathbf{y}_{1:E} = [\mathbf{c}_{1:E}^T \varphi_{\varepsilon_e-2:\varepsilon_e,1:E}^T \psi_{\varepsilon_e-2:\varepsilon_e,1:E}^T]^T$

$$\min_{\mathbf{x}_{0:T}, \mathbf{y}_{1:E}} \sum_t^T J_t(\mathbf{x}_{t-2:t}, \mathbf{y}_{1:E}) \in \mathcal{Q}_+ \quad (3.86)$$

$$\text{s.t. } 1) \text{ Eq. (3.76)} \quad (3.87)$$

$$\Leftrightarrow \mathbf{0} \leq \mathbf{g}_t(\mathbf{x}_{t-2:t}, \mathbf{y}_{1:E}) \in \mathcal{Q}_\pm \quad (3.88)$$

$$2) \text{ Eqs. (3.68)} \quad (3.89)$$

$$\Leftrightarrow \mathbf{0} \leq \mathbf{A}_t \mathbf{x}_{t-2:t} \in \mathcal{A} \quad (3.90)$$

We can see that similar to the simultaneous formulation, summands of the objective and constraint functions only depend on  $\mathbf{x}_{t-2:t}, \mathbf{y}_{1:E}$  rather than on all time steps. The quadratic constraints in Eq. (3.87) can be decomposed efficiently as discussed in Sec. III.3.3. We will now write out the sparse constraint Jacobian

$$\left[ \begin{array}{ccc|c} \frac{\partial g_0}{\partial \mathbf{x}_0} & \frac{\partial g_0}{\partial \mathbf{x}_1} & \frac{\partial g_0}{\partial \mathbf{x}_2} & \frac{\partial g_0}{\partial \mathbf{y}_{1:E}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{\partial g_1}{\partial \mathbf{x}_1} & \frac{\partial g_1}{\partial \mathbf{x}_2} & \frac{\partial g_1}{\partial \mathbf{x}_3} & \frac{\partial g_1}{\partial \mathbf{y}_{1:E}} \\ \mathbf{A}_0 & & & \vdots \\ \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & & & \vdots \\ \frac{\partial g_{T-1}}{\partial \mathbf{x}_{T-3}} & \frac{\partial g_{T-1}}{\partial \mathbf{x}_{T-2}} & \frac{\partial g_{T-1}}{\partial \mathbf{x}_{T-1}} & \frac{\partial g_{T-1}}{\partial \mathbf{y}_{1:E}} \\ \mathbf{A}_{T-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right]$$

We will now write out the Hessian of the Lagrangian

$$\left[ \begin{array}{ccc|c} \nabla_{0,0}^2 & \nabla_{0,1}^2 & \nabla_{0,2}^2 & \nabla_{\mathbf{x},\mathbf{y}}^2 \mathcal{L} \\ \nabla_{0,1}^2 & \nabla_{1,1}^2 & & \\ \nabla_{0,2}^2 & & \ddots & \\ \hline & \nabla_{T,T-2}^2 & \nabla_{T,T-1}^2 & \nabla_{\mathbf{x},\mathbf{y}}^2 \mathcal{L} \\ \hline \nabla_{\mathbf{x},\mathbf{y}}^2 \mathcal{L}^T & & & \nabla_{\mathbf{y},\mathbf{y}}^2 \mathcal{L} \end{array} \right]$$

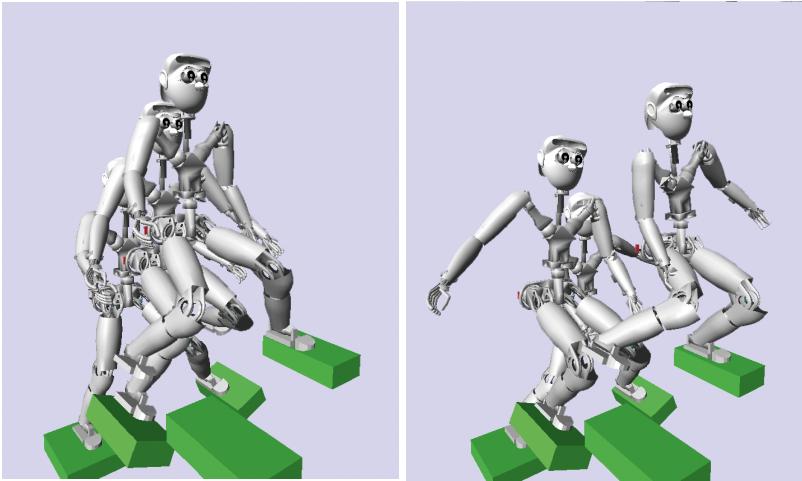
$$\begin{aligned}\nabla_{t,t'}^2 &= \nabla_{\mathbf{x}_t \mathbf{x}_{t'}}^2 \mathcal{L} = \sum_{k=0}^2 \nabla_{t,t'}^2 J_{t+k} + \sum_{k=0}^2 \phi_{t+k,i} \nabla_{t,t'}^2 \mathbf{g}_{t+k,i}, \\ \nabla_{t,t'}^2 \mathbf{g}_{t+k,i} &= (\mathbf{Q}_{t+k,i} - \mathbf{P}_{t+k,i}) \in \mathcal{Q}_{\pm}\end{aligned}\quad (3.91)$$

#### III.4.4. Comparison of sparsity patterns

As we analyzed in the previous sections, both the simultaneous as well as sequential formulation of the mathematical program (3.57) have sparse constraint Jacobians and Lagrangian Hessians. A common approach to solve optimal control formulations of that kind is to use interior point methods [35] as computation time benefits from these sparsity properties. Usually, the computation time is dominated by a decomposition of an *augmented system* matrix  $\nabla^2 \tilde{\mathcal{L}} + \mathbf{G}^T \boldsymbol{\Sigma} \mathbf{G}$ , where  $\mathbf{G}$  is the constraint Jacobian and  $\nabla^2 \tilde{\mathcal{L}}$  is a (preferably convex) approximation to the Lagrangian Hessian. Comparing the sparsity patterns of the two formulations from Sec. III.4, we see that both problems result in augmented systems that have a block structure with a band-diagonal matrix on the top left. The computational complexity for decomposing the augmented system is in both formulations linear w.r.t. the discretization parameter  $T$ . Finding an approximation  $\nabla^2 \tilde{\mathcal{L}}$  that both makes progress towards a solution and can be robustly decomposed is usually not trivial. In our problem however, a convex approximation can be constructed exploiting the fact that  $\nabla^2 \mathcal{L}$  can be decomposed into a difference of p.s.d matrices (cf. Eq. (3.91)). We can obtain  $\nabla^2 \tilde{\mathcal{L}}$  by simply dropping the negative part of  $\nabla^2 \mathcal{L}$  from Eq. (3.91). Note that we do not require expensive operations such as eigenvalue decompositions to be carried out online, but obtain the convex part of  $\nabla^2 \mathcal{L}$  through consequently applying Eq. (3.69) and Theorem (1) as we construct the problem. This can be done for both, the simultaneous as well as the sequential formulation.

The number of variables is very different for the sequential and simultaneous formulations and has an impact on the computational complexity of resulting optimization algorithms. In the simultaneous case, we have to consider 6 force variables for each end effector and 9 state variables. For instance a biped that remains in double support at all time requires  $21T$  variables. On the other hand, the sequential method does not require variables for states and thus contains only  $12T$  variables for the aforementioned example. Further, we drop the  $6T$  dynamics constraints. As a consequence computational complexity is reduced with a sequential formulation as we solve for longer and finer grained time horizons  $T$ .

It is also worth mentioning that since the momentum dynamics generalize the commonly used LIPM, the sparse structure in our sequential formulation will be similar using LIPM dynamics. This nicely complements the discussion in [34].



**Figure 3.19.:** A visualization of a kinematic plan generated by Algorithm 1. After the first iteration (left) the CoM remains above the center between the feet and the arms are at rest. After the last iteration (right) the kinematic plan is in coherence with constrained force profiles. The CoM has a wider sway in lateral direction and the robot uses the arms to account for angular momentum.

## III.5. Experiments

We demonstrate the proposed motion optimization Algorithm 1 on a stepping task for a humanoid robot. Further, we convince ourselves that the sparse structure in our momentum problem results in linear computational complexity w.r.t. time discretization  $T$ . All experiments were performed on a notebook with a 2.7 GHz intel i7 processor with 16gb RAM.

### III.5.1. Optimization Algorithm

In this section we would like to discuss our choice of optimization algorithms for solving the two subproblems in (3.57),(3.59). In both algorithms, we exploit the sparse structure resulting from the optimal control formulation. Further, we condense the number of variables in the momentum optimal control problem (3.57) as we discussed in Sec. III.4.3. We solve the optimization problem (3.59) with a Gauss-Newton method, where the Hessian is a block tridiagonal positive definite matrix that can be factorized with a dedicated Cholesky Decomposition [149]. The

bottleneck in our implementation is a numerical differentiation of the momentum Jacobian  $\frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}}$ , which should be constructed directly from the dynamics model. Nevertheless, with our naive implementation we were able to generate whole-body motion and force plans in 30 s of computation time.

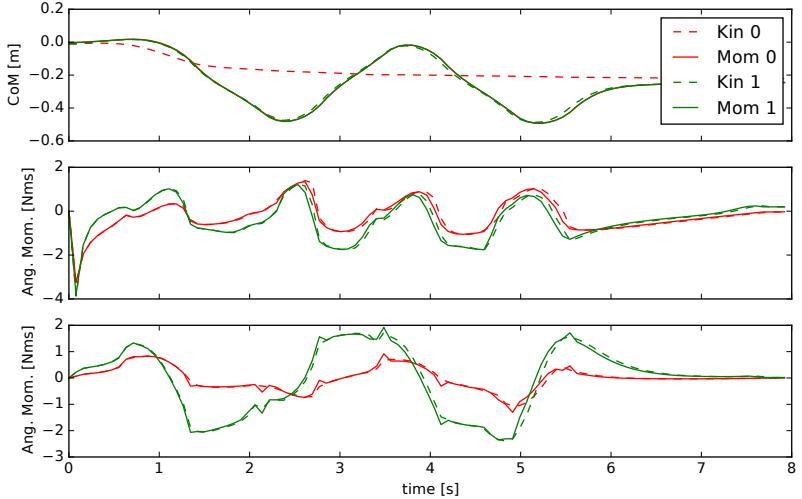
For the momentum optimization (3.57) we implement a primal-dual interior point method [107, Algo. 19.2], which allows us to preserve the band-diagonal structure in our problem. The algorithm is modified as discussed in Sec III.4.4 to exploit the convex approximation of the Lagrangian Hessian. We run the momentum optimization for increasing values of  $T$  and plot convergence of the Lagrangian gradient in Fig. 3.21. It becomes evident from the plot that computation time increases linearly as we increase  $T$ . Note that sequential methods in general scale cubic with  $T$ , however, our sparse formulation allows to preserve linear complexity potentially leading to more efficient algorithms.

In a side experiment we investigated the effect of our approximation  $\nabla^2 \tilde{\mathcal{L}}$  on convergence to a solution. We implemented a very basic sequential quadratic program [107, Algo. 18.1] that was using dense operations, meaning it was not exploiting the sparsity structure. Instead we focused on the benefits of the approximation  $\nabla^2 \tilde{\mathcal{L}}$ . Although much slower than the sparse interior-point method, our algorithm still converged by an order of magnitude faster than an off-the-shelf SQP method, SNOPT, [51] motivating the need for dedicated solvers in robotics.

### III.5.2. Motion Generation

We construct a stepping scenario<sup>6</sup> for a simulation of our Sarcos humanoid robot (cf. Fig. 3.19). The humanoid is to step on stepping stones that increase in height and are angled as illustrated in the figure. Note that here the LIPM assumption does not hold, but instead non-coplanar contacts have to be considered together with angular momentum that is required for leg swing motions. In our experiments, we fix the footstep locations  $\mathbf{c}_e$  to predefined values for a simpler implementation and compute coherent joint trajectories and force profiles using Algorithm 1. Every 0.7s contact is broken or created switching from single to double support or vice versa. Our initial guess for the CoM and momentum  $\bar{\mathbf{h}}$  is to simply have zero momentum and keep the CoM centered above the feet. We introduce weights into the cost function in Eq. (3.59) giving more importance to footstep locations. In our example, Algorithm 1 converges already after 30s for  $T = 100$ , despite the inefficient implementation and requires only 2 passes to acquire consistent force and joint trajectories. The first kinematic plan realizes our naive CoM motion moderately well, which however cannot be accomplished given the contact constraints. Thus, the momentum optimization introduce a CoM sway in lateral direction, which is then realized by the kinematic optimizer in the consecutive iteration as can be seen in Fig. 3.20. Additionally, the kinematic plan introduces an arm sway motion after the first pass to compensate for angular momentum introduced by the swing legs. After the motion generator converges it yields a kinematic plan that is coherent

<sup>6</sup>The result is summarized in <https://youtu.be/NkP5Z9MfRRw>



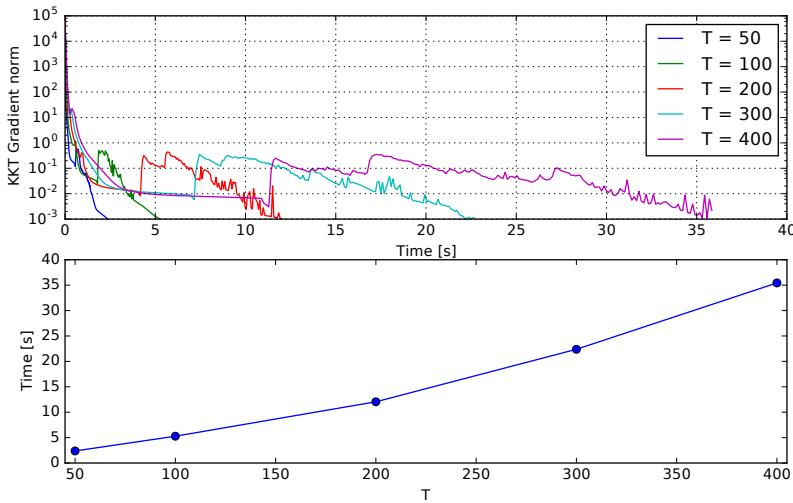
**Figure 3.20.:** Momentum plans generated by the motion optimization algorithm 1. The kinematic pass optimizes for an unfeasible CoM lateral motion, which is corrected after the first momentum optimization. In the second iteration of the algorithm, the kinematic plan is adapted to realize the CoM sway. Horizontal angular momentum (bottom two plots) is also found to be consistent across kinematic as well as momentum optimization.

with bounded force and CoP profiles (cf. Fig. 3.22). In this example, we keep the ankle joints fixed and do not account for object collision. However, collision avoidance can be added inside the kinematic optimization using trajectory optimization techniques [75].

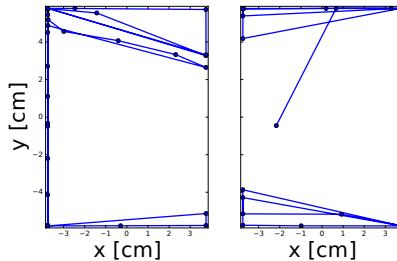
Our experiments demonstrate that constrained force profiles and consistent whole-body motion plans can be generated efficiently with the proposed algorithm. The structure identified in the momentum optimization led to more efficient optimization algorithms.

### III.6. Conclusion

We presented a contact-centric motion generation algorithm for floating-base robots. It consists of two sub-problems optimizing for kinematic trajectories and momentum profiles iteratively. An analysis of the momentum optimization is presented



**Figure 3.21.:** Computation time of the momentum optimization (3.57). We execute the optimization for increasing values  $T$  and plot the resulting norm of the Lagrangian gradient (top) and the overall computation time (bottom). Although our implementation still requires improvements, we can already see a linear trend in the computation complexity confirming our analysis in Sec. III.4.3



**Figure 3.22.:** Footprints of the robot with optimized CoP trajectories. As can be seen from the plot, the CoPs never exceed the foot support.

### *3. Optimization-based motion generation for multiped robots in contact scenarios*

---

and the relevance for optimal control formulations is discussed. In our experiments, we demonstrate the momentum-centric motion generator on a stepping task for a humanoid robot simulation. Indeed, consistent momentum and whole-body motion plans can be acquired efficiently. Further, our experiments show favorable scaling properties of our sparse optimal control formulation.

# 4

## Chapter

## Conclusion and future directions

---

### 4.1. Summary

This thesis presented a motion generation approach for legged robots performing tasks in contact-rich scenarios. We propose a control architecture that is formulated as an optimal control problem over a hierarchical task description. The algorithms we presented are specialized for legged robots performing dynamic and contact-interaction tasks. We break down the overall optimization into optimal control problems that are solved at different frequencies with increasing problem granularity. Our architecture is computationally very efficient and can be used for the realtime generation of complex movements for any robot with arms and legs. At the fastest time-scale we solve a hierarchical inverse dynamics problem with our QP cascade variant. The algorithm is implemented efficiently in a 1 kHz feedback loop on our robot. We demonstrate robustness of our controller in a series of balancing tasks on a torque controlled humanoid robot pushing the state-of-the-art in the area.

The hierarchical controller was used in combination with our whole-body trajectory optimizer and enabled our humanoid to step over difficult terrain in simulation. We decomposed the whole-body trajectory optimization problem into two sub-problems of reduced complexity resolved iteratively with specialized optimizers. Therefore, optimization over whole-body motion trajectories is decoupled from computation of contact force profiles. Resulting motion and contact force profiles were computed in a few iterations for a difficult contact-interaction task.

At the intermediate level of our motion generation architecture (see layer 2 in Figure 2.1) we suggest using a model predictive controller over the centroidal dynamics of legged robots. We demonstrated important properties of the problem that proved to be very beneficial for the design of an efficient specialized solver. Overall, we addressed all the core algorithmic problems that allow the realization of

the motion generation architecture presented in this thesis (and outlined in Figure 2.1) on multiped robots. Our theoretical contribution to numerical optimal control for legged robots and our experimental work on whole-body balancing, now, render the prospect of realtime whole body control tangible. However, enabling robots to plan and control tasks from general descriptions reliably in complex and dynamic environments remains a challenge and further research in that direction will be necessary.

## 4.2. Future directions

In conclusion, this thesis has proposed computationally efficient whole-body optimal control methods, and demonstrated their validity in robot experiments. However, several important challenges remain to be addressed.

Our implementation of layer 3 of the proposed motion generation architecture (see Figure 2.1) has been shown to work reliably on our robot and is now a common tool in the ‘movement generation and control group’. Although, simulation results of the overall control framework are promising, experimental evaluation of complex contact-interaction tasks on the robot requires more implementation work and is part of future work.

Our structural analysis of optimal control with centroidal dynamics was shown to be beneficial for algorithm design. However, a more efficient implementation of our solvers is desirable for application in even faster control loops. For example, profiling our code showed that more efficient automatic differentiation algorithms [105], could significantly improve performance. Another option is to approximate sub-parts of our whole-body trajectory optimizer in layer 1. For example, our results led to a convex approximation of the centroidal dynamics for efficient motion planning [112] making the problem easier to solve. In future work, we will integrate our results into the full architecture.

Although, robustness of our whole-body controller was demonstrated in a series of push experiments, extension to trajectory tracking in face of modeling errors may require additional work. Faster control cycles of layer 1 and layer 2 could lead to better reaction to unforeseen disturbances. However, each control command is computed under the assumption of a perfectly known world. Instead, taking into account model uncertainty in the *task feedback construction* (see Figure 2.1) can lead to more robust control policies [44, 113].

Throughout this thesis, we used model-based approaches that rely on accurate models of the robot. Our humanoid model was sufficiently accurate for balancing. However, realizing a stepping task such as the one discussed in Paper III may require more accurate parameter identification. System identification methods [9] could be utilized to improve accuracy of mass distribution parameters. On the other hand, sensor data collected online can account for task dependent changes in the model, for example the weight of a tool. An interesting research direction is the extension of methods addressing this problem for manipulators [78]. In addition, estimating the configuration of legs and arms relative to the environment

is required in the during approach. Exploiting information from a depth camera [48, 155] mounted on the robot can improve state estimation of limb configurations and contact surfaces.

The control architecture presented in this thesis, addresses many tasks that require autonomous grasping of tools or objects in the environment, such as ladders or valves. Deciding where to grasp and how to approach a grasping point poses a challenging problem [18]. In previous work [58], we proposed a data-driven method to detect grasping points and to shape the endeffector configuration accordingly. Our experiments on a mobile manipulation platform have demonstrated reliable grasp prediction for a large variety of objects. An integration of our method into the whole-body control framework presented here is an interesting future direction.



## Appendices



# Appendix A

## TV-LQR for hierarchical task-space policies

---

Here we extend the discussion of the motion generation framework proposed in Section 2.1 and visualized in Figure 2.1. It is defined over several levels of optimal control problems solved at different time-scales. The output of layers higher up are optimized state and control trajectories. The input into the bottom layer are task-space feedback policies. In the following, we propose how optimized trajectories from slower-time scales can be used to compute feedback policies for the fast prioritized time-scale in layer 3.

The cost function of our first layer optimizer is expressed over  $H$  state and control features (or state and control tasks)

$$\mathbf{x}_h(\mathbf{y}) := \mathbf{x}_h(\mathbf{q}, \dot{\mathbf{q}}), \quad (\text{A.1})$$

$$\mathbf{x}_h(\mathbf{u}) := \mathbf{x}'_h(\boldsymbol{\tau}, \boldsymbol{\lambda}), \quad h = 1 \dots H, \quad (\text{A.2})$$

where  $\mathbf{q} \in \mathbb{SE}(3) \times \mathbb{R}^n$  is the base and joint configuration,  $\boldsymbol{\tau} \in \mathbb{R}^n$  the joint torques and  $\boldsymbol{\lambda} \in \mathbb{R}^m$  the vector of contact forces acting on the robot and  $\mathbf{x}_h \in \mathbb{R}^l, \mathbf{x}'_h \in \mathbb{R}^{l'}$  mappings from states and controls into task-spaces. We summarize states and controls into vectors  $\mathbf{y} = [\mathbf{q} \quad \dot{\mathbf{q}}]^T, \mathbf{u} = [\boldsymbol{\tau} \quad \boldsymbol{\lambda}]^T$ . The dynamics of the state-feature is

$$\dot{\mathbf{x}}_h = \frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial t} = \frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} \dot{\mathbf{y}} \quad (\text{A.3})$$

where we assume to have partial derivatives of  $\mathbf{x}_h$ .

The first layer of our motion generation framework optimizes over a cost function with weighted prioritization

$$c = \sum_h w_h c_h(\mathbf{x}_h), \quad (\text{A.4})$$

### A. TV-LQR for hierarchical task-space policies

---

with weights  $w_h > 0$ . The result of the first layer are optimal state and control trajectories  $\mathbf{y}^*(t), \mathbf{u}^*(t)$  and also optimal feature trajectories  $\mathbf{x}_h^*(t), \mathbf{x}'_h^*(t)$ . In order to track  $\mathbf{x}_h^*(t), \mathbf{x}'_h^*(t)$ , our hierarchical solver at the fastest time-scale expects a desired task-space dynamics. We will construct it in the form

$$\begin{bmatrix} \mathbf{A}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_h \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_h \\ \mathbf{b}_h \end{bmatrix} \quad (\text{A.5})$$

where we need to design projectors  $\mathbf{A}_h, \mathbf{B}_h$  and biases  $\mathbf{a}_h, \mathbf{b}_h$ . We propose to compute an optimal feedback policy on the full dynamics and project them into task space defined by  $\mathbf{x}_h, \mathbf{x}'_h$ . We use time-varying LQR over a linearization of the full dynamics to compute an optimal (time-varying) affine control policy. First, we linearize the equations of motion (1.1) and expand the cost (A.4) up to second order around the optimal state and control trajectories  $\mathbf{y}^*(t), \mathbf{u}^*(t)$ . Resulting time-varying linear dynamics and quadratic cost are optimized with TV-LQR to obtain an optimal feedback policy and optimal closed-loop system

$$\bar{\mathbf{u}} = \mathbf{D}_t \mathbf{y} + \mathbf{d}_t \quad (\text{A.6})$$

$$\dot{\mathbf{y}} = \mathbf{E}_t \mathbf{y} + \mathbf{e}_t \quad (\text{A.7})$$

Optimal policy (A.6) and closed-loop dynamics (A.7) are now projected into task-space resulting in

$$\dot{\bar{\mathbf{x}}}_h = \frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} \dot{\bar{\mathbf{y}}} \quad (\text{A.8})$$

$$\bar{\mathbf{x}}'_h \approx \mathbf{x}'_h(\bar{\mathbf{u}}) + \frac{\partial \mathbf{x}'_h}{\partial \mathbf{u}}(\mathbf{u} - \bar{\mathbf{u}}) \quad (\text{A.9})$$

Here, we substituted Equation (A.7) into (A.3) and obtained the optimal closed-loop dynamics of the state feature  $\dot{\bar{\mathbf{x}}}_h$ . The optimal control feature  $\bar{\mathbf{x}}'_h$  is approximated by a first order tailor-series expansion around the optimal control  $\bar{\mathbf{u}}$ . Finally, we obtain task projections for Equation (A.5) as follows

$$\dot{\bar{\mathbf{x}}}_h = \dot{\bar{\mathbf{x}}}_h \quad (\text{A.10})$$

$$\iff \frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} \dot{\bar{\mathbf{y}}} = \frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} \dot{\bar{\mathbf{y}}} \quad (\text{A.11})$$

$$\iff \underbrace{\frac{\partial \mathbf{x}_h}{\partial \dot{\mathbf{q}}}}_{\mathbf{A}_h} \ddot{\mathbf{q}} = \underbrace{\frac{\partial \mathbf{x}_h}{\partial \mathbf{y}} (\mathbf{E}_t \mathbf{y} + \mathbf{e}_t)}_{\mathbf{a}_h} - \underbrace{\frac{\partial \mathbf{x}_h}{\partial \mathbf{q}} \dot{\mathbf{q}}}_{\mathbf{b}_h} \quad (\text{A.12})$$

---

and similarly for  $\mathbf{B}_h, \mathbf{b}_h$

$$\mathbf{x}'_h = \bar{\mathbf{x}}'_h \quad (\text{A.13})$$

$$\Rightarrow \frac{\partial \mathbf{x}'_h}{\partial \mathbf{u}} (\mathbf{u} - \bar{\mathbf{u}}) \approx \mathbf{0} \quad (\text{A.14})$$

$$\Rightarrow \underbrace{\frac{\partial \mathbf{x}'_h}{\partial \mathbf{u}}}_{\mathbf{B}_h} \mathbf{u} = \underbrace{\frac{\partial \mathbf{x}'_h}{\partial \mathbf{u}} (\mathbf{D}_t \mathbf{y} + \mathbf{d}_t)}_{\mathbf{b}_h} \quad (\text{A.15})$$

The result are feedback policies passed on to the hierarchical solver in layer 3 of our motion generation algorithm (see Section 2.1). We projected the optimal joint-space policy in Equation (A.6) back into prioritized task spaces  $\mathbf{x}_h, \mathbf{x}'_h$ . Throughout this derivation we worked with a continuous time representation of states and controls  $\mathbf{y}, \mathbf{u}$  for simplicity. In practice, discretization is required to solve the time-varying LQR problem.



## Bibliography

---

- [1] Fetch robotics. URL <http://fetchrobotics.com/>. Online, accessed: 2017-06-22.
- [2] Lula robotics inc. URL <https://www.lularobotics.com/>. Online, accessed: 2017-06-22.
- [3] Mujin. URL <http://mujin.co.jp>. Online, accessed: 2017-06-22.
- [4] Rethink robotics. URL <http://www.rethinkrobotics.com/>. Online, accessed: 2017-06-22.
- [5] H. H. Asada and J.-J. E. J.-J. E. Slotine. *Robot analysis and control*. J. Wiley, 1986. ISBN 0471830291.
- [6] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, and X. Xinjilefu. No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 623–630, 2015. doi: 10.1109/HUMANOIDS.2015.7363436.
- [7] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida. Model preview control in multi-contact motion-application to a humanoid robot. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 4030–4035, 2014.
- [8] H. Audren, A. Kheddar, and P. Gergondet. Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1037–1044, 2016. doi: 10.1109/HUMANOIDS.2016.7803399.
- [9] K. Ayusawa, G. Venture, and Y. Nakamura. Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems. *The International Journal of Robotics Research*, 33: 446–468, 2014.

- [10] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell. A reactive controller framework for quadrupedal locomotion on challenging terrain. In *2013 IEEE International Conference on Robotics and Automation*, pages 2554–2561, 2013. doi: 10.1109/ICRA.2013.6630926.
- [11] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716–9, aug 1952.
- [12] D. P. Bertsekas. *Dynamic programming and optimal control, Volume I*. Athena Scientific, 1996. ISBN 1886529264.
- [13] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, mar 1998. doi: 10.2514/2.4231.
- [14] M. Bloesch, M. Hutter, M. H. Hoepflinger, C. D. Remy, C. Gehring, and R. Siegwart. State estimation for legged robots-consistent fusion of leg kinematics and IMU. In *Robotics: Science and Systems (R:SS)*, 2012.
- [15] T. Boaventura, M. Focchi, M. Frigerio, J. Buchli, C. Semini, G. A. Medrano-Cerda, and D. Caldwell. On the role of load motion compensation in high-performance force control. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2012.
- [16] T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi, and D. Caldwell. Dynamic Torque Control of a Hydraulic Quadruped Robot. In *IEEE Intl Conf on Robotics and Automation*, 2012.
- [17] J. Bobrow. Optimal robot plant planning using the minimum-time criterion. *IEEE Journal on Robotics and Automation*, 4(4):443–450, 1988. doi: 10.1109/56.811.
- [18] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, apr 2014. doi: 10.1109/TRO.2013.2289018.
- [19] J. Bohg, J. Romero, A. Herzog, and S. Schaal. Robot arm pose estimation through pixel-wise part classification. In *IEEE Intl Conf on Robotics and Automation*, 2014.
- [20] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar. Potential field guide for humanoid multicontacts acyclic motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 1165–1170, 2009. doi: 10.1109/ROBOT.2009.5152353.

- [21] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. ISBN 9780521833783.
- [22] T. Bretl. Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *The International Journal of Robotics Research*, 25(4):317–342, apr 2006. doi: 10.1177/0278364906063979.
- [23] T. Bretl, S. Rock, J. Latombe, B. Kennedy, and H. Aghazarian. Free-climbing with a multi-use robot. In *International Symposium on Experimental Robotics ISER*, 2004.
- [24] S. Caron, Q.-C. Pham, and Y. Nakamura. Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas. In *IEEE Intl Conf on Robotics and Automation*, May 2015. doi: 10.1109/ICRA.2015.7139910.
- [25] S. Caron, Q.-C. Pham, and Y. Nakamura. Zmp support areas for multicontact mobility under frictional constraints. *IEEE Transactions on Robotics*, 33(1):67–80, feb 2017. doi: 10.1109/TRO.2016.2623338.
- [26] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard. A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion. In *IEEE Intl Conf on Robotics and Automation*, 2016.
- [27] G. Cheng, H. Sang-Ho, A. Ude, J. Morimoto, J. G. Hale, J. Hart, J. Nakanishi, D. Bentivegna, J. Hodgins, C. Atkeson, M. Mistry, S. Schaal, and M. Kawato. CB: Exploring Neuroscience with a Humanoid Research Platform. In *IEEE Intl Conf on Robotics and Automation*, 2008.
- [28] A. Crespi and A. Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics*, 24(1):75–87, feb 2008. doi: 10.1109/TRO.2008.915426.
- [29] H. Dai and R. Tedrake. Planning robust walking motion on uneven terrain via convex optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 579–586, 2016. doi: 10.1109/HUMANOIDS.2016.7803333.
- [30] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302, 2014. doi: 10.1109/HUMANOIDS.2014.7041375.
- [31] M. de Lasas, I. Mordatch, and A. Hertzmann. Feature-Based Locomotion Controllers. *ACM Transactions on Graphics*, 29(3), 2010.

- [32] R. Deits and R. Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 279–286, 2014. doi: 10.1109/HUMANOIDS.2014.7041373.
- [33] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics*, pages 65–93. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi: 10.1007/978-3-540-36119-0\_4.
- [34] D. Dimitrov, A. Sherikov, and P.-B. Wieber. A sparse model predictive control formulation for walking motion generation. *IROS*, 2011.
- [35] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. *CDC*, pages 668–674, 2012.
- [36] A. El Khoury, F. Lamiraux, and M. Taix. Optimal motion planning for humanoid robots. In *2013 IEEE International Conference on Robotics and Automation*, pages 3136–3141, 2013. doi: 10.1109/ICRA.2013.6631013.
- [37] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schaffer. Overview of the torque-controlled humanoid robot toro. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 916–923, 2014. doi: 10.1109/HUMANOIDS.2014.7041473.
- [38] J. Englsberger, C. Ott, and A. Albu-Schaffer. Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion. *IEEE Transactions on Robotics*, 31(2):355–368, 2015.
- [39] T. Erez, Y. Tassa, and E. Todorov. Infinite-horizon model predictive control for periodic tasks with contacts. In *Robotics: Science and Systems VII*, pages 73–79, 2011. doi: 10.15607/RSS.2011.VII.010.
- [40] A. Escande and A. Kheddar. Contact planning for acyclic motion with tasks constraints. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 435–440, 2009. doi: 10.1109/IROS.2009.5354371.
- [41] A. Escande, N. Mansard, and P.-B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, May 2014.
- [42] S. Faraji, S. Pouya, C. Atkeson, and A. Ijspeert. Versatile and robust 3d walking with a simulated humanoid robot (atlas): a model predictive control approach. In *IEEE International Conference on Robotics and Automation*, 2014.

- [43] S. Faraji, S. Pouya, and A. Ijspeert. Robust and Agile 3D Biped Walking With Steering Capability Using a Footstep Predictive Approach. In *Robotics: Science and Systems (R.SS)*, Berkeley, USA, July 2014.
- [44] F. Farshidian and J. Buchli. Risk sensitive, nonlinear optimal control: Iterative linear exponential-quadratic optimal control with gaussian noise. *ArXiv e-prints*, dec 2015.
- [45] F. Farshidian, M. Kamgarpour, D. Pardo, and J. Buchli. Sequential linear quadratic optimal control for nonlinear switched systems. *ArXiv e-prints*, sep 2016.
- [46] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim. Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1028–1035, 2015. doi: 10.1109/HUMANOIDS.2015.7363480.
- [47] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice-a survey. *Automatica*, 25(3):335–348, may 1989. doi: 10.1016/0005-1098(89)90002-2.
- [48] C. Garcia Cifuentes, J. Issac, M. Wuthrich, S. Schaal, and J. Bohg. Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robotics and Automation Letters*, 2(2):577–584, apr 2017. doi: 10.1109/LRA.2016.2645124.
- [49] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart. Control of dynamic gaits for a quadrupedal robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 3287–3292, 2013. doi: 10.1109/ICRA.2013.6631035.
- [50] M. Giftthaler, T. Sandy, K. Doerfler, I. Brooks, M. Buckingham, G. Rey, M. Kohler, F. Gramazio, and J. Buchli. Mobile robotic fabrication at 1:1 scale: the in situ fabricator. *Construction Robotics*, pages 1–12, jun 2017. doi: 10.1007/s41693-017-0003-5.
- [51] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.
- [52] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, Sept. 1983.

- [53] A. Goswami. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The Intl Journal of Robotics Research*, 18(6):523–533, 1999.
- [54] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox. Motion planning for legged robots on varied terrain. *The International Journal of Robotics Research*, 27(11-12):1325–1349, nov 2008. doi: 10.1177/0278364908098447.
- [55] A. Herdt, N. Perrin, and P.-B. Wieber. Walking without thinking about it. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 190–195, 2010.
- [56] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-Based Learning of Grasp Selection. In *IEEE Intl Conf on Robotics and Automation*, 2012.
- [57] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Momentum-based balance control for torque-controlled humanoids. <http://arxiv.org/abs/1305.2042v1>, 2013.
- [58] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal. Learning of grasp selection based on shape-templates. *Autonomous Robots*, 36(1):51–65, 2014. ISSN 1573-7527. doi: 10.1007/s10514-013-9366-8.
- [59] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2014.
- [60] A. Herzog, N. Rotella, S. Schaal, and L. Righetti. Trajectory generation for multi-contact momentum-control. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2015.
- [61] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti. Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid. *Autonomous Robots*, 40(3):473–491, 2016. ISSN 1573-7527. doi: 10.1007/s10514-015-9476-6.
- [62] A. Herzog, S. Schaal, and L. Righetti. Structured contact force optimization for kino-dynamic motion generation. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 2703–2710, Oct 2016. doi: 10.1109/IROS.2016.7759420.
- [63] N. Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1, mar 1985. doi: 10.1115/1.3140702.

- [64] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25(7):627–643, jul 2006. doi: 10.1177/0278364906067174.
- [65] M. Hutter, M. A. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart. Hybrid Operational Space Control for Compliant Legged Systems. In *Robotics: Science and Systems (R:SS)*, 2012.
- [66] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart. Efficient and versatile locomotion with highly compliant legs. *IEEE/ASME Transactions on Mechatronics*, 18(2):449–458, apr 2013. doi: 10.1109/TMECH.2012.2222430.
- [67] S.-H. Hyon, J. G. Hale, and G. Cheng. Full-Body Compliant Human–Humanoid Interaction: Balancing in the Presence of Unknown External Forces. *IEEE Transactions on Robotics*, 23(5):884–898, 2007.
- [68] D. H. Jacobson. New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach. *Journal of Optimization Theory and Applications*, 2(6):411–440, nov 1968. doi: 10.1007/BF00925746.
- [69] G. Jarquin, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega. Real-time smooth task transitions for hierarchical inverse kinematics. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2013.
- [70] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt. Team ihm’s lessons learned from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208, mar 2015. doi: 10.1002/rob.21571.
- [71] P. Kaiser, M. Grotz, E. E. Aksoy, M. Do, N. Vahrenkamp, and T. Asfour. Validation of whole-body loco-manipulation affordances for pushability and liftability. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 920–927, 2015. doi: 10.1109/HUMANOIDS.2015.7363471.
- [72] S. Kajita, F. Kanehiro, and K. Kaneko. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Intl Conf on Robotics and Automation*, 2003.
- [73] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2003.

- [74] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal. Learning, Planning, and Control for Quadruped Locomotion over Challenging Terrain. *The Intl Journal of Robotics Research*, 30(2):236–258, 2011.
- [75] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. stomp: stochastic trajectory optimization for motion planning. In *IEEE Intl Conf on Robotics and Automation*, 2011.
- [76] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira. Humanoid robot hrp-4 - humanoid robotics platform with lightweight and slim body. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4400–4407, 2011. doi: 10.1109/IROS.2011.6094465.
- [77] O. Kanoun, F. Lamiraux, and P.-B. Wieber. Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.
- [78] D. Kappler, F. Meier, N. Ratliff, and S. Schaal. A New Data Source for Inverse Dynamics Learning. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2017.
- [79] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, jun 2011. doi: 10.1177/0278364911406761.
- [80] M. Khadiv, A. Herzog, S. A. Moosavian, and L. Righetti. Step timing adjustment: A step toward generating robust gaits. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2016.
- [81] M. Khadiv, S. Kleff, A. Herzog, S. A. Moosavian, and L. Righetti. Stepping stabilization using a combination of dcm tracking and step adjustment. In *RSI International Conference on Robotics and Mechatronics*, 2016.
- [82] M. Khadiv, A. Herzog, S. A. Moosavian, and L. Righetti. A robust walking controller based on online step location and duration optimization for bipedal locomotion. *ArXiv e-prints*, Apr 2017.
- [83] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, feb 1987. doi: 10.1109/JRA.1987.1087068.
- [84] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351, 2015. doi: 10.1109/IROS.2015.7353843.

- [85] K. Koyanagi, H. Hirukawa, S. Hattori, M. Morisawa, S. Nakaoka, K. Harada, and S. Kajita. A pattern generator of humanoid robots walking on a rough terrain using a handrail. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2617–2622, 2008. doi: 10.1109/IROS.2008.4650686.
- [86] S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *IEEE International Conference on Robotics and Automation*, 2014.
- [87] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, mar 2016. doi: 10.1007/s10514-015-9479-3.
- [88] S. M. LaValle and S. M. *Planning algorithms*. Cambridge University Press, 2006. ISBN 0521862051.
- [89] S. M. LaValle, J. J. Kuffner, and Jr. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.
- [90] S.-H. Lee and A. Goswami. Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 3157–3162, 2010.
- [91] S.-H. Lee and A. Goswami. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, 33:399–414, 2012.
- [92] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar. Generation of whole-body optimal dynamic multi-contact motions. *The International Journal of Robotics Research*, 32(9-10):1104–1119, aug 2013. doi: 10.1177/0278364913478990.
- [93] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Lee, Y. Bok, D.-G. Choi, B. Cho, S. Kim, J. Heo, I. Kim, J. Lee, I. S. Kwon, and J.-H. Oh. Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals. *Journal of Field Robotics*, 34(4):802–829, jun 2017. doi: 10.1002/rob.21673.
- [94] S. Lohmeier, T. Buschmann, and H. Ulbrich. System design and control of anthropomorphic walking robot lola. *IEEE/ASME Transactions on Mechatronics*, 14(6):658–666, dec 2009. doi: 10.1109/TMECH.2009.2032079.

- [95] N. Mansard. A dedicated solver for fast operational-space inverse dynamics. In *IEEE Intl Conf on Robotics and Automation*, 2012.
- [96] S. Mason, L. Righetti, and S. Schaal. Full dynamics lqr control of a humanoid robot: An experimental study on balancing and squatting. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2014.
- [97] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Hierarchical planning of dynamic movements without scheduled contact sequences. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4636–4641, 2016. doi: 10.1109/ICRA.2016.7487664.
- [98] J. Mattingley and S. Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, mar 2012. doi: 10.1007/s11081-011-9176-9.
- [99] M. Mistry, S. Schaal, and K. Yamane. Inertial Parameter Estimation of Floating Base Humanoid Systems using Partial Force Sensing. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2009.
- [100] K. Mombaur and Katja. Using optimization to create self-stable human-like running. *Robotica*, 27(03):321, may 2009. doi: 10.1017/S0263574708004724.
- [101] I. Mordatch, E. Todorov, and Z. Popovi?. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics*, 31(4):1–8, jul 2012. doi: 10.1145/2185520.2185539.
- [102] I. Mordatch, K. Lowrey, and E. Todorov. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5307–5314, 2015. doi: 10.1109/IROS.2015.7354126.
- [103] F. Moro, M. Gienger, A. Goswami, N. Tsagarakis, and D. Caldwell. An attractor-based whole-body motion control (wbmc) system for humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2013.
- [104] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6:3–15, 1987.
- [105] M. Neunert, M. Gifthaler, M. Frigerio, C. Semini, and J. Buchli. Fast derivatives of rigid body dynamics for control, optimization and estimation. In *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pages 91–97, 2016. doi: 10.1109/SIMPAR.2016.7862380.

- [106] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli. Trajectory optimization through contacts and automatic gait discovery for quadrupeds. *IEEE Robotics and Automation Letters*, 2(3):1502–1509, jul 2017. doi: 10.1109/LRA.2017.2665685.
- [107] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
- [108] D. E. Orin and A. Goswami. Centroidal Momentum Matrix of a humanoid robot: Structure and Properties. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2008.
- [109] C. Ott, M. A. Roa, and G. Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2011.
- [110] D. Pardo, L. Moller, M. Neunert, A. W. Winkler, and J. Buchli. Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *IEEE Robotics and Automation Letters*, 1(2):946–953, jul 2016. doi: 10.1109/LRA.2016.2527062.
- [111] Q.-C. Pham, S. Caron, P. Lertkultanon, and Y. Nakamura. Admissible velocity propagation: Beyond quasi-static path planning for high-dimensional robots. *The International Journal of Robotics Research*, 36(1):44–67, jan 2017. doi: 10.1177/0278364916675419.
- [112] B. Ponton, A. Herzog, S. Schaal, and L. Righetti. A convex model of momentum dynamics for multi-contact motion generation. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2016.
- [113] B. Ponton, S. Schaal, and L. Righetti. Risk sensitive nonlinear optimal control with measurement uncertainty. *ArXiv e-prints*, may 2016.
- [114] L. S. Pontryagin. *The Mathematical Theory of Optimal Processes*. CRC Press (reprint), 1962. ISBN 2881240771.
- [115] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, jan 2014. doi: 10.1177/0278364913506757.
- [116] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid. *The Intl Journal of Robotics Research*, 31(10), 2012.
- [117] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2):10822–10825, 2008. doi: 10.3182/20080706-5-KR-1001.01833.

- [118] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, 2009. doi: 10.1109/ROBOT.2009.5152817.
- [119] L. Righetti and A. Herzog. *Momentum-Centered Control of Contact Interactions*, pages 339–359. Springer International Publishing, Cham, 2017. ISBN 978-3-319-51547-2. doi: 10.1007/978-3-319-51547-2\_14".
- [120] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse Dynamics Control of Floating-Base Robots with External Constraints: a Unified View. In *2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, pages 1085–1090, 2011.
- [121] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Control of legged robots with optimal distribution of contact forces. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 318–324, 2011.
- [122] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal. Optimal distribution of contact forces with inverse-dynamics control. *The Intl Journal of Robotics Research*, 32(3):280–298, 2013.
- [123] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal. State Estimation for a Humanoid Robot. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2014.
- [124] N. Rotella, A. Herzog, S. Schaal, and L. Righetti. Humanoid momentum estimation using sensed contact wrenches. In *IEEE-RAS Intl Conf on Humanoid Robots*, 2015.
- [125] L. Saab, O. Ramos, N. Mansard, P. Soueres, and J.-Y. Fourquet. Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. In *IEEE Transactions on Robotics*, 2013.
- [126] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent asimo: system overview and integration. In *IEEE/RSJ International Conference on Intelligent Robots and System*, pages 2478–2483, 2012. doi: 10.1109/IRDS.2002.1041641.
- [127] J. Salini, V. Padois, and P. Bidaud. Synthesis of Complex Humanoid Whole-Body Behavior: A Focus on Sequencing and Tasks Transitions. In *IEEE Intl Conf on Robotics and Automation*, 2011.
- [128] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, aug 2014. doi: 10.1177/0278364914528132.

- [129] G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, oct 2010. doi: 10.1109/TMECH.2009.2035112.
- [130] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell. Design of hyq - a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(6):831–849, sep 2011. doi: 10.1177/0959651811402275.
- [131] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, pages 505–518, 2005.
- [132] A. Sherikov, D. Dimitrov, and P.-B. Wieber. Whole body motion controller with long-term balance constraints. *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2014.
- [133] A. Sideris and J. Bobrow. An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2275–2280, 2005. doi: 10.1109/ACC.2005.1470308.
- [134] T. Simeon, J.-P. Laumond, J. Cortés, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):729–746, aug 2004. doi: 10.1177/0278364904045471.
- [135] Siyuan Feng, X. Xinjilefu, Weiwei Huang, and C. G. Atkeson. 3d walking based on online optimization. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 21–27, 2013. doi: 10.1109/HUMANOIDS.2013.7029950.
- [136] O. Stasse, F. Morsillo, M. Geisert, N. Mansard, M. Naveau, and C. Vassallo. Airbus/future of aircraft factory hrp-2 as universal worker proof of concept. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1014–1015, 2014. doi: 10.1109/HUMANOIDS.2014.7041488.
- [137] B. J. Stephens and C. G. Atkeson. Dynamic Balance Force Control for compliant humanoid robots. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2010.
- [138] M. Stilman. Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics*, 26(3):576–584, jun 2010. doi: 10.1109/TRO.2010.2044949.

- [139] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012. doi: 10.1109/IROS.2012.6386025.
- [140] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175, 2014. doi: 10.1109/ICRA.2014.6907001.
- [141] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, sep 2006. doi: 10.1002/rob.20147.
- [142] E. Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 1071–1076, 2011. doi: 10.1109/ICRA.2011.5979814.
- [143] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettre. A reachability-based planner for sequences of acyclic contacts in cluttered environments. In *International Symposium on Robotics Research ISRR*, 2015.
- [144] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard. An efficient acyclic contact planner for multipied robots. *hal-01267345*, 2016.
- [145] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba. Online Walking Pattern Generation for Push Recovery and Minimum Delay to Commanded Change of Direction and Speed. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, 2012.
- [146] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, K. Kaneko, M. Morisawa, E. Yoshida, and F. Kanehiro. Vertical ladder climbing by the hrp-2 humanoid robot. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 671–676, 2014. doi: 10.1109/HUMANOIDS.2014.7041435.
- [147] M. Vucobratovic and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01):157–173, mar 2004. doi: 10.1142/S0219843604000083.
- [148] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, mar 2006. doi: 10.1007/s10107-004-0559-y.

- [149] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, March 2010. ISSN 1063-6536. doi: 10.1109/TCST.2009.2017934.
- [150] P. Wensing and D. Orin. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In *2013 IEEE International Conference on Robotics and Automation*, 2013.
- [151] P. M. Wensing and D. E. Orin. Development of high-span running long jumps for humanoids. In *IEEE Intl Conf on Robotics and Automation*, pages 222–227, 2014.
- [152] P. M. Wensing and D. E. Orin. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *International Journal of Humanoid Robotics*, 13(01):1550039, mar 2016. doi: 10.1142/S0219843615500395.
- [153] E. Westervelt, J. Grizzle, and D. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, jan 2003. doi: 10.1109/TAC.2002.806653.
- [154] P. Wieber. Holonomy and nonholonomy in the dynamics of articulated motion. *Fast motions in biomechanics and robotics*, pages 411–425, 2006.
- [155] M. Wuthrich, P. Pastor, M. Kalakrishnan, J. Bohg, and S. Schaal. Probabilistic object tracking using a range camera. In *IEEE/RSJ Intl Conf on Intelligent Robots and Systems*, pages 3195–3202, 2013. doi: 10.1109/IROS.2013.6696810.
- [156] T. Yoshikawa. Force control of robot manipulators. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, pages 220–226, 2000. doi: 10.1109/ROBOT.2000.844062.
- [157] D. Zimmermann, S. Coros, Y. Ye, R. W. Sumner, and M. Gross. Hierarchical planning and control for complex motor tasks. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation - SCA '15*, pages 73–81, 2015. doi: 10.1145/2786784.2786795.



# Curriculum Vitae

---

## Alexander Herzog

born May 23, 1986

citizen of Germany

- since 2017    *X Inc. (Google)*, Mountain View, California, USA  
                    Roboticist
- 2016 - 2017    *ETH Zurich*, Switzerland  
                    Doctoral studies carried out externally at the Max Planck Institute for Intelligent Systems (Prof. Dr. Ludovic Righetti) in collaboration with the ETH Zurich (Prof. Dr. Jonas Buchli); graduated with Dr. sc. ETH Zurich
- 2012 - 2017    *Max Planck Inst. Intelligent Systems, Tübingen, Germany*  
                    Doctoral studies in the Movement Generation and Control Group (Prof. Dr. Ludovic Righetti), in the Autonomous Motion Department (Prof. Dr. Stefan Schaal)
- 2016            *X Inc. (Google)*, Mountain View, California, USA  
                    Internship in Robotics
- 2011 - 2016    *University of Southern California, Los Angeles, USA*  
                    Collaborative research at the Computational Learning and Control Lab (Prof. Dr. Stefan Schaal)
- 2007 - 2011    *Karlsruhe Institute of Technology (KIT), Germany*  
                    Undergraduate and graduate studies; graduated with Dipl.-Inform. (equivalent to M.Sc.) in Computer Science
- 1997 - 2006    *Goethe Gymnasium, Germersheim, Germany*  
                    High-school education; graduated with Allgemeine Hochschulreife (German A-levels)