BRAHAYAM PONTON

MOTION PLANNING AND FEEDBACK CONTROL OF SIMULATED ROBOTS IN MULTI-CONTACT SCENARIOS

MOTION PLANNING AND FEEDBACK CONTROL OF SIMULATED ROBOTS IN MULTI-CONTACT SCENARIOS

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

MSc. BRAHAYAM PONTON

aus Quito / Ecuador

Tüebingen
2019

Gedrückt mit Genehmigung der Mathematisch - Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	16.07.2019
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Ludovic Righetti
2. Berichterstatter:	Prof. Dr. Hendrik Lensch

To my beloved parents and little sister

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

Computer Graphics Group
Department of Computer Science
Eberhard Karls Universität Tübingen
Tübingen, Germany

ABSTRACT

This dissertation presents an optimal control-based architecture for motion planning and feedback control of simulated robots in multi-contact scenarios. Motion planning and control are fundamental building-blocks for the creation of truly autonomous robots. While there has been tremendous progress in these areas for fixed-base manipulators and wheeled robots, the motion planning and control problem for floating-base robots such as those with arms and legs is still an open one, as experienced in recent robotic challenges where the need for efficient and robust algorithmic approaches was made evident. Under this context, this dissertation proposes an architecture to tackle two main challenges, namely: the efficient planning of contact sequences and whole-body movements for floating-base robots as well as its successful execution using feedback control policies capable of coping with environmental uncertainties.

One of the first steps to successfully realize complex motions is the ability to efficiently synthesize them, which if achievable in a close to real-time receding-horizon fashion makes it possible to reactively control a robot based on its current state. Traditional whole-body trajectory optimization approaches allow to fully exploit the robot's capabilities; however, they are computationally very costly to be run in an online fashion. To overcome this algorithmic challenge, the motion planning architecture proposed in this dissertation exploits the geometrical properties of the problem to efficiently approximate a local solution using a polynomial-time algorithm, but without sacrificing expressiveness for synthesizing complex motion behaviors. Using the proposed approach, it will be shown that a wide range of time-optimal highly dynamic motions can be generated for humanoid as well as for quadruped robots.

Furthermore, based on the efficient computation of solutions for dynamic feasibility of movement plans and thanks to its convex convergence properties, it will be shown that it is also possible to efficiently select sequences of contacts in complex terrains based on a measure of dynamical robustness using a mixed-integer solver. This constitutes a very important step towards designing contact planners that go beyond traditional approaches based on quasi-static stability assumptions and that can be used in more complex and dynamic scenarios such as crossing a wide gap, going down a steep slope or reactively selecting a new contact location in a push-recovery task.

Despite the benefits that a fast contacts and motion replanning loop could bring at enhancing the execution's success rate of generated motions; due to environmental uncertainties and the still non real-time replanning rates of the planning algorithms, a feedback control architecture for the execution of movement plans is advantageous

and can greatly contribute to the successful accomplishment of motion tasks. To this end, a feedback control architecture is proposed to execute movement plans in tasks involving multi-contact interactions. This architecture builds upon two key elements: Firstly, it exploits the superposition property of impedances to stabilize desired movement plans based on its kino-dynamic nature. Secondly, it can generate feedback behaviors adaptive to the uncertainty type (e.g. within dynamics or measurement model) using a noise-sensitive algorithm that introduces a systematic way to react in the presence of uncertainties by incorporating higher-order statistics of the performance measure within the policy optimization. The resulting feedback law not only exploits all the couplings between the degrees of freedom of the robot for optimal tracking of kino-dynamic movement plans, but also provides the robot a policy sensitive to the type of uncertainty it can find in contact interaction tasks.

In summary, a motion planning and feedback control architecture for legged robots based on efficiently exploiting the mathematical structure of the problems with dedicated solvers is proposed. This contribution brings the state-of-the-art closer to the vision of real-time whole-body control for robots with arms and legs in complex and highly dynamic environments.

ZUSAMMENFASSUNG

Diese Dissertation präsentiert eine optimale steuerungsbasierte Architektur für die Bewegungsplanung und Rückkopplungssteuerung simulierter Roboter in Multikontaktszenarien. Bewegungsplanung und -steuerung sind grundlegende Bausteine für die Erstellung wirklich autonomer Roboter. Während in diesen Bereichen enorme Fortschritte für Manipulatoren mit festem Sockel und Radrobotern in den letzten Jahren erzielt wurden, besteht das Problem der Bewegungsplanung und -steuerung für Roboter mit Armen und Beinen immer noch ein ungelöstes Problem, das die Notwendigkeit effizienterer und robusterer Algorithmen belegt. In diesem Zusammenhang wird in dieser Dissertation eine Architektur vorgeschlagen, mit der zwei Hauptherausforderungen angegangen werden sollen, nämlich die effiziente Planung von Kontaktsequenzen und Ganzkörperbewegungen für Floating-Base-Roboter sowie deren erfolgreiche Ausführung mit Rückkopplungsregelungsstrategien, die Umgebungsunsicherheiten bewältigen können.

Einer der ersten Schritte zur erfolgreichen Verwirklichung komplexer Bewegungsabläufe ist die Fähigkeit, diese effizient zu synthetisieren. Wenn dies in Echtzeit möglich ist, kann der Roboter basierend auf seinem aktuellen Zustand reaktiv gesteuert werden. Herkömmliche Ansätze zur Optimierung der Ganzkörperflugbahn ermöglichen es, die Fähigkeiten des Roboters voll auszunutzen, jedoch sind sie für die Online-Ausführung sehr zeitintensiv. Um dieser algorithmische Herausforderungen in Angriff zu nehmen, nutzt die in dieser Dissertation vorgeschlagene Bewegungsplanungsarchitektur die geometrischen Eigenschaften des Problems, um eine lokale Lösung mit einem Polynomialzeit-Algorithmus effizient anzunähern, ohne jedoch die Ausdruckskraft für die Synthese komplexer Bewegungsverhalten abzuwägen. Mit dem vorgeschlagenen Ansatz wird gezeigt, dass ein breites Spektrum an zeitoptimalen hochdynamischen Bewegungen sowohl für humanoide als auch für vierbeinige Roboter erzeugt werden kann.

Basierend auf der effizienten Berechnung von Lösungen für die dynamische Machbarkeit von Bewegungsplänen und dank der konvexen Konvergenzeigenschaften wird außerdem gezeigt, dass es auch möglich ist, Kontaktsequenzen in komplexen Gebieten basierend auf einer Messgröße von dynamische Robustheit mithilfe eines Mixed-Integer-Solvers optimal auszuwählen. Dies ist ein sehr wichtiger Schritt bei der Gestaltung von Kontaktplanern, die über herkömmliche Ansätze hinausgehen, die auf quasi-statischen Stabilitätsannahmen basieren, und die in komplexeren und dynamischeren Szenarien verwendet werden können, z. B. beim Überqueren einer großen Lücke, bei einem steilen Abhang oder bei der reaktiven Auswahl eines neuen Kontaktorts in eine Push-Recovery-Aufgabe.

Trotz der Vorteile einer schnellen Kontakt- und Bewegungsplanungsschleife bei der Verbesserung der Erfolgsrate der durchgeführten Bewegungen, aufgrund von Umgebungsunsicherheiten und den noch nicht in Echtzeit erfolgenden Neuplanungsraten der Architektur ist eine Rückkopplungssteuerungsarchitektur für die Ausführung von Bewegungsplänen von Vorteil und kann erheblich zur erfolgreichen Durchführung von Bewegungsaufgaben beitragen. Zu diesem Zweck wird eine Rückkopplungssteuerungsarchitektur vorgeschlagen, um Bewegungspläne in Aufgaben, die Multikontaktinteraktionen umfassen, auszuführen. Die Architektur baut auf zwei Schlüsselelementen auf: Erstens nutzt sie die Überlagerungseigenschaft von Impedanzen zur Stabilisierung gewünschter Bewegungspläne aufgrund ihrer kinodynamischen Natur. Zweitens kann sie Rückkopplungsverhalten erzeugen, die an den Unsicherheitstyp angepasst sind (z. B. innerhalb eines Dynamik- oder Messmodells). Dazu wird ein geräuschsensitiver Algorithmus verwendet, der eine systematische Methode zur Reaktion auf Ungewissheiten einführt, indem Statistiken der Leistungsmessung höherer Ordnung in die Optimierung einbezogen werden. Das daraus resultierende Regelgesetz nutzt nicht nur alle Kopplungen zwischen den Freiheitsgraden des Roboters zur optimalen Verfolgung kino-dynamischer Bewegungspläne, sondern bietet dem Roboter auch ein Regelgesetz, das auf die Art der Unsicherheit anspricht, die er bei Kontaktaufgaben finden kann.

Zusammenfassend wird eine Architektur für Bewegungsplanung und Rückkopplungssteuerung für Floating-Base-Roboter vorgeschlagen, die auf einer effizienten Ausnutzung der mathematischen Struktur der Probleme mit dedizierten Lösern basiert. Dieser Beitrag hat uns die Vision der Echtzeit Ganzkörper Steuerung für Roboter mit Armen und Beinen in komplexen und hochdynamischen Umgebungen näher gebracht.

ACKNOWLEDGEMENTS

The presentation of this dissertation is one step more in my search for academic growth and in the development of the person I strongly desire to be since my childhood. The help of many people made this long journey possible, and this is the reason why I would like to use this space to express them my gratitude.

First of all, I would like to express my deepest gratitude to my Doktorväter, Prof. Dr. Ludovic Righetti. I would like to thank him for giving me the opportunity to do a PhD within his group at the Max-Planck Institute for Intelligent Systems. During these years, his ideas and vision inspired me, his happiness and excitement for science motivated me, and his aim for excellence and endless knowledge challenged and encouraged me for it as well. This dissertation would certainly not have been possible without him. It has been an honor and a pleasure to work with him.

I would like to thank the members of my committee for reviewing this dissertation, for their invested time and effort, and for coming together to my doctoral defense.

I am thankful to Prof. Dr. Stefan Schaal for sharing his deep knowledge and unconventional ideas, as well as his honest and thoughtful comments. A special thanks to Leila, whose support beyond academics helped me to get here.

This work would not have been possible without the contributions and support of countless people. I would like to thank all my friends and members of the Movement Generation and Control Group at the Max Planck Institute for Intelligent Systems in Tübingen, the Machines in Motion Lab at New York University, and the CLMC Lab at USC in Los Angeles. I am deeply thankful to my collaborators and co-authors: Alex, Jeannette, Daniel, Manuel, Franzi, Sean, Nick, John, Vince, Max, Majid, Andrea, Felix, Joel, Vincent, Bharath, Bilal, Julian, Alonso, Sebastian, Alina, Christina, Dominik, Giacomo, Lahiru ... I am greatly thankful to Yu-Chi and Lukas wo worked with me and contributed to the results of this dissertation.

I would like to thank my parents that have been my strength and one of my greatest sources of inspiration. They taught me that it is possible to realize our childhood dreams as a product of the effort and passion professed to one's activities. I also thank my little sister, because since the moment she came into my life, she has been an endless source of happiness that enlightened my best and worst days. Thanks to my family for their support and encouragement to pursue my dreams despite the implications of this decision such as stop sharing the daily meals and celebrating each other's birthday.

Brahayam Ponton
Tübingen, July 2019

Financial Support

This research was supported by University of Tuebingen, New York University, the Max-Planck Society, the European Union's Horizon 2020 research and innovation programme (grant agreement No 780684 and European Research Council's grant No 637935), and the US National Science Foundation grant DCSD 1825993.

CONTENTS

Abstract i

Zusammenfassung iii

Acknowledgements v

Preface 1

1 Introduction 3

1.1 Scope of research work . . . 4

1.2 Planning of contact sequences . . . 6

1.3 Optimal control for feedback design . . . 8

1.4 Motion planning for floating-base robots . . . 9

1.5 Numerical optimization and computational complexity . . . 11

1.6 Background material . . . 12

2 Contributions 19

2.1 Contacts and motion - planning and control architecture . . . 20

2.2 Contribution . . . 23

2.3 List of contributions . . . 27

3 Dynamic Motion Planning 29

3.1 Introduction . . . 30

3.2 Optimization of Dynamic Motion Plans . . . 32

3.3 Optimization of Contact Plans . . . 39

3.4 Experimental Results . . . 43

3.5 Discussion . . . 54

3.6 Conclusion . . . 55

4 Kinematic Motion Planning 57

4.1 Introduction . . . 58

4.2 Optimization of Kinematic Motion Plans . . . 60

4.3 Experimental Results . . . 67

4.4 Discussion . . . 74

4.5 Conclusion . . . 74

5 Feedback Control Design 75

5.1 Introduction . . . 76

5.2 Related Work . . . 77

5.3 Problem Formulation . . . 79
5.4 Algorithmic Derivation . . . 81
5.5 Feedback Control Architecture for Kino-Dynamic Movement Plans . . . 92
5.6 Experimental Results . . . 97
5.7 Discussion . . . 107
5.8 Conclusion . . . 107

6 Summary and Conclusion 109
6.1 Summary . . . 110
6.2 Conclusion . . . 111

Bibliography 113

PREFACE

This dissertation proposes an optimal control based architecture for motion planning and feedback control for robots with arms and legs. It consists of six chapters. Chapter 1 is an introductory chapter that highlights the potential of legged robots in our society, but also describes the problems for contacts planning, whole-body motion generation and feedback control design, which embeds this dissertation into the context of related approaches and motivates the open challenges to be addressed in the following chapters. It also defines the scope of this research in terms of the type of problems considered and the selected optimization-based approach. Chapter 2 formally presents the proposed motion planning and feedback control architecture, as well as a summary of the contributions. Chapters 3, 4 and 5 present the theoretical and experimental contribution of this dissertation in terms of contacts and whole-body motion generation algorithms, as well the feedback control architecture for stabilization of whole-body movement plans. Finally, Chapter 6 summarizes and concludes this dissertation.

Chapter 1

INTRODUCTION

My heart is on the work!
— Andrew Carnegie, Scottish - American industrialist
and philanthropist, 1835 - 1919

Given the rapid development of our society, it is not unrealistic to believe that in a not too distant future, robots will have an important place in our daily lives by supporting our society within a broad spectrum of applications ranging from industry and service to common household scenarios and elderly care [1]. Consequently, the design of algorithms for efficient planning of complex whole-body robotic behaviors as well as of feedback control strategies for the safe operation of robots among humans in unstructured, uncertain and highly dynamic environments are fundamental pieces for the future of autonomous robots [2, 3, 4] and really exciting research problems with far-reaching applications in our society.

For instance, robots (guided by high-level task descriptions) could assist their human supervisors: in factory or industry settings within intensive or repetitive manufacturing tasks [5, 6, 7, 8], in construction sites with transport and operation of heavy tools throughout the environment [9], in rescue or disaster relief scenarios within risky and very dangerous tasks for humans [10, 11, 12], in elderly care with daily chores and company [13], among many others. In the way to fulfilling our vision of robots that achieve human levels of competence at movement generation and control of complex tasks, and are thus capable of successfully filling-in many useful roles in society, we first need to understand the underlying principles that would allow us to overcome the major challenges for efficiently planning and controlling the task-oriented movement of a robot with arms and legs.

Among the major challenges that a legged robot faces when performing a task are to decide: where and when to contact the environment to avoid falling, in which direction and how fast to move its body (specially in the case of highly dynamic motions such as jumps), how compliant to behave with an uncertain environment, among others. A freely moving legged robot needs to coherently coordinate its movement and its interaction with the environment via intermittent physical contacts to successfully perform a desired task while keeping its balance. Furthermore, in real-world scenarios it needs to quickly adapt to uncertainties and react to sudden changes in the environment to successfully complete the desired task. In this context, this dissertation proposes an architecture that tries to provide answers to these questions by addressing the following problems: planning of contact sequences based on a criteria of dynamical robustness, efficient generation of time-optimal movement plans and whole-body feedback control design for robots performing multi-contact interaction tasks in uncertain environments.

1.1 SCOPE OF RESEARCH WORK

This section describes the main focus and some key aspects and assumptions of the architecture proposed in this dissertation.

Models based on physical knowledge: One of the core challenges for a legged robot is to keep its balance: to move in a direction different than gravity, the robot needs to interact with the environment via contact forces limited by the mechanical laws of unilateral contact [14]: e.g. feet can only push and not pull on the ground, thus arbitrary motions are not possible [15]. For this reason, the proposed architecture makes use of mathematical models that express physical knowledge about the problem to quickly and continuously answer questions such as: where and when to break or make a contact, in which direction and how fast to move, or how hard or compliant to push against the environment? and in this way it prevents the robot from falling and successfully achieves a task.

Robot as floating-base rigid-body system: Traditional sampling-based approaches would try to answer these questions by first selecting a sequence of contacts and joint configurations in a discrete approximation of the continuous space, and then the sequence of controls that would realize the sampled trajectories [16]. The difficulty of applying this approach to a legged robot is that it is subject to nonlinear dynamic constraints such as friction forces, under-actuation (meaning that there is no direct control over the robot's pose in space "the floating base") or equations of motion constraints (that describe a robot as a tree of rigid bodies connected through joints) [17, 18]; which implies that either the sequences found could be inconsistent with the nonlinear dynamic constraints or the difficulty of sampling consistent sequences or trajectories in highly dimensional spaces would be a limiting factor. This makes traditional sampling-based motion generation algorithms not well suited for the kind of problems considered in this dissertation, such as whole-body motion generation for floating-base systems.

Discrete nature of contact models: Another challenge of motion planning for legged robots is the inherent discrete nature of contacts switching: they introduce instantaneous events (making or breaking of a contact) due to which the system exhibits hybrid dynamics [19] (including continuous and instantaneous changes) that are difficult to handle during planning, estimation and control phases [11, 20]. Despite the challenges they pose, physical contacts are the at the core of the proposed architecture, because they allow a robot to interact with the environment and in this way to dynamically move and control its floating-base. Here, they are considered to be rigid, static and perfectly known by the motion planning algorithm and soft by the feedback control method and within the physical simulation. In theoretical and practical terms, flat ground walking of legged robots has been robustly demonstrated [21, 22, 23, 24, 25]. However, going to a broader range of tasks where robots interact with a non-coplanar environment using multiple endeffectors still remains a challenging problem and is part of the main focus of this dissertation.

Model-based trajectory optimization: One of the main goals of this dissertation is to design a contacts and whole-body motion planning algorithm for robots with arms and legs that perform tasks by physically interacting with the environment through the creation of intermittent contacts. The approach for motion planning of the proposed architecture belongs to the family of model-based optimal control techniques that describe a task by means of a performance criteria and a set of task constraints [26, 27, 28]; i.e. given a user-defined performance criteria in terms of features of the robot's state and environment, the proposed architecture computes state and control sequences that maximize the performance measure while satisfying task constraints given by the models of system dynamics and contact geometries. The approach is limited to locally optimal solutions for floating-base robots such as humanoids or quadrupeds [29, 30, 31, 32]. It is further assumed perfect knowledge about mass distribution of robot models, on which model-based control approaches rely.

Feedback policy based on Bellman's optimality principle: Besides motion planning, to execute a complex behavior, the proposed architecture processes noisy sensory information and using its control policy decides upon the appropriate actions. For instance, a robot estimates its state (e.g. joint configuration and floating-base pose) and the state of the world (e.g. the locations of stepping stones) by processing sensor readings. Then, based on the estimated state, the architecture decides over the control actions (e.g. joint torques) that would allow the robot to track the desired motion plan [33, 34, 35]. A second goal of this dissertation is then to design a feedback control module that on the one hand is capable of exploiting the kino-dynamic nature of motion plans by means of the superposition property of desired task impedances, and on the other hand is capable of reasoning about noise, the cost of this uncertainty and of adapting its control policy according to the environmental uncertainty (e.g. noise within its dynamics or measurement model) to control the interactions of the robot with its uncertain environment. The world is not considered to be perfectly known within the physical simulator and the focus of the approach is on locally optimal control policies based on applying Bellman's principle of optimality around the nominal planned trajectories [36].

Computationally tractable algorithms: Finding computationally efficient solutions to these optimal control problems [37, 38] is also the focus of this dissertation, because the applicability and effectiveness of the developed algorithms at reactively controlling a robot will to a great extent depend on it [30, 39, 40].

Experimental validation: The effectiveness of the proposed architecture and the validity of the assumptions will be demonstrated through experimental validation on a simulated torque controlled humanoid robot [41].

OUTLINE

This dissertation proposes an architecture for motion planning and control of legged robots in multi-contact scenarios composed by two modules: one for contacts and whole-body motion planning, and other for feedback control (specially designed based on the kino-dynamic nature of motion plans and to cope with environmental uncertainty). In the remainder of this chapter, some of the challenges of the addressed problem will be introduced in more detail and standard planning and control approaches will be discussed to place the contributions into context.

Section 1.2 discusses the difficulties involved in the planning of contacts sequences for legged robots. Techniques for design of feedback controllers are introduced in Section 1.3. In Section 1.4, whole-body motion planning approaches will be reviewed to motivate the design of the proposed architecture in the following chapters. Section 1.5 will examine numerical methods and computational complexity and finally, a brief overview of background material is presented in Section 1.6. In the following, traditional approaches to contacts and motion generation and control will be discussed to highlight the open problems that this dissertation will address.

1.2 PLANNING OF CONTACT SEQUENCES

Physical interaction via intermittent contacts allows a robot to dynamically navigate an environment. A rich contact interaction with the world using multiple endeffectors is favorable in terms of controllability and balance; however, it also presents challenges within its planning phase, such as: it requires computationally demanding balance checks due to the non-coplanar nature of contacts in complex terrains, and the problem's planning complexity increases combinatorially because any combination of the available endeffector contacts can be selected [42, 43, 44].

To overcome these challenges, in the computer graphics community the first successful contributions to contacts and (in configuration-space) motion planning techniques came from the animation of virtual characters that exploited motion graphs built based on motion capture data [45], or its combination with early locomotion controllers [46]. They excelled at generating realistic motions, but were limited by the capture data that could not easily generalize to unforeseen scenarios. In the robotics community, the emphasis was initially placed on real-time contact planners for flat-terrain bipedal locomotion [47], but is now starting to focus on the still unsolved general acyclic contacts planning problem [48, 49].

To tackle the combinatorial problem of planning acyclic contact sequences, [50] suggested its formulation as two simultaneous sub-problems: the search of a guide trajectory for the robot's floating-base, and the selection of balanced contact configurations along this trajectory. Based on this problem's description later works proposed solutions that reached a compromise between planning time and optimality, such as solving both sub-problems sequentially [51, 52, 53] or simultaneously either over a restricted search space that limits the discrete set of possible contacts [54] or over a continuous search space that relaxes the problem and relies on high-level heuristics to converge to a local solution [55]. A particularly interesting solution to the sequential approach was proposed in [48] because of its guarantees (offline motion guide trajectory guarantees the existence of a contact sequence to generate it) and real-time capabilities (online selection along the guide trajectory of a contact sequence that optimizes a desired performance measure). These approaches however rely on static stability criteria to scale down the computational demands of selecting a contact sequence, thus favoring time complexity over dynamic feasibility of contact plans [49].

Other interesting line of research are contact planners based on mixed-integer solvers, capable of modeling discrete changes in the contact state using integer variables, but also continuous postural adjustments and contact forces. They introduce in this way the possibility to optimize a discrete sequence of contacts using a measure of dynamic stability and robustness, previously not considered. For instance, [56, 57] demonstrated the advantages of simultaneous adaptation of discrete gait patterns and continuous force-motion trajectories in a humanoid robot locomoting over flat-ground by using a simplified dynamics model (Linear Inverted Pendulum) as a criteria to evaluate the desirability of a gait pattern within a mixed-integer solver. [58] did not include a dynamics model, however it proposed a novel approach to select footsteps on a complex uneven terrain using integer decision variables to model the set of convex terrain regions, over which it is possible to step.

From what we have discussed, it is evident that a measure of dynamic stability and robustness is important at selecting contact sequences that support dynamic motions such as walking down a steep slope, crossing a wide gap, walking on uneven terrain or recovering from a push. To this end, this dissertation extends the state-of-the-art by proposing a mixed-integer program that efficiently plans short contact sequences over complex uneven terrains using an approximate centroidal dynamics model as a measure of dynamic feasibility. Further, in [44] we have shown that significant speed-up can be achieved by training neural networks at predicting the evolution of the robot centroidal momenta, which can then be used for the generation of dynamically robust contact sequences using a search-based planner.

1.3 OPTIMAL CONTROL FOR FEEDBACK DESIGN

The theory of optimal control evolved from Pontryagin's [59] and Bellman's [36] optimality principles. This section will review standard optimal control techniques based on Bellman's principle of optimality as they have been a very successful and widely used tool in robotics for trajectory generation and control of legged robots [31, 39, 60].

Bellman's approach consisted in decomposing a problem into smaller nested sub-problems in such a way that the optimal control action for any sub-problem is independent of the way one arrived to the current state and minimizes the sum of the cost at the current state plus the least cost incurred at the state at which one arrives as a consequence of this action (nested sub-problem) [36, 61]. The recursive and backwards in time application of this principle (to build an overall solution "the control policy") leads in the discrete case to the Dynamic Programming algorithm [62], very successful at solving many multi-stage decision problems. The global nature of the approach however makes it intractable for high-dimensional or in continuous state space-defined problems due to the curse of dimensionality [63]. In this case, maintaining a single trajectory that can be iteratively improved along its neighborhood based on the optimality principle falls back to local methods such as Differential Dynamic Programming (DDP) [64] or iterative Linear-Quadratic Gaussian (LQG) [65], capable of successfully synthesizing motions for large dimensional robotic systems [30, 31, 39, 65, 66, 67].

Typically, these techniques optimize a local control policy by iteratively solving a low-order Taylor approximation of the nonlinear problem, as in [67, 68, 69]. Its sequential nature (meaning that they optimize only over control policies, in contrast with trajectory optimization methods that do it over control-state trajectories) limits its applicability as state constraints cannot be easily incorporated; however, as they build a local approximation of the cost-to-go, a feedback law for closed loop control can be derived to stabilize the implicitly defined state trajectories.

These standard optimal control approaches do not consider or are indifferent to the statistical properties of the noise that affects the system, i.e. control design is independent from noise which is only handled by the estimator. In [70], Jacobson proposed a risk-sensitive optimal control technique that broke this paradigm and made the resulting feedback law explicitly dependent on the statistical properties of the noise by including higher-order statistics of the performance measure in the policy optimization. It was shown that under low noise levels standard and risk-sensitive policies agree, but they differ the more as noise increases. [65, 71, 72] also proposed a noise-sensitive algorithm, synthesized by considering multiplicative noise in the parameters of its model, which allowed them to explain the robust behavior of human muscle models at movement generation and control in noisy environments. [73], further extended Jacobson's idea to nonlinear continuous-time systems using iterative methods.

Further efforts centered on sensitivity for systems under measurement noise have produced finite dimensional solutions under special cases [74, 75, 76], but also general solutions [77, 78]. Its applicability to robotic domains however has not been shown, one of the reasons being that it relies on the definition of risk-sensitive estimators, for which it is not clear how its mathematical construction will affect in practice the control of a legged robot. In this context, this dissertation proposes a scheme for feedback control of movement plans that exploits the superposition property of impedances based on the kino-dynamic structure of motion plans [79]. It further exploits a novel formulation of a controller sensitive to measurement noise that maintains the usual definition of estimators to model and control the dynamic interaction of a humanoid robot with an uncertain environment (fundamental in robotics) as an optimal control problem with measurement uncertainty (e.g. uncertain distance to contact locations).

1.4 motion planning for floating-base robots

This section will review trajectory optimization methods, that in contrast to sequential optimal control techniques (discussed in the previous section), attempt to find a state-control trajectory instead of a control policy, which makes them well suited to handle constraints over states and controls [28]. This is of particular interest in motion planning problems for floating-base robots as it allows to easily express e.g. contact properties such as friction force constraints or task-relevant state-space limits such as kinematic or actuation restraints.

A common trajectory optimization approach for dynamic locomotion is a sequential contacts-movement architecture [80, 81, 82] that in the first place optimizes a contact plan (as studied in section 1.2) and then a whole-body movement [82, 83]. Success stories [80, 81, 82, 83] of this approach motivate its use within the proposed architecture, as it allows to break down the problem's complexity without sacrificing neither the notions of global optimality and dynamical robustness when selecting contact surfaces nor the possibility to further optimize contact locations when optimizing the whole-body movement plan.

A simultaneous contacts-movement architecture has also been successfully implemented [29, 55, 84, 85, 86, 87]. In these approaches, discrete in nature contact decision variables are carefully introduced so as not to harm convergence properties of gradient-based optimization routines. For instance, contacts are modeled using complementarity constraints [29, 84] that imply that non-zero forces are only possible if the distance to the contact surface is zero (and viceversa), or using smooth contact models [69, 88] as in [55, 85, 86]. These contact models e.g. may require a compromise between physical accuracy and numeric stability [14], or could miss solutions that exploit force redundancy. As a consequence, simultaneous contacts-movement approaches that on the positive side can fully leverage the robot's capabilities, also require careful initialization and parameterization to converge and are still computationally costly for close to real-time control [40, 86, 89, 90].

More recent trajectory optimization approaches exploit the strong torque capabilities of nowadays robots and instead of considering the full dynamics equations of motion of a floating-base robot, they look into its structure and focus on its key part, the centroidal momentum [91]. The equations of motion describe how contact interaction forces affect the kinematic state of all rigid bodies. For instance, the dynamics of a single rigid body can be described in terms of its momentum: changes to it are determined by forces acting on the body, but it also relates to the body velocity via a spatial inertia tensor [18]. Consequently, for a floating-base system, in contrast with a fixed-base manipulator, the only way to move and steer its floating-base body is by applying forces on the environment. This suggested to consider only the dynamics of the floating-base body expressed at the center of mass (centroidal dynamics) for motion planning, as the rest of degrees of freedom can be resolved with techniques used for manipulators assuming enough torque authority [84, 92]. The centroidal dynamics [91, 93] have thus become very popular, as they serve as the bridge between the interaction of the robot with the environment through external contact forces and its motion in the world [23, 79, 84]. They capture the couplings between a full kinematics model and the Newton-Euler equations of the robot's center of mass (CoM), which are required to describe limb motions and contacts in 3d spaces [20].

Several works [29, 84, 94, 95, 96, 97, 98] have realized about the importance of time-optimality in movement plans, not only to improve motion trajectories in terms of smoothness and robustness, but also to make possible motions that would otherwise be infeasible. For instance, highly constrained motions such as non-coplanar walking under surfaces with low friction coefficient would be impossible if time cannot be adapted, so as to allow the robot to move along a trajectory consistent with the limited available contact forces [99]. Time optimization plays also a crucial role in highly dynamic motions such as jumps with full flight phases, because momentum conservation implies that e.g. the CoM motion cannot be changed as no contact is available. For instance, to perform a jump without optimization of timings, the robot would need to propel itself high enough such that its CoM does not end up under the ground after landing, which might require excessively large contact forces. In contrast, if timings are decision variables, the robot can decide upon the appropriate flight time to e.g. minimize contact forces and satisfy torque constraints [32, 84]. But because trajectory optimization problems can quickly become intractable, specially in robotic applications that involve large dimensional systems, including time optimization has been usually ignored due to its high computational demands.

The architecture proposed in this dissertation reasons about the centroidal dynamics of the robot to generate time-optimal whole-body movement plans. Inspired by the kino-dynamic structure proposed in [79], this architecture designs specialized numerical solvers for both sub-problems exploiting their underlying mathematical structures. As will be shown in later chapters, this methodology can very efficiently optimize time-optimal contact force and whole-body movement trajectories, as well as respect torque actuation limits if required.

1.5 NUMERICAL OPTIMIZATION AND COMPUTATIONAL COMPLEXITY

Numerical optimization is a fundamental tool in decision science with countless applications in many fields [37, 38]. Transferring its principles to robotic domains is still a work in progress, because of the nonlinear and hybrid nature of robotic systems that cannot be neglected and that solutions are required at fast timescales [19]. However, the everyday increasing computational power of modern personal computers and the technological maturity and availability of efficient optimization software [100, 101, 102, 103, 104] have promoted the use of numerical optimization tools in many robotic applications, as they allow to easily model problems as the optimization of an objective function (a quantitative measure of the performance of the system in terms of features of the optimization variables) subject to constraints, that describe the set of feasible solutions.

The apparently straightforward formulation of the optimization problem is a very important step that will play a fundamental role throughout this dissertation. If the problem description is too simple, it might be of limited applicability; but if it is too complex, it might be too difficult to solve [38]. For instance, trajectory optimization problems based on the full equations of motion of the floating-base rigid-body system are non-convex [38], very difficult to solve and in most cases involve a compromise between very long computation times and not always finding a solution. On the opposite end of the spectrum, less detailed models (that describe only the essence of a behavior such as optimization problems based on the linear inverted pendulum model) [105, 106] are convex [37], very efficiently and reliably solvable, but limited in terms of the behaviors that they can synthesize. Models based on the centroidal dynamics constitute a middle-ground level of complexity between these two extremes [79, 84] and while they are still non-convex, its structure can be exploited to derive accurate but efficient approximations with convex convergence properties, as will be shown in this dissertation.

The nonconvexities due to the bilinear expressions of the dynamic constraints imposed by the centroidal momentum dynamics appear in the optimization problems in the form of quadratic equality constraints (linear either in the states or in the controls, but not in both [107]). The algorithms used in the proposed architecture for time-optimal whole-body motion planning exploit this special problem structure by approximating the solution of the originally non-convex trajectory optimization problem via a sequence of convex second-order cone programs that can be very efficiently solved using a sparse implementation of an interior-point polynomial-time method for convex programming (based on [104, 108]). A convex second-order cone program is composed by a linear or quadratic objective function and by a set of constraints that include: linear equality and inequalities as well as convex quadratic inequality constraints.

In a similar way, the formulation of the combined estimation-control problem used to derive sensitive feedback controllers in chapter 4 has a bilinear structure (linear in the parameters of the estimation problem or in those of the control problem, but not in both). In this case however, an alternating procedure is used to solve the problem, because in this way the solution to one of the sub-problems, namely the estimation problem, is already provided by the known Kalman Filter and then only the control problem needs to be solved, which can be efficiently performed by adapting standard optimal control techniques.

Within the kinematics optimization problem, sources of nonconvexity such as obstacle avoidance constraints can also be found. In this case, the problem is decomposed in such a way that nonconvex constraints are treated separately from convex ones, using a nonlinear solver for unconstrained minimization LBFGS [109] for the nonconvex part and an efficient interior-point method for conic programming for the convex part. The solutions are combined together into an overall solution in the spirit of distributed consensus optimization [110, 111, 112, 113, 114, 115, 116]. To conclude, discrete decisions to select contact surfaces based on a measure of dynamical robustness are taken using a mixed-integer solver. To compare the difference among optimized contact plans depending on the dynamics model used to evaluate its desirability (e.g. including or not time optimization) a custom implementation of a mixed-integer solver based on a Branch and Bound method [117] that internally solves a centroidal dynamics problem using the previously mentioned algorithms is used. Mixed-integer solvers have in the worst case exponential-time complexity, but in the range of problems of interest in this dissertation they have shown good performance, which made them well suited for the applications.

1.6 BACKGROUND MATERIAL

This section will present background introductory material including a general formulation of the trajectory optimization problem, its notation (used throughout the following chapters) and, its structural analysis and reformulation as an alternating kino-dynamic optimization problem [79], extensively used in this dissertation.

1.6.1 *Trajectory Optimization Problem*

A general formulation of the trajectory optimization problem for motion planning can be stated as follows [29, 31, 55, 87]:

$$\begin{aligned} \min_{\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\tau_\mathrm{j},\lambda} \quad & \phi_\mathrm{T}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\tau_\mathrm{j},\lambda) + \sum_{\mathrm{t}=1}^{\mathrm{T}-1} \phi_\mathrm{t}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\tau_\mathrm{j},\lambda) && \text{(1a)} \\ \text{subject to} \quad & \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{S}^T\tau_\mathrm{j} + \mathbf{J}(\mathbf{q})^T\lambda && \text{(1b)} \\ & \text{Kinematic Limits} && \text{(1c)} \\ & \text{Actuation Limits} && \text{(1d)} \end{aligned}$$

It minimizes the sum of a terminal cost ϕ_T and a running cost ϕ_t (1a), that expresses a cost over the robot kinematic (posture $\mathbf{q}$, velocity $\dot{\mathbf{q}}$ and acceleration $\ddot{\mathbf{q}}$) and dynamic variables (torques τ_j and forces λ), while satisfying dynamics constraints (1b), kinematic (1c) and actuation limits (1d). Kinematic limits (1c) include: constraints relating postures $\mathbf{q}$, velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ defined for all robot's joints and the floating-base as well as acceleration, velocity or joint limits $\mathbf{q}_\mathrm{j} \in [\mathbf{q}_\mathrm{j}^{\mathrm{min}}, \mathbf{q}_\mathrm{j}^{\mathrm{max}}]$. Actuation limits (1d) include: torque constraints $\tau_\mathrm{j} \in [\tau_\mathrm{j}^{\mathrm{min}}, \tau_\mathrm{j}^{\mathrm{max}}]$, center of pressure constraints to ensure stationary contacts, friction force constraints for the endeffector not to slip, and contact constraints $\mathbf{J}_\mathrm{e}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_\mathrm{e}(\mathbf{q})\dot{\mathbf{q}} = \ddot{\mathbf{p}}_\mathrm{e}$ to keep the endeffectors stationary during contact (e.g. zero acceleration $\ddot{\mathbf{p}}_\mathrm{e} = 0$) or control its motion in cartesian space during flight phases (e.g. $\ddot{\mathbf{p}}_\mathrm{e} = \ddot{\mathbf{p}}_\mathrm{e}^{\mathrm{ref}}$) [92].

Finally, the dynamic constraints or equations of motion (EoM) for a floating-base rigid-body system are given by equation (1b) [17, 18]. A robot's posture defined as $\mathbf{q} = \begin{bmatrix}\mathbf{x}^T & \mathbf{q}_\mathrm{j}^T\end{bmatrix}^T$ is composed by the pose $\mathbf{x} \in SE(3)$ of a floating-base frame in the robot relative to an inertial frame, and by the joint positions $\mathbf{q}_\mathrm{j} \in \mathbb{R}^n$, where n denotes the number of actuated joints. $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{(n+6)\times(n+6)}$ stands for the inertia matrix of the rigid-body system, $\mathbf{N}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n+6}$ is a vector of nonlinear terms (including Coriolis, gravity and friction forces). The selection matrix $\mathbf{S} = \begin{bmatrix}\mathbf{0}^{n\times 6} & \mathbf{I}^{n\times n}\end{bmatrix}$ represents the system under-actuation [118], i.e. that the floating-base $\mathbf{x}$ is not directly actuated by joint torques $\tau_\mathrm{j} \in \mathbb{R}^n$, but indirectly only via contact interaction forces $\lambda = \begin{bmatrix}\dots & \mathbf{f}_\mathrm{e}{}^T & \gamma_\mathrm{e}^T & \dots\end{bmatrix}^T \in \mathbb{R}^{6\mathrm{e}}$ (e being the number of endeffectors) and the jacobian of the contact constraints $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\mathrm{e}\times(n+6)}$ formed by stacking endeffector jacobians $\mathbf{J}_\mathrm{e}(\mathbf{q}) \in \mathbb{R}^{6\times(n+6)}$. The pair $\begin{bmatrix}\mathbf{f}_\mathrm{e}{}^T & \gamma_\mathrm{e}^T\end{bmatrix}^T$ denotes the endeffector wrench (force and torque) with respect to the frame at the endeffector position $\mathbf{p}_\mathrm{e}$.

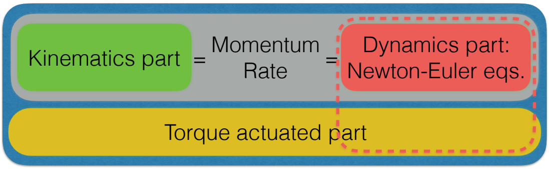


FIGURE 1: Block decomposition of the equations of motion (1b) of a floating-base rigid-body system (that describes a robot with arms and legs) into its kinematic, dynamic and torque actuation components. The gray box denotes the under-actuated dynamics equations 2a that can further be decomposed into its kinematic (green box) and dynamic (red box) components, as shown in equation 3, used in the alternating kino-dynamic motion optimization approach [79]. The yellow box denotes the actuated part of the equations of motion 2b. Finally, the dashed red box denotes the combination of Newton-Euler equations and a linearized version of the torque actuated constraints along a nominal kinematics trajectory.

1.6.2 *Structure in Equations of Motion*

This section presents an structural analysis of the EoM (1b) and how they can be reformulated in a way more amenable for structured optimization (as in Figure 1).

System Under-actuation

The nature of the selection matrix **S** leads to a dynamics decomposition of the EoM (1b) into an actuated (superscript a) and un-actuated parts (superscript u) [119], as:

$$\mathbf{M}^{\mathrm{u}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}^{\mathrm{u}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J}^{\mathrm{u}}(\mathbf{q})^{T}\lambda \tag{2a}$$

$$\mathbf{M}^{\mathrm{a}}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}^{\mathrm{a}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J}^{\mathrm{a}}(\mathbf{q})^{T}\lambda + \tau_{\mathrm{j}} \tag{2b}$$

Eq. (2a) (first 6 rows of EoM (1b)) relates the acceleration of the floating-base body located at $\mathbf{x} \in SE(3)$ to contact forces, i.e. external forces are needed to accelerate this body in a direction other than gravity. Eq. (2b) (last n rows of EoM (1b)) asserts that any combination of forces λ and accelerations $\ddot{\mathbf{q}}$ is possible, as long as they are consistent with (2a) and, satisfy kinematic and actuation limits (1c)-(1d) [17, 92]. Thus, under the assumption of enough torque authority eq. (2a) is sufficient to plan dynamically feasible motions [84]. In other words, robots with strong actuators can neglect eq. (2b) and use only eq. (2a) for motion planning, while robots with limited torque capability can benefit from it to plan contact force distribution across the available endeffectors in a way that satisfies torque limits.

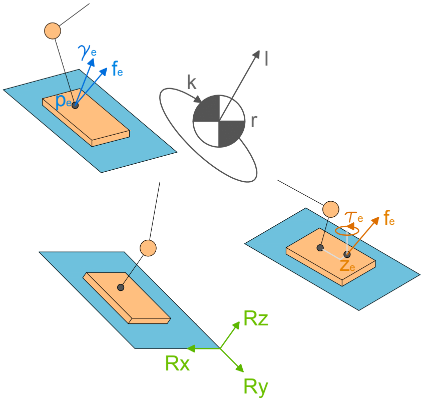


FIGURE 2: A summary of the notation used throughout this dissertation. It shows the center of mass $\mathbf{r}$, linear $\mathbf{l}$ and angular momentum $\mathbf{k}$. It also depicts control variables of an endeffector over a flat contact ($\mathbf{f}_e$, τ_e, $\mathbf{z}_e$), force, normal torque and center of pressure (in orange). These same control variables can also be expressed at the endeffector position $\mathbf{p}_e$ as a force $\mathbf{f}_e$ and torque γ_e (as shown in blue). Each axis of the endeffector orientation is shown in green.

Centroidal Momentum Dynamics

As noted earlier, eq. (2a) is sufficient for planning dynamically feasible motions. Its mapping to a frame at the robot CoM (3), makes its kino-dynamic structure apparent; namely, the rate of momentum around the CoM computed from the robot joint angles and velocities (kinematic quantities) matches the total wrench generated by external and gravitational forces (dynamic quantities) [84].

$$\underbrace{\frac{d}{dt}\left[\mathbf{A}(\mathbf{q})\dot{\mathbf{q}}\right]}_{\text{From kinematics}} = \begin{bmatrix}\dot{\mathbf{l}}\\ \dot{\mathbf{k}}\end{bmatrix} = \underbrace{\begin{bmatrix} m\mathbf{g} + \sum_e \mathbf{f}_e \\ \sum_e(\mathbf{p}_e + \mathbf{R}_e^{x,y}\mathbf{z}_e - \mathbf{r}) \times \mathbf{f}_e + \mathbf{R}_e^{z}\tau_e \end{bmatrix}}_{\text{From dynamics}} \qquad (3)$$

The dynamics centroidal momentum (right hand side (3)) is also known as Newton-Euler eqns. Fig. 2 shows a summary of some of its variables. Further, "m" denotes the robot mass and $\mathbf{g}$ the gravity vector. $\mathbf{R}_e^{x,y} \in \mathbb{R}^{3\times 2}$ and $\mathbf{R}_e^{z} \in \mathbb{R}^{3\times 1}$ are the first two and third columns of the rotation matrix $\mathbf{R}_e \in \mathbb{R}^{3\times 3}$ that rotates quantities from endeffector- to inertial frame. Local forces are given by ${}^{L}\mathbf{f}_e = \mathbf{R}_e^{T}\mathbf{f}_e$ and the connection between torques γ_e and τ_e is given by $\gamma_e = \mathbf{R}_e^{x,y}\mathbf{z}_e \times \mathbf{f}_e + \mathbf{R}_e^{z}\tau_e$. Within the kinematics centroidal momentum (left hand side (3)), $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{6\times(n+6)}$ is the centroidal momentum matrix [93] that maps joint velocities $\dot{\mathbf{q}}$ to momentum $\begin{bmatrix}\mathbf{l} \ \mathbf{k}\end{bmatrix}$.

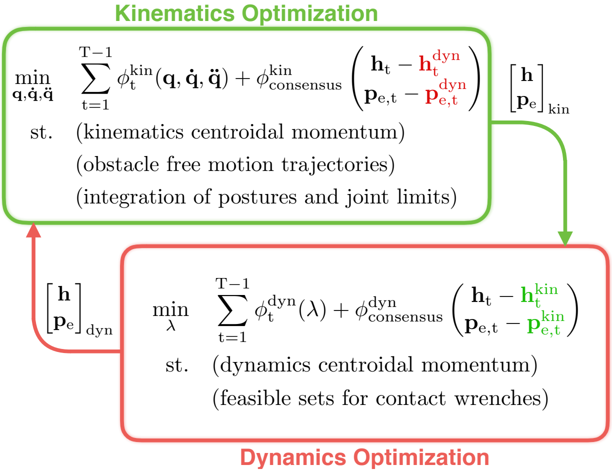

Kinematics Optimization
$\min_{\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}} \sum_{t=1}^{T-1} \phi_t^{kin}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) + \phi_{consensus}^{kin}\begin{pmatrix}\mathbf{h}_t - \mathbf{h}_t^{dyn}\\ \mathbf{p}_{e,t} - \mathbf{p}_{e,t}^{dyn}\end{pmatrix}$
st. (kinematics centroidal momentum)
(obstacle free motion trajectories)
(integration of postures and joint limits)
$\begin{bmatrix}\mathbf{h}\\ \mathbf{p}_e\end{bmatrix}_{kin}$
$\begin{bmatrix}\mathbf{h}\\ \mathbf{p}_e\end{bmatrix}_{dyn}$
$\min_{\lambda} \sum_{t=1}^{T-1} \phi_t^{dyn}(\lambda) + \phi_{consensus}^{dyn}\begin{pmatrix}\mathbf{h}_t - \mathbf{h}_t^{kin}\\ \mathbf{p}_{e,t} - \mathbf{p}_{e,t}^{kin}\end{pmatrix}$
st. (dynamics centroidal momentum)
(feasible sets for contact wrenches)
Dynamics Optimization

FIGURE 3: Schematic of kino-dynamic motion optimization approach [79] to iteratively compute contact force and whole-body trajectories until convergence of the common set of variables, namely robot centroidal momenta $\mathbf{h}$ and endeffector poses $\mathbf{p}_e$.

The whole-body trajectory optimization algorithm based on the kino-dynamic separability of the centroidal momentum equations was proposed in [79] and allows to find whole-body and contact force trajectories via an alternating procedure that independently optimizes kinematic and dynamic quantities, that only need to agree on a set of common variables: CoM, momentum and contact locations.

1.6.3 *Trajectory Optimization*

This section focuses on the trajectory optimization problem formulated as an alternating optimization between contact forces (dynamics optimization) and whole-body trajectories (kinematics optimization). This alternating kino-dynamic approach for trajectory optimization was originally proposed in [79]. It does not constitute a contribution of this dissertation, but it is widely used throughout it. Figure 3 schematically depicts its main idea; however, the specific details of the formulations may vary as the architecture proposed in this dissertation further extends this work.

First, the differential equations (3) are discretized into algebraic ones and then the optimization problems are formulated.

Dynamics Optimization

This section describes the problem formulation used to dynamically optimize a motion plan (that includes timings, contact wrenches and momentum trajectories) under the centroidal dynamics constraints (3) and that minimizes a desired objective. Formally, the goal is to search a local solution for problem (4).

$$\min_{\substack{\mathbf{h},\Delta,\mathbf{p}_e\\ \mathbf{f}_e,\mathbf{z}_e,\tau_e}} \sum_{t=1}^{T} \phi_t^{\text{dyn}} \begin{pmatrix} \mathbf{h}, \Delta, \mathbf{p}_e, \\ \mathbf{z}_e, \mathbf{f}_e, \tau_e \end{pmatrix} + \phi_{\text{consensus}}^{\text{dyn}} \begin{pmatrix} \mathbf{h}_t - \mathbf{h}_t^{\text{des}} \\ \mathbf{p}_{e,t} - \mathbf{p}_{e,t}^{\text{des}} \end{pmatrix} \tag{4a}$$

$$\text{subject to} \quad \mathbf{h}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{k}_t \\ \mathbf{l}_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \frac{1}{m}\mathbf{l}_t \Delta_t \\ \mathbf{k}_{t-1} + (\sum_e \kappa_{e,t})\Delta_t \\ \mathbf{l}_{t-1} + (m\mathbf{g} + \sum_e \mathbf{f}_{e,t})\Delta_t \end{bmatrix} \tag{4b}$$

$$\kappa_{e,t} = (\mathbf{p}_{e,t} - \mathbf{r}_t) \times \mathbf{f}_{e,t} + \gamma_{e,t} \tag{4c}$$

$$\gamma_{e,t} = \mathbf{R}_{e,t}^{x,y} \mathbf{z}_{e,t} \times \mathbf{f}_{e,t} + \mathbf{R}_{e,t}^{z} \tau_{e,t} \tag{4d}$$

$$\mathbf{p}_e \in \mu(\mathcal{S}) \tag{4e}$$

$$\Delta_t \in [\Delta_{\min}, \Delta_{\max}] \tag{4f}$$

$$\mathbf{z}_{e,t}^{x,y} \in \left[\mathbf{z}_{\min}^{x,y}, \mathbf{z}_{\max}^{x,y}\right] \tag{4g}$$

$$\|\mathbf{p}_{e,t} - \mathbf{r}_t\| \leq \ell_e^{\max} \tag{4h}$$

$$\left\| {}^{L}\mathbf{f}_{e,t}^{x} + {}^{L}\mathbf{f}_{e,t}^{y} \right\|_2 \leq \mu_f \, {}^{L}\mathbf{f}_{e,t}^{z}, \quad {}^{L}\mathbf{f}_{e,t}^{z} \geq 0 \tag{4i}$$

It minimizes a quadratic cost (4a) composed of a running cost ϕ_t^{dyn} that regularizes controls (such as wrenches or duration of timesteps Δ_t) and includes user-defined rewards (such as reaching a CoM position or moving through a waypoint), and a consensus cost $\phi_{\text{consensus}}^{\text{dyn}}$ that rewards good tracking of desired CoM and momentum trajectories.

The constraints (formulated for all active endeffectors and timesteps $\forall e, t$) include a dynamics model (4b)-(4d) that coupled with the consensus cost guide dynamic solutions towards kinematic ones, without explicitly including all kinematic variables, thus satisfying (3). Other constraints include: (4e) that constrains endeffector locations to assigned contact surfaces, to be detailed in 3.3; (4f) restricts time variables to a box constraint; (4g) maintains CoP within the endeffector support region; (4h) heuristically keeps the contact locations reachable from the CoM and avoids singularities due to over-extensions, and (4i) defines friction cones [120].

Kinematics Optimization

This section describes the problem formulation used to kinematically optimize a motion plan (including posture, velocity and acceleration trajectories) under the centroidal momentum dynamics (3) and that minimizes a desired objective. Formally, a local solution for problem (5) is to be found.

$$\min_{\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{h},\mathbf{p}_e} \quad \sum_{t=1}^{n} \phi_t^{\text{kin}}(\mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t) + \phi_{\text{cons}}^{\text{kin}} \begin{pmatrix} \mathbf{h}_t - \mathbf{h}_t^{\text{des}} \\ \mathbf{p}_{e,t} - \mathbf{p}_{e,t}^{\text{des}} \end{pmatrix} \tag{5a}$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{q}_t \\ \dot{\mathbf{q}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{t-1} + \dot{\mathbf{q}}_t \Delta_t \\ \dot{\mathbf{q}}_{t-1} + \ddot{\mathbf{q}}_t \Delta_t \end{bmatrix} \tag{5b}$$

$$\mathbf{q}_{j,t} \in [\mathbf{q}_j^{\text{min}}, \mathbf{q}_j^{\text{max}}] \ \forall j \in \mathbb{R}^n \tag{5c}$$

$$\mathbf{p}_{e,t} = \text{ForwKin}(\mathbf{q}_t) \ \in \mathcal{O}_{\text{free}} \tag{5d}$$

$$\mathbf{h}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{l}_t \\ \mathbf{k}_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \frac{1}{m} \mathbf{l}_t \Delta_t \\ \mathbf{A}(\mathbf{q}_t)\dot{\mathbf{q}}_t \end{bmatrix} \tag{5e}$$

This problem formulation minimizes the cost (5a) composed of a running cost ϕ_t^{kin} that regularizes controls (such as joint velocities and accelerations) and includes user-defined rewards (such as staying close to a desired robot posture at some trajectory point), and a consensus cost $\phi_{\text{cons}}^{\text{kin}}$ that rewards good tracking of CoM, momentum and endeffector trajectories.

The constraints (formulated for all endeffectors and timesteps $\forall e, t$) include a simple dynamics model (5b) that coupled with the consensus cost guide kinematic solutions towards dynamic ones, without explicitly including all dynamic variables, thus satisfying (3). Other optimization constraints include: (5c) that limits joint positions to its motion range; (5d) which expresses that endeffector locations computed using a forward kinematics algorithm should lie in obstacle free spaces; and (5e) that defines how CoM and robot momenta are computed using kinematic information. Δ_t denotes the timestep discretization between trajectory points.

The dynamics (equations (4)) and kinematics sub-problems (equations (5)) compose the kino-dynamic approach for optimization of contact force and whole-body motion trajectories that will be the focus within the following chapters.

In summary, this chapter introduced the topic and scope of this dissertation. It also motivated the importance and impact of legged robots in our society, and discussed the challenges to be faced by contacts and motion - planning and control approaches. Background material was also introduced in order to place into context of related works the architecture proposed in this dissertation. The following chapter will discuss in more detail the proposed planning and control architecture and will highlight its contribution to the state-of-the-art in the field.

Chapter 2

CONTRIBUTIONS

This chapter states the contributions of the architecture proposed in this dissertation to the state-of-the-art in the field. Section 2.1 introduces the planning and control architecture. In Section 2.2, each contribution is presented within the context of the overall architecture. Finally, specific contributions are listed in Section 2.3.

2.1 CONTACTS AND MOTION - PLANNING AND CONTROL ARCHITECTURE

This section presents the architecture proposed in this dissertation to optimize contact and movement plans as well as to control in closed-loop its execution, which will be described in detail within the following chapters. The proposed motion planning and control architecture is based on optimal control to describe in an easy yet general form the desired motion optimization problem and on dedicated solvers that exploit the problem's mathematical structure to efficiently find local solutions. Figure 4 highlights within red boxes the main components of the proposed architecture where theoretical contributions have been attained, namely: a contacts planning approach based on a measure of dynamical robustness, dedicated solvers that exploit the structure of the kino-dynamic motion planning approach, and feedback control design methods to execute whole-body movement trajectories.

Optimal control formulation

This dissertation proposes an optimization-based motion planning and control architecture that takes as input a high-level task description and outputs torque commands to successfully execute it on a physically simulated robot. This input-output mapping happens in an efficient yet general manner thanks to the architecture composed by three optimization-based controllers: for contacts planning, for time-optimal whole-body motion generation and for feedback control. Each controller, as shown in Figure 4, makes use of an appropriate level of model complexity so as to remain efficiently solvable, but at the same time expressive enough so as not to limit the range of possible behaviors that can be generated.

In general, if it were doable in an efficient manner, we (the robotics community) would like to solve problem (1) without approximations or trade-offs, but in practice it is intractable for real-time control. For this reason, in order to efficiently find solutions to the motion planning problem, the proposed architecture approximates a local solution using the structure depicted in Figure 4. In the first place, out of the description of a complex terrain, the proposed architecture efficiently selects a set of contacting surfaces that supports a contact sequence leading to a dynamically feasible motion. To this end, it makes use of a mixed-integer solver that internally evaluates the desirability of a contact sequence based on an efficient yet accurate approximation of the centroidal momentum dynamics model, which allows to find a globally optimal or ϵ-suboptimal set of contacting surfaces $\mathcal{S}_\mathrm{r}$.

This set can then be used in the second step to only locally optimize a time-optimal whole-body movement plan in a much more efficient way than if problem (1) were to be naively solved using an off-the-shelf solver that does not exploit its structure. This second phase, in contrast with the first one, includes a full kinematics model but does not include anymore the discrete decision variables to select contacting surfaces. The third and last stage of the architecture makes use of

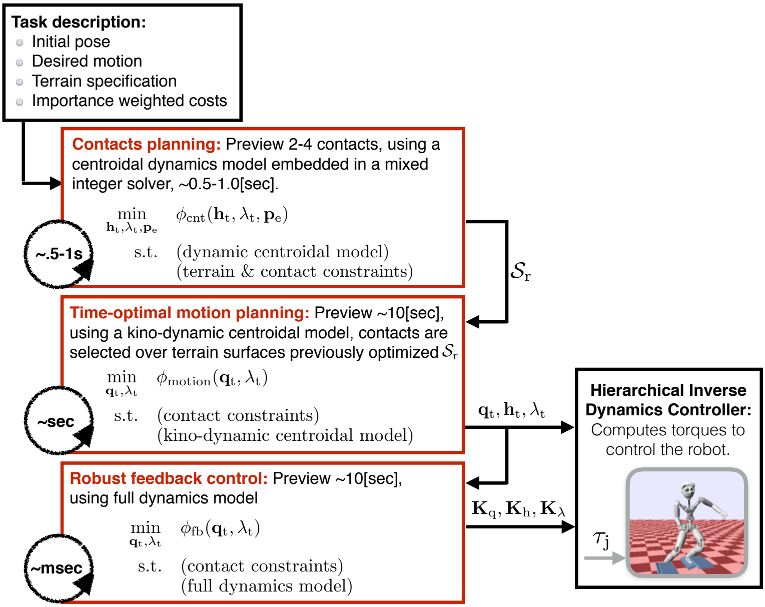

Task description:
Initial pose
Desired motion
Terrain specification
Importance weighted costs
Contacts planning: Preview 2-4 contacts, using a centroidal dynamics model embedded in a mixed integer solver, ~0.5-1.0[sec].
$\min_{\mathbf{h}_t,\lambda_t,\mathbf{p}_e} \phi_{cnt}(\mathbf{h}_t,\lambda_t,\mathbf{p}_e)$
s.t. (dynamic centroidal model)
(terrain & contact constraints)
~.5-1s
$\mathcal{S}_r$
Time-optimal motion planning: Preview ~10[sec], using a kino-dynamic centroidal model, contacts are selected over terrain surfaces previously optimized $\mathcal{S}_r$
$\min_{\mathbf{q}_t,\lambda_t} \phi_{motion}(\mathbf{q}_t,\lambda_t)$
s.t. (contact constraints)
(kino-dynamic centroidal model)
~sec
$\mathbf{q}_t, \mathbf{h}_t, \lambda_t$
Hierarchical Inverse Dynamics Controller: Computes torques to control the robot.
Robust feedback control: Preview ~10[sec], using full dynamics model
$\min_{\mathbf{q}_t,\lambda_t} \phi_{fb}(\mathbf{q}_t,\lambda_t)$
s.t. (contact constraints)
(full dynamics model)
~msec
$\mathbf{K}_q, \mathbf{K}_h, \mathbf{K}_\lambda$
τ_j

FIGURE 4: Proposed architecture for motion planning and control, highlighting within red boxes the components where theoretical contributions have been achieved. The input to the proposed architecture is a task description composed by the definition of the initial and desired final states, a given contact sequence to follow or a terrain description from where to select it, and a set of importance weighted costs that define the desirability of each term of the objective functions of the contacts and motion - planning and control trajectory optimization problems.

the previously optimized contact force, momentum and whole body trajectories (λ_t, $\mathbf{h}_t$, $\mathbf{q}_t$) to design impedance policies ($\mathbf{K}_\lambda$, $\mathbf{K}_h$, $\mathbf{K}_q$) for optimal tracking of the desired kino-dynamic behaviors even in the presence of uncertainty or model inaccuracies.

In its current state, the proposed architecture optimizes a motion and then executes it, i.e. the motion is optimized only once instead of in a receding horizon fashion. In spite of this limitation, it is capable of successfully generating and executing many motions. However, thanks to the computational efficiency of the algorithms composing the proposed architecture, it could be possible to use it in a receding horizon fashion (using a shorter time-horizon, benefiting from warm-starts using previous solutions and using updated measured states and constraints) to achieve close to real-time truly robust and adaptive control capabilities.

Cost and constraints

Task performance descriptors as well as feasible sets for the optimization variables constitute the inputs to the proposed architecture. Task performance descriptors allow to measure the quality of a solution and thus to select the best one among the set of feasible ones. They are built in terms of state and control features and express its desirability. For instance, within the architecture shown in Figure 4, the functions $\phi_{\text{cnt}}(\cdot)$, $\phi_{\text{motion}}(\cdot)$ and $\phi_{\text{fb}}(\cdot)$ express the desirability of state-control features (contact force λ_t, momentum $\mathbf{h}_t$, whole-body trajectories $\mathbf{q}_t$, and endeffector contacts $\mathbf{p}_e$) within the solutions of the contacts planning, combined contact force and motion planning, and kino-dynamic feedback design problems.

Problem constraints instead define the sets of admissible solutions without indicating any preference among them. For instance, in a kinematic optimization problem a task performance descriptor "cost" would describe the desirability of a robot posture over others, while a "task constraint" would describe e.g. joint limits, location of an endeffector over a surface to be in contact, or non-penetration obstacle constraints when not in contact. In a dynamics problem, a cost could e.g. describe the desirability of low energy solutions, while a constraint could describe the set of forces admissible to generate a motion or torque actuation limits.

Task feedback

The optimized time-optimal contact and movement plans define a desired nominal behavior for the robot motion in the environment (robot postures and velocities) as well as for the robot interaction with the environment (contact forces). The successful realization of these desired behaviors requires stabilization around the nominal trajectories to cope with environmental uncertainties and mismatches between the planning model and reality (e.g. a static and rigid contact model used for motion planning that differs from a spring-damper system contact model used for execution within the physical simulator).

The first idea that comes to mind is the use of hand-designed proportional-derivative (PD) feedback controllers to track the optimal feature trajectories; however, their applicability is limited and the required tuning effort might be high. For these reasons, the architecture proposed in this dissertation makes use of time-varying linear quadratic regulators (TV-LQR) for optimal tracking of the desired whole-body motion trajectories. They exploit all the couplings between the robot's degrees of freedom and the superposition property of impedances makes it possible to further exploit the kino-dynamic separability of motion plans for feedback control. Furthermore, these feedback control laws can be synthesized using noise-sensitive algorithms in order to allow the robot to select the appropriate impedance according to its knowledge about the uncertainty in the environment.

2.2 CONTRIBUTION

This section states the contributions of this dissertation in terms of the structural analysis of the problem, the construction of dedicated solvers to exploit the understanding of this structure and the evaluation of the proposed architecture in multi-contact interaction tasks.

The proposed architecture formulates and implements contacts and whole-body motion planners as well as kino-dynamic feedback controllers to efficiently optimize and successfully execute desired movement plans (*Contr. A1*). At the core of the motion planning and control architecture are optimal control methods based on the centroidal momentum dynamics that exploit the geometrical properties of the dynamics model to achieve very high levels of computational efficiency (*Contr. A2*). The methods to be introduced allow to study and optimize the effects of time at generating highly dynamic and robust motion plans that include full flight phases while preserving computational efficiency (*Contr. A3*). Further, the proposed formulation leads to a dedicated solver with polynomial-time convergence properties that could be exploited for receding horizon control or the efficient selection of contacting surfaces in uneven terrains based on a measure of dynamical robustness using a mixed-integer solver (*Contr. A4*).

Feedback controllers sensitive to uncertainties in the environment are derived and used to select the impedance or compliance policy according to the task's uncertainty type, such as e.g in tasks involving contact interactions with an uncertain environment modeled as optimal control problems with measurement noise (*Contr. A5*). It is further shown in simulation experiments that the feedback controller is applicable to each of the sub-problems composing the whole-body motion planning algorithm in a very large dimensional system such as a humanoid robot and that the super-position of the optimized impedances leads to robust tracking performance of the desired kino-dynamic movement plans (*Contr. A6*). Finally, the kinematics solver, inspired on how a consensus solver finds a solution to a problem, is able to incorporate nonconvexities in a simple yet efficient distributed manner (*Contr. A7*).

Additionally, the proposed theoretical contributions have found application outside of this dissertation, such as in a search-based contact planner where neural networks are trained at predicting the optimal evolution of the robot centroidal momenta and then used to generate dynamically robust contact sequences using a search-based planner [44], or in practical applications of the algorithms on a quadruped robot [121, 122] that dynamically traverses uneven terrains.

To summarize, the contributions of the proposed motion planning and control architecture presented in this dissertation extensively exploit the mathematical structure of the trajectory optimization problems to develop efficient but accurate approximations. The remaining part of this section provides a summary of the contributions to be presented in the following chapters.

Contribution I: Dynamic motion and contacts planning

Background

Optimal control theory is a fundamental tool in the robotics community [123] that has been extensively used for solving contacts and motion - planning and control problems described using dynamics models with different levels of complexity. Some approaches use simplified contacts or dynamics models to efficiently find a solution, but potentially limiting their applicability to specialized tasks [106, 124, 125]. Other approaches attempt to find a solution to the trajectory optimization problem using off-the-shelf solvers without simplifying assumptions and using a full dynamics model, leading to solutions that can leverage the full capabilities of the robot but that are potentially very costly to optimize [29, 89, 126] and might miss the benefits of exploiting structural properties of the problem [79]. Given the goal of this dissertation being the close to real-time planning of complex multi-contact behaviors and fast control loops, this chapter will propose a middle ground optimal control formulation based on the centroidal dynamics and focus on its structural analysis to devise efficient yet accurate approximations that do not sacrifice generality.

Contribution

The proposed approach is based on a decomposition of the whole-body finite-horizon trajectory optimization problem over joint and contact force trajectories in two sub-problems: a trajectory optimization over the whole body and an optimal control problem over the centroidal dynamics [79]. Taking this formulation as starting point, the proposed architecture suggests to solve the centroidal dynamics problem using a sequential convex approximation of its nonconvexities within its convex envelope and tailored to the curvature knowledge. An efficient and sparse implementation of the method achieves polynomial-time convergence rates, which make it possible to optimize 10 seconds of motion in around a second (*Contr. A2*). The general nature of the approximation allows to include time as an optimization variable (*Contr. A3*), which significantly improves feasibility and robustness of the generated motions making it possible to e.g. generate highly dynamic movements such as jumps with full flight-phases without affecting convergence properties.

The gained efficiency in the optimization of movement plans using a centroidal dynamics model can also be exploited to optimize discrete contact decisions over complex uneven terrains based on a criteria of dynamical robustness using a mixed-integer solver (*Contr. A4*). This solver remains computationally efficient for short contact sequences and its capabilities go beyond traditional approaches that limit themselves to quasi-static assumptions. This chapter will show that discrete contact decision can be made in dynamically challenging situations such as crossing a wide gap, going down a steep slope or recovering from a push. In summary, the proposed approach is general enough such that highly dynamic time-optimal multi-contact whole-body motions over challenging scenarios can be optimized out of task and terrain descriptions, but at the same time it is computationally very efficient for close to real-time applications (*Contr. A1*).

Context

The proposed formulations are motivated by the kino-dynamic decomposition presented in [79], that allows the focus of this chapter to be on the efficient solution of the dynamics problem exploiting its rich structure. Hopefully, the efficiency and convergence properties of the proposed approximations can in the future be further exploited to reactively and robustly control robotic platforms in real-world scenarios in a receding horizon fashion. The contents of this chapter constitute an extended version of the conference paper [127]. Plans resulting from this algorithm can be stabilized using the feedback controllers to be described in chapter 5 and executed on a torque controlled humanoid robot using inverse dynamics algorithms [92].

Contribution II: Feedback control design

Background

The proposed architecture takes advantage of the problem's structure to efficiently find feasible whole-body motion and contact force trajectories. However, despite the progress at improving the computational efficiency of the derived algorithms, they are still not able to achieve real-time control rates of 1KHz, but lower ones. Thus, to cope with the discrepancy between the model for planning and the simulated environment for execution, the proposed architecture includes a module for feedback control design to stabilize the desired kino-dynamic trajectories. In previous work, a reinforcement learning approach was used to optimize impedance policies for robust trotting on a quadruped robot [121, 128] by sampling from real-world and simulated data. In this chapter, the aim is to derive based on first principles a model-based optimal control algorithm for feedback design capable of synthesizing policies sensitive to environmental uncertainty for kino-dynamic movement plans.

Contribution

This chapter will present a computationally efficient feedback control design method that distinguishes itself from standard approaches in that it is sensitive to two sources of uncertainty coming from the environment (process and measurement noise), which allows it to adapt the impedance behavior of its feedback control law to the environmental uncertainty (*Contr. A5*). By exploiting the separability of a trajectory optimization problem into a whole-body and a contact force problems, the proposed architecture builds feedback controllers to stabilize the desired motion and contact force interaction trajectories of kino-dynamic movement plans for a high-dimensional humanoid robot (*Contr. A6*). This chapter will also show that in the presence of environmental uncertainties such as unknown distances to the locations of stepping stones, sensitivity to measurement noise naturally leads to compliant policies to control the interaction between the robot and an uncertain environment.

Context

The proposed approach for feedback design is based on the superposition property of impedances and the kino-dynamic separability of trajectory optimization problems into a whole-body motion and a centroidal dynamic problem. The proposed architecture exploits this separability to independently design feedback policies that generate optimal closed-loop behaviors for motion and contact interaction tasks, which further exploit all the couplings between the system states (in contrast with naive diagonal PD controllers). The contents of this chapter constitute an extended version of the conference papers [23, 129], as it proposes and demonstrates a feedback control architecture for a high-dimensional humanoid robot traversing challenging and uncertain scenarios. The desired kino-dynamic feedback behaviors can be realized on a torque controlled robot with inverse dynamics algorithms [92].

Contribution III: Kinematic motion planning

Background

Multi-contact trajectory optimization for robots traversing dynamically challenging scenarios is typically performed using complex models to describe the dynamic and kinematic interaction among all rigid bodies of a robot [29]. However, thanks to recent work [79] that suggested the possibility to take advantage of the kino-dynamic structure of the centroidal momentum dynamics [130] to independently optimize whole-body motion and contact force trajectories, it is possible to focus in this chapter on the whole-body motion optimization problem. Desirable properties of a whole-body trajectory include e.g. smoothness that comes from considering a time-horizon in the optimization [131, 132], as well as the ability to incorporate non-convex constraints such as the ones for obstacle avoidance, without turning the entire problem nonconvex and thus intractable for the problem sizes of interest.

Contribution

This chapter presents an algorithm for whole-body motion planning (*Contr. A7*). The most important characteristics of the approach are: a) Its capability of considering a time-horizon within the optimization that leads to smooth whole-body motion trajectories (when possible given the aim for highly dynamic motions such as galloping on a quadruped robot), and b) the possibility to include nonconvex constraints (such as the ones for obstacle avoidance) by taking inspiration on how a consensus algorithm solves a big problem by decomposing it into smaller ones (easier to solve, even though they might still be nonconvex). The proposed algorithm combines a Limited-Memory BFGS solver to approximate the solution of small nonconvex problems (solvable in a distributed manner) with an efficient Interior-Point method that will bring the solutions to consensus. The algorithm offers a compromise between computational efficiency and suitability for generating complex kinematic motions.

Context

The proposed optimal control formulation is motivated by the separability of trajectory optimization problems based on the kino-dynamic structure of the centroidal momentum dynamics [79] that can be exploited using dedicated solvers. In a future, when much more complex and dynamic motions (such as parkour maneuvers, back-flips or traversing complex scenarios with obstacles) are possible in terms of hardware control (e.g. new ideas are needed to model hard contact constraints of inverse dynamics algorithms [92] in more flexible ways, such as [129, 133]), the proposed algorithm could be useful at finding feasible kinematic plans for complex motions as efficiently as possible.

2.3 LIST OF CONTRIBUTIONS

The contributions of the architecture proposed in this dissertation are as follows:

Contr. A1 It formulates and implements computationally efficient contacts and whole-body motion planning as well as feedback control methods for robots performing multi-contact interaction tasks. At the core of the architecture are efficient optimal control methods that exploit the problem mathematical structure.

Contr. A2 It analyses the geometrical properties of the centroidal dynamics, identifies its basic sources of nonconvexity (quadratic equality constraints), and builds efficient yet accurate approximations using sequential convex methods that exploit the knowledge about the problem's curvature.

Contr. A3 It can generate time-optimal motions without affecting convergence properties, which opens the possibility to the generation of highly complex and dynamic motions including full flight phases or very limiting constraints.

Contr. A4 It shows that the dedicated solver with polynomial-time convergence properties for centroidal dynamics optimization can be integrated in an mixed-integer solver to efficiently select contacting surfaces in complex uneven terrains based on a measure of dynamical robustness.

Contr. A5 It introduces feedback controllers sensitive to uncertainties in the environment as a way to optimally select the impedance or compliance policy according to the task's uncertainty type, which leads to the idea of modeling uncertain contact interactions as optimal control problems with measurement uncertainty.

Contr. A6 It shows in simulation experiments that the superposition of feedback controllers for each sub-problem (contact force and whole-body trajectory optimization) in a very large dimensional system such as a humanoid robot leads to robust tracking performance of the desired kino-dynamic movement plans.

Contr. A7 It proposes a kinematics solver, inspired on how a consensus solver finds a solution to a problem, that is capable of handling soft nonconvexities in a simple yet efficient distributed manner.

Chapter 3

DYNAMIC MOTION PLANNING

Abstract

The synthesis of complex whole-body movements requires solving hard nonconvex trajectory optimization problems. Among its most challenging issues are the high problem's dimensionality, a limited time budget to find a solution amenable for execution, nonconvexities coming from nonlinear dynamic constraints and discontinuities due to intermittent contacts. This chapter takes as starting point the kino-dynamic approach [79] that scales down the large dimensionality of a problem that simultaneously optimizes kinematic and dynamic variables by reformulating it in an alternating manner, where kinematic and dynamic variables are sequentially optimized until convergence to a common set of bridging variables, namely the robot centroidal momentum and endeffector poses. It then proposes a solution for the dynamics part of the trajectory optimization problem that systematically addresses the challenges by exploiting the problem's structure. The nonconvexities of the Newton-Euler equations are approximated using an iterative convex model built based on its curvature properties, which improves convergence for online generation of time-optimized movement plans. Furthermore, the use of this model to efficiently check feasibility of dynamic motions within a mixed-integer solver allows to take discrete decisions over the contact surfaces from a terrain that support the generation of a desired dynamic motion. To sum up, the algorithm decides over contacting surfaces (globally or with ϵ sub-optimality using a mixed integer solver) and then computes locally time optimal motions (that can also respect actuation limits) fast enough for online control using an Interior Point method for convex programming. The capabilities and limitations of the approach are evaluated in several multi-contact scenarios traversed by a physically simulated humanoid robot.

Notes:

The contents of this chapter and the introductory section 1.6 will be submitted as a journal article, and constitute an extended version of the conference paper [127].

3.1 INTRODUCTION

Many approaches have been proposed in the literature for the optimization of complex robotic behaviors. One of the fundamental differences among these approaches is the level of expressiveness of the dynamics model used within the trajectory optimization problem, as its degree of complexity implies a compromise or trade-off between the variety of movements that can be synthesized and the difficulty of solving the corresponding trajectory optimization problem [98].

Among the first approaches proposed in the literature are those based on linear models [124, 125, 134, 135] that e.g. describe only the essence of a behavior such as the motion of its Center of Mass (CoM). They offer great computational advantages that make them suitable for receding horizon control [136, 137, 138], but also they have enough model expressiveness to describe a desired task that makes them well suited to successfully control the performance of the desired (usually specialized) robotic task such as e.g. flat-ground walking in an uncertain environment [139, 140, 141, 142].

Researchers have also considered trajectory optimization problems where the full set of equations of motion of a floating-base system are used to describe the physical interaction among all robot bodies, because this allows them to synthesize motions that exploit the full robot's capabilities and are thus methods useful for a wider and more complex range of behaviors. Despite the enormous challenges of solving such problems, great results have been achieved. To name a few: [67, 86, 96] efficiently optimize and execute complex tasks for a quadruped robot that include gait and motion discovery. [40, 89, 143] show theoretical and experimental results on online movement generation and robust control for legged robots. Further works [29, 55, 87, 126, 144, 145] present theoretical approaches on how to efficiently generate contact and movement plans for much more complex behaviors such as the cooperation of several characters on a common task or the automatic discovery of contacts.

Trajectory optimization problems with an intermediate level of complexity have also been studied [95, 98, 146, 147]. One of these approaches based on the centroidal momentum dynamics [91, 93] has become very popular in the robotics community [79, 84, 148], because under the assumption of actuation feasibility (enough torque authority), it provides sufficient conditions to plan kino-dynamically consistent whole-body motions [84]. Trajectory optimization approaches based on the momentum dynamics are simple enough to be solved online to control the performance of a legged robot [32, 94], but at the same time expressive enough to plan complex behaviors [23, 32, 98, 147]. [84] e.g. shows that a trajectory optimization approach based on the centroidal momentum dynamics can be used to generate time-optimal whole-body motions using a general nonlinear solver for nonconvex programming [100]. Furthermore, it is capable of selecting contacts formulated as complementarity constraints and can handle obstacle avoidance.

Approaches based on the momentum dynamics model are still nonconvex and thus hard to solve, which led researchers to focus on the problem mathematical structure to derive more efficient methods. [130] e.g. uses a convex bound on the angular momentum rate (that maximizes the contact wrench cone margin) to minimize a worst-case bound on the l_1 angular momentum norm via convex optimization. In [32, 94], the bilinear terms of the momentum dynamics and timings are handled by a dedicated solver based on multiple-shooting, proxy constraints are used for handling whole-body limits based on an offline learning method, and experiments on a humanoid robot exemplify the online control capabilities of the approach. [147] exploits a linear formulation of the momentum dynamics based on a lower dimensional space projection and an adaptive method for timing optimization to control a robot traversing multi-contact scenarios in a receding horizon fashion. In [95, 146], the interpretation of friction cones as dual twists allows to compute online cones of feasible CoM accelerations. The resulting bilinear constraints are decoupled into linear pairs via a conservative trajectory-wide contact-stability criterion for online motion generation. Timings between contact switches are optimized online using a nonlinear but fast to solve problem.

[79], similarly to [84], proposed a kino-dynamic trajectory optimization method based on the momentum dynamics solvable in an alternating manner. It further proposed an analytical decomposition of positive and negative definite terms of the problem's Lagrangian based on the decomposition of angular momentum nonconvexities to improve solver convergence properties. Our previous work [149] proposed a relaxation of the problem that suggested the use of a proxy function to minimize angular momentum, namely the sum of the squares of the terms composing its nonconvexity. [122] showed experimental evidence about the application of this approach [149] to control a quadruped robot and determined that not being able to include an explicit target momentum in the cost function limits the applicability of the approach, as it is not well suited for the alternating whole-body motion optimization method [79]. For this reason, in [127] we proposed a more general approximation of the problem's nonconvexities: two methods for convex relaxation of the dynamics problem that allowed to include an explicit angular momentum objective, and thus to use it in the alternating approach [79] to compute consistent whole-body time-optimal kino-dynamic motions. Recently, [98] presented a novel phase-based parameterization of endeffectors and a smooth terrain description to formulate an optimization problem, online solvable with a general nonlinear solver, able to select gait-sequences, momentum and timings, swing-motions and body poses, thus bringing to mind the question if a problem approximation is needed.

This chapter presents an extended version of the previous problem formulation [127] that includes the following contributions:

- A convex, accurate and efficient modeling of the nonconvex centroidal momentum dynamics, based on two convex relaxations, that allows to online generate time-optimized dynamic motions.

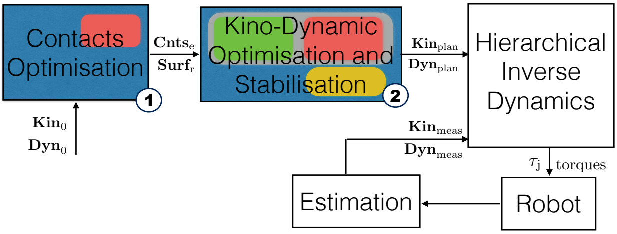


FIGURE 5: Control diagram highlighting inputs and outputs to each stage, and colored parts of the equations of motion used at each stage. For instance, to optimize contacts only a dynamics model is used (red color), while for whole-body optimization, a full kino-dynamic centroidal momentum dynamics model is used (red and green colors). In addition, actuation limits could be included (yellow box).

- It shows that by exploiting the problem's mathematical structure (Fig. 1), this formulation can be used in a kino-dynamic approach to generate whole-body motions, it can also include actuation limit constraints, as well as it can be used in a mixed-integer solver to select contact locations based on a measure of dynamical robustness.

- The capabilities and limitations of the algorithm are evaluated using a physical simulator where a humanoid robot traverses several multi-contact scenarios.

To sum up, the algorithm is capable of selecting contact locations over a discrete set of terrain surfaces using a measure of dynamical robustness. It is capable of generating whole-body time-optimal motions that satisfy actuation limits (if required) by exploiting the problem mathematical structure to render it tractable and very efficiently solvable.

The remainder of this chapter is structured as follows: Section 3.2 details the motion optimization approach to solve problem 4 presented in the background introduction section 1.6, based on convex relaxations of its nonconvexities. Section 3.3 presents a formulation for the efficient planning of dynamically consistent contact sequences. Finally, experimental results on a physical simulator are shown in section 3.4 and conclusions in section 3.6.

3.2 OPTIMIZATION OF DYNAMIC MOTION PLANS

This section presents a method to optimize motion plans based on an analytical decomposition of nonconvex bilinear expressions as a difference of convex functions, whose known curvature can be exploited to design efficient iterative convex approximations. The remainder of this section will analyze the nature of the nonconvexities

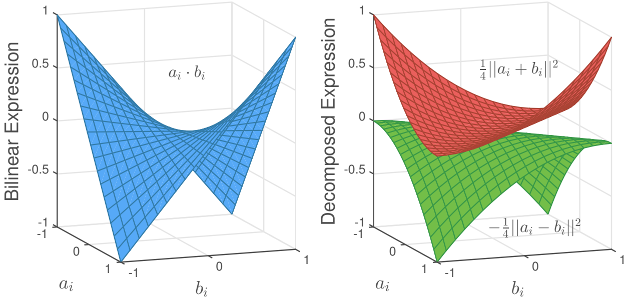


FIGURE 6: Decomposition of bilinear form $\mathbf{a}_i \cdot \mathbf{b}_i$ into a difference of quadratic expressions $\mathbf{a}_i \cdot \mathbf{b}_i = \mathbf{s}(\cdot) - \mathbf{r}(\cdot)$, where $\mathbf{s}(\cdot) = \frac{1}{4}\left\|\mathbf{a}_i + \mathbf{b}_i\right\|^2$ and $\mathbf{r}(\cdot) = \frac{1}{4}\left\|\mathbf{a}_i - \mathbf{b}_i\right\|^2$ are quadratic functions $\in \mathcal{Q}^+$ (i.e. its quadratic parameter matrices are positive semi-definite).

of problem (4), propose two convex relaxations to approximate them and, detail the numerical optimization procedures and its convergence criteria.

3.2.0.1 *Bilinear Terms as Difference of Convex Functions*

Some constraints in problem (4) are affine (4e)-(4g) or second-order cones (SOC) (4h)-(4i) and thus convex; others however describe nonconvex constraints such as the momentum dynamics evolution when considering variable timesteps (4b) or torque cross products (4c)-(4d), and are thus nonconvex. In the following its common nature is made clear and its reformulation is presented, in a way that is more amenable for its approximation using iterative convex models.

The torque cross product $\ell \times \mathbf{f}$ between a length ($\mathbf{p}_e - \mathbf{r}$ in equation (4c) or $\mathbf{R}_e^{x,y}\mathbf{z}_e$ in equation (4d)) and the force $\mathbf{f}_{e,t}$ can be written in the following way:

$$\ell \times \mathbf{f} = \begin{bmatrix} 0 & -\ell_z & \ell_y \\ \ell_z & 0 & -\ell_x \\ -\ell_y & \ell_x & 0 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$$= \left[\overbrace{\begin{bmatrix} -\ell_z & \ell_y \end{bmatrix}}^{\mathbf{a}_x^T} \overbrace{\begin{bmatrix} f_y \\ f_z \end{bmatrix}}^{\mathbf{b}_x}, \overbrace{\begin{bmatrix} \ell_z & -\ell_x \end{bmatrix}}^{\mathbf{a}_y^T} \overbrace{\begin{bmatrix} f_x \\ f_z \end{bmatrix}}^{\mathbf{b}_y}, \overbrace{\begin{bmatrix} -\ell_y & \ell_x \end{bmatrix}}^{\mathbf{a}_z^T} \overbrace{\begin{bmatrix} f_x \\ f_y \end{bmatrix}}^{\mathbf{b}_z} \right]$$

It is possible to see that each term is a scalar product $\mathbf{a}_i \cdot \mathbf{b}_i$ between two vectors composed by elements of ℓ and $\mathbf{f}$. Similarly, notice that the nonconvexities in (4b) can be written as scalar products between the timestep Δ and linear momentum $\mathbf{l}$, contact forces $\mathbf{f}_{e,t}$ or torque variables $\kappa_{e,t}$. It means that all nonconvex constraints solely include equality constraints with bilinear terms. Thus, in the following a reformulation of bilinear expressions in terms of mathematical expressions with known curvature is presented (in the spirit of [150, 151]), as performed in [79] as a difference of convex quadratic functions [152].

Difference of Convex Quadratic Functions $\mathcal{Q}^{\pm}$

Let V be a real vector space. The set $\mathcal{Q}^{\pm}$ is then

$$\begin{split}\mathcal{Q}^{\pm}(\mathbf{u}) = \Big\{ V \to \mathbb{R} \mid \exists\ \mathbf{s}(\cdot), \mathbf{r}(\cdot) \in \mathcal{Q}^{+}(\text{quadratic func.}) \\ \mathcal{Q}^{\pm}(\mathbf{u}) = \mathbf{s}(\mathbf{u}) - \mathbf{r}(\mathbf{u}) \quad \forall \mathbf{u} \in V \Big\}\end{split} \tag{6}$$

where $\mathcal{Q}^{+}$ is the set of quadratic functions whose quadratic parameter matrices are positive semi-definite. The set $\mathcal{Q}^{\pm}$ is closed under scalar multiplication, addition and composition with affine functions $\mathcal{A}(\cdot)$:

$$\begin{gathered}(\lambda(\mathbf{s} \circ \mathbf{v}) + \beta(\mathbf{r} \circ \mathbf{w})) \in \mathcal{Q}^{\pm} \\ \forall \mathbf{s}(\cdot), \mathbf{r}(\cdot) \in \mathcal{Q}^{+}; \quad \mathbf{v}(\cdot), \mathbf{w}(\cdot) \in \mathcal{A}; \quad \lambda, \beta \in K\end{gathered}$$

Figure 6 shows an example of a bilinear form $\mathbf{a}_i \cdot \mathbf{b}_i$ (in blue) analytically decomposed as the difference of the quadratic functions $\frac{1}{4}\left\|\mathbf{a}_i + \mathbf{b}_i\right\|^2$ (in red) and $\frac{1}{4}\left\|\mathbf{a}_i - \mathbf{b}_i\right\|^2$ (in green). By using this tool, it is possible to decompose a bilinear term with an indefinite curvature into a difference of quadratic terms with known curvature, which is key for the efficiency of the proposed algorithm.

Let's consider e.g. equation (4d) and assume for simplicity that $\mathbf{R}^{x,y}_{e,t}\mathbf{z}_{e,t} \times \mathbf{f}_{e,t}$ is represented by the decomposition $\ell \times \mathbf{f}$, then each endeffector torque component becomes $\gamma^{i}_{e,t} = \mathbf{a}_i^T \mathbf{b}_i + (\mathbf{R}^{z}_{e,t}\tau_{e,t})^{i}$. The torque component $\tau_{e,t}$ can be formulated as a difference of positive components $\tau^{+}_{e,t} - \tau^{-}_{e,t}$, as in [29]. Then, $\gamma^{i}_{e,t}$ can be written as:

$$\gamma^{i}_{e,t} = \overbrace{\underbrace{\left[\frac{1}{4}\left\|\mathbf{a}_i + \mathbf{b}_i\right\|^2 + (\mathbf{R}^{z}_{e,t})^{i}\tau^{+}_{e,t}\right]}_{\in\mathcal{Q}^{+}} - \underbrace{\left[\frac{1}{4}\left\|\mathbf{a}_i - \mathbf{b}_i\right\|^2 + (\mathbf{R}^{z}_{e,t})^{i}\tau^{-}_{e,t}\right]}_{\in\mathcal{Q}^{+}}}^{\in\mathcal{Q}^{\pm}} \tag{8}$$

In a similar manner, each endeffector torque component $\kappa^{i}_{e,t}$ (4c), can be decomposed parameterizing its cross product $(\mathbf{p}_e - \mathbf{r}) \times \mathbf{f}_e$ with vectors $\mathbf{c}_i$ and $\mathbf{d}_i$ as $\kappa^{i}_{e,t} = \mathbf{c}_i^T \mathbf{d}_i + \gamma^{i}_{e,t}$. Then, the endeffector torque component $\kappa^{i}_{e,t}$ can be written as:

$$\kappa^{i}_{e,t} = \overbrace{\overbrace{\underbrace{\left[\frac{1}{4}\left\|\mathbf{c}_i + \mathbf{d}_i\right\|^2\right]}_{\in\mathcal{Q}^{+}} - \underbrace{\left[\frac{1}{4}\left\|\mathbf{c}_i - \mathbf{d}_i\right\|^2\right]}_{\in\mathcal{Q}^{+}}}^{\in\mathcal{Q}^{\pm}} + \overbrace{\gamma^{i}_{e,t}}^{\in\mathcal{Q}^{\pm}}}^{\in\mathcal{Q}^{\pm}} \tag{9}$$

A similar analysis holds for each of the cartesian components of the bilinear expressions within the dynamic constraints (4b), which can be decomposed into a difference of convex functions as elements of $\mathcal{Q}^{\pm}$ in the following way:

$$\mathbf{l}^{i}_{t}\Delta_t = \frac{1}{4}\left\|\mathbf{l}^{i}_{t} + \Delta_t\right\|^2 - \frac{1}{4}\left\|\mathbf{l}^{i}_{t} - \Delta_t\right\|^2 \tag{10a}$$

$$\left(\sum_e \check{}^{i}_{e,t}\right)\Delta_t = \frac{1}{4}\left\|\sum_e \check{}^{i}_{e,t} + \Delta_t\right\|^2 - \frac{1}{4}\left\|\sum_e \check{}^{i}_{e,t} - \Delta_t\right\|^2 \tag{10b}$$

$$\left(\sum_e \mathbf{f}^{i}_{e,t}\right)\Delta_t = \frac{1}{4}\left\|\sum_e \mathbf{f}^{i}_{e,t} + \Delta_t\right\|^2 - \frac{1}{4}\left\|\sum_e \mathbf{f}^{i}_{e,t} - \Delta_t\right\|^2 \tag{10c}$$

The next section shows how the quadratic expressions $\mathcal{Q}^{+}$ (in which the bilinear terms have been decomposed) can be efficiently approximated using sequential convex relaxations that exploit the curvature knowledge of the quadratic expressions.

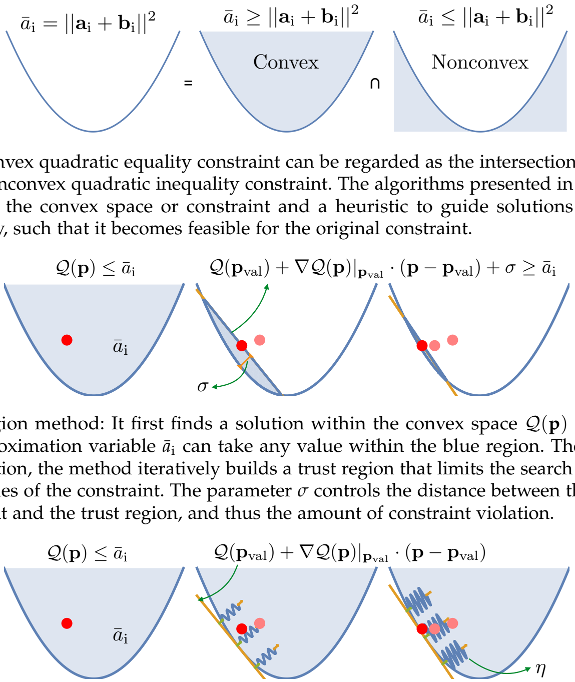
3.2.0.2 *Sequential Convex Approximation of Quadratic Terms*

In this section, the known curvature of the quadratic terms $\mathcal{Q}^{+}$ is used to build its approximation. First, they are isolated via a change of variables, e.g. equation (8) can be written as (note that the method is valid for any $\mathcal{Q}^{\pm}$ such as (9)-(10)):

$$\bar{a}_i = \left\|\mathbf{a}_i + \mathbf{b}_i\right\|^2 \;, \quad \bar{b}_i = \left\|\mathbf{a}_i - \mathbf{b}_i\right\|^2 \tag{11a}$$

$$\gamma^{i}_{e,t} = \left[\frac{1}{4}\bar{a}_i + (\mathbf{R}^{z}_{e,t})^{i}\tau^{+}_{e,t}\right] - \left[\frac{1}{4}\bar{b}_i + (\mathbf{R}^{z}_{e,t})^{i}\tau^{-}_{e,t}\right] \tag{11b}$$

where the introduction of the variables $\bar{a}_i, \bar{b}_i \in \mathbb{R}$ allowed to isolate the nonconvex expressions with known curvature (11a) from its original equation, which is now linear (11b). There are now two additional variables and constraints, but with the difference and benefit that each nonconvex constraint from the original problem can now be written in the very simple form of equation (11a), which will allow to find efficient sequential convex approximations.


$\bar{a}_i = ||\mathbf{a}_i + \mathbf{b}_i||^2$
=
$\bar{a}_i \geq ||\mathbf{a}_i + \mathbf{b}_i||^2$
Convex
∩
$\bar{a}_i \leq ||\mathbf{a}_i + \mathbf{b}_i||^2$
Nonconvex
$\mathcal{Q}(\mathbf{p}) \leq \bar{a}_i$
$\bar{a}_i$
$\mathcal{Q}(\mathbf{p}_{\text{val}}) + \nabla\mathcal{Q}(\mathbf{p})|_{\mathbf{p}_{\text{val}}} \cdot (\mathbf{p} - \mathbf{p}_{\text{val}}) + \sigma \geq \bar{a}_i$
σ
$\mathcal{Q}(\mathbf{p}) \leq \bar{a}_i$
$\bar{a}_i$
$\mathcal{Q}(\mathbf{p}_{\text{val}}) + \nabla\mathcal{Q}(\mathbf{p})|_{\mathbf{p}_{\text{val}}} \cdot (\mathbf{p} - \mathbf{p}_{\text{val}})$
η

(a) A nonconvex quadratic equality constraint can be regarded as the intersection of a convex and a nonconvex quadratic inequality constraint. The algorithms presented in this chapter use only the convex space or constraint and a heuristic to guide solutions towards its boundary, such that it becomes feasible for the original constraint.

(b) Trust region method: It first finds a solution within the convex space $\mathcal{Q}(\mathbf{p}) \leq \bar{a}_i$, where the approximation variable $\bar{a}_i$ can take any value within the blue region. Then based on this solution, the method iteratively builds a trust region that limits the search space to the boundaries of the constraint. The parameter σ controls the distance between the quadratic constraint and the trust region, and thus the amount of constraint violation.

(c) Soft-constraint method: It first finds a solution within the convex space $\mathcal{Q}(\mathbf{p}) \leq \bar{a}_i$ as in the previous method and then based on this solution, it iteratively builds a function underestimator (in orange), whose purpose is to act as a cost that rewards the selection of values close to it and thus close to the constraint boundary. Parameter η controls the desirability of selecting solutions close to the underestimator of the quadratic constraint.

FIGURE 7: Approximation of quadratic equality constraints $\in \mathcal{Q}^+$ within its convex space using sequential convex relaxation methods: trust region and soft constraints.

Fig. 7a sketches the hyperplane defined by the nonconvex constraint (11a), which can be conceived as the intersection of two inequalities, a convex $\bar{a}_i \geq \|\mathbf{a}_i + \mathbf{b}_i\|^2$ and a nonconvex one $\bar{a}_i \leq \|\mathbf{a}_i + \mathbf{b}_i\|^2$. While it is difficult to search a solution in a high dimensional nonconvex space, it is easier to search within the space defined by the convex inequality and guide the optimization towards the constraint boundary as illustrated in Figure 7, approaching in this way towards solutions with practical feasibility for the original nonconvex quadratic equality constraint. In the following, the two proposed relaxation methods based on SOC programs to deal with the quadratic equality constraints are presented.

TRUST REGION METHOD In this approach the main idea is to use a primal constraint to limit the convex search space to values close to the boundaries (Fig. 7b). In mathematical terms, the trust region should constrain the problem to values of $\bar{a}_i$ near $\mathcal{Q}(\mathbf{p})$ (for simplicity of notation we define $\mathbf{p} = \mathbf{a}_i + \mathbf{b}_i$ and $\mathcal{Q}(\cdot) = \|\cdot\|^2$). In the first iteration, as no guess for the optimal problem values is available, a solution is found without additional constraints over the entire relaxed convex search space. From there on, the trust region is built based on the optimal vector from the previous iteration and by reducing the allowed amount of constraint violation σ. The benefits of constraining the problem in this way are twofold: in the first place, it is easy to refine the solution with values of $\mathbf{p}$ around the optimal values of the previous iteration $\mathbf{p}_{\text{val}}$ that satisfy the desired amount of constraint violation σ, and secondly, it provides a method to iteratively increase the approximation accuracy by reducing the value of σ, as required by convergence tolerances. Note that if the hessian of the nonconvex constraint were an indefinite matrix, this trust region would lead to unbounded regions instead of constraining the problem as desired.

Trust Region Approximation of $\mathcal{Q}^+$ Expressions

In the case of $\mathcal{Q}^+$ expressions, thanks to the positive curvature of the constraint hessian, a linear inequality constraint suffices to constrain the problem as desired [153].

$$\mathcal{Q}(\mathbf{p}) = \bar{a}_i \rightarrow \begin{cases} \mathcal{Q}(\mathbf{p}) \leq \bar{a}_i \\ \mathcal{Q}(\mathbf{p}_{\text{val}}) + \nabla \mathcal{Q}(\mathbf{p})|_{\mathbf{p}_{\text{val}}} \cdot (\mathbf{p} - \mathbf{p}_{\text{val}}) \geq \bar{a}_i - \sigma \end{cases}$$

The linear constraint is built based on the current optimal values $\mathbf{p}_{\text{val}}$ taken by the optimization variables $\mathbf{p}$ and σ is a positive threshold, big enough to provide a feasible interior to the intersection of the constraints. Figure 7b graphically depicts the approximation.

SOFT-CONSTRAINT METHOD Alternatively, a hard restriction of the search space could be replaced with a cost that biases the optimizer towards finding solutions close to the boundary of the constraint by pulling optimization variables towards a function underestimator (Fig. 7c).

Soft-Constraint Approximation of $\mathcal{Q}^+$ Expressions

A cost heuristic rewards the selection of values of $\bar{a}_i$ close to the function underestimator ($\mathcal{Q}(\mathbf{p}_{val}) + \nabla\mathcal{Q}(\mathbf{p})|_{\mathbf{p}_{val}} \cdot (\mathbf{p} - \mathbf{p}_{val})$), hyperplane that supports the function and was built based on the current optimal values $\mathbf{p}_{val}$ taken by the optimization variables $\mathbf{p}$ in the previous iteration.

$$\mathcal{Q}(\mathbf{p}) = \bar{a}_i \rightarrow \begin{cases} \mathcal{Q}(\mathbf{p}) \leq \bar{a}_i \\ \eta \left\| \mathcal{Q}(\mathbf{p}_{val}) + \nabla\mathcal{Q}(\mathbf{p})|_{\mathbf{p}_{val}} \cdot (\mathbf{p} - \mathbf{p}_{val}) - \bar{a}_i \right\|^2 \end{cases}$$

η defines the desirability of selecting optimization values close to the underestimator, and that thus enjoy of practical feasibility for the nonconvex constraint. Figure 7c graphically depicts the approximation.

As shown in Fig. 7, both methods are iterative convex approximations of the problem nonconvexities. They are based on looking for a solution over the convex search space and thus enjoy of good convergence convexity properties, but at the same time they are general approximations of the nonconvexities that can capture its nature all over its range.

3.2.0.3 *Numerical Optimization*

This section describes numerical aspects such as convergence criteria and algorithmic implementation details used within the trajectory optimization problem.

CONVERGENCE CRITERIA The amount of constraint violation is used as the measure to decide upon convergence. It is defined as the supremum among the average errors of state variables (12a), which are computed by comparing the values of optimization variables ($\mathbf{r}_t$, $\mathbf{l}_t$, $\mathbf{k}_t$) that solve the approximate problem and the values obtained by integrating endeffector wrenches ($\mathbf{r}_t^{seq}$, $\mathbf{l}_t^{seq}$, $\mathbf{k}_t^{seq}$) that satisfy exactly all nonconvex constraints, as in (12).

$$\mathbf{l}_i^{seq} = \mathbf{l}_0 + \sum_{s=1}^{i} \left(m\mathbf{g} + \sum_e \mathbf{f}_{e,s} \right) \Delta_s$$

$$\mathbf{r}_i^{seq} = \mathbf{r}_0 + \frac{1}{m} \sum_{r=1}^{i} \left(\mathbf{l}_0 + \sum_{s=1}^{r} \left(m\mathbf{g} + \sum_e \mathbf{f}_{e,s} \right) \Delta_s \right) \Delta_r$$

$$\mathbf{k}_{\mathrm{i}}^{\mathrm{seq}} = \mathbf{k}_0 + \sum_{\mathrm{s}=1}^{i} \left(\sum_{\mathrm{e}} (\mathbf{p}_{\mathrm{e,s}} + \mathbf{R}_{\mathrm{e,s}}^{\mathrm{x,y}} \mathbf{z}_{\mathrm{e,s}} - \mathbf{r}_{\mathrm{s}}^{\mathrm{seq}}) \times \mathbf{f}_{\mathrm{e,s}} + \mathbf{R}_{\mathrm{e,s}}^{\mathrm{z}} \tau_{\mathrm{e,s}} \right) \Delta_{\mathrm{s}}$$

$$e = \max \left(\sum_{\mathrm{t}=1}^{T} \underbrace{\frac{\|\mathbf{r}_{\mathrm{t}} - \mathbf{r}_{\mathrm{t}}^{\mathrm{seq}}\|^2}{T}}_{e_{\mathbf{r}}}, \sum_{\mathrm{t}=1}^{T} \underbrace{\frac{\|\mathbf{l}_{\mathrm{t}} - \mathbf{l}_{\mathrm{t}}^{\mathrm{seq}}\|^2}{T}}_{e_{\mathbf{l}}}, \sum_{\mathrm{t}=1}^{T} \underbrace{\frac{\|\mathbf{k}_{\mathrm{t}} - \mathbf{k}_{\mathrm{t}}^{\mathrm{seq}}\|^2}{T}}_{e_{\mathbf{k}}} \right) \tag{12a}$$

When the errors (12a) fall below a certain threshold for the constraint violation to be considered negligible for practical purposes, the algorithm has converged. While errors are normalized to make them comparable, in practice angular momentum trajectories converge after CoM and linear momentum trajectories, thus it is possible to only focus on convergence of angular momentum to desired thresholds of constraint violation to decide upon algorithmic convergence.

algorithmic implementation details To approximate the solution of problem (4), the proposed approach iteratively solves an approximate problem (using an interior point solver for SOC programs based on [104]), where each nonconvex constraint (4b)-(4d) has been replaced by a convex approximation. At each iteration, the approximations and its parameters are updated (based on the optimal values of the previous iteration) to reduce the constraint violation amount. The procedure is then repeated until convergence.

The problem is solved in two phases (in the spirit of [55]): In the first one, the algorithm discovers a good motion using a full quadratic approximation, i.e. all nonconvex constraints are approximated using the methods described in section 3.2.0.2. In the second, the focus is more on convergence to physical consistency and approximation accuracy of (4c)-(4d) which are approximated using a second-order model as in the previous phase, and use only a first-order approximation of time-related constraints (4b) to allow the optimizer to still adjust CoM, linear momentum and timings, while speeding up and increasing accuracy of angular momentum convergence. Within the trust region method, the parameter σ is decreased using iteratively increasing powers of a value less than one, e.g. $\sigma \propto c^i$, where c is a value less than one, and i denotes the iteration number. Within the soft-constraint method, a different value for the parameter η is selected at each phase between $[1e4, 1e6]$ to be higher than other optimization costs.

3.3 optimization of contact plans

This section will explain how contact locations can be optimized within problem (4) when they are considered optimization variables but given the contact surfaces to which they should belong to. Then, this section will also describe an algorithm to efficiently select out of a terrain description a discrete set of contact surfaces and locations that would support the generation of a dynamic motion.

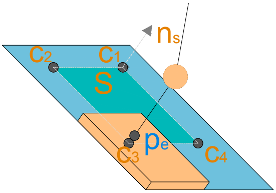


FIGURE 8: The description of a safe contact surface $\mathcal{S}$ is a set of coplanar corners $\mathbf{c}_i, i \in [1, q]$, out of which the following quantities can be computed: surface normal $\mathbf{n}_s$, surface rotation $\mathcal{S}^{\text{rot}}$, any surface point $\mathbf{p}_s = \mathbf{c}_i$ and a membership constraint $\mathbf{x} \in \mu(\mathcal{S}), \forall \mathbf{x} \in \mathcal{S}$, that simply defines the set of points $\mathbf{x}$ that lie on the safe surface.

3.3.1 *Membership of contact locations to terrain surfaces*

Given a description of the terrain surface $\mathcal{S}$ (over which it is safe to make contact), a contact location can be optimized by including its membership constraint to surface $\mathcal{S}$ to the optimization problem. A contact surface $\mathcal{S}$ (as defined in Fig. 8) is such that any contact point $\mathbf{p}_e$ selected from its interior guarantees that the entire endeffector is in contact. The expression $\mathbf{p}_e \in \mu(\mathcal{S})$ constrains an endeffector position $\mathbf{p}_e$ to belong to surface $\mathcal{S}$ (see Fig. 8 for notation):

$$\mathbf{p}_e \in \mu(\mathcal{S}) := \begin{bmatrix} \Lambda_s \\ \mathbf{n}_s \\ -\mathbf{n}_s \end{bmatrix} \mathbf{p}_e \leq \begin{bmatrix} \lambda_s \\ \mathbf{n}_s \cdot \mathbf{p}_s \\ -\mathbf{n}_s \cdot \mathbf{p}_s \end{bmatrix} \tag{13}$$

Equation (13) defines a set of halfspaces, whose intersection constrains a contact point $\mathbf{p}_e$ to lie on a safe contact surface. For instance, $\Lambda_s \mathbf{p}_e \leq \lambda_s$ denote the halfspaces that define lateral limits of the contact surface, while $\mathbf{n}_s \cdot \mathbf{p}_e = \mathbf{n}_s \cdot \mathbf{p}_s$ implies that the normal distance from the plane should be zero, i.e. the contact point has to lie on the contact surface.

Impact on convergence speed of this linear constraint is small and can be easily included, assuming a terrain description in terms of safe contact surfaces is given.

3.3.2 *Dynamics-Based Contacts Planning*

Robots dynamically navigate an environment by interacting with it via intermittent contacts. Thus, for dynamic motions such as walking down a steep slope, crossing a wide gap or recovering from a push, it makes sense to select them using a measure of dynamical robustness. To this end, ideally one would like to use a mixed-integer solver that evaluates the desirability of a set of contact surfaces by solving problem (4), as shown in section 3.2. In the following, such an approach is described.

3.3.2.1 *Terrain Description and Contact Model*

This section describes how a terrain is modeled and how contacts are selected within this description using the notation of [58].

The terrain consists of a set of convex, obstacle free regions $\mathcal{S}_{\mathrm{r}} \in \{1, R\}$ and the approach considers a sequence of $\mathbf{p}_{\mathrm{n}} \in \{1, N\}$ contacts. Binary variables $\mathbf{H}_{n,r} \in \{0,1\}^{N-N_0,R}$ define the contact surface $\mathcal{S}_{\mathrm{r}}$, whose domain contains contact $\mathbf{p}_{\mathrm{n}}$ (N_0 are contacts initially active and thus have a predefined pose). $\mathbf{H}_{n,r} \rightarrow \mathbf{p}_{\mathrm{n}} \in \mu(\mathcal{S}_{\mathrm{r}})$ and $\sum_r \mathbf{H}_{n,r} = 1$ constrain contact $\mathbf{p}_{\mathrm{n}}$ to belong to only one contact surface. The mapping between index n of contact location $\mathbf{p}_{\mathrm{n}}$ and the range of timesteps t, in which this endeffector location $\mathbf{p}_{\mathrm{e,t}}$ is active, is predefined.

Binary variables $\mathbf{H}$ decide upon the terrain region from where a contact location $\mathbf{p}$ can be selected. Integrality constraints guarantee that only one region is active for feet contacts and either one or none for hand contacts.

$$
\begin{aligned}
\mathbf{H}_{n,r} &\implies \mathbf{p}_n \in \mu(\mathcal{S}_r) \\
\textstyle\sum_r \mathbf{H}_{n,r} &\begin{cases} = 1, & \text{for feet contacts} \\ \leq 1, & \text{for hands contacts} \end{cases} \\
1 - \textstyle\sum_r \mathbf{H}_{n,r} &\implies (\mathbf{f}_{\mathrm{e,t}} = 0), \quad \text{for hands} \\
\mathbf{H}_{n,r} &\implies F_\mu^{\mathrm{cone}} \mathcal{S}_r^{\mathrm{rot}} \mathbf{f}_{\mathrm{e,t}} \leq 0, \quad \text{friction cone}
\end{aligned}
$$

When no contact region is selected, control variables such as contact forces are zeroed. Local endeffector forces $\mathcal{S}^{\mathrm{rot}}\mathbf{f}_{\mathrm{e,t}}$ also depend on the selected region to enforce friction cone constraints. F_μ^{cone} is a matrix such that when multiplied by the local force, the result is less than zero when the force respects the friction cone.

Reachability constraints between footstep locations can be based on kinematic reachability using linear inequalities [154] or on the intersection of SOC constraints [58]. They can be described in a convex form using linear inequalities based on kinematic reachability such as in

$$\Delta \mathbf{p}_{\mathrm{min}} \leq (\mathbf{p}_{\mathrm{i}} - \mathbf{p}_{\mathrm{i-1}}) \leq \Delta \mathbf{p}_{\mathrm{max}}$$

or they could be described as in [58] using an intersection of second-order cone constraints as:

$$
\begin{aligned}
\mathbf{S}_{\mathrm{h,n}}^{\mathrm{sec}} &\implies \begin{cases} \phi_h^s \leq \theta_n \leq \phi_{h+1}^s \\ \mathbf{s}^\theta = g_h^s \theta_n + b_h^s \end{cases}, \quad \textstyle\sum_{\mathrm{h}} \mathbf{S}_{\mathrm{h,n}}^{\mathrm{sec}} = 1 \\
\mathbf{C}_{\mathrm{h,n}}^{\mathrm{sec}} &\implies \begin{cases} \phi_h^c \leq \theta_n \leq \phi_{h+1}^c \\ \mathbf{c}^\theta = g_h^c \theta_n + b_h^c \end{cases}, \quad \textstyle\sum_{\mathrm{h}} \mathbf{C}_{\mathrm{h,n}}^{\mathrm{sec}} = 1 \\
\left\| \begin{bmatrix} \mathbf{p}_{\mathrm{i}}^{\mathrm{x}} \\ \mathbf{p}_{\mathrm{i}}^{\mathrm{y}} \end{bmatrix} \right. &- \left. \left(\begin{bmatrix} \mathbf{p}_{\mathrm{i-1}}^{\mathrm{x}} \\ \mathbf{p}_{\mathrm{i-1}}^{\mathrm{y}} \end{bmatrix} + \begin{bmatrix} \mathbf{c}^\theta & -\mathbf{s}^\theta \\ \mathbf{s}^\theta & \mathbf{c}^\theta \end{bmatrix} \mathrm{p}_{1,2} \right) \right\| \leq \mathrm{d}_{1,2}
\end{aligned}
$$

In the latter case e.g., piece-wise affine approximations of sine and cosine functions are used to model footsteps' rotation θ in a convex form. Binary variables $\mathbf{S}^{\text{sec}}_{\text{h,n}}, \mathbf{C}^{\text{sec}}_{\text{h,n}} \in \{0,1\}^{H,N_{\text{f}}}$ are used to select the active approximation h for each footstep n. Integrality constraints $\sum_{\text{h}} \mathbf{S}^{\text{sec}}_{\text{h,n}} = \sum_{\text{h}} \mathbf{C}^{\text{sec}}_{\text{h,n}} = 1$ guarantee that only one approximation is active at each footstep. Each affine approximation is defined by a region of validity of the yaw angle $\phi^s_h \leq \theta_n \leq \phi^s_{h+1}$ (for sine) or $\phi^c_h \leq \theta_n \leq \phi^c_{h+1}$ (for cosine) and, the corresponding linear approximation of the yaw angle of rotation $\mathbf{s}^\theta = g^s_h \theta_n + b^s_h$ (for sine) or $\mathbf{c}^\theta = g^c_h \theta_n + b^c_h$ (for cosine). Finally, these variables are used to model the range of available positions for the next footstep based on the current footstep position and yaw angle as the intersection of two SOC constraints.

3.3.2.2 *Dynamics Model and Objective Function*

The contacts planning algorithm makes use of a light version of problem (4) that does not consider e.g. endeffector torques $\gamma_{\text{e,t}}$ (without loss of generality contacts are modeled as points), uses a linear model of friction cones and, either a centroidal momentum dynamics model with fixed or non-fixed timings. The objective function ϕ^{cnt}_t similarly to (4a) regularizes states and controls and, also incorporates user-defined objectives.

The objective function ϕ^{cnt}_t regularizes available controls (timesteps Δ, contact locations $\mathbf{p}_{\text{n}}$, endeffector forces $\mathbf{f}_{\text{e}}$), as well as states (CoM and momentum).

$$\begin{aligned}
\min_{\mathbf{h},\Delta,\mathbf{p}_{\text{n}},\mathbf{f}_{\text{e}}} \quad & \sum_{t=1}^{T} \phi^{\text{cnt}}_t \Big(\mathbf{h}, \Delta_{\text{t}}, \mathbf{p}_{\text{n}}, \mathbf{f}_{\text{e}}\Big) \\
\text{subject to} \quad & \mathbf{h}_{\text{t}} = \begin{bmatrix} \mathbf{r}_{\text{t}} \\ \mathbf{k}_{\text{t}} \\ \mathbf{l}_{\text{t}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\text{t}-1} + \frac{1}{\text{m}} \mathbf{l}_{\text{t}} \Delta_{\text{t}} \\ \mathbf{k}_{\text{t}-1} + (\sum_{\text{e}} \kappa_{\text{e,t}}) \Delta_{\text{t}} \\ \mathbf{l}_{\text{t}-1} + (\text{m}\mathbf{g} + \sum_{\text{e}} \mathbf{f}_{\text{e,t}}) \Delta_{\text{t}} \end{bmatrix} \\
& \kappa_{\text{e,t}} = (\mathbf{p}_{\text{e,t}} - \mathbf{r}_{\text{t}}) \times \mathbf{f}_{\text{e,t}} \\
& \Delta_{\text{t}} \in [\Delta_{\text{min}}, \Delta_{\text{max}}] \\
& \|\mathbf{p}_{\text{e,t}} - \mathbf{r}_{\text{t}}\| \leq \ell^{\text{max}}_{\text{e}}
\end{aligned}$$

Physical constraints include a momentum dynamics model, a box constraint for timestep discretizations and limits between endeffector positions and CoM.

3.3.2.3 *Numerical Optimization*

To compare the effect of time-optimal vs. non time-optimal, as well as the effect of the number of iterations used to approximate the dynamics model over optimized contact plans, a mixed-integer solver capable of solving a sequence of SOC programs is used (not available commercially). It relies on two functions to bound the optimal

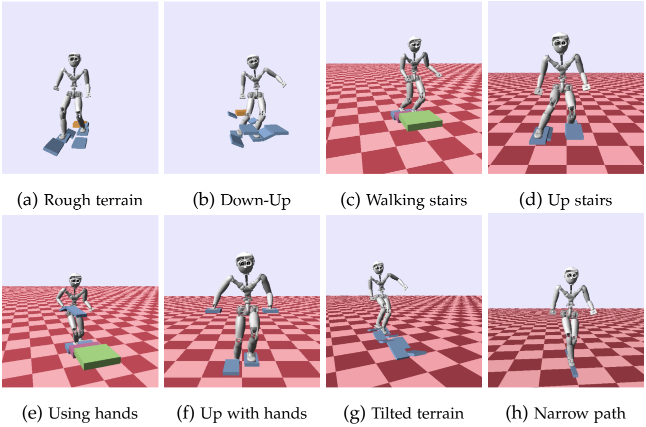

(a) Rough terrain (b) Down-Up (c) Walking stairs (d) Up stairs (e) Using hands (f) Up with hands (g) Tilted terrain (h) Narrow path

FIGURE 9: Examples of time-optimized dynamic movement plans.

value of a search space. The lower bound comes from a relaxation of the search space binary variables and the upper bound by any solution where binary variables are binary. The rest of constraints are treated using an iterative or non-iterative model as previously described. Then, the feasible search space is partitioned into convex sets and each partition bounded. The algorithm converges once global lower and upper bounds are close enough, otherwise the partitions are refined and the search process is repeated. The implementation of this mixed-integer solver is based on a branch and bound method for global nonconvex optimization, as detailed in [117]. In simple scenarios linear reachability constraints suffice, and SOC constraints are used in more complex ones, as will be shown in the experimental section 3.4.

3.4 EXPERIMENTAL RESULTS

This section shows experimental results about the optimization of contact and motion plans. It demonstrates the capabilities of the algorithm in multi-contact scenarios including walking on uneven terrain, climbing stairs using hands and many others.

3.4.1 *On the Optimization of Dynamic Movement Plans*

This section will demonstrate the capabilities of the algorithm (soft and trust region-based) at optimizing several centroidal momentum trajectories as defined by problem (4). In particular, it looks at convergence to feasibility (measured by the amount of constraint violation of the solution) and time complexity to converge to a desired feasibility threshold. This section will also present results regarding the qualitative improvement of motions that include time and/or contact locations in the optimization. Finally, it will show how whole-body motions can be optimized using a kino-dynamic approach, how actuation limits can be included in the dynamics optimization, and tracking performance of time-optimized motions.

3.4.1.1 *Feasibility Convergence and Time Complexity*

To analyze convergence properties and computational complexity of the algorithm, a set of 8 optimized motions (shown in Fig. 9) are used to gather statistics about algorithmic performance.

Figure 10 presents statistics about time complexity, convergence to feasibility and relative cost reduction when using the same objective function but different number of discretization timesteps and algorithmic settings. In particular, it shows information about what happens when the optimization includes or not time as a variable *Time* vs. *Mom*, includes or not optimization of contact locations *Cnt*, using soft-constraint or trust-region heuristics *Sc* vs. *Tr*. First of all, in the center plot the amount of constraint violation of the optimized solutions is shown, as measured by (12a). Note that the algorithm converges when the error or its reduction within an iteration fall below a desired threshold and, as visible on the plot, the method converges in all experiments to the desired feasibility thresholds under all settings.

The uppermost plot shows statistics about the time-complexity of the algorithm for convergence to the desired feasibility thresholds; in particular, it shows its linear-tendency in momentum and time optimization problems. Notice that for fixed-time optimization problems, neither the employed heuristic (trust region or soft-constraint) nor the inclusion in the optimization of contact locations affect the solving time performance. A similar behavior can be seen for time optimization problems with the difference that trust-regions are slightly faster than soft-constraints. Finally, the bottom plot quantifies numerically the relative reduction of the cost when optimizing time and contact locations. In orange tones, reference normalized costs of momentum optimization problems using fixed contact locations and timings for trust-region and soft-constraint heuristics are shown. As expected both achieve a similar minimum and have thus similar normalized costs (close to one). Considering contact locations as optimization variables has minimum impact on solving time performance, yet it significantly reduces the objective value (around 35 percent) because this degree of freedom allows the optimizer to select motions that require lower values of momentum trajectories, e.g. motions with less lateral sway of the center of mass.

The effect of time optimization on the objective value is dependent on the problem time horizon (for simplicity, one can assume that the value of one timestep is 0.1 seconds and thus the horizontal axis spans between 2 and 20 seconds). For instance,

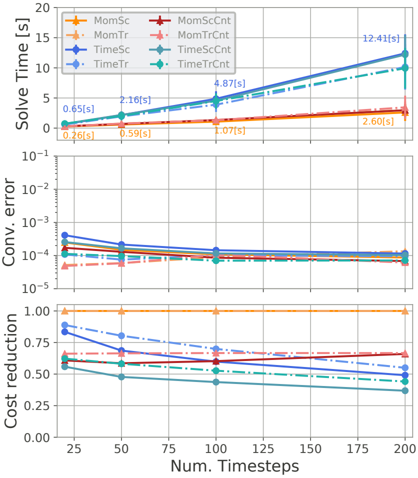


Figure 10: *Top:* Linear-time complexity of movement plans: with or without time optimization *Time* − *Mom*, using soft-constraint or trust-region heuristics *Sc* − *Tr*, and with or without optimization of contact locations *Cnt*. *Center:* Corresponding normalized convergence errors e as given by (12a) and *Bottom:* numerical relative cost reduction of motions optimized including time and/or contact locations with respect to motions using fixed contacts and timings. Each data-point summarizes information from 8 experiments (shown in Fig. 9) optimized using the same objective function but different number of timesteps and heuristics.

in problems with short-time horizons such as at the leftmost side of the plot, the cost difference between motions that consider or not time as an optimization variable is modest, but as the look-ahead horizon increases (right side) time optimization becomes a powerful way of shaping the motion to achieve lower costs. Notice that in this case the soft-constraint heuristic finds in average slightly lower local minima than the trust-region heuristic. Figure 11 shows the average number of iterations required to solve a momentum or time optimization problem for varying number of timesteps, as well as the average time required to solve each of these iterations. For instance, momentum optimization problems require between 2-3 iterations, while time optimization problems 7-10. However, the difference in solving times of one iteration is not significant, e.g. for a time horizon of 2 seconds (20 timesteps) the solving times are 80 ms and 100 ms for momentum and time optimization problems respectively. This suggests that the approach could be used in a receding horizon

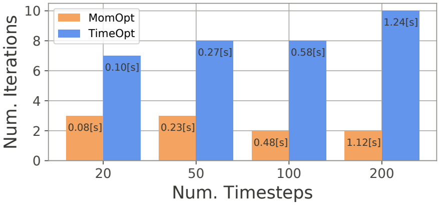


FIGURE 11: Average number of iterations required to solve an optimization problem with or without time optimization for different number of discretization timesteps and, average time to solve each iteration. Each data-point is based on 32 experiments, with different heuristic $Sc - Tr$ and with our without optimization of contacts.

setting, where the optimizer can be warm-started from a previous solution and thus only needs to solve one iteration for a short look-ahead horizon.

3.4.1.2 *Qualitative Improvement of Solutions*

In this space, qualitative results on a pair of motions are shown to illustrate the working principles and benefits of the approach.

For example, Figure 12 shows time optimal results for a walking up tilted stairs motion traversed with two different values of the friction coefficient μ. In the first case ($\mu = 0.35$), the tendency is to increase the duration of timestep discretizations during double supports to have enough time to slowly accelerate the CoM while respecting physical constraints, resembling statically stable motions. In an environment with flat surfaces, the same approach would be valid even if the friction coefficient is further reduced (e.g. $\mu = 0.25$). However, in a terrain with tilted surfaces such a strategy might not be viable. In such a setting, even the fixed-time version of the algorithm might struggle at finding a dynamically feasible solution. However, the time optimization approach is able to find a solution, whose main strategy is to quickly traverse the tilted surfaces to get to the uppermost flat contact surfaces. During this phase, lateral contact forces are exploited to the limit, and then a similar strategy to the previous case is used.

Figure 13 shows a walking up stairs motion, where hand contacts are used. Within the performed experiments, in such multi-contact scenarios time optimization does not significantly change motion timings. However, optimizing contact locations allows to find motions with less CoM sway (compare e.g. CoM trajectories for a momentum optimization without optimization of contact locations *MomSc* and a time optimization that includes optimization of contact locations *TimeScCnt*), which is more energetically efficient with a small additional computational cost in the optimization.

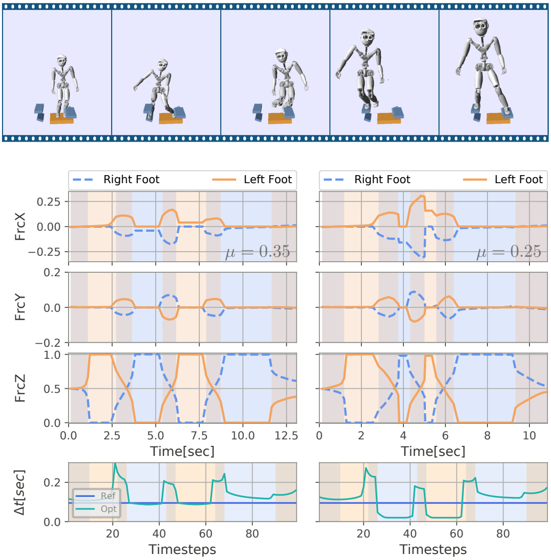


FIGURE 12: Comparison of optimal normalized endeffector forces and timing results for two different values of friction coefficient μ. *Ref* timings are the initial ones and *Opt* the final optimized ones.

3.4.1.3 *Kino-Dynamic Whole-Body Optimization*

This section will show how the proposed algorithm can be used in a kino-dynamic approach to generate whole-body motions. A climbing stairs motion (Fig. 9d) is used to illustrate convergence of the method to kino-dynamic consistency. Figure 14 graphically compares kinematic *Kin* and dynamic momentum trajectories *Dyn* at the end of each dynamics optimization, and on the bottom, it shows a quantitative comparison that depicts how error norms, obtained by comparing desired kinematic momentum trajectories and dynamic ones obtained by integrating optimal controls, decrease until convergence at each kino-dynamic iteration. Linear momentum converges to high levels of precision, while angular momentum only to modest levels of precision. Note as well that the first dynamics optimization (shown in red) takes the longest to converge. This plot also shows how trajectories optimized in subsequent iterations converge faster without using any information from previous ones. In practice, however by exploiting this information to construct optimization heuristics, they can be solved much faster. Dark colors are used to show dynamic trajectories *Dyn*, and the same, but light color, for kinematic ones *Kin*. Solid lines

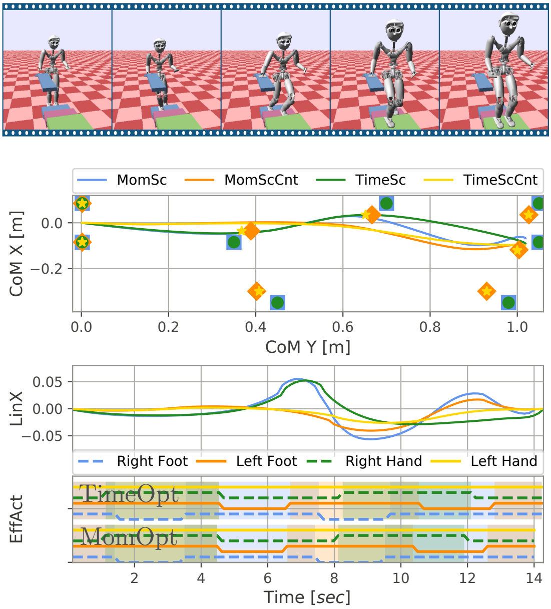


Figure 13: Comparison between CoM and normalized linear momentum in the lateral direction for a walking up stairs motion using hands. Bottom plot shows activations of endeffectors over the time horizon for momentum and time optimization problems.

correspond to motions optimized using soft-constraints, and dashed lines to motions optimized using trust regions. Qualitatively and quantitatively they converge to similar solutions, as can be seen in the plots, where it is difficult to distinguish them from each other. Finally, notice that at each kino-dynamic iteration kinematic and dynamic momentum trajectories match; however, in practice at least a couple of iterations are used to converge to a motion easily executable on a physical simulator.

3.4.1.4 *Execution of Movement Plans*

This section shows that optimal motion plans optimized in the previous section using a kino-dynamic approach can be executed on a physical simulator. Figure 15 shows tracking of optimized movement plans using feedback [129] and inverse dynamics controllers [92] using the architecture described in Fig. 5. The top three rows show desired momentum trajectories ($\mathbf{Dyn}_{\text{plan}}$ in orange) as well as its tracking

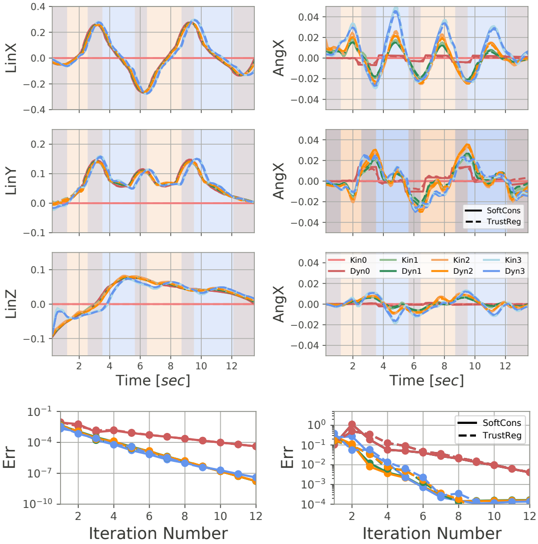


FIGURE 14: This figure shows convergence to feasibility of each dynamics optimization along three kino-dynamic iterations. It compares desired kinematic momentum trajectories *Kin* and dynamic momentum trajectories obtained by integrating optimal controls *Dyn* at the end of each dynamic optimization. Bottom plots show how error norms decrease until convergence along each iteration. Momentum values are normalized by robot mass. Vertical colored bars show the activation of each endeffector over time.

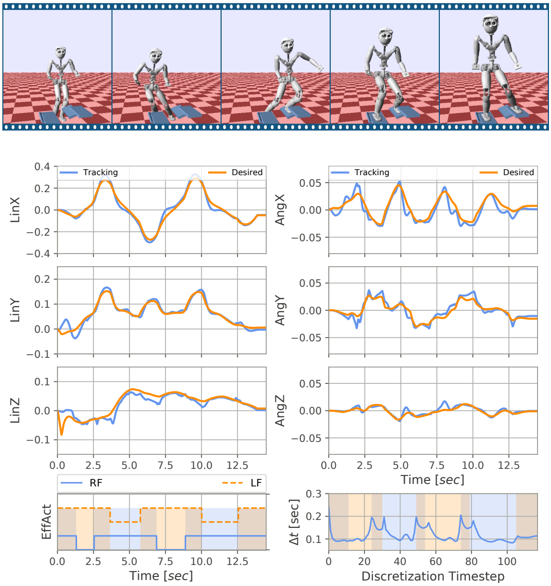


Figure 15: Tracking of desired momentum trajectories for the climbing up stairs motion (shown in Fig. 9d) using time optimization.

(**Dyn**$_{\text{meas}}$ in blue). At the bottom left corner, the activation of the endeffectors over time can be seen, as given by the optimal timings, shown at the bottom right corner. At the beginning of the motion execution, the robot transitions from a stiff feedback controller to a pure feedforward one; in this phase, executed momentum trajectories differ from desired ones.

Actuation limits are not always satisfied, if they are not explicitly considered in the optimization. For instance, in the climbing up stairs motion the knee flexion-extension KFE joint torque exceeds its limits by around 30[N], as shown in Fig. 16 in orange. If the torque limit constraint is included, linearized around the current kinematic trajectory, the dynamics optimizer can adapt other available degrees of freedom, such as timings and endeffector wrenches to satisfy it, as seen by the executed torque in green. Another way to satisfy actuation limit constraints is by redistribution of contact forces among the available endeffectors (Fig. 17). In this case, timestep discretizations were kept constant, and the optimizer distributes contact forces in such a way that the left leg is supported by the left hand in order to synthesize a motion within the leg torque actuation limits.

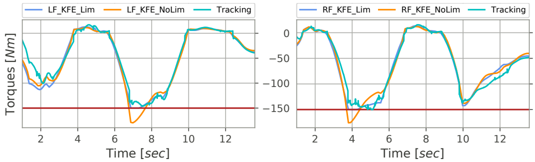


Figure 16: Satisfaction of actuation limits in walking motion (Fig. 9d) by considering joint torque limit constraints in the dynamics optimization problem that adapts endeffector wrenches and timings.

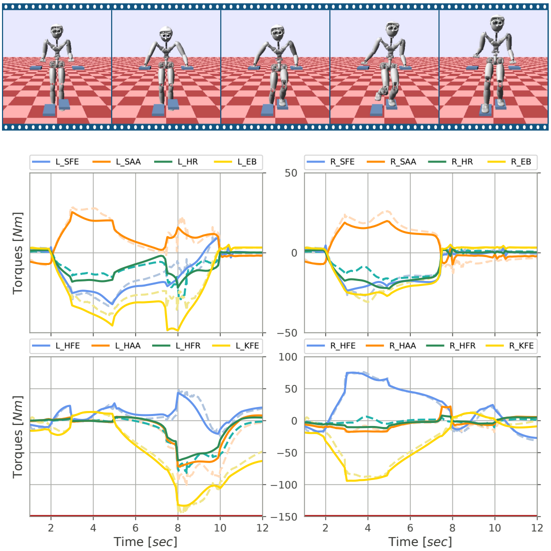


Figure 17: Satisfaction of actuation limits by redistribution of contact forces among available endeffectors in multi-contact fixed-time motion. Torque references are shown in solid lines, and execution torques in dashed lines. *Top:* left and right hands - shoulder and elbow joints; *Bottom:* left and right feet - hip and knee joints.

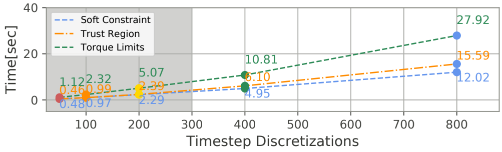


FIGURE 18: Effect of actuation limit constraints on solving time of fixed-time optimization problems for different number of discretization timesteps. Results shown correspond to a walking down and up motion (Fig. 9b) using soft-constraints for torque limits.

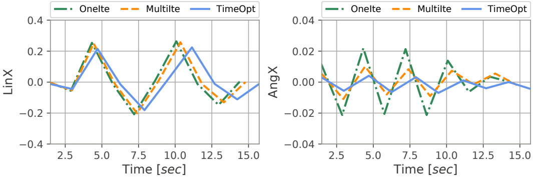
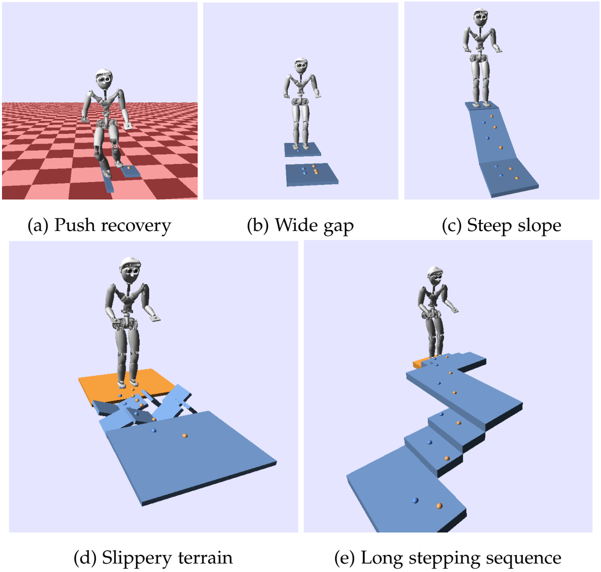


FIGURE 19: Comparison of normalized linear and angular momentum trajectories of dynamically optimized contact plans for a walking up motion (Fig. 9d). In green dash-dotted lines, results when using only one iteration are shown; in dashed orange lines, when using a multi-iteration dynamics model; and in blue solid lines results when time is also considered an optimization variable are displayed.

3.4.2 *On the Optimization of Contact Plans*

This section analyzes the results of the contacts planning algorithm at finding a set of contact surfaces and possibly an initial guess for a sequence of contacts that support the generation of a dynamic motion.

In the first place, Figure 19 compares momentum trajectories of dynamically optimized contact plans for a walking up motion using different dynamics models, namely an iterative model that optimizes over time variables (in blue), an iterative model with fixed timings (in orange), and a single-iteration dynamics model with fixed timings (in green). It is possible to observe that momentum trajectories are similar and thus generate similar contact plans. Yet, they significantly differ in the time required to solve the corresponding mixed-integer problem ranging from one second for the single-iteration model to a few seconds for the time-varying dynamics model. This suggests that for contacts planning it suffices to consider a single-iteration dynamics model, which has the benefits of being quickly solvable, thus making the contacts planner reactive for online planning of contact locations, but at the same time still considers a dynamic measure of motion robustness. The



(a) Push recovery (b) Wide gap (c) Steep slope

(d) Slippery terrain (e) Long stepping sequence

FIGURE 20: Examples of dynamically optimized contact plans.

details of the motion plan such as optimal timings are then deferred to the motion planning stage, which can be initialized using the dynamics information of the current contacts planning stage.

Fig. 20 shows a few examples of dynamically optimized contact plans, where a static contact planner is likely to fail. On *top*, it shows contact plans optimized using linear inequalities for reachability constraints, while on the *bottom*, SOC reachability constraints are used, as they naturally allow the selection of foot rotations to traverse difficult environments. Short contact plans (2-4 contacts) as in Fig. 20a can be solved in a second; medium-size plans as in Fig. 20b-20c in a few seconds; and long contact sequences with many contact surfaces as in Fig. 20d-20e can take a minute to solve. In summary, short contact plans can be optimized online using the current approach, while long contact sequences can be learned using e.g. the search-based approach [44].

3.5 DISCUSSION

This section discusses about time complexity and ways to speed up convergence, as well as limitations of the approach, in the sense that it works well within the tested problems, but it does not mean that it can solve any general nonconvex problem.

3.5.0.1 *Time and Computational Complexity*

In general, finding a solution to the dense version of any of the convex approximations solved, requires a polynomial time algorithm (of order $\mathcal{O}(p^{\frac{1}{2}}[p+q]q^2) \approx \mathcal{O}(p^{\frac{3}{2}})$, p being the number of quadratic constraints and q its size) [108]. However, within the problem size ranges of interest to this work and thanks to the exploited problem sparsity patterns, it can be observed (Fig. 10) that the problem has approximately linear time complexity. Notice this linear tendency for both momentum and time optimization problems, despite their different rates of grow due to distinct problem sizes. Even problems that consider actuation limits show this linear tendency (Fig. 18).

In practice, speed up of time optimization problems can be achieved by appropriately warm-starting the optimizer using solutions from previous iterations to build optimization heuristics. When considering torque limits the doubled computational effort due to the addition of $2nT$ inequality constraints for a problem with T timesteps and robot with n joints (≈ 30 in our case) can be reduced by considering only the weakest joints or only those involved in the motion. All in all, computation times are still lower than the planned horizon, making it possible to run the algorithm online (the next plan can be computed, while the current one is being executed).

3.5.0.2 *On Limitations of the Approximations*

Problem (4) is nonconvex and thus hard to solve. The proposed heuristics lighten to some extent the effort required to find a solution by searching for an approximate one within the convex space of the problem. This however comes with certain limitations. For instance, when using trust regions, they might be inappropriately built leading to non-optimal solutions, or even unsuitably initialized which could render the interior of the convex cone empty leading to primal infeasibility. For the soft-constraint method, the difficulty lies in finding an appropriate trade-off between two competing objectives: amount of constraint violation and problem conditioning. An adaptive solution that iteratively reduces the value of the allowed amount of constraint violation σ works well for the trust region heuristic. For the soft-constraint method, a constant value for each phase of the problem (tuned for a good objective trade-off) suffices in practice.

3.6 conclusion

This chapter presented a structured and efficient algorithm for generating time-optimal whole-body movement plans for a humanoid robot, as well as an approach to select a set of contact surfaces from a terrain description that supports such a motion. Finally, experimental evidence on a physical simulator has been shown to exemplify the capabilities of the algorithm at generating online motion plans to successfully control a simulated humanoid robot.

Chapter 4

KINEMATIC MOTION PLANNING

Abstract

The synthesis of kino-dynamically consistent whole-body motion behaviors for robots with arms and legs usually requires solving trajectory optimization problems based on a highly complex dynamics model (such as the equations of motion) or a middle complexity one (such as the centroidal momentum dynamics), which are able to capture how contact force interaction with the environment relates to whole-body motion of the robot. The alternating kino-dynamic approach for independent optimization of whole-body motion and contact force trajectories based on the centroidal dynamics model [79] made it possible in the previous chapter to only focus on the efficient and structured optimization of the problem's dynamics side. In the same spirit, this chapter focuses only on the kinematics side of the problem, whose main objective will be to find whole-body movement plans in consensus with the dynamic momentum (computed from contact wrenches), center of mass and desired endeffector motion trajectories for floating-base rigid-body systems. The algorithm presented in this chapter also considers a look-ahead horizon as well as proposes a methodology to incorporate soft nonconvexities inspired by how consensus algorithms solve a problem in a distributed manner. The computational efficiency, capabilities and limitations of the proposed algorithm are evaluated in several multi-contact scenarios traversed by legged robots.

4.1 INTRODUCTION

Many approaches have been proposed in the literature for the synthesis of complex kinematic behaviors for robots with arms and legs, ranging from sampling-based [48, 51, 53] to optimization-based techniques [29, 55, 84, 87, 126] to name a few. The range of possible behaviors that can be synthesized by the different methods as well as the computational complexity of generating a motion behavior largely depend on the dimensionality of the search space and the expressiveness of the model used for movement generation (e.g. equations of motion or centroidal dynamics model in an optimization-based method) [98].

Sampling-based techniques [16] sample from the robot's configuration space (floating-base and joints configuration) to build a discrete graph (called a roadmap) composed by vertices that represent obstacle free configurations and edges that denote the existence of an obstacle-free path between two vertices. The graph is built as a discrete approximation of the connectivity of the obstacle-free space and used to find a path between the initial and desired final configuration [51]. Sampled configurations are usually further processed to increase the sampling success rate of zero-measure manifolds [155]: for instance sampled configurations can be projected so as to fulfill e.g. static stability conditions [156], contact conditions [157], or dynamic constraints [158]. In the last years sampling based approaches, such as Probabilistic Roadmaps (PRM) [159, 160, 161] or Rapidly-exploring Random Trees (RRT) [162, 163], have been shown to provide not only theoretical guarantees such as probabilistic completeness but also to work well in practice [164]. However, its extension to sampling in state space (including robot posture and velocity) while satisfying complex dynamic constraints (such as contact constraints and equations of motion for floating-base robots) remains a challenging problem [165].

Optimization-based approaches on the other hand allow to easily formulate constrained problems that include e.g. the dynamics model of a floating-base rigid-body system and geometric contact constraints [29, 40]. These optimization problems are able to describe the dynamic and kinematic interaction of every link and actuator of the robot and can thus exploit the full robot's capabilities to synthesize a wide variety of motions [86, 87]. However, solving an optimization problem that considers as dynamic model the equations of motion is a very challenging task for many reasons, including high problem dimensionality, discontinuities due to intermittent contacts, nonconvexities that make the optimizer prone to poor local minima from where it is hard to recover, limited time budget to find a solution amenable for execution on a robot, among others [55, 126].

Middle-ground approaches have also been considered [23, 96, 98, 146, 147]. Among them, the momentum dynamics model has [91, 93] has raised a lot of attention [79, 84, 148], because it offers [79, 84, 148] sufficient conditions to plan whole-body motions under the assumption of enough torque authority [84]. This model allows the planning of complex behaviors [23, 98, 147], while at the same time it can be solved online to control a robot [32, 94]. [84] proposed for the first time to formulate a trajectory optimization problem based on the kinematic and dynamics side of the centroidal dynamics and was able to plan very complex and highly dynamic motions, select contact sequences, timings and handle obstacle avoidance constraints at orders of magnitude faster compared to approaches based on the full equations of motion. [79] further proposed that the kino-dynamic problem can be solved in an alternating fashion, where kinematic and dynamic variables are optimized independently until convergence to a common set of variables.

The alternating formulation of whole-body kino-dynamic movement plans [79] has thus opened the possibility of solving the kinematics problem by exploiting highly developed algorithms for motion planning in complex environments [166, 167, 168, 169], capable of efficiently handling obstacle avoidance or nonlinear dynamic constraints. The algorithm proposed in this chapter belongs to the family of optimization-based methods such as [29, 84, 168] and is built as a dedicated solver for the kinematics' side [79] of the trajectory optimization problem of floating-base rigid-body systems. The contributions of the algorithm are as follows:

- It finds a locally optimal solution of the kinematics trajectory optimization problem using a sequential convex formulation. The approach is capable of generating a wide variety of whole-body motions satisfying kinematic feasibility constraints (such as joint posture or velocity constraints as in [29, 84]) with the efficiency of a motion planning algorithm [168], but in addition it also includes tracking objectives for dynamics momentum (computed from contact wrenches) to ensure whole-body feasibility of the kino-dynamic movement plans for floating-base systems.

- Typically, motion planning algorithms [166, 167, 168] handle nonconvex constraints such as those for obstacle avoidance without any difference from others thus creating bigger and bigger problems which are difficult and expensive to solve. The proposed approach instead reformulates the problem in a way inspired by distributed optimization [110], where soft-nonconvexities (such as obstacle avoidance constraints) can be solved separately in small distributed sub-problems, whose solutions can then be used in the main optimization problem to bring an overall solution to consensus with distributed ones. In summary, small distributed problems handle soft-nonconvexities using a general nonlinear solver, and then the main optimization problem makes use of these solutions to build an overall solution using an efficient interior-point method for convex optimization.

- The capabilities and limitations of the algorithm are evaluated on several multi-contact scenarios traversed by legged robots.

To sum up, the algorithm presented in this chapter for kinematics optimization combined with the algorithm for dynamics optimization from chapter 3, is capable of generating whole-body motions that satisfy kinematic constraints such as joint limits, as well as dynamic constraints such as those for friction force limits. This is achieved by using the kino-dynamic formulation [79] that by exploiting the separability of the momentum dynamics makes it possible to focus only on each sub-problem and build a dedicated solver for it based on its mathematical structure.

The remainder of this chapter is structured as follows: Background material was presented in section 1.6. In section 4.2 the motion optimization algorithm is detailed. Experimental results are shown in section 4.3 and conclusions in section 4.5.

4.2 OPTIMIZATION OF KINEMATIC MOTION PLANS

This section will detail the proposed algorithm for kinematic motion planning (that attempts to find a locally optimal solution of problem (5)) tailored for floating-base rigid-body systems such as robots with arms and legs. The algorithm is based on a sequential convex approximation of the problem and the reformulation of soft-nonconvexities as way-points in obstacle-free configuration space.

The main goal of the kinematics optimization problem is to find a trajectory of robot postures that satisfies the following conditions: good tracking of center of mass, dynamic momentum and collision-free endeffector motion trajectories, and the generation of smooth profiles of joints velocities and accelerations if possible. To accomplish these objectives, the proposed approach will first find a sequence of robot postures using an iterative inverse kinematics approach. Then, if required it will update the sequence of postures so as to e.g. satisfy obstacle avoidance constraints, improve smoothness of the solution, and achieve good tracking performance of the desired robot's dynamics momenta (computed from contact wrenches).

The next subsection will describe a typical iterative inverse kinematics approach and how it is updated within the proposed formulation to obtain a first guess of a locally optimal trajectory of robot postures. Then, the strategies to handle soft-nonconvexities and a looking-ahead horizon will be introduced.

4.2.1 *Finding a Local Trajectory of Robot Postures*

In this subsection, first a standard iterative inverse kinematics algorithm is introduced and then its updated version, capable of generating trajectories of robot postures that satisfy the conditions required to synthesize consistent kino-dynamic whole-body movement plans.

Algorithm 1 describes the basic procedure in which a standard iterative inverse kinematics (IK) algorithm optimizes a movement trajectory [170]. It assumes a desired motion in cartesian space for reference points is given (such as the center of mass and endeffector trajectories for a legged robot) and resolves the configuration in joint space that achieves as close as possible the reference cartesian trajectories. The desired center of mass motion trajectory can be obtained e.g. as explained in

Chapter 3, and endeffector trajectories by interpolating between contact locations using way-points to guide the endeffector motion through obstacle-free spaces.

Algorithm 1 Standard Iterative Inverse Kinematics Algorithm

```
Given:
    • Desired CoM $\mathbf{r}_t^{\text{des}}$ and endeffector $\mathbf{p}_{e,t}^{\text{des}}$ trajectories $\forall e, t$.
Initialize:
    • $\mathbf{q}_t = \mathbf{q}_{\text{nom}}$, to nominal or initial robot's posture.
    • $\delta t$ to the timestep for iterative inverse kinematics.
Optimize:
for $t = 1, ..., T$ do
    do
        • Update desired cartesian motion velocities
              $\dot{\mathbf{r}}^{\text{des}} = \frac{1}{\delta t}\left[\mathbf{r}_t^{\text{des}} - \mathbf{r}(\mathbf{q}_t)\right], \quad \dot{\mathbf{p}}_e^{\text{des}} = \frac{1}{\delta t}\left[\mathbf{p}_{e,t}^{\text{des}} - \mathbf{p}_e(\mathbf{q}_t)\right]$
        • Find robot velocity solving the optimization problem
              $\min_{\dot{\mathbf{q}}} \quad \mathcal{Q}(\dot{\mathbf{r}}^{\text{des}} - \mathbf{J}_{\mathbf{q}}^{\text{CoM}}\dot{\mathbf{q}}) + \sum_{\text{eff}} \mathcal{Q}(\dot{\mathbf{p}}_e^{\text{des}} - \mathbf{J}_{\mathbf{q}}^{\text{eff}}\dot{\mathbf{q}})$
        • Update posture $\mathbf{q}_t$ with velocity $\dot{\mathbf{q}}$ during timestep $\delta t$.
    while $\mathcal{Q}(\dot{\mathbf{r}}^{\text{des}}) + \sum_e \mathcal{Q}(\dot{\mathbf{p}}_e^{\text{des}}) < \epsilon$
end for
```

At each timestep $t \in [1, T]$ of the movement trajectory, a standard iterative IK algorithm computes the required center of mass $\dot{\mathbf{r}}_{\text{des}}$ and endeffector $\dot{\mathbf{p}}_e^{\text{des}}$ velocities that would move the robot from the current CoM $\mathbf{r}(\mathbf{q}_t)$ and endeffector $\mathbf{p}_e(\mathbf{q}_t)$ locations to the desired CoM $\mathbf{r}_t^{\text{des}}$ and endeffector $\mathbf{p}_{e,t}^{\text{des}}$ configurations. Then, using a local approximation of the velocities at the current robot posture $\mathbf{q}_t$, the joint space velocities are optimized so that they realize the desired cartesian space velocities. $\mathcal{Q}(\cdot)$ denotes a general quadratic penalty cost; $\mathbf{J}_{\mathbf{q}}^{\text{CoM}}$ and $\mathbf{J}_{\mathbf{q}}^{\text{eff}}$ denote the CoM and endeffector jacobians at the current robot posture, which is updated at each iteration by integrating the optimized robot velocity $\dot{\mathbf{q}}$ over the timestep discretization δt. At each timestep, the algorithm converges when the norm of the required velocities or its rate of change is below a desired threshold ϵ.

Further improvements to the basic algorithm have been proposed in the literature, such as the introduction of redundancy resolution strategies via optimal control principles [171] (very successful in humanoid robotics [172, 173]) or the use of strictly prioritized tasks using hierarchies [174, 175, 176].

The following optimization problem is an adaptation of the iterative inverse kinematics algorithm 1, originally described in a differential form [170], to be used in the proposed architecture for the construction of a sequence of kino-dynamically consistent robot postures. It is applied at each timestep t for a number of iterations k until its convergence to the desired tolerance level.

$$\min_{\mathbf{v}_\mathrm{t}, \mathbf{q}_\mathrm{t}^\mathrm{k}, \dot{\mathbf{q}}_\mathrm{t}^\mathrm{k}, \ddot{\mathbf{q}}_\mathrm{t}^\mathrm{k}, \mathbf{h}_\mathrm{t}^\mathrm{k}, \mathbf{p}_\mathrm{e,t}^\mathrm{k}} \quad \phi_{\mathrm{reg}}^{\mathrm{kin}}(\mathbf{q}_\mathrm{t}^\mathrm{k}, \dot{\mathbf{q}}_\mathrm{t}^\mathrm{k}, \ddot{\mathbf{q}}_\mathrm{t}^\mathrm{k}) + \phi_{\mathrm{track}}^{\mathrm{kin}} \begin{pmatrix} \mathbf{h}_\mathrm{t}^\mathrm{k} - \mathbf{h}_\mathrm{t}^\mathrm{des} \\ \mathbf{p}_\mathrm{e,t}^\mathrm{k} - \mathbf{p}_\mathrm{e,t}^\mathrm{des} \end{pmatrix} \tag{17a}$$

$$\text{st.} \quad \begin{bmatrix} \mathbf{q}_\mathrm{t}^\mathrm{k} \\ \mathbf{p}_\mathrm{e,t}^\mathrm{k} \\ \mathbf{r}_\mathrm{t}^\mathrm{k} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_\mathrm{t}^\mathrm{k-1} \\ \mathbf{p}_\mathrm{e,t}^\mathrm{k-1} \\ \mathbf{r}_\mathrm{t}^\mathrm{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_\mathrm{t} \\ \mathbf{J}_\mathbf{q}^\mathrm{eff} \mathbf{v}_\mathrm{t} \\ \mathbf{J}_\mathbf{q}^\mathrm{CoM} \mathbf{v}_\mathrm{t} \end{bmatrix} \delta t \tag{17b}$$

$$\dot{\mathbf{q}}_\mathrm{t}^\mathrm{k} = \frac{1}{\Delta_\mathrm{t}} \left[\mathbf{q}_\mathrm{t}^\mathrm{k} - \mathbf{q}_{\mathrm{t}-1} \right] \tag{17c}$$

$$\ddot{\mathbf{q}}_\mathrm{t}^\mathrm{k} = \frac{1}{\Delta_\mathrm{t}} \left[\dot{\mathbf{q}}_\mathrm{t}^\mathrm{k} - \dot{\mathbf{q}}_{\mathrm{t}-1} \right] \tag{17d}$$

$$\begin{bmatrix} \mathbf{l}_\mathrm{t}^\mathrm{k} \\ \mathbf{k}_\mathrm{t}^\mathrm{k} \end{bmatrix} = \mathbf{A}_\mathbf{q} \dot{\mathbf{q}}_\mathrm{t}^\mathrm{k} \tag{17e}$$

$$\mathbf{q}_\mathrm{j,t}^\mathrm{k} \in [\mathbf{q}_\mathrm{j}^\mathrm{min}, \mathbf{q}_\mathrm{j}^\mathrm{max}] \; \forall \mathrm{j} \in \mathbb{R}^n \tag{17f}$$

At each timestep t, the algorithm builds a sequence of length $k + 1$ of kinematic solutions ($[\mathbf{q}_\mathrm{t}^\mathrm{o}, \mathbf{p}_\mathrm{e,t}^\mathrm{o}]$, ... , $[\mathbf{q}_\mathrm{t}^\mathrm{k}, \mathbf{p}_\mathrm{e,t}^\mathrm{k}]$) taking as starting point the kinematic solution of the previous timestep ($[\mathbf{q}_\mathrm{t}^\mathrm{o}, \mathbf{p}_\mathrm{e,t}^\mathrm{o}]$ = $[\mathbf{q}_\mathrm{t\text{-}1}, \mathbf{p}_\mathrm{e,t\text{-}1}]$) and as output the final kinematic state ($[\mathbf{q}_\mathrm{t}, \mathbf{p}_\mathrm{e,t}]$ = $[\mathbf{q}_\mathrm{t}^\mathrm{k}, \mathbf{p}_\mathrm{e,t}^\mathrm{k}]$), such that the total robot velocity required to generate the kinematic motion from $[\mathbf{q}_\mathrm{t\text{-}1}, \mathbf{p}_\mathrm{e,t\text{-}1}]$ to $[\mathbf{q}_\mathrm{t}, \mathbf{p}_\mathrm{e,t}]$ satisfies the kinematic limits and produces the desired momentum and cartesian endeffector motion.

In this optimization problem, the algorithm starts from a previous solution ($\mathbf{q}_\mathrm{t}^\mathrm{k-1}$, $\mathbf{p}_\mathrm{e,t}^\mathrm{k-1}$, $\mathbf{r}_\mathrm{t}^\mathrm{k-1}$), which can be modified using an instantaneous robot velocity $\mathbf{v}_\mathrm{t}$ integrated during a small timestep δt to compose a new solution ($\mathbf{q}_\mathrm{t}^\mathrm{k}$, $\mathbf{p}_\mathrm{e,t}^\mathrm{k}$, $\mathbf{r}_\mathrm{t}^\mathrm{k}$), as shown in equation (17b). The newly composed solution will be updated so as to satisfy joint limit constraints (17f), and will be used to update total velocities (17c), accelerations (17d) and generated robot momenta (17e) in such a way that regularization of the state and tracking performance of the solution are improved as desired by the objective function (see equation (17a)). For simplicity, quaternions representing base orientations can be treated as 3D-vectors using logarithmic and exponential transformations as needed [177]. Δ_t, different from the small timestep δt, corresponds to the total time elapsed between the solution at timestep $t - 1$ and the one at t. Despite not explicitly shown, constraints on joint velocities or acceleration could also be enforced if required, in the same way as for joint limits. Finally, the algorithm converges when the norm of the robot velocity $\mathbf{v}_t$ or the difference between two consecutive solutions falls below a desired threshold.

4.2.2 *Optimizing a Time-Horizon Trajectory*

The last subsection presented an approach to obtain a sequence of robot postures capable of satisfying the required conditions to generate a kino-dynamically consistent whole-body motion. This subsection takes this approach one step forward, in the sense that by making use of the previously found solution, it is capable of constructing a kinematic motion based on a time-horizon trajectory optimization. One of the advantages of this improvement is the capability to look ahead and thus of considering actions for long-term optimality that can improve e.g. the tracking performance of momentum trajectories or achieve smoother movement plans. This algorithm starts from a previously found locally optimal solution $k-1$ and, iteratively updates it to improve momentum tracking and smoothness of the motion plan. Formally, the following problem is solved:

$$\min_{\mathbf{v}_t, \mathbf{q}_t^k, \dot{\mathbf{q}}_t^k, \ddot{\mathbf{q}}_t^k, \mathbf{h}_t^k, \mathbf{p}_{e,t}^k} \quad \sum_{t=1}^{T} \phi_{\text{reg}}^{\text{kin}}(\mathbf{q}_t^k, \dot{\mathbf{q}}_t^k, \ddot{\mathbf{q}}_t^k) + \phi_{\text{cons}}^{\text{kin}} \begin{pmatrix} \mathbf{h}_t^k - \mathbf{h}_t^{\text{des}} \\ \mathbf{p}_{e,t}^k - \mathbf{p}_{e,t}^{\text{des}} \end{pmatrix} \tag{18a}$$

$$\text{st.} \quad \begin{bmatrix} \mathbf{q}_t^k \\ \mathbf{p}_{e,t}^k \\ \mathbf{r}_t^k \end{bmatrix} = \begin{bmatrix} \mathbf{q}_t^{k-1} \\ \mathbf{p}_{e,t}^{k-1} \\ \mathbf{r}_t^{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_t \\ \mathbf{J}_{\mathbf{q}}^{\text{eff}} \mathbf{v}_t \\ \mathbf{J}_{\mathbf{q}}^{\text{CoM}} \mathbf{v}_t \end{bmatrix} \delta t \tag{18b}$$

$$\dot{\mathbf{q}}_t^k = \frac{1}{\Delta_t} \left[\mathbf{q}_t^k - \mathbf{q}_{t-1}^k \right] \tag{18c}$$

$$\ddot{\mathbf{q}}_t^k = \frac{1}{\Delta_t} \left[\dot{\mathbf{q}}_t^k - \dot{\mathbf{q}}_{t-1}^k \right] \tag{18d}$$

$$\begin{bmatrix} \mathbf{l}_t^k \\ \mathbf{k}_t^k \end{bmatrix} = \mathbf{A}_{\mathbf{q}} \dot{\mathbf{q}}_t^k \tag{18e}$$

$$\mathbf{q}_{j,t}^k \in [\mathbf{q}_j^{\min}, \mathbf{q}_j^{\max}] \ \forall \mathrm{j} \in \mathbb{R}^n \tag{18f}$$

In this optimization problem, the algorithm starts from a previous solution ($\mathbf{q}_t^{k-1}$, $\mathbf{p}_{e,t}^{k-1}$, $\mathbf{r}_t^{k-1}$) such as the one obtained in the previous subsection, which can be iteratively improved using an instantaneous robot velocity $\mathbf{v}_t$ integrated during a small timestep discretization δt to compose an updated solution ($\mathbf{q}_t^k$, $\mathbf{p}_{e,t}^k$, $\mathbf{r}_t^k$), as shown in equation (18b). The newly composed solution is also subject to kinematic constraints such as joint limits (18f), and is used to update total robot velocities (18c), accelerations (18d) and generated momenta (18e) in such a way that the problem objectives are optimized for long-term optimality. These goals might include e.g. smoothness of the solution and good tracking performance of desired quantities such as momentum and endeffector motion trajectories (see equation (18a)).

So far, the algorithm described is capable of producing kinematic movement plans considering actions for long-term optimality. It could be further improved using slack variables [29] or soft-constraints [55] to enhance convergence properties or ease the finding of a solution. However, it would still be difficult to handle soft-nonconvexities, such as those present in obstacle avoidance constraints [178, 179] or manipulability objectives [180], because the nonconvex nature of these constraints and objectives as well as the problem's dimensionality would make it hard to solve such a problem. Under this setting, the following subsection presents an alternative to consider soft-nonconvexities, yet to conserve computational efficiency.

4.2.3 *Incorporating Soft-Nonconvexities*

This subsection presents an approach in which soft-nonconvexities, such as nonconvex terms in the objective function (e.g. rewards for staying away from singular configurations) or nonconvex inequality constraints (e.g. those for obstacle avoidance), could be taken into account in a trajectory optimization problem. The main idea of the approach, inspired on how distributed or consensus optimization methods find a solution to a given problem [110], is to reformulate the nonconvexities of the kinematics trajectory optimization as independent separate problems, whose solutions can then be used as way-points in configuration space that bias an overall solution towards those that satisfy the nonconvex constraints.

For instance, to solve a problem with obstacle avoidance constraints, one could reformulate them as way-points through obstacle-free configuration spaces for the endeffector motion. The approach could for example iterate between 1) a whole-body motion optimization that attempts to find an overall solution close to the in configuration space given way-points that satisfy obstacle avoidance constraints, and then 2) a set of smaller distributed problems for each endeffector that search for a solution that satisfies the non-penetration obstacle avoidance constraint for each endeffector and as close as possible to the overall solution (this projection over the feasible manifold can even in some simple cases be performed analytically [111]). Naturally, the first iteration of the whole body optimization could be performed without considering obstacle constraints. Then, the algorithm iterates until a convergence criteria is met such as finding a feasible solution.

Before presenting the proposed approach, in the following a parenthesis is made to motivate the approach by explaining the way in which a consensus optimization algorithm solves a problem.

Consensus Optimization

A general optimization problem can be written as the minimization of a cost $\mathrm{f}(\mathbf{x})$, subject to a set of i constraints, that denote the membership of the optimization vector $\mathbf{x}$ to the possibly non-convex sets $\mathcal{S}_i$.

$$\begin{array}{lll} \min_{\mathbf{x}} & \mathrm{f}(\mathbf{x}) \\ \text{st.} & \mathbf{x} \in \mathcal{S}_i \end{array} \implies \begin{array}{ll} \min_{\mathbf{x},\mathbf{z}_i} & \mathrm{f}(\mathbf{x}) + \sum_i \mathrm{g}_i(\mathbf{z}_i) \\ \text{st.} & \mathbf{z}_i = \mathbf{x} \quad \forall i \end{array}$$

In consensus optimization, each constraint $\mathbf{x} \in \mathcal{S}_i$ is replaced by the indicator function $\mathrm{g}_i(\mathbf{z}_i)$, which is zero if the constraint is satisfied and infinity otherwise. Additionally, the variables $\mathbf{z}_i$ are used to make the objective separable and the consensus constraints $\mathbf{z}_i = \mathbf{x}$ are introduced to make the variables $\mathbf{x}$ and $\mathbf{z}_i$ agree upon convergence. Using this canonical form, a consensus algorithm can be derived by minimizing its Augmented Lagrangian, given by:

$$\mathcal{L} = \mathrm{f}(\mathbf{x}) + \sum_i \left(\mathrm{g}_i(\mathbf{z}_i) + \alpha_i^T(\mathbf{z}_i - \mathbf{x}) + \frac{\rho}{2} \|\mathbf{z}_i - \mathbf{x}\|_2^2 \right)$$

where α_i denote the dual variables of the consensus constraints, and the quadratic terms robustify convergence. Out of the optimality conditions, the following iterates for primal ($\mathbf{x}$, $\mathbf{z}_i$) and dual variables ($\mathbf{u}_j$) can be obtained:

$$\mathbf{z}_i^{k+1} \leftarrow \operatorname{argmin}_{\mathbf{z}_i} \left\{ \mathrm{g}_i(\mathbf{z}_i) + \frac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{x}^k + \mathbf{u}_i^k \right\|_2^2 \right\} , \forall i \tag{19a}$$

$$\mathbf{x}^{k+1} \leftarrow \operatorname{argmin}_{\mathbf{x}} \left\{ \mathrm{f}(\mathbf{x}) + \frac{\rho}{2} \sum_i \left\| \mathbf{z}_i^{k+1} - \mathbf{x} + \mathbf{u}_i^k \right\|_2^2 \right\} \tag{19b}$$

$$\mathbf{u}_i^{k+1} \leftarrow \mathbf{u}_i^k + \mathbf{z}_i^{k+1} - \mathbf{x}^{k+1} , \forall i \tag{19c}$$

The form of these iterates is known as the scaled version of the algorithm, because it makes use of the scaled dual variables $\mathbf{u}_i = \frac{\alpha_i}{\rho}$. Notice that the updates of consensus variables $\mathbf{z}_i$ can be performed in parallel.

To sum up, a consensus method 1) benefits from parallel updates of its primal variables $\mathbf{z}_i$ (see equation (19a)) based on the projection of its desired value $\mathbf{x}^k - \mathbf{u}_i^k$ over its feasible manifold $g_i(\mathbf{z}_i)$. Then, 2) it updates the primal variable $\mathbf{x}$ (equation (19b)) so as to minimize its objective $f(\mathbf{x})$ selecting values close to $\mathbf{z}_i + \mathbf{u}_i^k$ $\forall i$ that satisfy the constraints $g_i(\mathbf{z}_i)$. Notice that solving small distributed problems and combining its solutions into an overall solution until consensus can be much less computationally expensive than solving the joint problem. Also note that as this first-order algorithm progresses, its iterates approach feasibility and its objective the optimal value; however the optimization vectors $\mathbf{z}_i$ do not necessarily approach its optimal value, but biased values $\mathbf{x}^k - \mathbf{u}_i^k$; and thus the optimization vector $\mathbf{x}$ does not converge to the optimal values.

What is very remarkable from this approach is the fact that it makes it possible to formulate and solve an optimization problem in a different way and that by means of this alternative way it is capable of building the solution for an optimization problem that can incorporate soft-nonconvexities without the need of solving the entire problem as a nonconvex one. In this way, part of the problem can be solved exploiting its sparsity and convexity using a dedicated solver for convex programming, and the other parts using a nonlinear nonconvex solver in the general case. In particular, for the kinematics trajectory optimization problem, an overall solution can be constructed by iterating between solving an instance of the whole-body motion optimization problem as shown in section 4.2.1 or 4.2.2, and updating the desired cartesian references for the endeffector motions such that they satisfy obstacle avoidance constraints, as shown in the following.

An obstacle avoidance constraint can be expressed as the distance between an obstacle and an endeffector position being greater than zero $\mathcal{D}(\mathbf{p}_{\text{obs}}, \mathbf{p}_{\text{e,t}}) \geq 0$. This constraint has to be satisfied simultaneously for all timesteps and endeffectors. In the proposed approach, they can be updated separately for each endeffector using the following optimization problem:

$$\begin{aligned} \min_{\mathbf{q}_{\text{e,t}}} \quad & \sum_{\text{t}} \mathcal{Q}(\mathbf{p}_{\text{e,t}} - \mathbf{q}_{\text{e,t}}) \\ \text{st.} \quad & \mathcal{D}(\mathbf{p}_{\text{obs}}, \mathbf{q}_{\text{e,t}}) \geq 0 \quad \forall \text{t} \end{aligned}$$

such that it finds the set of points $\mathbf{q}_{\text{e,t}}$ that are the closest ones to $\mathbf{p}_{\text{e,t}}$ through which the original endeffector trajectories originally pass, but that satisfy obstacle avoidance constraints. Notice that, the optimization problems that need to be solved to find the points $\mathbf{q}_{\text{e,t}}$ are still nonconvex, but smaller and can thus be solved faster than a big single joint problem. Further, note that the whole-body motion trajectory is then optimized so as to realize a collision free trajectory, which is much easier than directly finding a collision-free trajectory within whole-body joint configuration space. Notice as well that endeffector trajectories need to be updated only if there is a collision. A similar procedure could be applied at each timestep to bias the robot posture towards that one that maximizes the robot's manipulability a a function of the set of contacts.

To sum up, Figure 21 summarizes the algorithm for the optimization of a sequence of robot postures that satisfy the requirements to generate consistent kino-dynamic whole-body motions.

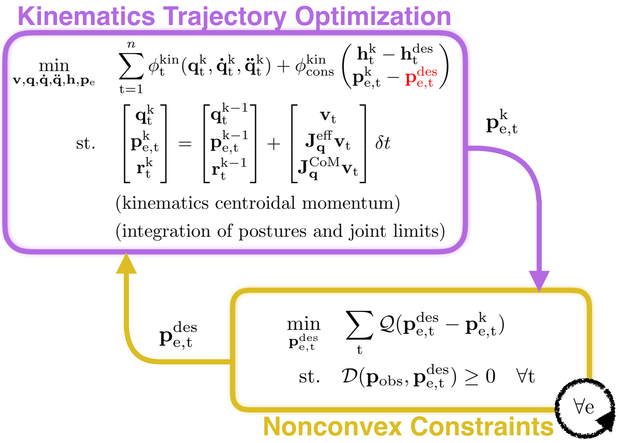

Kinematics Trajectory Optimization
$\min_{\mathbf{v},\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},\mathbf{h},\mathbf{p}_e} \sum_{t=1}^{n} \phi_t^{kin}(\mathbf{q}_t^k, \dot{\mathbf{q}}_t^k, \ddot{\mathbf{q}}_t^k) + \phi_{cons}^{kin}\begin{pmatrix}\mathbf{h}_t^k - \mathbf{h}_t^{des}\\ \mathbf{p}_{e,t}^k - \mathbf{p}_{e,t}^{des}\end{pmatrix}$
$\text{st.} \begin{bmatrix}\mathbf{q}_t^k\\ \mathbf{p}_{e,t}^k\\ \mathbf{r}_t^k\end{bmatrix} = \begin{bmatrix}\mathbf{q}_t^{k-1}\\ \mathbf{p}_{e,t}^{k-1}\\ \mathbf{r}_t^{k-1}\end{bmatrix} + \begin{bmatrix}\mathbf{v}_t\\ \mathbf{J}_\mathbf{q}^{eff}\mathbf{v}_t\\ \mathbf{J}_\mathbf{q}^{CoM}\mathbf{v}_t\end{bmatrix} \delta t$
(kinematics centroidal momentum)
(integration of postures and joint limits)
$\mathbf{p}_{e,t}^k$
$\mathbf{p}_{e,t}^{des}$
$\min_{\mathbf{p}_{e,t}^{des}} \sum_t \mathcal{Q}(\mathbf{p}_{e,t}^{des} - \mathbf{p}_{e,t}^k)$
$\text{st.} \quad \mathcal{D}(\mathbf{p}_{obs}, \mathbf{p}_{e,t}^{des}) \geq 0 \quad \forall t$
$\forall e$
Nonconvex Constraints

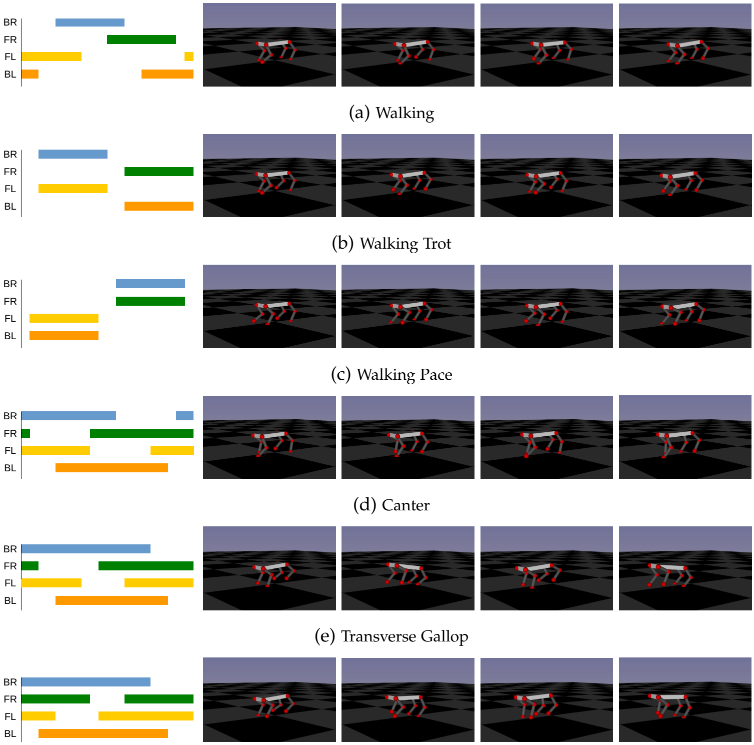
FIGURE 21: Summary of the kinematics optimization algorithm: 1) whole-body trajectory optimization, and 2) reformulation of soft-nonconvexities as way-points in obstacle-free configuration space.

4.3 EXPERIMENTAL RESULTS

This section will present experimental results about the optimization of kinematic movement plans using the algorithms previously described. Its capabilities and limitations are tested in several multi-contact scenarios including several gaits for a quadruped robot such as walking, trotting, galloping and many others.

4.3.1 *On the Optimization of Kinematic Movement Plans*

This section demonstrates the capabilities of the proposed algorithms at generating whole-body kino-dynamically consistent motions. The algorithm of section 4.2.2 was used to optimize motions for the quadruped robot. Figure 22 shows a few snapshots and gait graphs of these optimized gaits. Later, two motions will be examined in detail: a transverse galloping motion (that involves simultaneous flight phases for all endeffectors) and an acrobatic motion (where two endeffectors always remain in contact and the other two have a flight phase).



(a) Walking

(b) Walking Trot

(c) Walking Pace

(d) Canter

(e) Transverse Gallop

(f) Rotary Gallop

Figure 22: Examples of kino-dynamically movement plans: Several optimized gaits for a quadruped robot are shown, including walk, trot, pace, canter, transverse- and rotaty gallop. *To the left*, gait graphs display in color the flight phases of each endeffector during one stepping cycle. *BR* denotes the back-right foot, *BL* the back-left foot, *FR* the front-right foot, and *FL* the front-left foot. *To the right*, snapshots of the optimized movement plans during a gait cycle are shown.

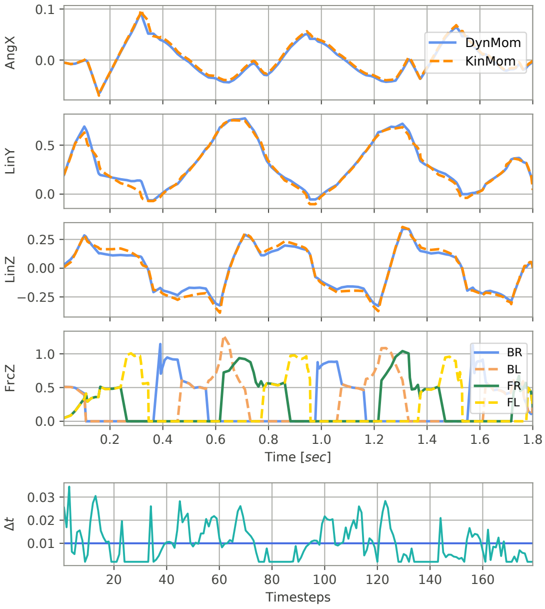


FIGURE 23: Whole-body motion and contact force trajectories of a transverse galloping movement. A full description of the plots in this figure are given in section 4.3.1.1.

4.3.1.1 *Transverse Galloping Motion*

This section analyzes a galloping motion, which is a very fast and dynamic movement and thus allows to test the capabilities of the approach proposed in section 4.2.2 at generating whole-body motion trajectories consistent with the required dynamic momentum (computed from gravitational and external contact wrenches) in a very challenging motion. A galloping motion is a gait that involves flight phases for three and four endeffectors simultaneously with only short stages where at most there are two endeffectors simultaneously in contact. Figure 22e depicts its gait graph, where flight phases are visualized in color for each endeffector, while stance phases in white color along a gait cycle.

Figure 23 shows optimization results for a transverse galloping motion with flight phases for all endeffectors simultaneously, as well as its corresponding momentum trajectories normalized by the robot mass. The top three rows show the matching of

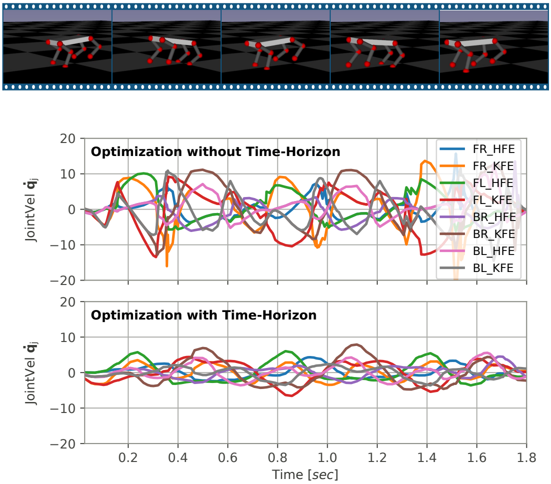


FIGURE 24: Comparison of joint velocities in a kinematic optimization that does and does not consider a time-horizon. *BR* denotes the back-right endeffector foot, *BL* the back-left foot, *FR* the front-right foot, and *FL* the front-left foot. While *HFE* denotes the hip-flexion-extension joint and *KFE* the knee flexion-extension joint.

momentum trajectories computed kinematically and dynamically. The Y direction is for forward motion, Z for upward-downward movements and X the transversal direction around which base and joints rotate generating angular momentum. Vertical forces for all endeffectors (normalized by the robot weight) as well as optimal durations of timestep discretizations are shown in the two bottom plots. Notice that momentum trajectories $DynMom$ computed from dynamic quantities (external and gravitational forces) and $KinMom$ computed from kinematic quantities (robot postures and velocities) can be brought to agreement, despite the dynamic motion being very hard to optimize. Key elements for the algorithmic convergence are the ability of the dynamics optimizer to adapt timestep discretizations during stance and flight phases, as well as the ability of the kinematics optimizer to consider a time-horizon when updating the trajectories of robot postures and velocities.

Notice as well in Figure 24 that an additional benefit of considering a sequential convex approximation of the time-horizon kinematic optimization problem is that the degree of smoothness in joint trajectories can be improved, leading to motion trajectories that are potentially easier to track and execute.

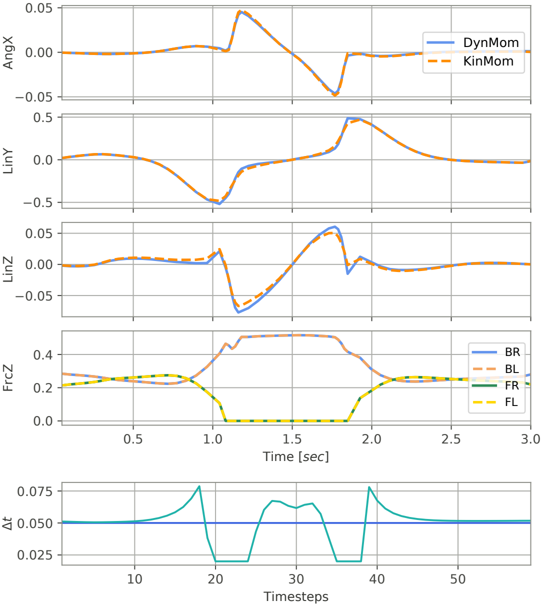


Figure 25: Whole-body motion and contact force trajectories of an acrobatic movement where the quadruped robot stands in only two feet. A full description of the plots in this figure are given in section 4.3.1.2.

4.3.1.2 *Acrobatic Motion - Standing in Two Feet*

The acrobatic motion, as the previous one, is also a fast and dynamic movement. In this motion, the quadruped robot keeps always its front feet in contact, while the other two are lifted up simultaneously. To this end, the robot moves its center of mass forward close to the line formed by the front feet and adapts the timings in which the motion is performed, so as to generate a consistent whole-body motion.

Figure 25 displays the optimization results for an acrobatic motion where the quadruped robot stands in only two feet. In the top three rows, momentum trajectories computed kinematically and dynamically are shown. The Y direction is for forward motion, Z for upward-downward movements and X the transversal direction around which base and joints rotate generating angular momentum. Vertical forces for all endeffectors (normalized by the robot's weight) as well as optimal durations of timestep discretizations are shown in the two bottom plots. Figure 25

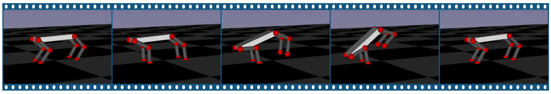


Figure 26: Snapshots of an acrobatic movement.

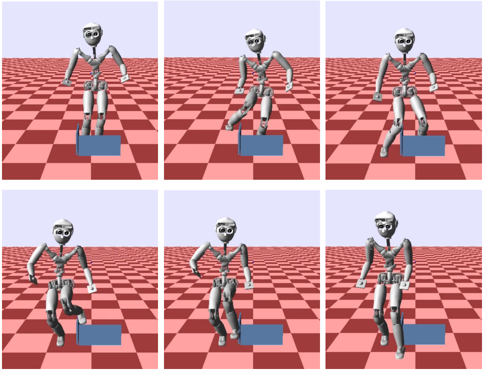


Figure 27: Execution on physical simulator of obstacle avoidance task.

shows the matching of kinematic and dynamic momentum trajectories (normalized by the robot's mass). Note that endeffector contact forces are equal for front and back feet because of the symmetry of the motion. It is worth mentioning that being able to optimize over time, helps to reduce the values of momentum that need to be kinematically generated (for instance, by reducing the flight time for getting up or down), and optimizing over a horizon makes it easier to find such a kinematic trajectory able to generate the desired momentum. Finally, Figure 26 shows snapshots of the motion.

4.3.1.3 *Obstacle Avoidance Task*

This section shows a task where a humanoid robot plans and executes a motion where it needs to get out of a box and thus needs to appropriately decide on the endeffector trajectories that will allow it to generate the motion without colliding with the environment. It does so by combining its ability to discover a motion (section 4.2.1) and avoid obstacles (section 4.2.3) Snapshots of this simple yet exemplary motion are visible in Figure 27 and show the proposed approach's ability to find a collision free motion.

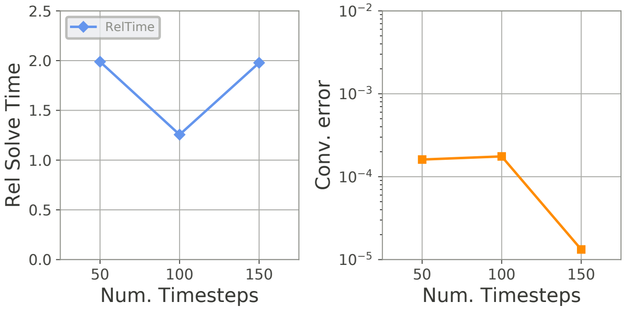


FIGURE 28: Statistics about the relative time required to solve a kinematics optimization problem in comparison with the time required to solve the corresponding time-optimal dynamics optimization problem of the same size as well as the achieved levels of precision to kinematic feasibility upon convergence.

4.3.2 *Solving Time and Convergence*

This section presents statistics about the required time to solve a kinematics optimization problem and the achieved values of precision upon algorithmic convergence. Figure 28 summarizes these two results.

To the left, it shows statistics about the relative time required to solve a kinematics optimization problem in comparison with the time required to solve the corresponding time-optimal dynamics optimization problem of the same size. For instance, for kinematic optimization problems that consider a time-horizon of 200 timesteps, the solving time is twice the time required to solve a dynamics trajectory optimization problem of the same size. *To the right*, this figure shows the errors in kinematic feasibility achieved once the algorithm has converged, i.e. the difference between the center of mass an momentum trajectories generated by a kinematics and a dynamics optimization problem. To generate these statistics, the problems solved include those in Figure 22 and Figure 26, which are 7 in total. Out of these optimized motions, the best and worst outlier results were removed and then the rest of samples were averaged to generate the results shown.

Notice that since the motions used to generate the statistics using the algorithm described in section 4.2.2 are still simple in terms of obstacle avoidance constraints, the algorithm only requires one pass. Also note the tendency that the time required to solve a kinematics trajectory optimization problem is more expensive than the time required to solve the time-optimal dynamics optimization problem, around twice as much expensive. However, it only required one iteration of kino-dynamic alternation with fixed contacts to find a solution to the described levels of precision. The implementation of the kinematics optimizer was done based on the fast analytical derivatives of rigid body algorithms provided by [181]. It is worth highlighting that the algorithm described in section 4.2.1 is less computationally expensive and is still able to generate solutions that converge to the desired precision of kino-dynamic feasibility; however more kino-dynamic iterations might be required to converge to a solution easily executable on a physical simulator.

4.4 DISCUSSION

The presented experiments have shown that the proposed approach for kinematics optimization is capable of synthesizing a whole-body motion trajectory that satisfies the constraints for kinematic feasibility as well as the momentum tracking objectives required to generate a kino-dynamically consistent movement plan.

From a computational point of view, the algorithm, as shown in the experimental section, scales to complex and large optimization problems. It is built based on the implementation of an interior-point method (based on [104]). This method works well for solving the problems described in section 4.2.2. However, to obtain an initial solution based on the method described in section 4.2.1 there are alternatives that should work faster. For instance, an interior point method cannot be warm-started as an active set one [182], which could be extensively exploited in this problem given its sequential nature to significantly speed up it solution.

In comparison with other approaches, the proposed method is a middle-ground one between: on the one hand, very flexible but also computationally costly methods based on the dynamics description of a rigid-body system [29, 55, 84] and on the other, faster methods for kinematic optimization of fixed-base manipulator trajectories [168, 169]. Thus, the proposed method offers a trade-off between flexibility or diversity of motions that can be synthesized and the time required to find such a solution. It is hard to justify the computational advantages of solving this problem in a distributed manner, given the limited number of experiments performed. But, the vision is to incorporate e.g. other optimization criteria such as the maximization of manipulability of center of mass and endeffectors [180] at given contact poses, and in cases like this, the distributed nature of the algorithm would come in very handy, as some problems could be independently solved and its information incorporated into a consensus problem, which is less computationally expensive to solve than a joint trajectory optimization one.

4.5 CONCLUSION

This chapter has presented a structured and efficient algorithm for kinematic planning of whole-body motions, and has shown experimental evidence that evaluates its performance at generating a variety of motions in several multi-contact scenarios traversed by a legged robot.

Chapter 5

FEEDBACK CONTROL DESIGN

Abstract

The physical interaction of robots with an uncertain environment through contacts is at the heart of many robotic applications. Environmental uncertainty can happen in the form of external disturbances (noise in the process model), but can also come from the lack of precise knowledge of the world (noise in the measurement model). Typically, feedback controllers ignore these noise statistics, which is a reasonable assumption for systems under low noise levels; however, a strategy able to reason about noise and the cost of this uncertainty would be more appropriate for systems under large noise intensity. Taking this observation into account, this chapter presents a feedback control architecture for a humanoid robot that exploits the kino-dynamic nature of movement plans using the superposition property of impedances to optimally execute whole-body motions, and furthermore, the proposed architecture can benefit from the use of feedback controllers sensitive to measurement noise to synthesize locally-optimal policies, whose compliance is a function of the uncertainty. The algorithm for risk-sensitive control exploits in the first place an exponential transformation of the performance criteria to synthesize policies that do not only optimize the expected value of the performance criteria, but also its higher order moments (cost of uncertainty), and second it includes the dynamics of an observer in such a way that the control law explicitly depends upon the covariance of estimation errors and process noise. Experimental results will show that high measurement uncertainty leads to low impedance behaviors, a result in contrast with the effects of process noise that creates stiff behaviors. Simulation results on a humanoid robot will also show the benefits of the proposed architecture over traditional approaches to control a humanoid robot that physically interacts with the environment under uncertain contact locations.

Note:
The contents of this chapter constitute an extended version of the conference paper [129].

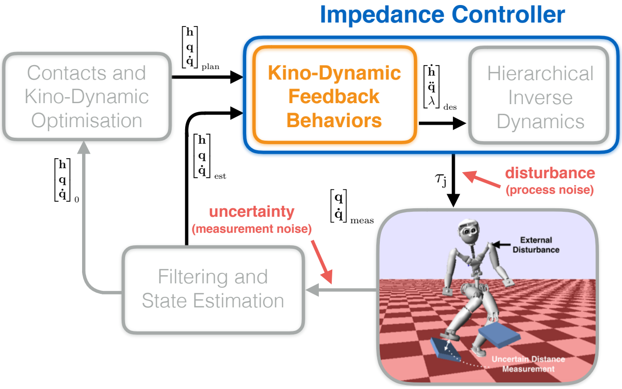

Impedance Controller
Contacts and Kino-Dynamic Optimisation
$\begin{bmatrix} h \\ q \\ \dot{q} \end{bmatrix}_{plan}$
Kino-Dynamic Feedback Behaviors
$\begin{bmatrix} \dot{h} \\ \ddot{q} \\ \lambda \end{bmatrix}_{des}$
Hierarchical Inverse Dynamics
$\begin{bmatrix} h \\ q \\ \dot{q} \end{bmatrix}_{est}$
$\begin{bmatrix} h \\ q \\ \dot{q} \end{bmatrix}_{0}$
τ_j
disturbance
(process noise)
uncertainty
(measurement noise)
$\begin{bmatrix} q \\ \dot{q} \end{bmatrix}_{meas}$
External Disturbance
Filtering and State Estimation
Uncertain Distance Measurement

FIGURE 29: Control diagram of a humanoid robot emphasizing where an how the two sources of uncertainty considered (process and measurement noise) enter the system. **h** denotes the robot's dynamic state (center of mass and centroidal momentum) and **q**, $\dot{\mathbf{q}}$ denote the robot's kinematic state (posture and velocity); subscripts denote measured, estimated, planned and initial states. The impedance controller (feedback control architecture) is composed by 1) noise-sensitive feedback controllers that define a control policy to optimally realize desired kinematic and dynamic movement plans coming from a high-level planner, and 2) by an inverse dynamics controller that maps desired kino-dynamic behaviors to robot torques τ_j via a convex constrained optimization problem that satisfies all physical constraints.

5.1 INTRODUCTION

One of the key elements in the way to fulfill the vision of robots that achieve human levels of competence at movement generation and control is the ability to safely interact with a dynamic environment by optimally adapting to its uncertainty. In this spirit, this chapter on the one hand proposes a feedback control architecture for the successful execution of kino-dynamic behaviors and on the other it studies the effect of considering measurement uncertainty upon the compliance or stiffness behavior of control policies in optimal control problems that involve contact interactions.

Figure 29 shows examples on a humanoid robot of the two sources of uncertainty analyzed in this chapter. First, external disturbance forces physically perturb the robot (as noise in the process model) and thus have a direct influence over its dynamic evolution. The goal of an impedance controller is then to scale down its effects by using a certain level of stiffness. Secondly, uncertainty in the belief about the robot's state inferred from noisy sensor readings (noise in the measurement model) has no direct effect over the system actuation, but instead it influences the decision making process of how control signals are selected based on the uncertain knowledge about the robot's state. For instance, in a hand-reaching task that involves an uncertain distance to the contact location due to visual noise, the reaching motion would require a gentle touch instead of an aggressive approach in order not to cause any self-harm.

As it will be shown within the experimental section, in general an increase in the magnitude of an external disturbance implies and increment of the robot's stiffness to maintain the desired tracking accuracy, while an increase in the magnitude of the uncertainty level on a measurement implies an increment on the level of compliance. Then, the distinction between the two sources of uncertainty examined in this chapter is very important, as it brings to mind the idea of modeling contact interactions between a humanoid robot and its uncertain environment (a fundamental problem in robotics [1]) as an optimal control problem with measurement uncertainty, out of which optimal impedance behaviors can be synthesized as a function of the uncertainty about the environment.

5.2 RELATED WORK

This section presents related works and discusses how the proposed architecture for feedback control distinguishes itself from other state-of-the-art techniques.

Optimal control approaches based on Bellman's Principle of Optimality [36], such as iterative Linear-Quadratic Gaussian (iLQG) [65] or Differential Dynamic Programming (DDP) [31, 39, 68], are very important in robotics due to its success at synthesizing complex motions for large dimensional systems [30, 66, 67]. These methods combine the benefits of a local method (as they maintain a single nominal trajectory) with those of a global method (as they improve the nominal trajectory iteratively along its neighborhood based on the optimality principle), which allows them to overcome to some extent the curse of dimensionality [63] of a global method and remain computationally efficient as a local one. The algorithm used in this chapter for the synthesis of feedback control policies belongs to this family with the added benefit that it can take into account and reason about measurement noise.

Generally speaking, these methods estimate the solution of the underlying nonlinear problem by iteratively solving a Taylor approximation of it [67, 68, 69, 183]. The linear (in the dynamics) and quadratic (in the cost) nature of this local problem description makes it possible to find solutions efficiently using a computationally tractable algorithm. However, this local approximation also poses limitations such as ignoring the effects of noise over the cost and dynamics, because it only considers the mean of a performance measure (expected value of a quadratic cost) for systems under purely additive noise, for which the solution of stochastic and deterministic problems is the same (Certainty Equivalence Principle). In other words, the estimator takes care of uncertainty by itself and control design is independent from noise. The Certainty Equivalence Principle is reasonable for systems under low noise levels; however, the same is not necessarily true for systems with large noise intensity where an approach sensitive to noise and the cost of this uncertainty would be more appropriate. This reasoning leads us to favor risk- or noise-sensitive approaches over neutral ones within the proposed architecture for feedback control. Furthermore, risk-sensitive approaches provide robustness guarantees such as a small gain theorem and a stochastic robust stability result [184].

Risk-sensitive methods capture the effects of uncertainty upon controls by invalidating the assumptions of the Certainty Equivalence Principle [185]. They make use of formulations that preserve the computational efficiency, but at the same time make controls dependent on statistical properties of the noise. For instance, [65, 71, 72] proposed to consider multiplicative noise in the process and measurement model parameters to study movement and control of a bio-mechanical arm where muscle models included multiplicative noise. Other different approaches exits [186], but the one taken in this chapter is the use of a non-quadratic performance criteria, able to capture noise effects on higher order statistics of the performance criteria.

Starting from a LQG controller (linear dynamics, quadratic cost and Gaussian additive noise), [70] introduced a method to consider higher order statistics of the cost by using an exponential transformation of the original quadratic performance criteria, and named it Linear-Exponential Gaussian (LEG). A LEG controller explicitly depends on process noise statistics, and is thus similar to LQG at low noise levels, but greatly differs as noise becomes larger. [73] extended the idea to nonlinear systems using an iterative method and illustrated risk-seeking and risk-averse behaviors in a continuous-time car on a cliff problem. Further, [187] presented a unified theory of linearly solvable control including both standard and risk-sensitive stochastic optimal control (SOC) problems for systems under process noise.

Later [74, 75, 76] considered systems under measurement noise and were able to find finite dimensional controllers in two special cases: when the objective depends only on the final state and when there is no process noise. A general solution of the SOC problem with measurement uncertainty was proposed in [77] for continuous time and in [78] for discrete time. The results in these works rely on the novel definition of risk-sensitive estimators (similar in spirit to noise-sensitive controllers), but for which it is not clear how the new mathematical construction of the estimators will affect in practice the control of a humanoid robot. For this reason, the architecture proposed in this chapter favors the formulation presented in [129], where measurement noise-sensitive controllers can be derived without redefining estimators. In this approach, as in [73], the nonlinear problem is sequentially approximated and risk-sensitive controllers are designed for the problem's Taylor approximation. However, to consider the effects of measurement uncertainty, it makes use of an extended dynamical system composed of the control and estimation problems [188, 189, 190]. The proposed approach limits the amount of information available for constructing the control policy to statistics that can be captured in the state estimate (mean and variance), but gains flexibility at defining the performance criteria and can capture simultaneously process and measurement noise.

The architecture for feedback control presented in this chapter is based on the previous problem formulations [23, 129], and presents the following contributions:

- A theoretical extension of recent work on risk-sensitive control [73] to construct feedback control laws that explicitly depend on statistical properties of measurement uncertainty by incorporating a state observer.
- The successful application of the proposed formulation to track and control the execution of kino-dynamic movement plans on a simulated humanoid robot shows that 1) the approach can optimally adapt impedance behaviors to traverse uncertain terrains (favorably suggesting that contact interaction tasks could be better formulated as optimal control problems with measurement uncertainty) and 2) that the proposed architecture for feedback control design with sensitivity to measurement uncertainty is applicable to high-dimensional robotic systems, which is generally not the case for noise-sensitive algorithms.

Section 5.3 presents background material for the synthesis of risk-sensitive feedback controllers for systems under process noise [73] and formulates in a general way the problem of interest including measurement uncertainty. Section 5.4 reminds the algorithm for noise-sensitive feedback control. Finally, in Section 5.5 the proposed architecture for feedback control and stabilization of kino-dynamic motion plans is discussed. Section 5.6 presents experimental results including the stabilization of optimal motions in a physically simulated humanoid robot. Finally, discussion and conclusions of this chapter are presented in Sections 5.7-5.8.

5.3 PROBLEM FORMULATION

This section presents the formulation of the optimal control problem with process and measurement noise considered in this chapter as well as recalls previous results [70, 73] useful to understand the derivation of the proposed algorithm.

The following stochastic differential equations (SDE's) define the dynamical evolution of the state and measurement models of a general nonlinear system:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})dt + \mathbf{g}^{\mathrm{x}}(\mathbf{x}, \mathbf{u})d\omega \tag{21}$$

$$d\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u})dt + \mathbf{g}^{\mathrm{y}}(\mathbf{x}, \mathbf{u})d\gamma \tag{22}$$

Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^p$ be state, control and measured output; $d\omega$, $d\gamma$ zero-mean Brownian motions with covariance Ωdt, Γdt respectively. $\mathbf{f}(\mathbf{x}, \mathbf{u})$ and $\mathbf{h}(\mathbf{x}, \mathbf{u})$ denote the drift coefficients or deterministic components of process and measurement models; and $\mathbf{g}^{\mathrm{x}}(\mathbf{x}, \mathbf{u})$, $\mathbf{g}^{\mathrm{y}}(\mathbf{x}, \mathbf{u})$ denote the diffusion coefficients of process and measurement models that encode the problem's stochasticity.

In optimal control, usually the objective consists in minimizing the performance criteria $\mathcal{J}^{\pi}$, which is a functional of the control policy $\mathbf{u} = \pi(\mathbf{x})$:

$$\mathcal{J}^{\pi}(\mathbf{x}, t) = \Phi_f(\mathbf{x}_{t_f}) + \int_t^{t_f} L(\mathbf{x}_t, \mathbf{u}_t, t)dt \tag{23}$$

where $L(\mathbf{x}_t, \mathbf{u}_t, t)$ denotes the rate at which cost increases and $\Phi_f(\mathbf{x}_{t_f})$ the measure of performance at the final time t_f and state $\mathbf{x}_{t_f}$. In general, the mean of the performance criteria $\mathbb{E}[\mathcal{J}^{\pi}]$ would be minimized; however, this is not enough to analyze the effects of uncertainty upon optimal control policies, as a term is required within the performance measure to capture the cost due to uncertainty. Tools from the risk-sensitive control literature make this possible as they allow to include higher order statistics of the performance criteria $\mathcal{J}^{\pi}$ in the optimization of control policies, e.g. via an exponential transformation of the original performance criteria [70]. The risk-sensitive cost would then be given by the following expression:

$$\mathbf{J} = \min_{\pi} \mathbb{E}\{\exp[\sigma \mathcal{J}^{\pi}]\} \tag{24}$$

$\mathcal{J}^{\pi}$ can be considered as a random variable functional of the control policy $\mathbf{u} = \pi(\mathbf{x})$; $\sigma \in \mathbb{R}$ is the risk-sensitive parameter; $\mathbb{E}$ the expectation operator; and finally $\mathbf{J}$ denotes the risk-sensitive cost that corresponds to the moment generating function, an alternative specification of the probability distribution of the random variable $\mathcal{J}^{\pi}$ [191]. The rest of this section reminds two previous results from [70, 73] useful to the architecture proposed in this chapter: the meaning of the exponential transformation of the performance criteria and the form that the Hamilton-Jacobi-Bellman (HJB) equation takes under this exponential transformation.

Meaning of the Exponential Transformation

It has been shown [73] that the logarithmic transformation of the risk-sensitive cost is proportional to a linear combination of the moments of the objective $\mathcal{J}^{\pi}$

$$\log[\mathbf{J}] \propto \mathbb{E}[\mathcal{J}^{\pi}] + \frac{\sigma}{2}\mu_2[\mathcal{J}^{\pi}] + \frac{\sigma^2}{6}\mu_3[\mathcal{J}^{\pi}] + \cdots \tag{25}$$

where μ_2, μ_3 denote second (variance) and third (skewness) order moments of $\mathcal{J}^{\pi}$, respectively. The parameter σ defines the role of higher order moments either as a penalty or a reward, which give rise to risk-averse or risk-seeking behaviors. Positive values of σ imply that control effort can be increased to narrow confidence intervals. The lower the value of σ, the lower the weight given to higher moments and thus the wider the confidence intervals. For negative σ, strong control effort is avoided in the presence of poor information (high variance), higher order moments even act as a reward leading to risk-seeking solutions.

Hamilton-Jacobi-Bellman (HJB) Equation under the Exponential Transformation

Another important result [73] is the form taken by the Hamilton-Jacobi-Bellman equation under the exponential transformation. Under only the process model dynamics of eq. (21) and objective function given by (23)-(24), it takes the form:

$$-\partial_t \Psi = \min_{u_t} \left\{ \underbrace{L + \nabla_{\mathbf{x}} \Psi^T \mathbf{f} + \frac{1}{2} \mathbf{Tr} \left(\nabla_{\mathbf{xx}} \Psi \mathbf{g}^{\mathrm{x}} \Omega \mathbf{g}^{\mathrm{x}T} \right)}_{\text{Usual HJB equation}} + \underbrace{\frac{\sigma}{2} \nabla_{\mathbf{x}} \Psi^T \mathbf{g}^{\mathrm{x}} \Omega \mathbf{g}^{\mathrm{x}T} \nabla_{\mathbf{x}} \Psi}_{\text{Term due to uncertainty}} \right\} \tag{26}$$

The value function Ψ is a function of state $\mathbf{x}$ and time t, that satisfies eq. (26). It is composed by the usual HJB eq. for a stochastic system with cost rate L due to the current state and control, the free drift and control benefit costs, and the diffusion cost where noise enters the system but without effect upon controls. The new term due to uncertainty captures the effects of noise over the statistical properties of the cost and because it is a function of the state and noise, it makes controls explicitly dependent on the process noise covariance Ω. For $\sigma = 0$, the problem reduces to the minimization of the expected value of the performance criteria $\mathbb{E}[\mathcal{J}]$ (25)-(26). It is worth highlighting once more that these results model only process noise.

Problem Statement

Finally, the problem definition can be stated as follows: Under the state and measurement dynamics (eqns. (21)-(22)), and risk-sensitive cost to be minimized (eqns. (23)-(24)), the goal is to find the optimal control law π^* that minimizes the cost $\mathbf{J}^{\pi}(\mathbf{x}_0, t_0)$ for the stochastic system in the presence of additive process and measurement noise. The globally optimal control law $\pi^*(\mathbf{x}, t)$ does not depend on an initial state. However, finding it is in general intractable. Instead, it suffices to find a locally-optimal solution that approximates the global one in the vicinity of a nominal trajectory $\mathbf{x}_t^n$. Since the nominal trajectory depends on the system's initial state, so does the control law. Note that this formulation differs from the one used in the previously shown results [73], because it includes a measurement model; however, it will be shown that it is still possible to exploit these results under a suitable reformulation.

5.4 algorithmic derivation

This section's goal is to describe how a measurement model is included into the problem in such a way that results on risk-sensitive control with process noise [73, 187] can be used to derive feedback controllers for problems with measurement uncertainty. The main idea is to extend the state dynamics (21) with the dynamics of a state estimator (22) and by means of a forward propagation of measurement uncertainty along a nominal trajectory to precompute optimal estimation gains. This allows then the use of standard techniques to compute backwards in time optimal feedback controllers [68] for the composite system (state and state estimate dynamics). The rest of this section presents a detailed derivation of the continuous time algorithm, as well as its formulation in discrete time for computational purposes.

5.4.1 *Continuous-Time Algorithm*

At each iteration, the algorithm begins with a nominal control sequence $\mathbf{u}_t^n$ and the corresponding zero-noise state trajectory $\mathbf{x}_t^n$, obtained by applying the control sequence to the system through the dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ with initial state $\mathbf{x}(0) = \mathbf{x}_0$. Then, following standard iterative control approaches [68, 69], a linear dynamics approximation and a quadratic cost approximation are composed along the nominal trajectories $\mathbf{x}_t^n$ and $\mathbf{u}_t^n$, in terms of control and state deviations $\delta\mathbf{u}_t = \mathbf{u}_t - \mathbf{u}_t^n$, $\delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}_t^n$. Linearized dynamics and measurement models evaluated along the nominal state and control trajectories $\mathbf{x}_t^n$, $\mathbf{u}_t^n$ then become:

$$d(\delta\mathbf{x}_t) = (\mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t)dt + \mathbf{C}_t d\omega_t \tag{27}$$

$$d(\delta\mathbf{y}_t) = (\mathbf{F}_t\delta\mathbf{x}_t + \mathbf{E}_t\delta\mathbf{u}_t)dt + \mathbf{D}_t d\gamma_t \tag{28}$$

The system's matrices are defined as follows:

$$\begin{aligned} \mathbf{A}_t &= \partial\mathbf{f}(\mathbf{x},\mathbf{u})/\partial\mathbf{x}^T|_{\mathbf{x}_t^n,\mathbf{u}_t^n} & \mathbf{B}_t &= \partial\mathbf{f}(\mathbf{x},\mathbf{u})/\partial\mathbf{u}^T|_{\mathbf{x}_t^n,\mathbf{u}_t^n} & \mathbf{C}_t &= \mathbf{g}^{\mathrm{x}}(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^n,\mathbf{u}_t^n} \\ \mathbf{F}_t &= \partial\mathbf{h}(\mathbf{x},\mathbf{u})/\partial\mathbf{x}^T|_{\mathbf{x}_t^n,\mathbf{u}_t^n} & \mathbf{E}_t &= \partial\mathbf{h}(\mathbf{x},\mathbf{u})/\partial\mathbf{u}^T|_{\mathbf{x}_t^n,\mathbf{u}_t^n} & \mathbf{D}_t &= \mathbf{g}^{\mathrm{y}}(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^n,\mathbf{u}_t^n} \end{aligned}$$

Similarly, a general quadratic approximation of the cost $\mathcal{J}$ can be written as:

$$\tilde{\ell}(\mathbf{x},\mathbf{u},t) = q_t + \mathbf{q}_t^T\delta\mathbf{x}_t + \mathbf{r}_t^T\delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{x}_t^T\mathbf{Q}_t\delta\mathbf{x}_t + \delta\mathbf{x}_t^T\mathbf{P}_t\delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{u}_t^T\mathbf{R}_t\delta\mathbf{u}_t \tag{29}$$

$$\tilde{\ell}_f(\mathbf{x}) = q_f + \mathbf{q}_f^T\delta\mathbf{x}_t + \frac{1}{2}\delta\mathbf{x}_t^T\mathbf{Q}_f\delta\mathbf{x}_t \tag{30}$$

By including the dynamics of a state observer (22) within an extended dynamical system composed of the control and estimation problems, noise of the measurement model is explicitly considered when computing the optimal control law and previous results can be extended, while remaining computationally efficient. The observer could in principle be of any type, but for simplicity an Extended Kalman Filter (EKF) is used in the following derivation. Its dynamics are given by:

$$d(\delta\hat{\mathbf{x}}_t) = (\mathbf{A}_t\delta\hat{\mathbf{x}}_t + \mathbf{B}_t\delta\mathbf{u}_t)dt + \mathbf{K}_t[d(\delta\mathbf{y}_t) - d(\delta\hat{\mathbf{y}}_t)]$$

More compactly, the control-estimation dynamics of the extended dynamical system (composed by state and state estimate dynamics) can be written as:

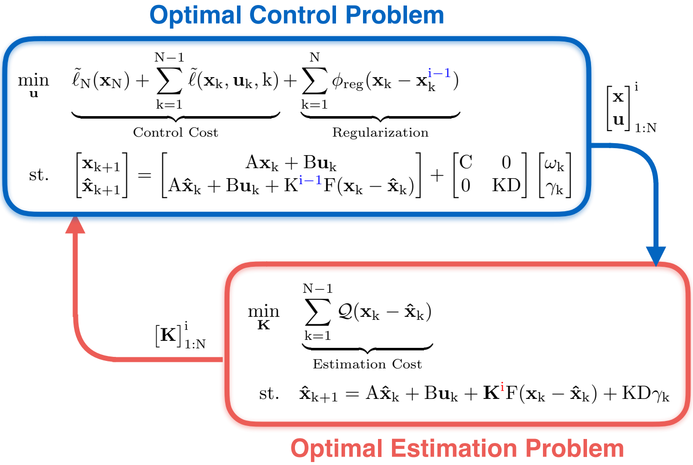

Optimal Control Problem
$\min_{\mathbf{u}} \underbrace{\tilde{\ell}_N(\mathbf{x}_N) + \sum_{k=1}^{N-1} \tilde{\ell}(\mathbf{x}_k, \mathbf{u}_k, k)}_{\text{Control Cost}} + \underbrace{\sum_{k=1}^{N} \phi_{reg}(\mathbf{x}_k - \mathbf{x}_k^{i-1})}_{\text{Regularization}}$
$\text{st.} \begin{bmatrix} \mathbf{x}_{k+1} \\ \hat{\mathbf{x}}_{k+1} \end{bmatrix} = \begin{bmatrix} A\mathbf{x}_k + B\mathbf{u}_k \\ A\hat{\mathbf{x}}_k + B\mathbf{u}_k + K^{i-1}F(\mathbf{x}_k - \hat{\mathbf{x}}_k) \end{bmatrix} + \begin{bmatrix} C & 0 \\ 0 & KD \end{bmatrix} \begin{bmatrix} \omega_k \\ \gamma_k \end{bmatrix}$
$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}_{1:N}^{i}$
$[\mathbf{K}]_{1:N}^{i}$
$\min_{\mathbf{K}} \underbrace{\sum_{k=1}^{N-1} \mathcal{Q}(\mathbf{x}_k - \hat{\mathbf{x}}_k)}_{\text{Estimation Cost}}$
$\text{st.} \quad \hat{\mathbf{x}}_{k+1} = A\hat{\mathbf{x}}_k + B\mathbf{u}_k + \mathbf{K}^i F(\mathbf{x}_k - \hat{\mathbf{x}}_k) + KD\gamma_k$
Optimal Estimation Problem

FIGURE 30: This figure depicts the proposed approach to approximate a solution for the combined control-estimation problem for the discrete case.

$$\underbrace{\begin{bmatrix} d(\delta\mathbf{x}_t) \\ d(\delta\hat{\mathbf{x}}_t) \end{bmatrix}}_{d(\delta\tilde{\mathbf{x}}_t)} = \underbrace{\begin{bmatrix} \mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t \\ \mathbf{A}_t\delta\hat{\mathbf{x}}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{K}_t\mathbf{F}_t(\delta\mathbf{x}_t - \delta\hat{\mathbf{x}}_t) \end{bmatrix}}_{\tilde{\mathbf{f}}(\delta\tilde{\mathbf{x}}_t, \delta\mathbf{u}_t)} dt + \underbrace{\begin{bmatrix} \mathbf{C}_t & 0 \\ 0 & \mathbf{K}_t\mathbf{D}_t \end{bmatrix}}_{\tilde{\mathbf{g}}(t)} \begin{bmatrix} d\omega_t \\ d\gamma_t \end{bmatrix} \qquad (31)$$

where $\delta\hat{\mathbf{x}}_t$ is the estimate of $\delta\mathbf{x}_t$, and $\delta\tilde{\mathbf{x}}_t$ represents the vector $[\delta\mathbf{x}_t, \delta\hat{\mathbf{x}}_t]$. Eq. (31) is bilinear in $\mathbf{K}_t$ and $\delta\tilde{\mathbf{x}}_t$. Below, a detailed derivation is shown. However, the algorithm's main idea is to use this special problem structure to iteratively find a solution. It first forward propagates measurement noise and computes estimation gains $\mathbf{K}_t$ along the nominal trajectory. Then, using fixed estimation gains, it does a usual backward pass to compute feedback controllers [68, 69]. This eases the design of a locally optimal estimator and controller, while still considering the effects of process and measurement noise. As can be easily noticed, $\tilde{\mathbf{f}}(\delta\tilde{\mathbf{x}}_t, \delta\mathbf{u}_t)$ and $\tilde{\mathbf{g}}(\mathbf{t})$ correspond to what in (26) was called $\mathbf{f}$ and $\mathbf{g}^x$ respectively. However, now they include measurement noise by incorporating the dynamics of the state estimator.

Figure 30 graphically depicts a simple sketch on how the proposed approach approximates a local solution for the combined control-estimation problem. Because it is linear in the states $\mathbf{x}, \hat{\mathbf{x}}$, or in the estimation gains $\mathbf{K}$, but not in both due to its nonconvex bilinear nature, an iterative approach is used to exploit this fact by decomposing the combined problem into two linear more easily solvable subproblems, but that are still capable of finding a local solution for the combined problem. The optimal control problem is solved using its cost plus an additional one that regularizes new optimal trajectories towards previous ones (to remain close to the region of validity of the dynamics linearizations and estimation gains). As constraint it makes use of a dynamics model linear in its states given the fact that estimation gains $\mathrm{K}^{\mathrm{i}-1}$, output of the optimal estimation problem, are considered fixed here. The optimal estimation problem as usual tries to minimize the outer product of the error dynamics between the state and state estimate $(\mathbf{x}_\mathrm{k} - \hat{\mathbf{x}}_\mathrm{k})$. The constraints used are the usual definitions of the dynamics models. It would be possible to solve the estimation problem along the entire state trajectory to exploit all the available information [192]; however, this approach approximates its solution by a sequence of minimizations at each timestep that fall back to estimation by means of the well known Extended Kalman Filter. In the optimal estimation problem, its cost acts also as a regularization term towards the solution of the control problem, and thus this common consensus cost supports the algorithm convergence.

5.4.1.1 *Estimator Design*

An Extended Kalman Filter in continuous time is used for simplicity. Note however that other estimators could be used as long as it is possible to extract a sequence of estimation gains from them [193]. In this EKF, the main idea is to find the gain $\mathbf{K}_t$ that minimizes the expected outer product of the error dynamics between state and state-estimate dynamics [123]. Error dynamics are given by:

$$\begin{aligned} d(\delta\mathbf{e}_t) &= d(\delta\mathbf{x}_t) - d(\delta\hat{\mathbf{x}}_t) \\ &= \mathbf{A}_t\delta\mathbf{e}_t dt + \mathbf{C}_t d\omega_t - \mathbf{K}_t[d(\delta\mathbf{y}_t) - d(\delta\hat{\mathbf{y}}_t)] \\ &= (\mathbf{A}_t - \mathbf{K}_t\mathbf{F}_t)\delta\mathbf{e}_t dt + \mathbf{C}_t d\omega_t - \mathbf{K}_t\mathbf{D}_t d\gamma_t \end{aligned}$$

Then, the optimal estimation gains that minimize

$$\begin{aligned} \dot{\Sigma}^e_t =& \mathbb{E}[d(\delta\mathbf{e}_t)d(\delta\mathbf{e}_t)^T] \\ =& (\mathbf{A}_t - \mathbf{K}_t\mathbf{F}_t)\Sigma^e_t + \Sigma^e_t(\mathbf{A}_t - \mathbf{K}_t\mathbf{F}_t)^T \\ &+ \mathbf{K}_t\mathbf{D}_t\Gamma_t\mathbf{D}_t^T\mathbf{K}_t^T + \mathbf{C}_t\Omega_t\mathbf{C}_t^T \end{aligned} \tag{32}$$

are given by

$$\mathbf{K}_t = \Sigma^e_t\mathbf{F}_t^T(\mathbf{D}_t\Gamma_t\mathbf{D}_t^T)^{-1} \tag{33}$$

They are updated at each iteration in a forward pass along the nominal trajectories, and are then fixed for the backward pass. In this way, the combined estimation and control system (31), is linear in $\delta\tilde{\mathbf{x}}_t$ and $\delta\mathbf{u}_t$, which allows to make use of the HJB Eq. (26) to compute a control law π sensitive to both process and measurement noise of the original system. For notational convenience the following shortcuts are used in the rest of this section: $\alpha_t = \mathbf{C}_t\Omega_t\mathbf{C}_t^T$ and $\beta_t = \mathbf{K}_t\mathbf{D}_t\Gamma_t\mathbf{D}_t^T\mathbf{K}_t^T$.

5.4.1.2 *Controller Design*

The locally-optimal control law is an affine function of the state estimate $\delta\mathbf{u}_t = \mathbf{l}_t + \mathbf{L}_t\delta\hat{\mathbf{x}}_t$. The HJB equation for this system that allows to derive control laws is given by (26) (just remember that $\mathbf{f}$ and $\mathbf{g}^x$ correspond now to $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ respectively), the cost by (29)-(30) (remember that the HJB equation under the exponential transformation is used; thus, the cost need not to be exponentiated), and the dynamics by (31). The Ansatz for the value function $\Psi(\delta\tilde{\mathbf{x}}_t, t)$ is a quadratic form

$$\Psi(\delta\tilde{\mathbf{x}}_t, t) = \frac{1}{2}\begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix}^T \begin{bmatrix}\mathbf{S}_t^x & \mathbf{S}_t^{x\hat{x}}\\ \mathbf{S}_t^{\hat{x}x} & \mathbf{S}_t^{\hat{x}}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix} + \begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix}^T\begin{bmatrix}\mathbf{s}_t^x\\ \mathbf{s}_t^{\hat{x}}\end{bmatrix} + s_t$$

and the partial derivatives of the Ansatz Ψ are given by

$$\begin{aligned}
\partial_t\Psi &= \frac{1}{2}\begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix}^T \begin{bmatrix}\dot{\mathbf{S}}_t^x & \dot{\mathbf{S}}_t^{x\hat{x}}\\ \dot{\mathbf{S}}_t^{\hat{x}x} & \dot{\mathbf{S}}_t^{\hat{x}}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix} + \begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix}^T\begin{bmatrix}\dot{\mathbf{s}}_t^x\\ \dot{\mathbf{s}}_t^{\hat{x}}\end{bmatrix} + \dot{s}_t\\
\nabla_{\delta\tilde{\mathbf{x}}}\Psi &= \begin{bmatrix}\mathbf{S}_t^x & \mathbf{S}_t^{x\hat{x}}\\ \mathbf{S}_t^{\hat{x}x} & \mathbf{S}_t^{\hat{x}}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}_t\\ \delta\hat{\mathbf{x}}_t\end{bmatrix} + \begin{bmatrix}\mathbf{s}_t^x\\ \mathbf{s}_t^{\hat{x}}\end{bmatrix}\\
\nabla_{\delta\tilde{\mathbf{x}}\delta\tilde{\mathbf{x}}}\Psi &= \begin{bmatrix}\mathbf{S}_t^x & \mathbf{S}_t^{x\hat{x}}\\ \mathbf{S}_t^{\hat{x}x} & \mathbf{S}_t^{\hat{x}}\end{bmatrix}
\end{aligned}$$

Right super-scripts x and $\hat{x}$ for $\mathbf{S}$ and $\mathbf{s}$ denote that they are sub-blocks that multiply x and $\hat{x}$, respectively. Under the assumed linear dynamics and quadratic cost and value function, the HJB equation can be written as follows: The left-hand side (LHS) corresponds to the time derivative of the value function and is given by

$$-\frac{1}{2}\delta\mathbf{x}_t^T\dot{\mathbf{S}}_t^x\delta\mathbf{x}_t - \frac{1}{2}\delta\hat{\mathbf{x}}_t^T\dot{\mathbf{S}}_t^{\hat{x}}\delta\hat{\mathbf{x}}_t - \delta\mathbf{x}_t^T\dot{\mathbf{S}}_t^{x\hat{x}}\delta\hat{\mathbf{x}}_t - \delta\mathbf{x}_t^T\dot{\mathbf{s}}_t^x - \delta\hat{\mathbf{x}}_t^T\dot{\mathbf{s}}_t^{\hat{x}} - \dot{s}_t$$

and the right-hand-side (RHS) corresponds to the following minimization

$$\begin{aligned}
= \min_{\delta\mathbf{u}_t} \Bigg\{ & q_t + \mathbf{q}_t^T\delta\mathbf{x}_t + \mathbf{r}_t^T\delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{x}_t^T\mathbf{Q}_t\delta\mathbf{x}_t + \delta\mathbf{x}_t^T\mathbf{P}_t\delta\mathbf{u}_t + \frac{1}{2}\delta\mathbf{u}_t^T\mathbf{R}_t\delta\mathbf{u}_t\\
& + (\mathbf{S}_t^x\delta\mathbf{x}_t + \mathbf{S}_t^{x\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^x)^T(\mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\delta\mathbf{u}_t)\\
& + (\mathbf{S}_t^{\hat{x}x}\delta\mathbf{x}_t + \mathbf{S}_t^{\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^{\hat{x}})^T(\mathbf{A}_t\delta\hat{\mathbf{x}}_t + \mathbf{B}_t\delta\mathbf{u}_t + \mathbf{K}_t\mathbf{F}_t(\delta\mathbf{x}_t - \delta\hat{\mathbf{x}}_t))\\
& + \frac{\sigma}{2}(\mathbf{S}_t^x\delta\mathbf{x}_t + \mathbf{S}_t^{x\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^x)^T\alpha_t(\mathbf{S}_t^x\delta\mathbf{x}_t + \mathbf{S}_t^{x\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^x) + \frac{1}{2}\mathrm{Tr}\left(\mathbf{S}_t^x\alpha_t\right)\\
& + \frac{\sigma}{2}(\mathbf{S}_t^{\hat{x}x}\delta\mathbf{x}_t + \mathbf{S}_t^{\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^{\hat{x}})^T\beta_t(\mathbf{S}_t^{\hat{x}x}\delta\mathbf{x}_t + \mathbf{S}_t^{\hat{x}}\delta\hat{\mathbf{x}}_t + \mathbf{s}_t^{\hat{x}}) + \frac{1}{2}\mathrm{Tr}\left(\mathbf{S}_t^{\hat{x}}\beta_t\right) \Bigg\}
\end{aligned}$$

In order to perform the minimization of the right-hand side expression above, its control dependent terms need to be analyzed. They are given by:

$$\begin{aligned} V_{\delta \mathbf{u}_t} =& \frac{1}{2}\delta \mathbf{u}_t^T \underbrace{\mathbf{R}_t}_{\mathbf{H}_t} \delta \mathbf{u}_t + \delta \mathbf{u}_t^T \Big(\underbrace{\mathbf{r}_t + \mathbf{B}_t^T \left(\mathbf{s}_t^x + \mathbf{s}_t^{\hat{x}}\right)}_{\mathbf{g}_t} + \\ & \underbrace{\left(\mathbf{P}_t^T + \mathbf{B}_t^T \left(\mathbf{S}_t^x + \mathbf{S}_t^{\hat{x}x}\right)\right)}_{\mathbf{G}_t^x} \delta \mathbf{x}_t + \underbrace{\mathbf{B}_t^T \left(\mathbf{S}_t^{x\hat{x}} + \mathbf{S}_t^{\hat{x}}\right)}_{\mathbf{G}_t^{\hat{x}}} \delta \hat{\mathbf{x}}_t \Big) \end{aligned}$$

The above expression is quadratic in the controls $\delta \mathbf{u}_t$ and is thus easy to minimize. However, the minimum depends not only on the state estimate $\delta \hat{\mathbf{x}}_t$, but also on the state $\delta \mathbf{x}_t$. The algorithm presented in this chapter assumes however not to have access to full state information, but only to a statistical description of it, given by the state estimate. Therefore, in order to perform the minimization above, the expectation of $V_{\delta \mathbf{u}_t}$ over $\delta \mathbf{x}_t$ conditioned on $\delta \hat{\mathbf{x}}_t$ is taken.

$$\mathbb{E}_{\delta \mathbf{x}_t | \delta \hat{\mathbf{x}}_t} \left[V_{\delta \mathbf{u}_t} \right] = \frac{1}{2}\delta \mathbf{u}_t^T \mathbf{H}_t \delta \mathbf{u}_t + \delta \mathbf{u}_t^T (\mathbf{g}_t + (\mathbf{G}_t^x + \mathbf{G}_t^{\hat{x}}) \delta \hat{\mathbf{x}}_t)$$

This means that the cost of uncertainty due to measurement noise, considers only the effects of mean and variance of the measurement (captured by the EKF) when evaluating noise effects on the statistical properties of the performance criteria. Consequently, the risk-sensitive control law, considers only as cost of measurement uncertainty the one that can be computed by means of the state estimate, i.e. the one that can be extracted from using mean and variance of the state estimate and neglecting higher order terms. From the above expression, the minimizer can be analytically computed. In case of control constraints, a quadratic program can be used to solve for the constrained minimizer [31]. In both cases, the minimizer is an affine functional of the state-estimate. For the unconstrained case, it is given by

$$\delta \mathbf{u}_t = \mathbf{l}_t + \mathbf{L}_t \delta \hat{\mathbf{x}}_t = -\mathbf{H}_t^{-1} \mathbf{g}_t - \mathbf{H}_t^{-1} (\mathbf{G}_t^x + \mathbf{G}_t^{\hat{x}}) \delta \hat{\mathbf{x}}_t \tag{34}$$

$V_{\delta \mathbf{u}_t}$ can then be written in terms of the optimal control as

$$\begin{aligned} V_{\delta \mathbf{u}_t^*} =& \frac{1}{2} \delta \hat{\mathbf{x}}_t^T \left(\mathbf{G}_t^{x\,T} \mathbf{H}_t^{-1} \mathbf{G}_t^x - \mathbf{G}_t^{\hat{x}\,T} \mathbf{H}_t^{-1} \mathbf{G}_t^{\hat{x}} \right) \delta \hat{\mathbf{x}}_t - \\ & \delta \mathbf{x}_t^T \mathbf{G}_t^{x\,T} \mathbf{H}_t^{-1} (\mathbf{G}_t^x + \mathbf{G}_t^{\hat{x}}) \delta \hat{\mathbf{x}}_t - \frac{1}{2} \mathbf{g}_t^T \mathbf{H}_t^{-1} \mathbf{g}_t - \\ & \delta \mathbf{x}_t^T (\mathbf{G}_t^x)^T \mathbf{H}_t^{-1} \mathbf{g}_t - \delta \hat{\mathbf{x}}_t^T \mathbf{G}_t^{\hat{x}\,T} \mathbf{H}_t^{-1} \mathbf{g}_t \end{aligned}$$

The negative coefficients in the terms of $V_{\delta \mathbf{u}_t^*}$ are the benefit of control at reducing the cost. It should be noted that even setting measurement noise to zero does not give a control law equivalent to what was found in [73]. It should be clear from

(34) that mathematically they are not the same. However, it is worth pointing out that, [73] considers neither measurement noise, nor the combined effect of process and measurement noise over optimal controls. In the presence of measurement noise, this control law has more conservative gains than [73], in order to remain compliant enough for the measurement noise level. Writing these terms back into the right-hand side of the HJB, the minimization can be solved and it can be verified that the quadratic Ansatz for the value function remains quadratic and is thus valid. Finally, matching terms in left-hand and right-hand sides of the HJB equation, the backward pass recursion equations become:

$$\begin{aligned}
-\dot{\mathbf{S}}_t^x =& \mathbf{Q}_t + \mathcal{B}(\mathbf{A}_t, \mathbf{S}_t^x) + \mathcal{B}(\mathbf{S}_t^{x\hat{x}}, \tau_t) + \\
& \sigma({\mathbf{S}_t^x}^T \alpha_t \mathbf{S}_t^x + \mathbf{S}_t^{x\hat{x}} \beta_t {\mathbf{S}_t^{x\hat{x}}}^T) \\
-\dot{\mathbf{S}}_t^{\hat{x}} =& \mathcal{B}(\kappa_t, \mathbf{S}_t^{\hat{x}}) + {\mathbf{G}_t^x}^T \mathbf{H}_t^{-1} \mathbf{G}_t^x - {\mathbf{G}_t^{\hat{x}}}^T \mathbf{H}_t^{-1} \mathbf{G}_t^{\hat{x}} + \\
& \sigma({\mathbf{S}_t^{x\hat{x}}}^T \alpha_t \mathbf{S}_t^{x\hat{x}} + {\mathbf{S}_t^{\hat{x}}}^T \beta_t \mathbf{S}_t^{\hat{x}}) \\
-\dot{\mathbf{S}}_t^{x\hat{x}} =& \mathbf{A}_t^T \mathbf{S}_t^{x\hat{x}} + \mathbf{S}_t^{x\hat{x}} \kappa_t + \tau_t^T \mathbf{S}_t^{\hat{x}} - {\mathbf{G}_t^x}^T \mathbf{H}_t^{-1} (\mathbf{G}_t^x + \mathbf{G}_t^{\hat{x}}) + \\
& \sigma({\mathbf{S}_t^x}^T \alpha_t \mathbf{S}_t^{x\hat{x}} + \mathbf{S}_t^{x\hat{x}} \beta_t \mathbf{S}_t^{\hat{x}}) \\
-\dot{\mathbf{s}}_t^x =& \mathbf{q}_t + \mathbf{A}_t^T \mathbf{s}_t^x + \tau_t^T \mathbf{s}_t^{\hat{x}} - {\mathbf{G}_t^x}^T \mathbf{H}_t^{-1} \mathbf{g}_t + \\
& \sigma({\mathbf{S}_t^x}^T \alpha_t \mathbf{s}_t^x + \mathbf{S}_t^{x\hat{x}} \beta_t \mathbf{s}_t^{\hat{x}}) \\
-\dot{\mathbf{s}}_t^{\hat{x}} =& \kappa_t^T \mathbf{s}_t^{\hat{x}} - {\mathbf{G}_t^{\hat{x}}}^T \mathbf{H}_t^{-1} \mathbf{g}_t + \sigma({\mathbf{S}_t^{x\hat{x}}}^T \alpha_t \mathbf{s}_t^x + {\mathbf{S}_t^{\hat{x}}}^T \beta_t \mathbf{s}_t^{\hat{x}}) \\
-\dot{\mathbf{s}}_t =& q_t - \frac{1}{2} \mathbf{g}_t^T \mathbf{H}_t^{-1} \mathbf{g}_t + \frac{1}{2} \mathrm{Tr} \left(\mathbf{S}_t^x \alpha_t + \mathbf{S}_t^{\hat{x}} \beta_t \right) + \\
& \frac{\sigma}{2} \left({\mathbf{s}_t^x}^T \alpha_t \mathbf{s}_t^x + {\mathbf{s}_t^{\hat{x}}}^T \beta_t \mathbf{s}_t^{\hat{x}} \right)
\end{aligned} \tag{35}$$

where for convenience of presentation the following definitions were used: $\mathcal{B}(\mathbf{X}, \mathbf{Y}) = \mathbf{X}^T\mathbf{Y} + \mathbf{Y}^T\mathbf{X}$, $\tau_t = \mathbf{K}_t\mathbf{F}_t$ and $\kappa_t = \mathbf{A}_t - \mathbf{K}_t\mathbf{F}_t$. The integration runs backward in time with $\mathbf{S}_t^x = \mathbf{Q}_f$, $\mathbf{S}_t^{\hat{x}} = 0$, $\mathbf{S}_t^{x\hat{x}} = 0$, $\mathbf{s}_t^x = \mathbf{q}_f$, $\mathbf{s}_t^{\hat{x}} = 0$ and $s_t = q_f$. Despite being long, it is a very simple to implement solution, similar to any other LQR-style recursion.

Remark The effects of process and measurement noise appear in pairs because their Brownian motions were assumed to be uncorrelated (see $\tilde{\mathbf{g}}(t)$ in (31)). However, their combined effect is not just as having higher process noise. Estimation couples their effects, and this can be seen in the recursion eqns. that do not only contain costs for the state and its estimate $\mathbf{S}_t^x$ and $\mathbf{S}_t^{\hat{x}}$, but also the coupling cost $\mathbf{S}_t^{x\hat{x}}$, whose products with the covariances of process noise and estimation errors determine how process and measurement noise affect the value function and therefore the control law. As will be seen in the experimental section, the higher the process noise, the higher the feedback gains, while the opposite holds for measurement noise.

5.4.2 *Discrete-Time Algorithm*

This section will briefly derive a discrete-time version of the algorithm, amenable for computational implementation and control, but that remains similar in spirit to the continuous time version. Note that all the details that can be easily recovered from the continuous case discussion are skipped. The reading of this section can be omitted safely without hindering the understanding of the rest of the chapter, since the derivation to come synthesizes the same algorithm only in discrete form.

The algorithm begins with a nominal control $\mathbf{u}_k^n$ and state sequence $\mathbf{x}_k^n$. It also requires to obtain a discrete linear approximation of the dynamics and a quadratic approximation of the cost along the nominal trajectories $\mathbf{u}_k^n$, $\mathbf{x}_k^n$ in terms of state and control deviations $\delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_k^n$, $\delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_k^n$, which is given by:

$$\delta\mathbf{x}_{k+1} = \mathbf{A}_k\delta\mathbf{x}_k + \mathbf{B}_k\delta\mathbf{u}_k + \mathbf{C}_k\omega_k \tag{36}$$

$$\delta\mathbf{y}_{k+1} = \mathbf{F}_k\delta\mathbf{x}_k + \mathbf{E}_k\delta\mathbf{u}_k + \mathbf{D}_k\gamma_k \tag{37}$$

where $\omega_k \sim \mathcal{N}(0, \Omega_k)$ and $\gamma_k \sim \mathcal{N}(0, \Gamma_k)$ are zero-mean Gaussian variables. The matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$, $\mathbf{E}_k$, $\mathbf{F}_k$, discretized along the nominal trajectories, are:

$$\mathbf{A}_k = I + \Delta t\partial\mathbf{f}/\partial\mathbf{x},\ \mathbf{B}_k = \Delta t\partial\mathbf{f}/\partial\mathbf{u},\ \mathbf{C}_k = \sqrt{\Delta t}\mathbf{g}^x(\mathbf{x}, \mathbf{u})$$

$$\mathbf{F}_k = \Delta t\partial\mathbf{h}/\partial\mathbf{x},\ \mathbf{E}_k = \Delta t\partial\mathbf{h}/\partial\mathbf{u},\ \mathbf{D}_k = \sqrt{\Delta t}\mathbf{g}^y(\mathbf{x}, \mathbf{u})$$

The quadratic approximation of $\mathcal{J}$ is given by

$$\tilde{\ell}(\mathbf{x}, \mathbf{u}, k) = q_k + \mathbf{q}_k^T\delta\mathbf{x}_k + \mathbf{r}_k^T\delta\mathbf{u}_k + \frac{1}{2}\delta\mathbf{x}_k^T\mathbf{Q}_k\delta\mathbf{x}_k + \delta\mathbf{x}_k^T\mathbf{P}_k\delta\mathbf{u}_k + \frac{1}{2}\delta\mathbf{u}_k^T\mathbf{R}_k\delta\mathbf{u}_k \tag{38}$$

$$\tilde{\ell}_N(\mathbf{x}) = q_N + \mathbf{q}_N^T\delta\mathbf{x}_k + \frac{1}{2}\delta\mathbf{x}_k^T\mathbf{Q}_N\delta\mathbf{x}_k \tag{39}$$

where $q_k = q_t\Delta t$, $\mathbf{q}_k = \mathbf{q}_t\Delta t$, $\mathbf{r}_k = \mathbf{r}_t\Delta t$, $\mathbf{Q}_k = \mathbf{Q}_t\Delta t$, $\mathbf{P}_k = \mathbf{P}_t\Delta t$ and $\mathbf{R}_k = \mathbf{R}_t\Delta t$. As in the continuous time case, a state estimator is introduced using a discrete-time Extended Kalman Filter, whose equation is given by

$$\delta\hat{\mathbf{x}}_{k+1} = \mathbf{A}_k\delta\hat{\mathbf{x}}_k + \mathbf{B}_k\delta\mathbf{u}_k + \mathbf{K}_k\mathbf{F}_k(\delta\mathbf{x}_k - \delta\hat{\mathbf{x}}_k) + \mathbf{K}_k\mathbf{D}_k\gamma_k$$

Combining the estimator and the state dynamics, an extended system is formed by the composition of the control and estimation dynamics, that takes the form:

$$\underbrace{\begin{bmatrix}\delta\mathbf{x}_{k+1}\\ \delta\hat{\mathbf{x}}_{k+1}\end{bmatrix}}_{\delta\tilde{\mathbf{x}}_{k+1}} = \underbrace{\begin{bmatrix}\mathbf{A}_k\delta\mathbf{x}_k + \mathbf{B}_k\delta\mathbf{u}_k\\ \mathbf{A}_k\delta\hat{\mathbf{x}}_k + \mathbf{B}_k\delta\mathbf{u}_k + \mathbf{K}_k\mathbf{F}_k(\delta\mathbf{x}_k - \delta\hat{\mathbf{x}}_k)\end{bmatrix}}_{\tilde{\mathbf{f}}(\delta\tilde{\mathbf{x}}_k, \delta\mathbf{u}_k)} + \underbrace{\begin{bmatrix}\mathbf{C}_k & 0\\ 0 & \mathbf{K}_k\mathbf{D}_k\end{bmatrix}}_{\tilde{\mathbf{g}}(k)}\begin{bmatrix}\omega_k\\ \gamma_k\end{bmatrix}$$

where $\delta\hat{\mathbf{x}}_k$ is the estimate of the state $\delta\mathbf{x}_k$, and $\delta\tilde{\mathbf{x}}_k$ the composite vector $[\delta\mathbf{x}_k, \delta\hat{\mathbf{x}}_k]^T$.

5.4.2.1 *Estimator design*

An EKF that minimizes the expected outer-product of the error dynamics is used. The optimal estimation gains that minimize

$$\Sigma^e_{k+1} = (\mathbf{A}_k - \mathbf{K}_k\mathbf{F}_k)\Sigma^e_k(\mathbf{A}_k - \mathbf{K}_k\mathbf{F}_k)^T + \mathbf{K}_k\mathbf{D}_k\Gamma_k\mathbf{D}_k^T\mathbf{K}_k^T + \mathbf{C}_k\Omega_k\mathbf{C}_k^T \tag{40}$$

are given by

$$\mathbf{K}_k = \mathbf{A}_k\Sigma^e_k\mathbf{F}_t^T(\mathbf{F}_t\Sigma^e_k\mathbf{F}_t^T + \mathbf{D}_t\Gamma_t\mathbf{D}_t^T)^{-1} \tag{41}$$

They are updated at each iteration in a forward pass along the nominal trajectories, and then fixed during the backward pass to compute feedback control policies.

5.4.2.2 *Controller design*

This subsection presents the derivation of the backward pass recursion equations to compute the optimal feedback control law. The locally-optimal control law is affine, of the form $\delta\mathbf{u}_k = \mathbf{l}_k + \mathbf{L}_k\delta\hat{\mathbf{x}}_k$. The Ansatz for the value function is still quadratic and the recursion equation is given by the following expression

$$\Psi_\sigma(\delta\tilde{\mathbf{x}}_k, k) = \min_{\mathbf{u}}\left\{\tilde{\ell}(\mathbf{x}, \mathbf{u}, k) + \mathbb{E}[\Psi_\sigma(\delta\tilde{\mathbf{x}}_{k+1}, k+1)]\right\}$$

The subscript σ is a reminder that, although this eq. is similar to the usual Bellman eq., noise propagation is different. Here, besides the usual diffusion cost (noise effects on the mean), the cost of uncertainty given by the HJB Eq. (26) in the σ-dependent term is included. Using the linear dynamics of the extended system, the quadratic cost eqns. (38)-(39), and the quadratic Ansatz, it is easy, but long to write the Bellman eq., which is why we omit it here. It is worth pointing out, that the gradients of the Ansatz for evaluating how noise propagates, should not be evaluated at $\delta\tilde{\mathbf{x}}_k$, but at $\delta\tilde{\mathbf{x}}_{k+1}$. This allows to capture noise effects in the control terms, and derive a risk- sensitive control law. The control law has the form

$$\delta\mathbf{u}_k = \mathbf{l}_k + \mathbf{L}_k\delta\hat{\mathbf{x}}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k - \mathbf{H}_k^{-1}(\mathbf{G}_k^x + \mathbf{G}_k^{\hat{x}})\delta\hat{\mathbf{x}}_k \tag{42}$$

where $\mathbf{H}_k$, $\mathbf{g}_k$, $\mathbf{G}_k^x$ and $\mathbf{G}_k^{\hat{x}}$ are given by

$$\begin{aligned}
\mathbf{H}_k =& \mathbf{R}_k + \mathbf{B}_k^T\left(\mathbf{S}_k^x + \mathbf{S}_k^{\hat{x}} + \mathbf{S}_k^{x\hat{x}} + \mathbf{S}_k^{\hat{x}x}\right)\mathbf{B}_k + \\
& \sigma\mathbf{B}_k^T\left(\mathbf{S}_k^x + \mathbf{S}_k^{x\hat{x}}\right)^T\alpha_k\left(\mathbf{S}_k^x + \mathbf{S}_k^{x\hat{x}}\right)\mathbf{B}_k + \\
& \sigma\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}x} + \mathbf{S}_k^{\hat{x}}\right)^T\beta_k\left(\mathbf{S}_k^{\hat{x}x} + \mathbf{S}_k^{\hat{x}}\right)\mathbf{B}_k \\
\mathbf{g}_k =& \mathbf{r}_k + \mathbf{B}_k^T\left(\mathbf{s}_k^x + \mathbf{s}_k^{\hat{x}}\right) + \sigma\mathbf{B}_k^T\left(\mathbf{S}_k^x + \mathbf{S}_k^{x\hat{x}}\right)^T\alpha_k\mathbf{s}_k^x +
\end{aligned}$$

$$\begin{aligned}
&\sigma\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}x}+\mathbf{S}_k^{\hat{x}}\right)^T\beta_k\mathbf{s}_k^{\hat{x}}\\
\mathbf{G}_k^x =&\mathbf{P}_k^T+\mathbf{B}_k^T\left(\mathbf{S}_k^x+\mathbf{S}_k^{\hat{x}x}\right)\mathbf{A}_k+\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}}+\mathbf{S}_k^{x\hat{x}}\right)\mathbf{K}_k\mathbf{F}_k+\\
&\sigma\mathbf{B}_k^T\left(\mathbf{S}_k^x+\mathbf{S}_k^{x\hat{x}}\right)^T\alpha_k\left(\mathbf{S}_k^x\mathbf{A}_k+\mathbf{S}_k^{x\hat{x}}\mathbf{K}_k\mathbf{F}_k\right)+\\
&\sigma\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}x}+\mathbf{S}_k^{\hat{x}}\right)^T\beta_k\left(\mathbf{S}_k^{\hat{x}x}\mathbf{A}_k+\mathbf{S}_k^{\hat{x}}\mathbf{K}_k\mathbf{F}_k\right)\\
\mathbf{G}_k^{\hat{x}} =&\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}}+\mathbf{S}_k^{x\hat{x}}\right)\left(\mathbf{A}_k-\mathbf{K}_k\mathbf{F}_k\right)+\\
&\sigma\mathbf{B}_k^T\left(\mathbf{S}_k^x+\mathbf{S}_k^{x\hat{x}}\right)^T\alpha_k\mathbf{S}_k^{x\hat{x}}\left(\mathbf{A}_k-\mathbf{K}_k\mathbf{F}_k\right)+\\
&\sigma\mathbf{B}_k^T\left(\mathbf{S}_k^{\hat{x}x}+\mathbf{S}_k^{\hat{x}}\right)^T\beta_k\mathbf{S}_k^{\hat{x}}\left(\mathbf{A}_k-\mathbf{K}_k\mathbf{F}_k\right)
\end{aligned}$$

For notational convenience, the following shortcuts were defined: $\alpha_k = \mathbf{C}_k\Omega_k\mathbf{C}_k^T$, $\beta_k = \mathbf{K}_k\mathbf{D}_k\Gamma_k\mathbf{D}_k^T\mathbf{K}_k^T$, $\tau_k = \mathbf{K}_k\mathbf{F}_k$, $\kappa_k = \mathbf{A}_k - \mathbf{K}_k\mathbf{F}_k$, $\|\mathbf{X}\|_{\mathbf{Y}} = \mathbf{X}^T\mathbf{Y}\mathbf{X}$, $\rho_k = \mathbf{S}_{k+1}^{x\hat{x}}\mathbf{A}_k + \mathbf{S}_{k+1}^{\hat{x}}\tau_k$, $\eta_k = \mathbf{S}_{k+1}^{x}\mathbf{A}_k + \mathbf{S}_{k+1}^{x\hat{x}}\tau_k$. Finally, by inserting the optimal control in the RHS of the Bellman eq., and grouping together terms with similar coefficients of $\delta\mathbf{x}_k$ and $\delta\hat{\mathbf{x}}_k$, the following set of backward recursion eqns. can be obtained:

$$\begin{aligned}
\mathbf{S}_k^x =&\mathbf{Q}_k+\|\mathbf{A}_k\|_{\mathbf{S}_{k+1}^x}+(\tau_k^T\mathbf{S}_{k+1}^{\hat{x}}+2\mathbf{A}_k^T\mathbf{S}_{k+1}^{x\hat{x}})\tau_k+\\
&\sigma\left(\|\eta_k\|_{\alpha_k}+\|\rho_k\|_{\beta_k}\right)\\
\mathbf{S}_k^{\hat{x}} =&\|\kappa_k\|_{\mathbf{S}_{k+1}^{\hat{x}}}+\mathbf{L}_k^T\mathbf{H}_k\mathbf{L}_k+2\mathbf{G}_k^{\hat{x}}\mathbf{L}_k+\\
&\sigma\left(\left\|\mathbf{S}_{k+1}^{x\hat{x}}\kappa_k\right\|_{\alpha_k}+\left\|\mathbf{S}_{k+1}^{\hat{x}}\kappa_k\right\|_{\beta_k}\right)\\
\mathbf{S}_k^{x\hat{x}} =&\rho_k^T\kappa_k+\mathbf{G}_k^x\mathbf{L}_k+\\
&\sigma\left(\eta_k^T\alpha_k\mathbf{S}_{k+1}^{x\hat{x}}\kappa_k+\rho_k^T\beta_k\mathbf{S}_{k+1}^{\hat{x}}\kappa_k\right)\\
\mathbf{s}_k^x =&\mathbf{q}_k+\mathbf{A}_k^T\mathbf{s}_{k+1}^x+\tau_k^T\mathbf{s}_{k+1}^{\hat{x}}+\mathbf{G}_k^x\mathbf{l}_k+\\
&\sigma\left(\eta_k^T\alpha_k\mathbf{s}_{k+1}^x+\rho_k^T\beta_k\mathbf{s}_{k+1}^{\hat{x}}\right)\\
\mathbf{s}_k^{\hat{x}} =&\kappa_k^T\mathbf{s}_{k+1}^{\hat{x}}+\mathbf{L}_k^T\mathbf{H}_k^T\mathbf{l}_k+\mathbf{L}_k^T\mathbf{g}_k+\mathbf{G}_k^{\hat{x}}\mathbf{l}_k+\\
&\sigma\left((\mathbf{S}_{k+1}^{x\hat{x}}\kappa_k)^T\alpha_k\mathbf{s}_{k+1}^x+(\mathbf{S}_{k+1}^{\hat{x}}\kappa_k)^T\beta_k\mathbf{s}_{k+1}^{\hat{x}}\right)\\
\mathbf{s}_k =&\mathbf{s}_{k+1}+q_k+\frac{1}{2}\mathbf{l}_k^T\mathbf{H}_k\mathbf{l}_k+\mathbf{l}_k^T\mathbf{g}_k+\frac{1}{2}\mathrm{Tr}(\mathbf{S}_{k+1}^x\alpha_k+\\
&\mathbf{S}_{k+1}^{\hat{x}}\beta_k)+\frac{\sigma}{2}\left(\left\|\mathbf{s}_{k+1}^x\right\|_{\alpha_k}+\left\|\mathbf{s}_{k+1}^{\hat{x}}\right\|_{\beta_k}\right)
\end{aligned} \tag{43}$$

where the recursion runs backward in time from $\mathbf{S}_N^x = \mathbf{Q}_N$, $\mathbf{S}_N^{\hat{x}} = 0$, $\mathbf{S}_N^{x\hat{x}} = 0$, $\mathbf{s}_N^x = \mathbf{q}_N$, $\mathbf{s}_N^{\hat{x}} = 0$ and $s_N = q_N$.

5.4.3 *Implementation details and algorithm summary*

Algorithm 2 Risk-Sensitive Nonlinear Control

```
Given:
- Risk sensitivity parameter $\sigma$
- System dynamics (21), measurement model (22), and cost function (23)
Initialization:
- Start with a stable control law $\pi(t, \mathbf{x})$
repeat
    - Forward integrate or propagate the system dynamics to
      compute a nominal trajectory: $\mathbf{x}^n_{1:T}$, $\mathbf{u}^n_{1:T}$
    - Linear approximation of the dynamics. Eqs. (27)-(28) or (36)-(37)
    - Quadratic approximation of the cost. Eqs. (29)-(30) or (38)-(39)
    - Forward pass for estimation gains. Eqs. (32)- (33) or (40), (41)
    - Backward pass with regularization parameter $\lambda$. Eqs. (35) or (43)
    - Update control law with line search parameter $\alpha$,
      $\pi(t, \mathbf{x}) = \mathbf{u}^n(t) + \alpha \mathbf{l}(t) + \mathbf{L}(t)(\mathbf{x}(t) - \mathbf{x}^n(t))$
until convergence or a termination condition is satisfied
```

Algorithm 2 summarizes the procedure to compute locally-optimal feedback control policies sensitive to process and measurement noise. It is important to mention two implementation details used for the discrete-time case, as presented in [39]. First of all, when computing the optimal control in Eq. (42), $\mathbf{H}_k$ needs to be inverted. When it is positive-definite, the unique minimizer can be readily found. If it is not, there exist control directions which would allow to make the cost arbitrarily small. This is obviously not true for the nonlinear system, it appears because of the approximations (compare for example $\mathbf{H}_k$ with $\mathbf{H}_t$). To control this, a regularization term $\lambda\mathbf{I}$ is used to make the sum $\mathbf{H}_k + \lambda\mathbf{I}$ positive-definite. This brings an additional advantage: when the sensitivity parameter takes a large value so as to make the matrix $\mathbf{H}_k$ be negative-definite, by including the regularization term, the feedback gains being too loose does not happen anymore. An outer-loop regulates λ, similar to [39]. The second detail, is that when propagating the dynamics with the updated control sequence, the new state trajectory might diverge. By adding a line search with parameter $\alpha \in [0, 1]$, a control sequence that generates a reduction in the cost can still be found and progress in the optimization can be made.

5.5 FEEDBACK CONTROL ARCHITECTURE FOR KINO-DYNAMIC MOVEMENT PLANS

In the previous section, a method for synthesis of noise-sensitive feedback controllers was derived (as summarized in algorithm 2), that requires the definition of a state dynamics and measurement models ($\mathbf{f}(\mathbf{x},\mathbf{u})$ and $\mathbf{h}(\mathbf{x},\mathbf{u})$ respectively as defined in (21)-(22)), as well as of the desired cost to be minimized. This section will describe how the previously derived algorithm can be applied to stabilize whole-body movement plans on a simulated humanoid robot and will thus define the appropriate models and objective functions required by the algorithm. As explained in the introductory background section 1.6, the algorithm described in chapters 3-4 for whole-body motion optimization is based on exploiting the kino-dynamic separability of the centroidal momentum dynamics model, as shown in Figure 1. In the same spirit, the feedback control architecture for tracking of kino-dynamic movement plans, proposed in this chapter, makes use of this decomposition to synthesize independently impedance behaviors for tracking of the desired whole-body motion (kinematic plan) and momentum trajectory (dynamic plan). These optimally designed closed loop behaviors will finally serve as input to an inverse dynamics controller, that while respecting all physical constraints, will compute the torques required to generate in the robot a behavior as close as possible to the desired ones.

In the rest of this section, the design of kino-dynamic feedback controllers and its use within an inverse dynamics controller will be presented.

5.5.1 *Optimal Impedance Behavior for Centroidal Momentum Plans*

The dynamics side of the kino-dynamic trajectory optimization approach is in charge of finding dynamically feasible endeffector contact wrenches (forces and torques) as well as contact locations that support the generation of a desired dynamic motion while satisfying all physical dynamics constraints. Thus, the goal of the impedance controller for a dynamic plan is to stabilize the robot state trajectories $\mathbf{h}$ around the desired ones $\mathbf{h}^{\text{des}}$ using the available controls. In other words, the robot state trajectories to be stabilized using the available controls $\mathbf{u}_{\text{t}}$, namely contact forces $\mathbf{f}_{\text{e}}$ and contact torques γ_{e} (with respect to the frame at the endeffector pose), include the robot's center of mass $\mathbf{r}$ and centroidal momenta $\mathbf{l},\mathbf{k}$. Figure 31 shows the structure used to synthesize a dynamics feedback controller.

The most important components include:

- The performance criteria $\mathcal{J}_{\text{dyn}}$ composed by two terms: $\phi^{\text{dyn}}_{\text{track}}$ used to reward good tracking of the desired center of mass and momentum trajectories, and ϕ^{dyn}_{t} used to regularize controls and include user-defined rewards.

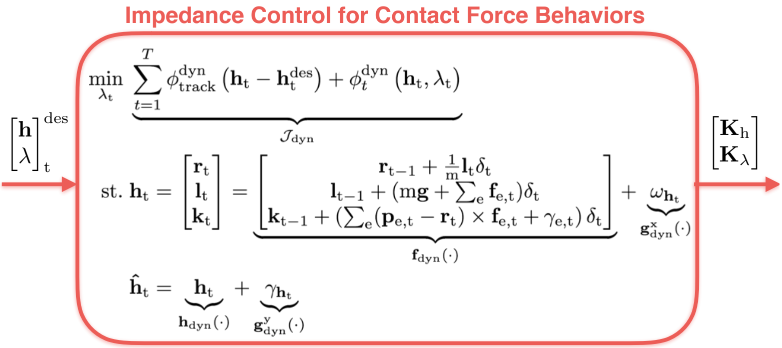

Impedance Control for Contact Force Behaviors
$\begin{bmatrix}\mathbf{h}\\ \lambda\end{bmatrix}_t^{\text{des}}$
$\underbrace{\min_{\lambda_t} \sum_{t=1}^{T} \phi^{\text{dyn}}_{\text{track}}\left(\mathbf{h}_t - \mathbf{h}^{\text{des}}_t\right) + \phi^{\text{dyn}}_t\left(\mathbf{h}_t, \lambda_t\right)}_{\mathcal{J}_{\text{dyn}}}$
$\text{st. } \mathbf{h}_t = \begin{bmatrix}\mathbf{r}_t\\ \mathbf{l}_t\\ \mathbf{k}_t\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{r}_{t-1} + \frac{1}{m}\mathbf{l}_t\delta_t\\ \mathbf{l}_{t-1} + (m\mathbf{g} + \sum_e \mathbf{f}_{e,t})\delta_t\\ \mathbf{k}_{t-1} + (\sum_e(\mathbf{p}_{e,t} - \mathbf{r}_t)\times\mathbf{f}_{e,t} + \gamma_{e,t})\,\delta_t\end{bmatrix}}_{\mathbf{f}_{\text{dyn}}(\cdot)} + \underbrace{\omega_{\mathbf{h}_t}}_{\mathbf{g}^x_{\text{dyn}}(\cdot)}$
$\hat{\mathbf{h}}_t = \underbrace{\mathbf{h}_t}_{\mathbf{h}_{\text{dyn}}(\cdot)} + \underbrace{\gamma_{\mathbf{h}_t}}_{\mathbf{g}^y_{\text{dyn}}(\cdot)}$
$\begin{bmatrix}\mathbf{K}_h\\ \mathbf{K}_\lambda\end{bmatrix}$

FIGURE 31: This figure shows the inputs (desired momentum and contact force trajectories), outputs (momentum and force impedances) and the definition of process and measurement models as well as of the objective used to optimize an impedance for tracking of momentum and contact force behaviors.

- The process model in this system defined by $\mathbf{f}_{\text{dyn}}$ that maps previous state $\mathbf{h}_{t-1}$ and control inputs $\mathbf{u} = \left[\dots\ \mathbf{f}_e{}^T\ \gamma_e^T\ \dots\right]^T$ to output state $\mathbf{h}_t$.

- The measurement model $\mathbf{h}_{\text{dyn}}$ that for simplicity, but without loss of generality, is assumed to consist of a noisy measurement of each state.

- The zero-mean noise parameters $\omega_{\mathbf{h}_t}$ and $\gamma_{\mathbf{h}_t}$ that encode the stochasticity within the process and measurement models respectively.

- The inputs to this system are the desired centroidal dynamics $\mathbf{h}^{\text{des}}_t$ and contact force λ^{des}_t trajectories generated by the kino-dynamic trajectory optimizer, and the outputs are a set of impedance policies $\mathbf{K}_{\mathbf{h}}, \mathbf{K}_\lambda$ for closed loop tracking of the desired dynamic behaviors.

The force gain $\mathbf{K}_\lambda$ is the control policy produced by the noise-sensitive algorithm mapping tracking errors in the state to corrections in the control actions (contact forces λ_t). The momentum gain $\mathbf{K}_{\mathbf{h}}$ is obtained by pre-multiplying the force gain $\mathbf{K}_\lambda$ by the control transition matrix of the linearized dynamics, and maps tracking errors in the state to corrections in the state (center of mass and momentum). Thus, together $\mathbf{K}_{\mathbf{h}}$ and $\mathbf{K}_\lambda$ allow to update nominal momentum trajectories and reference contact wrenches according to the current errors in state tracking, behaviors which serve then as input to an inverse dynamics controller that attempts to generate the torques to follow them as close as physically possible.

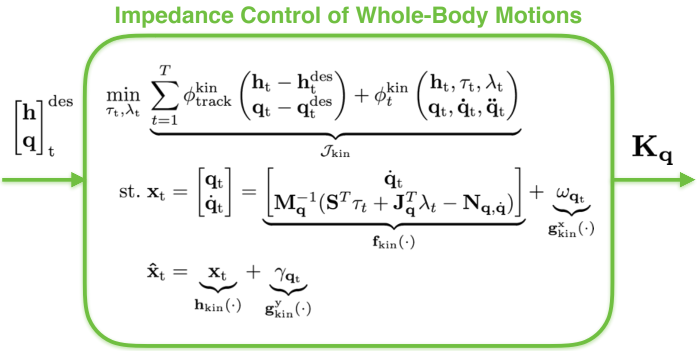

Impedance Control of Whole-Body Motions
$\begin{bmatrix}\mathbf{h}\\ \mathbf{q}\end{bmatrix}_{t}^{\text{des}}$
$\min_{\tau_t,\lambda_t} \underbrace{\sum_{t=1}^{T} \phi_{\text{track}}^{\text{kin}}\begin{pmatrix}\mathbf{h}_t - \mathbf{h}_t^{\text{des}}\\ \mathbf{q}_t - \mathbf{q}_t^{\text{des}}\end{pmatrix} + \phi_t^{\text{kin}}\begin{pmatrix}\mathbf{h}_t, \tau_t, \lambda_t\\ \mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t\end{pmatrix}}_{\mathcal{J}_{\text{kin}}}$
$\text{st. } \mathbf{x}_t = \begin{bmatrix}\mathbf{q}_t\\ \dot{\mathbf{q}}_t\end{bmatrix} = \underbrace{\begin{bmatrix}\dot{\mathbf{q}}_t\\ \mathbf{M}_{\mathbf{q}}^{-1}(\mathbf{S}^T\tau_t + \mathbf{J}_{\mathbf{q}}^T\lambda_t - \mathbf{N}_{\mathbf{q},\dot{\mathbf{q}}})\end{bmatrix}}_{\mathbf{f}_{\text{kin}}(\cdot)} + \underbrace{\omega_{\mathbf{q}_t}}_{\mathbf{g}_{\text{kin}}^{\text{x}}(\cdot)}$
$\hat{\mathbf{x}}_t = \underbrace{\mathbf{x}_t}_{\mathbf{h}_{\text{kin}}(\cdot)} + \underbrace{\gamma_{\mathbf{q}_t}}_{\mathbf{g}_{\text{kin}}^{\text{y}}(\cdot)}$
$\mathbf{K}_{\mathbf{q}}$

FIGURE 32: This figure shows the inputs (desired momentum and whole-body motion trajectories), output (whole-body gain) and the definition of process and measurement models as well as of the objective used to optimize an impedance for tracking of whole-body motion behaviors.

5.5.2 *Optimal Impedance Behavior for Whole-Body Motion Plans*

The kinematics side of the kino-dynamic trajectory optimization is in charge of finding a kinematically feasible sequence of robot postures $\mathbf{q}$, velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ within the robot limits that resemble the dynamically optimized center of mass motion and centroidal momentum. Thus, the goal of the impedance controller for a kinematic plan is to stabilize the robot state trajectories $\mathbf{q}, \dot{\mathbf{q}}$ around the desired ones $\mathbf{q}^{\text{des}}, \dot{\mathbf{q}}^{\text{des}}$ using the available controls, namely contact forces λ and joint torques τ_{j}, such that the motion-induced momentum $\mathbf{h} = \mathbf{A}(\mathbf{q})\dot{\mathbf{q}}$ follows the desired dynamics one $\mathbf{h}^{\text{des}}$. Consequently, to synthesize a kinematics feedback controller, the structure defined in Figure 32 is used.

Its most important components include:

- The performance criteria $\mathcal{J}_{\text{kin}}^{\pi}$ composed as in the previous case by two components, namely: $\phi_{\text{track}}^{\text{kin}}$ used to reward good tracking performance of desired momentum and kinematic state trajectories; and ϕ_t^{kin} used to regularize controls and include user-defined rewards.

- The process model corresponds to the equations of motion of a floating-base rigid-body system. The state dynamics are defined by $\mathbf{f}_{\text{kin}}$ that maps previous state $\mathbf{x}_{\text{t}-1}$ and control inputs $\mathbf{u} = \begin{bmatrix}\lambda_{\text{t}}^T & \tau_{\text{t}}^T\end{bmatrix}^T$ to the output state $\mathbf{x}_{\text{t}}$.

- The measurement model $\mathbf{h}_{\text{kin}}$ is for simplicity, but without loss of generality, assumed to consist of a noisy measurement of each state variable in $\mathbf{x}_t$.

- The zero-mean noise parameters $\omega_{\mathbf{h}_t}$ and $\gamma_{\mathbf{h}_t}$ that encode the stochasticity within the process and measurement models respectively.

- The inputs to this system are the desired centroidal dynamics $\mathbf{h}_t^{\text{des}}$ and whole-body motion trajectories $\mathbf{q}_t^{\text{des}}$ generated by the kino-dynamic trajectory optimizer, and the outputs are a set of impedance policies $\mathbf{K}_\mathbf{q}$ for closed loop tracking of the desired motion behavior.

It is worth highlighting that the objective function for tracking the kinematically optimized motion plans is defined for simplicity in joint space (stabilization of robot postures and velocities $\mathbf{q}, \dot{\mathbf{q}}$), but it could also be defined in cartesian space (in terms of tracking of endeffectors' cartesian trajectories $\mathbf{p}_{e,t} - \mathbf{p}_{e,t}^{\text{des}}$) as the endeffectors' locations can be easily computed applying a forward kinematics algorithm to the current joint configuration $\mathbf{p}_{e,t} = \text{ForwKin}(\mathbf{q}_t)$.

Note also that in this problem the equations of motion are not being used to optimize a motion from scratch, but instead only to find a locally optimal impedance controller around the trajectories provided by the kino-dynamic motion planner. To synthesize a feedback controller for a whole-body motion plan torque slack variables are used to have full control over the degrees of freedom of the floating-base; however, they are heavily penalized so as to guarantee that their value is negligible. In practice they make it much easier to perform a couple of iterations of the algorithm to synthesize a feedback controller. Alternatively a simplified kinematics model such as the one used for building the kinematic plan in chapter 4 could be used instead of the equations of motion to simplify the computation of derivatives, while still allowing to synthesize optimal control policies capable of stabilizing a whole-body motion behavior.

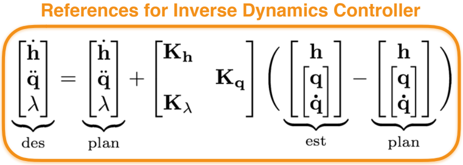
The noise-sensitive algorithm generates a torque gain $\mathbf{K}_\tau$ as the control policy. It maps tracking errors in the state to corrections in the control actions (contact forces λ_t and joint torques τ_t). The whole-body gain $\mathbf{K}_\mathbf{q}$ can be obtained by pre-multiplying the torque gain $\mathbf{K}_\tau$ by the control transition matrix of the linearized system dynamics, and it maps tracking errors in the state to corrections in the state (whole-body motion trajectories). Notice that only the whole-body gain $\mathbf{K}_\mathbf{q}$ is used to update the nominal motion plan as the inverse dynamics controller takes care of resolving the torques that track as good as physically possible the hierarchical task.

5.5.3 *Inverse Dynamics Controller*

The hierarchical inverse dynamics controller [92] is used to compute the torques to realize the desired kino-dynamic behaviors. This controller computes at each timestep the torques required to track as close as possible the desired references

Rank	Task/Constraint
1	Friction constraint
	Center of Pressure constraint
	Stance endeffector acceleration task
2	Force control task
	Momentum rate control task
	Base orientation control task
	Joint acceleration control task
	Swing endeffector acceleration task

TABLE 1: Definition of tasks and ranks for a hierarchical inverse dynamics controller.



$$\underbrace{\begin{bmatrix}\dot{\mathbf{h}}\\ \ddot{\mathbf{q}}\\ \lambda\end{bmatrix}}_{\text{des}} = \underbrace{\begin{bmatrix}\dot{\mathbf{h}}\\ \ddot{\mathbf{q}}\\ \lambda\end{bmatrix}}_{\text{plan}} + \begin{bmatrix}\mathbf{K_h} & \\ & \mathbf{K_q}\\ \mathbf{K}_\lambda & \end{bmatrix}\left(\underbrace{\begin{bmatrix}\mathbf{h}\\ \begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix}\end{bmatrix}}_{\text{est}} - \underbrace{\begin{bmatrix}\mathbf{h}\\ \begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix}\end{bmatrix}}_{\text{plan}}\right)$$

FIGURE 33: This figure shows how control references for the hierarchical inverse dynamics controller are computed. Given these references and a description of the ranks and importance weightings of tasks within each hierarchy, the hierarchical inverse dynamics controller [92] will compute the required joint torques to track the desired closed loop behaviors as close as possible within the feasible sets.

given a set of hierarchies that define strict task priorities and the importance weighting of each task within a hierarchy. The most important references provided include endeffector wrenches, joint accelerations, and momentum rate references, as shown in Figure 33. These references include a feedforward term provided by the motion planner and state-control corrections coming from the feedback controllers. Controls are interpolated and low-pass filtered to obtain smooth continuous trajectories.

$\mathbf{K}_\lambda$ and $\mathbf{K_h}$ are force and momentum gains respectively, used for optimal stabilization of the desired dynamic behaviors. Similarly, within the kinematics problem $\mathbf{K_q}$ is the whole-body gain to optimally track whole-body behaviors. The hierarchical inverse dynamics problem is solved using the set of ranks and tasks described in Table 1. Note that for simplicity only two hierarchies are used: one for constraints and one for tasks, but other architectures are also possible as well.

5.6 experimental results

The experimental results, to be shown in this section, include simple experiments to analyze the effect of noise sensitivity into the feedback control policies, but also more complex ones to evaluate the algorithm's performance at the stabilization of kino-dynamic motion plans on a simulated humanoid robot. The performed experiments demonstrate the effect of considering measurement uncertainty on impedance policies and the behaviors they generate. The capabilities of the proposed feedback control architecture are tested in several multi-contact scenarios (as shown in Figure 9) including climbing stairs with and without using hands, among others.

5.6.1 *Experiment 1: Process Noise Vs. Measurement Uncertainty*

In this subsection, the control algorithm is applied to a two degree of freedom (DOF) manipulator on a passing through a viapoint task. This simple setting allows the analysis of important properties of the algorithm.

Its equations of motion are given by $\mathrm{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathrm{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathrm{J}(\mathbf{q})^T \lambda + \tau$. The vector $\mathbf{q} = [\mathrm{q}_1, \mathrm{q}_2]^T$ contains the joints positions. $\mathrm{M}(\mathbf{q})$ denotes the inertia matrix, $\mathrm{C}(\mathbf{q}, \dot{\mathbf{q}})$ the vector of Coriolis and centrifugal forces, $\mathrm{J}(\mathbf{q})$ the endeffector Jacobian, $\lambda \in \mathbb{R}^2$ the external forces and $\tau \in \mathbb{R}^2$ the input torques. The system dynamics can be easily written in the form given by equation (21) with additive process noise $d\omega$ and state $\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$. The measurement model can also easily be written in the form given by equation (22) ($d\mathbf{y} = d\mathbf{x} + d\gamma$) with Brownian motion γ and variance Γdt.

The effects of process and measurement noise in the control law are compared in a motion task between two points with several viapoints. The objective function

$$\mathcal{J} = \sum_{0}^{t_f} \mathcal{Q}_u(\tau) + \sum_{i=1}^{N_{via}} \mathcal{Q}_i(\mathbf{x} - \mathbf{x}_i) + \mathcal{Q}_{t_f}(\mathbf{x} - \mathbf{x}_{t_f})$$

measures task performance. $\mathbf{x}, \mathbf{x}_i, \mathbf{x}_{t_f} \in \mathbb{R}^4$ are current, viapoints and final desired endeffector positions and velocities, respectively. $\mathcal{Q}_u$, $\mathcal{Q}_i$ and $\mathcal{Q}_{t_f}$ are quadratic cost functions that regularize control signals, reward passing through the viapoints and arriving to the final endeffector location given in cartesian space.

First, motion and optimal feedback behaviors for this system are shown under no process noise and no measurement uncertainty (Figure 34, to the left). Then, the center plot in Figure 34 presents the results of evaluating the effects of increasing process noise while keeping the level of measurement uncertainty at zero. It shows the nominal motion and optimal feedback gains for the motion task under process noise. As process noise increases, the cost of uncertainty does too, because the manipulator might miss passing through the viapoints or the goal due to disturbances. Under these circumstances, the trade-off between cost of uncertainty and control-

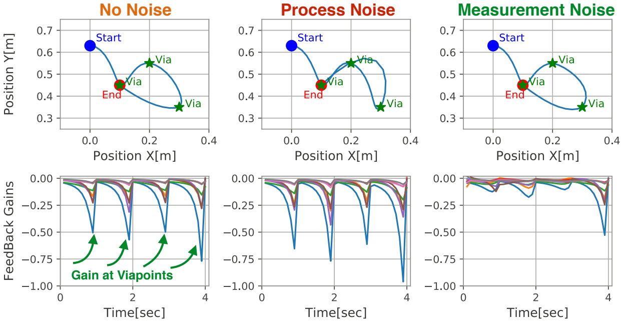


FIGURE 34: This figure depicts cartesian trajectories and impedances to control the motion of a 2-DOF finger moving from a Start to and End point passing through three Viapoints. *To the left*, optimal impedance behaviors under no noise in the system are shown; *in the center*, the impedance gains under process noise; and *to the right*, the values of feedback gains under measurement noise. Notice that the higher the process noise, the higher the optimal values for the impedances; while for measurement uncertainty: the higher the noise, the lower the feedback gains.

effort involves feedback gains proportional to the process noise, *the higher the process noise, the higher the feedback gains* in order to maintain the desired performance.

In a second set of simulations, the effect of increasing measurement uncertainty under no process noise are tested (Figure 34, to the right). Optimal feedback gains and nominal motion trajectories for the motion task under measurement noise are shown. Feedback gains are also higher near viapoints and goal position, as in the case of the controller sensitive to process noise. The difference is that the optimal control policy in this case is to rely on feedback proportionally to the information content of the measurements, i.e. *the higher the measurement uncertainty, the lower the feedback gains*. It shows how under low measurement noise, feedback control with higher gains is possible and optimal (as shown in the No Noise case). Under high measurement noise, lower impedance is better. In these experiments, the risk sensitive parameter was kept constant as it is not the focus of this paper (see for example [73]). However, the effects of process and measurement noise are qualitatively similar for the allowed values of σ (data not shown).

5.6.2 *Experiment 2: Process Noise Vs. Measurement Uncertainty with Control Constraints*

In this experiment, the algorithm is applied in a control-constrained task, namely a car parking maneuver [31], where front wheels' acceleration and turning angle are constrained. This experiment allows to show that the proposed algorithm can

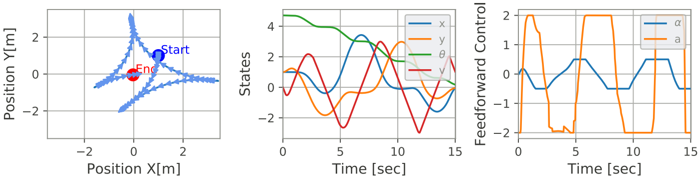


Figure 35: This figure shows examples of nominal trajectories for the car parking example: *to the left*, cartesian motion and orientation of the car moving from a Start to an End position are shown; *in the center*, the evolution of the state variables over time ("x, y" cartesian position, θ angle with respect to the x-axis, and the velocity of the forward wheels "v") is displayed; and *to the right*, nominal control feedforward trajectories ("a" acceleration of front wheels and "α" turning angle of the wheels).

still be applied to a more complex system that includes control constraints and the same tendency in the optimal noise-dependent impedance behaviors holds.

The system's state (x, y, θ, v) is composed by: "x, y" the position of the point in between the backward wheels, θ is the angle of the car with respect of the x-axis, and "v" denotes the velocity of the front wheels. The control vector is composed by: α denoting the angle of the front wheels, and "a" the acceleration of the front wheels. Similarly to the formulation of the car parking experiment presented in [31], the rolling distance of front and back wheels is defined as:

$$\begin{aligned} f &= \Delta v \\ b &= f\cos(\alpha) + d - \sqrt{d^2 - f^2\sin^2(\alpha)} \end{aligned}$$

where "d" denotes the distance between the front and back axles, and Δ the timestep discretization. The discrete-time system dynamics are given by:

$$\begin{aligned} x_{k+1} &= x_k + b\cos(\theta) \\ y_{k+1} &= y_k + b\sin(\theta) \\ \theta_{k+1} &= \theta_k + \arcsin\left(\sin(\alpha)\frac{f}{d}\right) \\ v_{k+1} &= v_k + \Delta a \end{aligned}$$

The cost for the parking task is written as a final cost on the distance from the last state from (0,0,0,0), being at the origin and facing towards east at rest.

Figure 35 shows examples of nominal optimized trajectories for the car parking experiment. The left picture shows the car's motion in cartesian space to reorient its angle θ so that it faces towards east as desired. The center plot shows how all states converge to zero as desired; and the right plot depicts the nominal control feedforward trajectories and shows how them respect control limits.

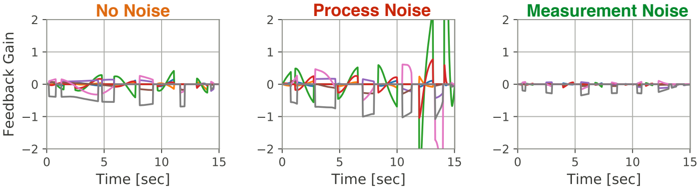


FIGURE 36: This figure shows examples of constrained feedback gains optimized with our algorithm and a constrained minimization of control signals. *To the left*, we show optimal impedance behaviors under no noise in the system; *in the center*, the impedance gains under process noise; and *to the right*, the values of feedback gains under measurement noise. We notice that under process noise, high impedance control is optimal; while under measurement uncertainty, compliant impedance control is optimal.

Finally, Figure 36 shows the effect of noise sensitivity to the different sources of uncertainty over impedance policies for tracking and stabilization of a desired behavior. Feedback control gains are constrained, similarly to [31], in such a way that they satisfy control constraints. The left plot shows feedback gains for a nominal zero noise case, where no process noise and no measurement noise are present into the system. It serves as a baseline for comparison to the gains of systems under different kinds of noise. The center plot presents feedback gains for this system under process noise. It is possible to see that the effect of process noise, such as external disturbances that can physically perturb the system, over the optimal control policy is the increase of the feedback gains as a function of the process noise level to scale down noise effects on the system. In contrast, in the measurement noise case (shown in the right plot), the effect of noise over the feedback policy is to decrease its value. One can interpret this result as: feedback based on noisy measurements is avoided, precisely because the confidence on the measurements is low and thus the system prefers to use feedforward control signals for control.

5.6.3 *Experiment 3: Stabilizing a Multi-Contact Interaction Task*

Our previous work [129] presented a simple contact interaction experiment where a 2-DOF manipulator established contact with a wall at an uncertain distance. The work in this chapter goes beyond this result and shows the results of applying the algorithm for controlling multi-contact interaction tasks on a humanoid robot.

5.6.3.1 *Nominal Impedance for Tracking Kino-Dynamic Plans*

This subsection presents in the first place the tracking performance of kino-dynamic motion plans using the described impedance controllers under nominal conditions (i.e. the evaluation of the impedance controllers' performance is under very small noise environments). As described in the background introductory section 1.6, the bridge between kinematic and dynamic variables is the robot centroidal momentum. Thus, its tracking allows to evaluate the success in the execution of a motion. Figure 37 and 38 show the tracking of optimized movement plans using the architecture described in Fig. 29. The top three rows show desired momentum trajectories (in orange) as well as its tracking (in blue). At the bottom left corner, the activation of the endeffectors over time can be seen, as given by the optimal timings, shown at the bottom right corner. At the beginning of the motion execution, the robot transitions from a stiff feedback controller to a pure feedforward one that includes feedforward and feedback control references; in this phase, executed momentum trajectories differ from desired ones. Figures 37 and 38 show how the proposed architecture for feedback control allows to track kino-dynamic behaviors as desired, given that planned force- and motion-induced momentum trajectories match at execution.

5.6.3.2 *Robust Impedance for Tracking of Kino-Dynamic Plans*

This subsection evaluates the effect of measurement uncertainty over optimal impedance gains for a multi-contact interaction task. The climbing up stairs motion task, shown in Fig. 9d, is used to illustrate these effects. Section 5.6.3.1 presented simulation experiments that showed that this motion can be successfully executed using the developed feedback controllers in an environment with very small noise levels. However, real world scenarios are noisy and thus now it will be shown that the use of this information is helpful at synthesizing noise-sensitive feedback controllers capable of successfully realizing a motion in such an environment. Environmental uncertainty is introduced by randomly perturbing the height of the stepping stones a few centimeters.

Under this setting, a comparison between a feedback controller sensitive to measurement uncertainty and one sensitive to process noise will be shown. As the uncertainty lies in the not exactly known contact locations (due to perturbations in stepping stone locations), uncertainty is modeled within the impedance controllers for tracking of kinematic plans (e.g. feedback controllers sensitive to noise in the legs' joints are synthesized) and feedback controllers for tracking of dynamic plans are synthesized with very small noise level. Figure 39a shows to the right a heat map of a force gain $\mathbf{K}_\lambda$ at one timestep of the trajectory. The top 6 and bottom 6 rows correspond to the wrench of the different feet. The first 3 columns correspond to gains for control corrections due to errors in the center of mass, the central 3 columns to corrections due to errors in linear momentum and the last 3 are gains for control corrections due to errors in angular momentum tracking. To the left a plot showing the norm of 3×3 sub-blocks of the force gains plotted over time can be visualized.

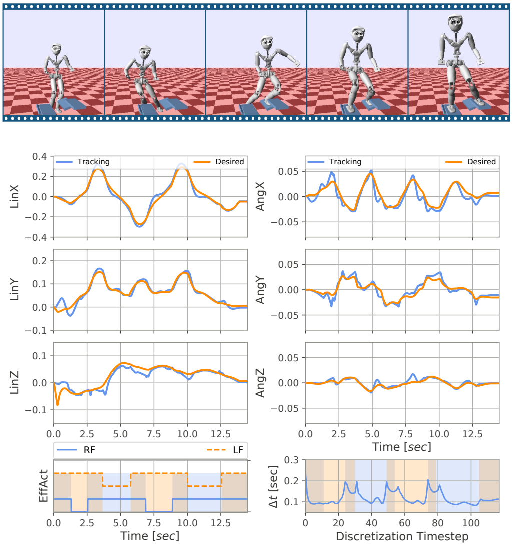


Figure 37: This figure shows examples of tracking desired centroidal momentum dynamic trajectories using the described kino-dynamic feedback controllers for a climbing up stairs motion. Momentum trajectories are shown normalized by the robot mass. EffAct plots show the activation of each endeffector over time. Vertical colored bars help to match optimal endeffector activations over time with bottom right plots displaying the optimal duration of each timestep discretization.

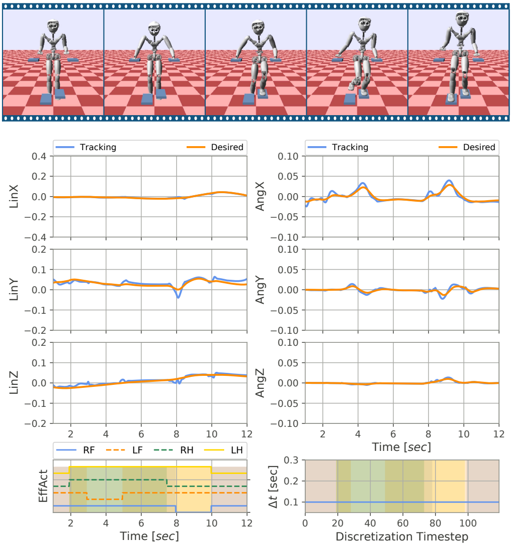
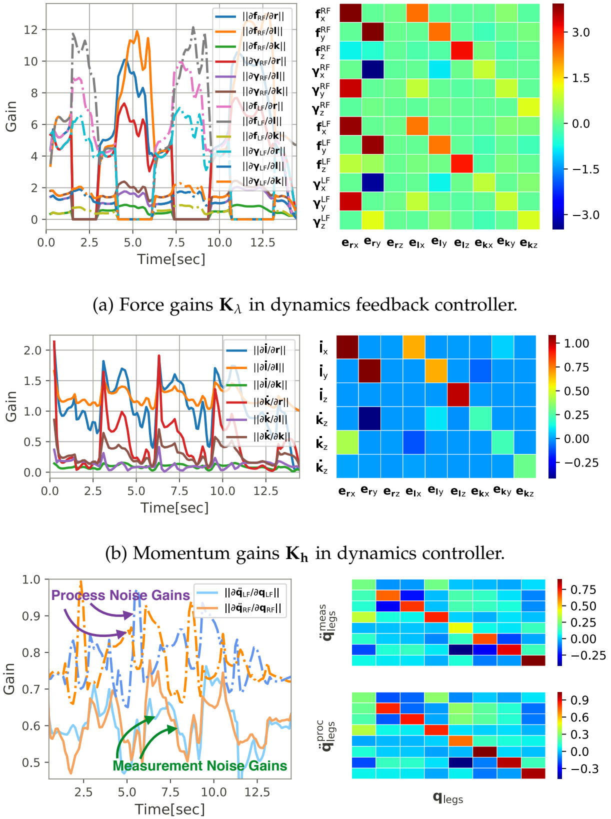


Figure 38: This figure shows examples of tracking desired centroidal momentum dynamic trajectories using the described kino-dynamic feedback controllers for a climbing up using hands motion. Momentum trajectories are shown normalized by the robot mass. EffAct plots show the activation of each endeffector over time. Vertical colored bars help to match optimal endeffector activations over time with bottom right plots displaying the optimal duration of each timestep discretization.



(a) Force gains $\mathbf{K}_\lambda$ in dynamics feedback controller.

(b) Momentum gains $\mathbf{K}_\mathbf{h}$ in dynamics controller.

(c) Whole-body gains $\mathbf{K}_\mathbf{q}$ in kinematics controller.

FIGURE 39: This figure shows time-indexed trajectory values and a heat map visualization of (39a) force, (39b) momentum and (39c) whole-body gains.

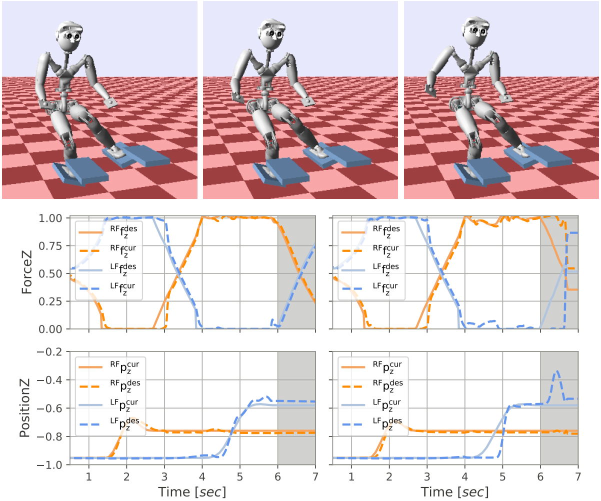


Figure 40: Cartesian force and location in the z direction of: measurement-noise sensitive feedback controller (to the left) and process-noise feedback controller (to the right).

Fig. 39b displays similar results but for the momentum gains $\mathbf{K_h}$, which are obtained by pre-multiplying the force gains by the control transition matrices of the linearized dynamics along the optimal trajectories. In the heat map, the first 3 rows correspond to linear momentum corrections and the last 3 rows to angular momentum corrections. As can be seen in the plots, the gains contain off-diagonal terms leading to optimal coupling between linear and angular momentum, which does not happen in a naive diagonal feedback control gain. To the left, the norm of 3x3 sub-blocks of the momentum gain plotted over time is shown. As can be seen, gain profiles change significantly over time and contact configurations to exploit the different couplings; however, discontinuities in the momentum trajectories were not observed.

While the dynamics feedback controllers are the same for both scenarios (modeling contact uncertainty either as measurement or process), the kinematics feedback controllers, shown in Fig. 39c, are not. To the right, heat maps of whole-body gains are shown at one timestep of the trajectories mapping errors in the joint configurations to corrections in joint accelerations. $\ddot{\mathbf{q}}^{\text{meas}}_{\text{legs}}$ corresponds to corrections in joint accelerations when contact location uncertainty has been modeled as measurement

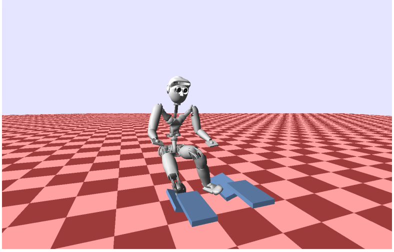


FIGURE 41: A standard feedback controller (optimized for negligible levels of noise) has also a similar problem to the controller sensitive to process-noise. Its optimal control policy implies also feedback gains high enough for a good tracking of the desired joint posture, reason for which when performing the motion in an uncertain environment, the robot hits the stepping stone laterally and fails.

noise, while $\ddot{\mathbf{q}}^{\text{proc}}_{\text{legs}}$ corresponds to corrections when contact location uncertainty has been modeled as without noise in the process model. Each leg is composed by four joints to control endeffector positions (HFE Hip Flexion-Extension, HAA Hip Abduction-Adduction, KR Knee Rotation, KFE Knee Flexion-Extension) and of three joints to control endeffector orientations. In these heat maps, only gains of the four joints in each leg that control positions and its couplings are shown. Notice that the off-diagonal terms significantly help to more easily track desired kinematic motions, while requiring lower gains than naive diagonal gains. To the left, the norm of the 4x4 diagonal sub-blocks of the whole-body gains plotted over time is shown. The dashed-dotted lines correspond to the norm of the controller gains that model uncertainty as process noise, and the solid lines correspond to the norm of the gains that model uncertainty as measurement noise. They have been normalized to one and show that measurement uncertainty leads to more compliant impedance behaviors that can ease the execution of a task in an uncertain environment.

In Fig. 40, an execution comparison of both feedback controllers in the uncertain environment is shown. On the gray highlighted region, it is shown how the impedance controller, sensitive to measurement uncertainty, smoothly handles landing on a stepping stone earlier than expected. However, the feedback controller sensitive to process-noise has a different behavior. Because it requires a stiffer policy, it impacts the stepping stone and bounces up again, which leads to higher forces on the active leg to support the robot's full weight. In the same figure, top pictures show how the controller sensitive to measurement uncertainty instead slides over the stepping stone after the early landing with a smooth interaction.

Finally Figure 41 shows how a standard feedback controller would also fail in such a noisy scenario as its control policy for the kinematic plan requires gains to track the desired postures that lead to the robot hitting laterally the stepping stone.

5.7 DISCUSSION

The presented experiments have shown that process noise is fundamentally different from measurement noise. While the first one is a dynamics disturbance that requires control using high feedback gains; the second one represents uncertainty in the state information, and requires compliance proportional to the uncertainty to dynamically interact with the world given our limited knowledge about it.

The fundamental difference between process and measurement noise effects on the control law comes from the cost they penalize. Cost of uncertainty due to process noise increases with terms of the form $\mathbf{C}_t\Omega_t\mathbf{C}_t^T$. Under no control action, process noise increases the cost. Thus, regulation with high gains is optimal. For measurement noise, cost grows with terms $\mathbf{K}_t\mathbf{D}_t\Gamma_t\mathbf{D}_t^T\mathbf{K}_t^T$ and estimation gains are inversely proportional to measurement noise. Thus, not using informative measurements is costly and requires high feedback gains. For poorly-informative measurements, cost is low and control with lower gains is optimal. This behavior can be used in robotic tasks with dynamic interactions. For example when making a contact, behaving compliant under poor contact-information could be the robust way of behaving. Once the contact is established and position certainty is higher, feedback gains could then be increased. In a receding horizon setup, gaining information about the current state of the world after contact would allow to online adapt the feedback control policy.

From a computational point of view, the algorithm, as shown in the experimental section, scales to complex systems. The complexity of a call to the dynamics can be approximated by its heaviest computation (factorization and back-substitution of $\mathbf{M}(\mathbf{q})$) as roughly $O(n^3)$, n being the number of states. The most expensive computation is that of first derivatives $O(Nn^4)$, N being the number of timesteps in the horizon. This is in the same order of complexity as other iterative approaches that show very good performance on also complicated robotic tasks [39], [31], although those examples did not exploit measurement uncertainty for control. While the proposed approach requires using twice the number of actual states, which increases the solving time a small amount, this is still fine given the impressive results of recent papers on high dimensional robotics problems with contacts [194].

5.8 CONCLUSION

An iterative algorithm for finding locally-optimal feedback controllers for nonlinear systems with additive measurement uncertainty was presented. In particular it was shown that measurement noise leads to very different behaviors than process noise and it can be exploited to create low impedance behaviors in uncertain environments (e.g. during contact interaction). This opens the possibility for planning and controlling contact interactions based on controllers sensitive to measurement noise. In a receding horizon setting, it could be possible to regulate impedance in a meaningful way depending on the current uncertainty about the environment.

Chapter 6

SUMMARY AND CONCLUSION

An approximate solution to the right problem is far better than an exact answer to an approximate problem.
— John Wilder Tukey, American statistician, 1915 -2000

This chapter concludes this dissertation by summarizing the approach proposed for motion planning and control of legged robots in multi-contact scenarios.

6.1 SUMMARY

This dissertation proposed an optimal control-based architecture for contacts and motion planning as well as for feedback control of floating-base robots performing complex tasks in multi-contact scenarios. The proposed architecture has been specially designed for floating-base rigid-body systems (such as robots with arms and legs) to exploit the geometrical properties of the trajectory optimization problem (dynamically challenging task in contact-rich scenarios) by efficiently building an approximate structured solution of the problem. The approach scales down the challenges of solving a very large nonlinear and nonconvex optimization problem by breaking it down into smaller problems and finding approximate solutions using efficient solvers with polynomial-time convergence properties. For instance, the discrete optimization problem, that in a first stage is in charge of selecting contact surfaces and locations from a terrain description, does not consider a full floating-base dynamics model that would render the optimization problem intractable. Instead it makes use of an approximate dynamics model based on the centroidal momentum dynamics and a simple kinematic reachability model, which give the algorithm a notion of dynamics and kinematics so as to keep its solution consistent and feasible for the next phase, but at the same time easy to find. Then in the next phase, the motion planning algorithm resolves in more detail whole-body movements, timings and contact interaction trajectories using solvers tailored to each sub-problem of the kino-dynamic structure and as a local solution around the previously selected contact surfaces.

The proposed architecture has been tested as a planning framework in simulation on a torque controlled humanoid robot performing several multi-contact tasks. However, its computational efficiency would make it well suited also as a framework for close to real-time receding horizon planning of complex tasks for any robot with arms and legs. The feedback control module of the proposed architecture, based on the superposition property of task impedances for optimal tracking of desired kino-dynamic behaviors as well as the adaptability of the control policies to environmental uncertainty, made it possible to successfully execute in a physical simulator several multi-contact motions over difficult terrains, even in the presence of discrepancies between the expected and actual locations of the stepping stones.

The performed experiments have demonstrated the benefits of the proposed architecture for motion planning and feedback control of legged robots thanks to the properties it exploits. The contribution of this dissertation brings the state-of-the-art in the field closer to the vision of real-time whole-body control of autonomous robots. However, the reliable execution of highly dynamic tasks in complex environments with real humanoid robots remains an open challenge for future research [*].

[*] Source code is made available at https://git-amd.tuebingen.mpg.de/bponton/timeoptimization in the hope that it can be useful to further progress of other interested researchers.

6.2 conclusion

To conclude, the most important contributions proposed in this dissertation include: a structured solution of a motion planning problem that makes use of an appropriate level of complexity for each planning stage, a contacts planner based on a measure of dynamical robustness which makes it better suited for selecting contacts in dynamically challenging scenarios, an approximation of the trajectory optimization problem for time-optimal whole-body movement generation, and a feedback control architecture to execute the desired kino-dynamic motion plans even in the presence of uncertainties. Thus, the proposed architecture for contacts and motion - planning and control of legged robots, based on the previously mentioned contributions, made it possible as a whole to efficiently generate a contacts and motion plan as well as to successfully execute it on a physical simulator. As future work remains the testing of the proposed architecture in robotic hardware experiments.

BIBLIOGRAPHY

[1] S. Schaal and C. Atkeson. “Learning control in robotics”. In: *2010 Robotics and Automation Magazine*, pages 20–29. doi 10.1109/MRA.2010.936957.

[2] C. Atkeson, B.P. Wisely Babu, N. Banerjee, D. Berenson, C.P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M.A. Gennert, J.P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, 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. doi 10.1109/HUMANOIDS.2015.7363436.

[3] M. Spenko, S. Buerger and K. Iagnemma. “The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue”. In: *2018 Springer Tracts in Advanced Robotics*. Springer, pages 1–684. doi 10.1007/978-3-319-74666-1.

[4] N. Alunni, H. Bener Suay, C. Phillips-grafflin and J. Mainprice. “DARPA Robotics Challenge: Towards a User-Guided Manipulation Framework for High-DOF Robots”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2088. doi 10.1109/ICRA.2014.6907141.

[5] 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. doi 10.1109/HUMANOIDS.2014.7041488.

[6] Fetch Robotics. `https://fetchrobotics.com/`. Online, accessed: 2019-01-23.

[7] Rethink Robotics. `https://www.rethinkrobotics.com/`. Online, accessed: 2019-01-23.

[8] Mujin Robotics. `https://mujin.co.jp/`. Online, accessed: 2019-01-23.

[9] 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”. In: *2017 Springer Construction Robotics*, pages 3–14. doi 10.1007/s41693-017-0003-5.

[10] M. Johnson, B. Shrewsbury, S. Bertrand, D. Calvert, T. Wu, D. Duran, D. Stephen, N. Mertins, J. Carff, W. Rifenburgh, J. Smith, C. Schmidt-Wetekam, D. Faconti, A. Graber-Tilton, N. Eyssette, T. Meier, I. Kalkov, T. Craig, N. Payton, S. McCrory, G. Wiedebach, B. Layton, P.D. Neuhaus and J.E. Pratt. "Team IHMC's Lessons Learned from the DARPA Robotics Challenge: Finding Data in the Rubble". In: *2017 Journal of Field Robotics*, pages 241–261. doi 10.1002/rob.21674.

[11] S. Kuindersma, R. Deits, M.F. 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". In: *2016 Autonomous Robots*, pages 429–455. doi 10.1007/s10514-015-9479-3.

[12] J. Lim, I.H. Lee, I. Shim, H. Jung, H. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Kyu Lee, Y. Bok, D.G. Choi, C. Buyoun, S. Kim, J. Heo, I. Kim, J. Lee, I.S. Kewon and J.H. Oh. "Robot System of DRC-HUBO+ and Control Strategy of Team KAIST in DARPA Robotics Challenge Finals". In: *2017 Journal of Field Robotics*, pages 802–829. doi https://doi.org/10.1002/rob.21673.

[13] SoftBank Robotics. `https://www.softbankrobotics.com/`. Online, accessed: 2019-01-23.

[14] B. Brogliato. Nonsmooth Mechanics - Models, Dynamics and Control. 2nd ed. Communications and Control Engineering. Springer-Verlag London. doi 10.1007/978-1-4471-0557-2.

[15] P.B. Wieber. "On the stability of walking systems". In: *2002 Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*. doi inria-00390866.

[16] S.M. LaValle. Planning Algorithms. 2006 Cambridge University Press, University of Illinois, pages 1–842. doi 10.1017/CBO9780511546877.

[17] P.B. Wieber. "Holonomy and Nonholonomy in the Dynamics of Articulated Motion". In: *2006 Fast Motions in Biomechanics and Robotics*. Engineering. Springer, Berlin, pages 411–425. doi 10.1007/978-3-540-36119-0_20.

[18] R. Featherstone. Rigid Body Dynamics Algorithms. 2008 Springer Verlag US. doi 10.1007/978-1-4899-7560-7.

[19] E.R. Westervelt, J.W. Grizzle and D.E. Koditschek. "Hybrid Zero Dynamics of Planar Biped Walkers". In: *2003 IEEE Transactions on Automatic Control*, pages 42–56. doi 10.1109/TAC.2002.806653.

[20] A. Herzog. Optimization-based motion generation for multiped robots in contact scenarios. 2017 ETH Collection. doi 10.3929/ethz-b-000199099.

[21] M. Raibert, K. Blankespoor, G. Nelson and R. Playter. "Bigdog, the rough terrain quadruped robot". In: *2008 IFAC Proceedings*, pages 10822–10825. doi 10.3182/20080706-5-KR-1001.01833.

[22] 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*. IEEE, pages 2554–2561. doi 10.1109/ICRA.2013.6630926.

[23] A. Herzog, N. Rotella, S. Schaal and L. Righetti. "Trajectory generation for multi-contact momentum-control". In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 874–880. doi 10.1109/HUMANOIDS.2015.7363464s.

[24] S. Lohmeier, T. Buschmann and H. Ulbrich. "System Design and Control of Anthropomorphic Walking Robot LOLA". In: *2009 IEEE/ASME Transactions on Mechatronics*, pages 658–666. doi 10.1109/TMECH.2009.2032079.

[25] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki and K. Fujimura. "The intelligent ASIMO: system overview and integration". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and System*, pages 2478–2483. doi 10.1109/IRDS.2002.1041641.

[26] Y. Wang and S. Boyd. "Fast Model Predictive Control Using Online Optimization". In: *2010 IEEE Transactions on Control Systems Technology*, pages 267–278. doi 10.1109/TCST.2009.2017934.

[27] M. Morari and J. Lee. "Model Predictive Control: Past, Present and Future". In: *1999 Elsevier Computers and Chemical Engineering*, pages 667–682. doi 10.1016/S0098-1354(98)00301-9.

[28] D. Mayne. "Model Predictive Control". In: *2014 Journal Automatica (IFAC)*, pages 2967–2986. doi 10.1016/j.automatica.2014.10.128.

[29] M. Posa, C. Cantu and R. Tedrake. "A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact". In: *2014 International Journal of Robotics Research*, pages 69–81. doi 10.1177/0278364913506757.

[30] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart and J. Buchli. "Fast Nonlinear Model Predictive Control for Unified Trajectory Optimization and Tracking". In: *2016 IEEE International Conference on Robotics and Automation*, pages 1398–1404. doi 10.1109/ICRA.2016.7487274.

[31] Y. Tassa, N. Mansard and E. Todorov. "Control-limited differential dynamic programming". In: *2014 International Conference on Robotics and Automation*, pages 1168–1175. doi 10.1109/ICRA.2014.6907001.

[32] J. Carpentier and N. Mansard. "Multi-contact Locomotion of Legged Robots". In: *2018 IEEE Transactions on Robotics*, pages 1441–1460. doi 10.1109/TRO.2018.2862902.

[33] S. Skogestad and I. Postlethwaite. Multivariable Feedback Control: Analysis and Design. 2005 USA: John Wiley and Sons, Inc.

[34] K.J. Astrom and R.M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Princeton, NJ, USA: 2008 Princeton University Press.

[35] G.F. Franklin, D.J. Powell and A. Emami-Naeini. Feedback Control of Dynamic Systems. 4th. Upper Saddle River, NJ, USA: 2001 Prentice Hall PTR.

[36] R. Bellman. On the Theory of Dynamic Programming. 1952 Proceedings of the National Academy of Sciences.

[37] S. Boyd and L. Vandenberghe. Convex Optimization. New York: Cambridge University Press (2009).

[38] J. Nocedal and S.J. Wright. Numerical Optimization. second. Springer Series in Operations Research and Financial Engineering. New York, NY, USA: 2006 Springer. doi 10.1007/978-0-387-40065-5.

[39] Y. Tassa. "Fast Model Predictive Control for Reactive Robotic Swimming". In: *2010 ResearchGate Contribution*.

[40] 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)*. IEEE, pages 3346–3351. doi 10.1109/IROS.2015.7353843.

[41] S. Schaal. The SL simulation and real-time control software package. Tech. rep. clmc. Los Angeles, CA.

[42] Y.C. Lin and D. Berenson. "Using previous experience for humanoid navigation planning". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots*. doi 10.1109/HUMANOIDS.2016.7803364.

[43] S.Y. Chung and O. Khatib. "Contact-consistent elastic strips for multi-contact locomotion planning of humanoid robots". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6289–6294. doi 10.1109/ICRA.2015.7140082.

[44] Y.C. Lin, B. Ponton, L. Righetti and D. Berenson. "Efficient Humanoid Contact Planning using Learned Centroidal Dynamics Prediction". In: *2019 international Conference on Robotics and Automation*. doi arxiv/1810.13082.

[45] L. Kovar, M. Gleicher and F. Pighin. "Motion Graphs". In: *2002 Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '02. San Antonio, Texas: ACM, pages 473–482. doi 10.1145/566570.566605.

[46] J. Pettré, J.P. Laumond and T. Siméon. "A 2-Stages Locomotion Planner for Digital Actors". In: *2003 Symposium on Computer Animation*. Ed. by D. Breen and M. Lin. The Eurographics Association. doi 10.2312/SCA03/258-264.

[47] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse and E. Yoshida. "Real-time replanning using 3D environment for humanoid robot". In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pages 584–589. doi 10.1109/Humanoids.2011.6100844.

[48] 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: *2015 Springer Proceedings in Advanced Robotics*, pages 287–303. doi 10.1007/978-3-319-60916-4_17.

[49] P. Fernbach, S. Tonneau, A. Del Prete and M. Taix. "A kinodynamic steering-method for legged multi-contact locomotion". In: *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 3701–3707. doi 10.1109/IROS.2017.8206217.

[50] T. Bretl, S. Rock, J.C. Latombe, B. Kennedy and H. Aghazarian. Free-climbing with a multi-use robot. English (US). Ed. by Marcelo Ang and Oussama Khatib. Springer Tracts in Advanced Robotics. Springer, pages 449–458. doi 10.1007/11552246_43.

[51] 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*. IEEE, pages 1165–1170. doi 10.1109/ROBOT.2009.5152353.

[52] A. Escande, A. Kheddar, S. Miossec and S. Garsault. "Planning Support Contact-Points for Acyclic Motions and Experiments on HRP-2". In: *ISER*. Springer Tracts in Advanced Robotics. Springer, pages 293–302. doi 10.1007/978-3-642-00196-3_35.

[53] K. Bouyarmane and A. Kheddar. "Humanoid Robot Locomotion and Manipulation Step Planning". In: *2012 Advanced Robotics*, pages 1099–1126. doi 10.1080/01691864.2012.686345.

[54] M. Stilman. "Global Manipulation Planning in Robot Joint Space with Task Constraints". In: *2010 IEEE Transacions on Robotics*. IEEE Press, pages 576–584. doi 10.1109/TRO.2010.2044949.

[55] I. Mordatch, E. Todorov and Z. Popovic. "Discovery of complex behaviors through contact-invariant optimization". In: *2012 ACM Transactions on Graphics*, pages 43:1–43:8. doi 10.1145/2185520.2185539.

[56] A. Ibanez, P. Bidaud and V. Padois. "Emergence of humanoid walking behaviors from Mixed-Integer Model Predictive Control". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4014–4021. doi 10.1109/IROS.2014.6943127.

[57] D. Dimitrov, A. Paolillo and P.B. Wieber. "Walking motion generation with online foot position adaptation based on l1- and linf-norm penalty formulations." In: *2011 IEEE International Conference on Robotics and Automation*, pages 3523–3529. doi 10.1109/ICRA.2011.5979671.

[58] 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. doi 10.1109/HUMANOIDS.2014.7041373.

[59] L.S. Pontryagin. The Mathematical Theory of Optimal Processes. 1962 CRC Press (reprint).

[60] N. Sugimoto and J. Morimoto. "Switching multiple LQG controllers based on Bellman's optimality principle: Using full-state feedback to control a humanoid robot". In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, pages 3185–3191. doi 10.1109/HUMANOIDS.2014.7041435.

[61] R. Bellman. Dynamic Programming. 1957 Dover Publications.

[62] D. Bertsekas. Dynamic programming and optimal control. 1996 Athena Scientific.

[63] E. Keogh and A. Mueen. "Curse of Dimensionality". In: *2017 Encyclopedia of Machine Learning and Data Mining*. Springer US, pages 314–315. doi 10.1007/978-1-4899-7687-1_192.

[64] D.H. Jacobson. "New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach". In: *1968 Journal of Optimization Theory and Applications*, pages 411–440. doi 10.1007/BF00925746.

[65] E. Todorov and E. Li. "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems". In: *Proceedings of the 2005, American Control Conference, 2005*, pages 300–306. doi 10.1109/acc.2005.1469949.

[66] S. Feng, X. X., W. 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. doi 10.1109/HUMANOIDS.2013.7029950.

[67] M. Giftthaler, M. Neunert, M. Staeuble, J. Buchli and M. Diehl. "A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. doi 10.1109/IROS.2018.8593840.

[68] D. Mayne. "A Second-order Gradient Method for Determining Optimal Trajectories of Non-linear Discrete-time Systems". In: *1966 International Journal of Control*, pages 85–95. doi 10.1080/00207176608921369.

[69] A. Sideris and J.E. Bobrow. "An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems". In: *2005 IEEE Transactions on Automatic Control*, pages 2043–2047. doi 10.1109/ACC.2005.1470308.

[70] D. Jacobson. "Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games". In: *1973 IEEE Transactions on Automatic Control*, pages 124–131. doi 10.1109/TAC.1973.1100265.

[71] W. Li and E. Todorov. "Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system". In: *2007 International Journal of Control*, pages 1439–1453. doi 10.1080/00207170701364913.

[72] E. Todorov. "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system". In: *2005 Neural Computation*. doi 10.1162/0899766053491887.

[73] F. Farshidian and J. Buchli. "Risk Sensitive, Nonlinear Optimal Control: Iterative Linear Exponential-Quadratic Optimal Control with Gaussian Noise". In: *2015 Arxiv contribution*, pages 1–7. doi arxiv/1512.07173.

[74] J. Speyer, J. Deyst and D. Jacobson. "Optimization of stochastic linear systems with additive measurement and process noise using exponential performance criteria". In: *1974 IEEE Transactions on Automatic Control*, pages 358–366. doi 10.1109/TAC.1974.1100606.

[75] J. Speyer. "An Adaptive Terminal Guidance Scheme based on an Exponential Cost Criterion with Application to Homing Missile Guidance". In: *1976 Transactions on Automatic Control*, pages 371–332. doi 10.1109/TAC.1976.1101206s.

[76] P.R. Kumar and J.H. Van Schuppen. "On the Optimal Control of Stochastic Systems with an Exponential-of-Integral Performance Index". In: *1981 Journal of Mathematical Analysis and Applications*, pages 312–332. doi 10.1016/0022-247X(81)90109-8.

[77] A. Bensoussan and J.H. Van Schuppen. "Optimal Control of Partially Observable Stochastic Systems with an Exponential-of-Integral Performance Index". In: *1985 Siam Journal Control and Optimization*, pages 599–613. doi https://doi.org/10.1137/0323038.

[78] P. Whittle. "Risk-Sensitive Linear/Quadratic/Gaussian Control". In: *1981 Advances in Applied Probability*, pages 764–777. doi 0001-8678/81/040764-14.

[79] A. Herzog, S. Schaal and L. Righetti. “Structured contact force optimization for kino-dynamic motion generation”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2703–2710. doi 10.1109/IROS.2016.7759420.

[80] F. Farshidian, D. Pardo and J. Buchli. “Sequential Linear Quadratic Optimal Control for Nonlinear Switched Systems”. In: *20th IFAC World Congress (2017)*, pages 1463–1469. doi 10.1016/j.ifacol.2017.08.291.

[81] D. Pardo, L. Moeller, M. Neunert, A. Winkler and J. Buchli. “Evaluating Direct Transcription and Nonlinear Optimization Methods for Robot Motion Planning”. In: *2016 IEEE Robotics and Automation Letters*, pages 946–953. doi 10.1109/LRA.2016.2527062.

[82] S. Lengagne, J. Vaillant, A. Kheddar and E. Yoshida. “Generation of Whole-body Optimal Dynamic Multi-Contact Motions”. In: *The International Journal of Robotics Research*. SAGE Publications (UK and US), pages 1104–1119. doi 10.1177/0278364913478990.

[83] 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: *Humanoids*. IEEE, pages 671–676. doi 10.1109/HUMANOIDS.2014.7041435.

[84] 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*. IEEE, pages 295–302. doi 10.1109/HUMANOIDS.2014.7041375.

[85] T. Erez, Y. Tassa and E. Todorov. “Infinite-Horizon Model Predictive Control for Periodic Tasks with Contacts”. In: *2011 Proceedings of Robotics: Science and Systems*. doi 10.15607/RSS.2011.VII.010.

[86] M. Neunert, F. Farshidian, A. Winkler and J. Buchli. “Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds”. In: *2017 IEEE Robotics and Automation Letters*, pages 1502–1509. doi 10.1109/LRA.2017.2665685.

[87] 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. doi 10.1109/IROS.2012.6386025.

[88] E. Todorov. “A convex, smooth and invertible contact model for trajectory optimization”. In: *2011 IEEE International Conference on Robotics and Automation*, pages 1071–1076. doi 10.1109/ICRA.2011.5979814.

[89] 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)*. IEEE, pages 5307–5314. doi 10.1109/IROS.2015.7354126.

[90] I. Mordatch and E. Todorov. "Combining the benefits of function approximation and trajectory optimization". In: *Robotics: Science and Systems 2014*, pages 242–248. doi 10.15607/RSS.2014.X.052.

[91] 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: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pages 1644–1650. doi 10.1109/IROS.2003.1248880.

[92] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti. "Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid". In: *2014 Autonomous Robots*, pages 473–491. doi 10.1007/s10514-015-9476-6.

[93] D. Orin, A. Goswami and S.H. Lee. "Centroidal Dynamics of a Humanoid Robot". In: *2013 Autonomous Robots*. Hingham, MA, USA: Kluwer Academic Publishers, pages 161–176. doi 10.1007/s10514-013-9341-4.

[94] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse and N. Mansard. "A versatile and efficient pattern generator for generalized legged locomotion". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3555–3561. doi 10.1109/ICRA.2016.7487538.

[95] S. Caron and Q.C. Pham. "When to make a step? Tackling the timing problem in multi-contact locomotion by TOPP-MPC". In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 522–528. doi 10.1109/HUMANOIDS.2017.8246922.

[96] F. Farshidian, M. Neunert, A. Winkler, G. Rey and J. Buchli. "An efficient optimal planning and control framework for quadrupedal locomotion". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 93–100. doi 10.1109/ICRA.2017.7989016.

[97] Q.C. Pham and O. Stasse. "Time-Optimal Path Parameterization for Redundantly Actuated Robots: A Numerical Integration Approach". In: *2015 IEEE/ASME Transactions on Mechatronics*, pages 3257–3263. doi 10.1109/TMECH.2015.2409479.

[98] A. Winkler, D. Bellicoso, M. Hutter and J. Buchli. "Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization". In: *2018 IEEE Robotics and Automation Letters*, pages 1560–1567. doi 10.1109/LRA.2018.2798285.

[99] M. Khadiv, S. Ali A. Moosavian, A. Yousefi-Koma, M. Sadedel and S. Mansouri. “Optimal gait planning for humanoids with 3D structure walking on slippery surfaces”. In: *2017 Robotica*, pages 569–587. doi 10.1017/S0263574715000715.

[100] P.E. Gill, W. Murray and M.A. Saunders. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization”. In: *2002 SIAM Journal on Optimization*, pages 979–1006. doi 10.1137/S0036144504446096.

[101] A. Waechter and L.T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *2006 Mathematical Programming*, pages 25–57. doi 10.1007/s10107-004-0559-y.

[102] J. Andersson, J. Gillis, G. Horn, J.B. Rawlings and M. Diehl. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *2018 Mathematical Programming Computation*, pages 1–36. doi 10.1007/s12532-018-0139-4.

[103] J. Mattingley and S. Boyd. “CVXGEN: A Code Generator for Embedded Convex Optimization”. In: *2012 Optimization and Engineering*, pages 1–27. doi 10.1007/s11081-011-9176-9.

[104] A. Domahidi, E. Chu and S. Boyd. “ECOS: An SOCP solver for embedded systems”. In: *2013 European Control Conference*, pages 3071–3076. doi 10.23919/ECC.2013.6669541.

[105] D. Dimitrov, A. Sherikov and P.B. Wieber. “A sparse model predictive control formulation for walking motion generation”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2292–2299. doi 10.1109/IROS.2011.6095035.

[106] A. Herdt, N. Perrin and P.B. Wieber. “Walking without thinking about it”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pages 190–195. doi 10.1109/IROS.2010.5654429.

[107] D. Hajinezhad and Q. Shi. “Alternating direction method of multipliers for a class of nonconvex bilinear optimization: convergence analysis and applications”. In: *2018 Journal on Global Optimization*, pages 261–288. doi 10.1007/s10898-017-0594-x.

[108] A. Nemirovski. Interior Point Polynomial Time Methods in Convex Programming. Online, accessed: 31.01.2019. SIAM.

[109] D.C. Liu and J. Nocedal. “On the limited memory BFGS method for large scale optimization”. In: *Mathematical Programming*, pages 503–528. doi https://doi.org/10.1007/BF01589116.

[110] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *2011 Foundations and Trends in Machine Learning*, pages 1–122. doi 10.1561/2200000016.

[111] N. Parikh and S. Boyd. "Proximal Algorithms". In: *2014 Foundations and Trends in Optimization*. Hanover, MA, USA: Now Publishers Inc., pages 127–239. doi 10.1561/2400000003.

[112] B. O'Donoghue, E. Chu, N. Parikh and S.P. Boyd. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding". In: *2016 Journal Optimization Theory and Applications*, pages 1042–1068. doi 10.1007/s10957-016-0892-3.

[113] K. Huang and N.D. Sidiropoulos. "Consensus-ADMM for General Quadratically Constrained Quadratic Programming". In: *2016 IEEE Transactions on Signal Processing*, pages 5297–5310. doi 10.1109/TSP.2016.2593681.

[114] J. Park and S. Boyd. "General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming". In: *2017 Arxiv Contribution*, pages 1–63. doi arxiv/1703.07870.

[115] K. Derinkuyu and M. Pinar. "On the S-procedure and Some Variants". In: *2006 Mathematical Methods of Operations Research*, pages 55–77. doi 10.1007/s00186-006-0070-8.

[116] A. Ben-Tal and A. Nemirovski. "On Polyhedral Approximations of the Second-Order Cone". In: *Math. Oper. Res.* pages 193–205. doi 10.1287/moor.26.2.193.10561.

[117] S. Boyd. Branch and Bound Methods. `https://stanford.edu/class/ee364b/lectures/bb_slides.pdf`. Online, accessed 31.01.2019.

[118] I. Fantoni and R. Lozano. Non-Linear Control for Underactuated Mechanical Systems. Berlin, Heidelberg: 2001 Springer-Verlag.

[119] R. Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on 2019-01-30 from http://underactuated.mit.edu.

[120] 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: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5107–5112. doi 10.1109/ICRA.2015.7139910.

[121] E. Heijmink, A. Radulescu, B. Ponton, V. Barasuol, D. Caldwell and C. Semini. "Learning Optimal Gait Parameters and Impedance Profiles for Legged Locomotion". In: *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2017*, pages 339–346. doi 10.1109/HUMANOIDS.2017.8246895.

[122] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D.G. Caldwell, J. Cappelletto, J.C. Grieco, G. Fernández-López and C. Semini. "Simultaneous Contact, Gait, and Motion Planning for Robust Multilegged Locomotion via Mixed-Integer Convex Optimization". In: *2018 IEEE Robotics and Automation Letters*, pages 2531–2538. doi 10.1109/LRA.2017.2779821.

[123] R. Stengel. Optimal Estimation and Control. New York: 1994 Dover Publications Inc.

[124] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. "Biped walking pattern generation by using preview control of zero-moment point". In: *2003 IEEE International Conference on Robotics and Automation*. IEEE, pages 1620–1626. doi 10.1109/ROBOT.2003.1241826.

[125] T. Sugihara. "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator." In: *2009 IEEE International Conference on Robotics and Automation*. IEEE, pages 1966–1971. doi 10.1109/ROBOT.2009.5152284.

[126] T. Erez and E. Todorov. "Trajectory optimization for domains with contacts using inverse dynamics". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4914–4919. doi 10.1109/IROS.2012.6386181.

[127] B. Ponton, A. Herzog, A. Del Prete, S. Schaal and L. Righetti. "On Time Optimization of Centroidal Momentum Dynamics". In: *2018 IEEE International Conference on Robotics and Automation ICRA*, pages 1–7. doi 10.1109/ICRA.2018.8460537.

[128] B. Ponton, F. Farshidian and J. Buchli. "Learning Compliant Locomotion on a Quadruped Robot". In: *2014 IROS Workshop on Compliant Manipulation Challenges in Learning and Control*. IEEE, pages 1–4.

[129] B. Ponton, S. Schaal and L. Righetti. "On the Effects of Measurement Uncertainty in Optimal Control of Contact Interactions". In: *2016 Workshop on Algorithmic Foundations of Robotics*. IEEE, pages 1–22. doi arxiv/1605.04344.

[130] 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)*. MIT, Cambridge, USA. IEEE, pages 579–586. doi 10.1109/HUMANOIDS.2016.7803333.

[131] P.B. Wieber. "Viability and predictive control for safe locomotion". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1103–1108. doi 10.1109/IROS.2008.4651022.

[132] J.P. Aubin. Viability Theory: New directions. 2nd ed. Systems Theory and Control. 1991 Springer-Verlag Berlin Heidelberg. doi 10.1007/978-3-642-16684-6.

[133] N. Hogan. "Impedance Control: An Approach to Manipulation: Part I—Theory". In: *1985 Journal of Dynamic Systems, Measurement, and Control*. ASME International. doi 10.1115/1.3140702.

[134] P. Sardain and G. Bessonnet. "Forces acting on a biped robot. Center of pressure-zero moment point". In: *2004 IEEE Trans. Systems, Man, and Cybernetics, Part A*, pages 630–637. doi 10.1109/TSMCA.2004.832811.

[135] M. Vukobratovic and B. Borovac. "Zero-Moment Point - Thirty Five Years of its Life." In: *2004 International Journal on Humanoid Robotics*, pages 157–173. doi 10.1142/S0219843604000083.

[136] P.B. Wieber. "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations". In: *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 137–142. doi 10.1109/ICHR.2006.321375.

[137] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Soueres. "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control". In: *2017 IEEE Robotics and Automation Letters*, pages 10–17. doi 10.1109/LRA.2016.2518739.

[138] R. Tedrake, S. Kuindersma, R. Deits and K. Miura. "A closed-form solution for real-time ZMP gait generation and feedback stabilization". In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, pages 936–940. doi 10.1109/HUMANOIDS.2015.7363473.

[139] A. Sherikov, D. Dimitrov and P.B. Wieber. "Whole body motion controller with long-term balance constraints". In: *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 444–450. doi 10.1109/HUMANOIDS.2014.7041399.

[140] S. Kuindersma, F. Permenter and R. Tedrake. "An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2594. doi 10.1109/ICRA.2014.6907230.

[141] J. Englsberger, C. Ott and A. Albu-Schaeffer. "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion". In: *2015 IEEE Transactions on Robotics*, pages 355–368. doi 10.1109/TRO.2015.2405592.

[142] I.G. Ramirez, M. Naveau, C. Benazeth, O. Stasse, J.P. Laumond, K. Harada and E. Yoshida. "Motion generation for pulling a fire hose by a humanoid robot". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, pages 1016–1021. doi 10.1109/HUMANOIDS.2016.7803396.

[143] I.R. Manchester, U. Mettin, F. Iida and R. Tedrake. "Stable dynamic walking over rough terrain: theory and experiment". In: *2009 Springer Tracts in Advanced Robotics*. Springer, pages 123–138. doi 10.1007/978-3-642-19457-3_8.

[144] J. Koschorreck and K. Mombaur. "Modeling and optimal control of human platform diving with somersaults and twists". In: *2012 Springer Optimization and Engineering*, pages 29–56. doi 10.1007/s11081-011-9169-8.

[145] K.H. Koch, K. Mombaur and P. Soueres. "Optimization-Based Walking Generation for Humanoid Robot". In: *10th IFAC Symposium on Robot Control International Federation of Automatic Control (2012)*. International Federation of Automatic Control, pages 498–504. doi 10.3182/20120905-3-HR-2030.00189.

[146] S. Caron and A. Kheddar. "Multi-contact walking pattern generation based on model preview control of 3D COM accelerations". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 550–557. doi 10.1109/HUMANOIDS.2016.7803329.

[147] 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: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4030–4035. doi 10.1109/IROS.2014.6943129.

[148] 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*, pages 3103–3109. doi 10.1109/ICRA.2013.6631008.

[149] B. Ponton, A. Herzog, S. Schaal and L. Righetti. "A Convex Model of Humanoid Momentum Dynamics for Multi-Contact Motion Generation". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 842–849. doi 10.1109/HUMANOIDS.2016.7803371.

[150] S. Diamond and S. Boyd. "CVXPY: A Python-Embedded Modeling Language for Convex Optimization". In: *2016 Journal of Machine Learning Research*, pages 1–5. doi PMCID: PMC4927437.

[151] X. Shen, S. Diamond, Y. Gu and S.P. Boyd. "Disciplined convex-concave programming". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pages 1009–1014. doi 10.1109/CDC.2016.7798400.

[152] F. Callier and C. Desoer. Linear Systems Theory. New York: 1994 Springer.

[153] D. Qian, J. Yi and S. Tong. "Understanding Neighborhood of Linearization in Undergraduate Control Education". In: *2013 IEEE Control Systems Magazine*, pages 54–60. doi 10.1109/MCS.2013.2258767.

[154] M. Khadiv, A. Herzog, S.A.A. Moosavian and L. Righetti. "Step Timing Adjustment: A Step Toward Generating Robust Gaits". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 35–42. doi 10.1109/HUMANOIDS.2016.7803251.

[155] N.M. Amato, O. Burchan Bayazit, L.K. Dale, C. Jones and D. Vallejo. "OBPRM: An Obstacle-based PRM for 3D Workspaces". In: *1998 Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective*. Houston, Texas, USA, pages 155–168.

[156] T. Bretl. "Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem". In: pages 317–342. doi 10.1177/0278364906063979.

[157] X. Ji and J. Xiao. "Towards random sampling with contact constraints". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation*, pages 1390–1395. doi 10.1109/ROBOT.2000.844792.

[158] D. Hsu, R. Kindel, J.C. Latombe and S. Rock. "Randomized Kinodynamic Motion Planning with Moving Obstacles". In: pages 233–255. doi 10.1177/027836402320556421.

[159] H. Choset. "Probabilistic Roadmaps of Trees for Parallel Computation of Multiple Query Roadmaps". In: Springer, pages 80–89. doi 10.1007/11008941_9.

[160] H. Choset. "Principles of Robot Motion: Theory, Algorithms and Implementations". In: New York, NY, USA: Cambridge University Press, pages 271–271. doi 10.1017/S0263574706212803.

[161] D. Hsu, J. Latombe and R. Motwani. "Path planning in expansive configuration spaces". In: *1997 Proceedings of International Conference on Robotics and Automation*, pages 2719–2726. doi 10.1109/ROBOT.1997.619371.

[162] D.H. Kim, S.J. Lim, D.H. Lee, J.Y. Lee and C.S. Han. "A RRT-based motion planning of dual-arm robot for (Dis)assembly tasks". In: *IEEE ISR 2013*, pages 1–6. doi 10.1109/ISR.2013.6695698.

[163] H. Kwon, S. Park, C. Moon and W. Chung. "Trajectory planning for mobile robot with kinodynamic constraints". In: *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 893–894. doi 10.1109/URAI.2017.7992859.

[164] O. Arslan, K. Berntorp and P. Tsiotras. "Sampling-based algorithms for optimal motion planning using closed-loop prediction". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4991–4996. doi 10.1109/ICRA.2017.7989581.

[165] M. Saha. Motion Planning with Probabilistic Roadmaps. 2006 Stanford University, pages 1–155.

[166] 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. doi 10.1109/ROBOT.2009.5152817.

[167] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor and S. Schaal. "STOMP: Stochastic trajectory optimization for motion planning". In: *2011 IEEE International Conference on Robotics and Automation*, pages 4569–4574. doi 10.1109/ICRA.2011.5980280.

[168] M. Toussaint, N. Ratliff, J. Bohg, L. Righetti, P. Englert and S. Schaal. "Dual execution of optimized contact interaction trajectories". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 47–54. doi 10.1109/IROS.2014.6942539.

[169] N. Ratliff, M. Toussaint and S. Schaal. "Understanding the geometry of workspace obstacles in Motion Optimization". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4202–4209. doi 10.1109/ICRA.2015.7139778.

[170] S.C.A Thomopoulos and R.Y.J Tam. "An iterative solution to the inverse kinematics of robotic manipulators". In: *1991 Mechanism and Machine Theory*, pages 359–373. doi https://doi.org/10.1016/0094-114X(91)90008-R.

[171] Y. Nakamura and H. Hanafusa. "Optimal Redundancy Control of Robot Manipulators". In: *1987 The International Journal of Robotics Research*, pages 32–42. doi 10.1177/027836498700600103.

[172] M. Gienger, H. Janssen and C. Goerick. "Task-oriented whole body motion for humanoid robots". In: *5th IEEE-RAS International Conference on Humanoid Robots, 2005.* pages 238–244. doi 10.1109/ICHR.2005.1573574.

[173] E.S. Neo, K. Yokoi, S. Kajita, F. Kanehiro and K. Tanie. "A switching command-based whole-body operation method for humanoid robots". In: *2005 IEEE/ASME Transactions on Mechatronics*, pages 546–559. doi 10.1109/TMECH.2005.856112.

[174] C. Samson, B. Espiau and M.L. Borgne. Robot Control: The Task Function Approach. New York, NY, USA: Oxford University Press, Inc. doi https://doi.org/10.1017/S0263574700000643.

[175] B. Siciliano and J. -. E. Slotine. "A general framework for managing multiple tasks in highly redundant robotic systems". In: *1991 Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pages 1211–1216. doi 10.1109/ICAR.1991.240390.

[176] N. Mansard, O. Stasse, P. Evrard and A. Kheddar. “A versatile Generalized Inverted Kinematics implementation for collaborative working humanoid robots: The Stack Of Tasks”. In: *2009 International Conference on Advanced Robotics*, pages 1–6.

[177] F.S. Grassia. “Practical Parameterization of Rotations Using the Exponential Map”. In: *1998 Journal of Graphical Tools*. Natick, MA, USA: A. K. Peters, Ltd., pages 29–48. doi 10.1080/10867651.1998.10487493.

[178] J. Chen, F. Ye and T. Jiang. “Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm”. In: *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1434–1438. doi 10.1109/ICCT.2017.8359869.

[179] M. Lin, J. Xiaoming and Q. Fei. “A Robot Obstacle Avoidance Method Based on Improved Genetic Algorithm”. In: *2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 327–331. doi 10.1109/ICICTA.2018.00081.

[180] K.L. Doty, C. Melchiorri, E.M. Schwartz and C. Bonivento. “Robot manipulability”. In: pages 462–468. doi 10.1109/70.388791.

[181] J. Carpentier and N. Mansard. “Analytical Derivatives of Rigid Body Dynamics Algorithms”. In: *2018 Robotics: Science and Systems*, pages 1–9. doi 10.15607/RSS.2018.XIV.038.

[182] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock and M. Diehl. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *2014 Mathematical Programming Computation*, pages 327–363.

[183] A. Ben-Tal, L. El Ghaoui and A. Nemirovski. Robust Optimization. 10th ed. Princeton Series in Applied Mathematics. 2009 Princeton University Press.

[184] P. Dupuis, M.R. James and I. Petersen. “Robust properties of risk-sensitive control”. In: *1998 Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, pages 2365–2370 vol.2. doi 10.1109/CDC.1998.758698.

[185] A. Bensoussan, J. Frehse and H. Nagai. “Some results on risk-sensitive control with full observation”. In: *Proceedings of 1995 34th IEEE Conference on Decision and Control*, pages 1657–1661. doi 10.1109/CDC.1995.480377.

[186] A. Bensoussan and R. J. Elliott. “General finite-dimensional risk-sensitive problems and small noise limits”. In: *1996 IEEE Transactions on Automatic Control*, pages 210–215. doi 10.1109/9.481520.

[187] K. Dvijotham and E. Todorov. “A Unifying Framework for Linearly Solvable Control”. In: *2011 Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 179–186.

[188] M.R. James, J.S. Baras and L.J. Elliot. "Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems". In: *1994 IEEE Transactions on Automatic Control*, pages 780–792. doi 10.1109/9.286253.

[189] P. Whittle and J. Kuhn. "A hamiltonian formulation of risk-sensitive linear quadratic gaussian control". In: *1986 International Journal on Control*, pages 1–12. doi 10.1080/00207178608933445.

[190] P. Whittle. "Risk-Sensitive Linear/Quadratic/Gaussian Control". In: *1981 Advances in Applied Probability*. Applied Probability Trust, pages 764–777. doi 10.2307/1426972.

[191] H. Wio. Path Integrals for Stochastic Processes - An Introduction. Cantabria, Spain: 2013 World Scientific. doi https://doi.org/10.1142/8695.

[192] J.B. Rawlings and B.R. Bakshi. "Particle filtering and moving horizon estimation". In: *2006 Papers from Chemical Process Control VII: CPC VII*, pages 1529–1541. doi https://doi.org/10.1016/j.compchemeng.2006.05.031.

[193] S. Bensaid and D. Slock. "Comparison of various approaches for joint Wiener/Kalman filtering and parameter estimation with application to BASS". In: *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 2159–2163. doi 10.1109/ACSSC.2011.6190413.

[194] S. Levine and V. Koltun. "Guided Policy Search". In: *2013 Proceedings of the 30th International Conference in Machine Learning*, pages 1–9.