Robust multi-contact dynamical motion planning using contact wrench set

by

Hongkai Dai

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2016

Author .....
Department of Electrical Engineering and Computer Science
July 14, 2016

Certified by .....
Russ Tedrake
Professor
Thesis Supervisor

Accepted by .....
Prof. Leslie A. Kolodziejski
Chair, Department Committee on Graduate Theses

Robust multi-contact dynamical motion planning using contact wrench set

by

Hongkai Dai

Submitted to the Department of Electrical Engineering and Computer Science
on July 14, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

In this thesis, we seek to plan a robust motion for robot with multiple non-coplanar contact on the environment. When the robot interacts with the environment through contact, it relies on the contact forces to generate the desired acceleration. The contact forces have to satisfy some physical constraints, such as lying within the friction cones. These constraints limit the robot acceleration. The robustness of the motion can be measured as the margin to the boundary of these constraints. By planning motion with a large preserved margin, we enable the robot to withstand large disturbance in the online motion execution.

In this thesis, we adopt the notion of contact wrench set to approximate the constraints on the robot dynamics. The margin of such set measures the capability of the motion to perfectly resist external wrench disturbance. We plan robust motion to increase this contact wrench set margin.

We present two planners to improve this robustness metric. For the first simple-model planner, we pre-specify the contact locations, and it generates a Center of Mass trajectory and an angular momentum trajectory, by solving a convex optimization problem. We show that this planner has similar output as the widely-used walking pattern generator that relies on Zero Moment Point (ZMP) on flat ground. Moreover, it can plan feasible motion on uneven ground with friction cone limits, while the ZMP planner fails.

For the second planner with robot whole-body model, we will search for the contact location and the robot whole-body motion simultaneously. We show that we can improve the robustness metric through certain non-convex optimization techniques. We apply our planner to three problems: 1) force closure grasp optimization, 2) static posture optimization, 3) trajectory optimization, achieving improved performance for all of them.

Thesis Supervisor: Russ Tedrake
Title: Professor

Acknowledgments

I have spent six incredible years in MIT, with the kind help and support from many people. I am extremely grateful to my advisor Russ Tedrake, for giving me this opportunity to work in one of the world's best place to do robotics research. Russ has always been super insightful in research. I lost count the number of times that he almost immediately pointed out possible improvements on my work, after just a quick discussion on the progress. Russ is also an amazing leader, he always encourages us with his trademark optimism and patience. I remember even during the competition of DARPA Robotics Challenge, when everyone of us got exhausted, still Russ never lost his warm smile when we talked, and kept the whole team in a high spirit. Russ cares his students not only professionally but also personally. I started running because Russ created the locomotion-runner mailing list, and led us to run along the Green line; now running becomes a habit.

I also want to thank the members of my thesis committee Prof. Emilio Frazzoli, Tomás Lozono-Pérez and Alberto Rodriguez, for their suggestions and directions to help make this thesis better. I want to thank Alberto for his guidance when I started to learn grasp optimization. Alberto wrote a long email elaborating on the previous related work in the grasping community, to help me familiarize with this topic.

I have been very fortunate to collaborate with Andrés Valenzuela and Anirudha Majumdar. Andrés and I spent a great amount of time writing code together during DRC. We sat on the second floor of the N9 warehouse, and built the inverse kinematics engine for Atlas robot together, during the hot summer that year. I learned a great deal of coding skills from Andrés through pair programming. I want to thank Ani for introducing me the idea of bilinear matrix inequality, which becomes the mathematical basis for our approach in grasp optimization. Ani has been incredibly sharp and patient, and he masters many tricks to call sum of squares solvers. There were many times that just by glancing the print out message from the solver, Ani could immediately spot the numerical pitfalls of our optimization problems, which had bothered me for days. Also, Ani is an amazing pianist, it was extremely enjoyable to listen to

his soothing rhythm, after a long day of work.

I want to thank the members of Robot Locomotion Group, for making the nose of Stata center such an enjoyable place to work at. I want to thank Andy Barry for introducing me the software pdfpc, and for his effort to maintain this software. It makes life much easier to embed video into beamer slides. I also want to thank him for visiting me and my family in my hometown, I hope he enjoyed the spicy food there, and forgave me for the “crying noodle" I bought for him. I want to thank Robin Deits for his humor, and for his patience when I came to him talking about half-baked ideas on motion planning. I want to thank John Carter, Greg Izatt, Geronimo Mirano and Alexander Chen for helping set up the ABB arm. I want to thank Pete Florence for teaching me how to do work out in a safe way. I want to thank Twan Koolen for teaching me the basics on robot dynamics, such as wrench, twist, geometric Jacobian, etc. I want to thank Lucas Manuelli for always asking me about the big picture and motivation of my approach, and for many discussions we had on the white board, which helped clarify my ideas. I want to thank Pat Marion for creating the awesome Director, and answering my questions in tremendous details about how to use this tool for visualization. I want to thank Frank Permenter for helping check the SOS conditions, and creating the facial reduction tool that simplifies a lot of my SOS problems. I want to thank Michael Posa for teaching me Lyapunov verification for hybrid systems. I want to thank Shen Shen for her jokes, and for helping proofread the thesis. I want to also thank Kathy Bates, Mieke Moran and Bryt Bradley for making the group functional.

I had the privilege to work in the MIT DRC team for almost three years. This is probably a once per life experience, to work on one of the world’s best humanoid with a group of top-notch roboticists for a competition. I want to thank the team leaders Russ Tedrake and Seth Teller giving me this opportunity to join the team. I want to thank the whole team for the devotion, the engineering, and those sleep-deprived nights we spent together in the N9 warehouse or bullpen before the competition.

Foremost, I want to thank Jiahui, my wife, and my best friend. It has been a long journey since the first day we held our hands, thank you for always being with me in

all these years, even we were geographically separated by the Pacific Ocean, the US Continent, or the Atlantic Ocean. I also want to thank my family, my parents, for their unconditional support and endless love.

Contents

List of Figures

List of Tables

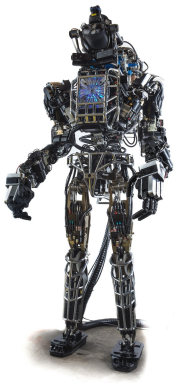
Chapter 1

Introduction

We have observed fast progress in robotics in recent years. For example, in DARPA Robotics Challenge, the human pilot can remotely control the robot to perform a variety of tasks, such as driving a car, getting off a car, openning a door, turning a valve, etc. All these tasks require the robot to make contact with the environment. One lesson we learned from the challenge, is that robustness is the key to success.

There are many aspects of robustness for robot motion. For example, for robots that do not get in contact with the environment, such as drones and underwater vehicles, a robust motion is the one such that the robot can dodge obstacles even under uncertainties. This "avoiding contact" problem can be analyzed through a geometric approach, by studying the distance to collision under uncertainties [107, 105]. On the other hand, for robots interacting with the environment through contacts, such as the Atlas robot (Fig.1-1a) or Little Dog (Fig.1-1b), they can only move their Center of Mass by controlling the contact forces, thus the robot has to exploit contacts to control its dynamics. As a result, the robustness of the motion is not just a geometric problem anymore, but rather a dynamical problem depending on the contact conditions [104, 142, 169, 23].

For dynamical systems, some of the robustness metrics, such as L_2 gain, require looking into the future dynamics, to evaluate the disturbance effect in the long horizon [180, 179, 27]. Such robustness metrics can be computational intractable, for complicated systems with nonlinear dynamics, such as humanoid robots. Instead in



(a) Atlas from Boston Dynamics Inc

(b) Little Dog from Boston Dynamics Inc

this thesis, we will consider robustness metric that does not involve integration the future dynamics, but focuses on each individual time instance separately. This allows us to handle complicated robots with nonlinear dynamics.

For a robot moving in the environment, at every time instance, the contacts impose constraints on the robot dynamics. For example, when walking on the flat ground, the normal contact force has to point upward, so the robot Center of Mass acceleration cannot be larger than the gravitational acceleration in the downward direction, so the acceleration of the robot has to be kept within a constraint set. If the acceleration falls outside this constraint set, then the robot cannot maintain its current contact condition, but instead loses contact by either tipping over or sliding. This could cause undesirable behaviors, such as the robot losing balance and falling down. To this end it is desirable to preserve some margin to the boundary of that set when planning the motion, such that the robot can instantaneously endure certain disturbance, without leaving that constrained set. In this thesis, we will compute an approximation of such set explicitly, as well as the margin of this constraint set. We will maximize this margin through optimization to achieve robustness.

1.1 Contribution

In this thesis, we present novel optimization-based approaches to plan a robust motion for a robot making multiple non-planar contacts with the environment. There are two distinct approaches to improve the robustness of a multi-contact robot and to

Planner	Contact locations	Time	Kinematics	Optimization type
1	pre-specified	pre-specified	simple Center of Mass model	convex
2	optimizing	optimizing	whole-body model	non-convex

Table 1.1: Comparison on two planners

prevent it from falling under external disturbances. One approach is to quickly change the contact location by moving the feet or the hands [153, 136, 87], this approach is computationally challenging, as it needs to predict effect on the future dynamics by changing the contact condition; the other approach is not to move the contact location, but instead attempt to stick to the planned contact location while moving the robot limbs and torso, as commonly used in ZMP-type walking [78, 145]. Compared to changing contact location, the second approach is more conservative, but computationally much more tractable. Here we will use this more conservative approach. We regard a planned motion being more robust if it is less vulnerable to losing planned contact with the environment. We adopt the idea of Contact Wrench Set (CWS) margin [5, 66] as the robustness metric, which measures the smallest magnitude of the wrench disturbance, that would cause the robot to lose contact.

We formulate two different planners to improve this robustness metric through numerical optimization, and we highlight the features of each planner in Table.1.1. To the best of our knowledge, this is the first work to optimize robust motion, aiming to improve the contact wrench set margin.

The first planner generates a center of mass trajectory for pre-specified contact locations, as shown in Fig. 1-2. The objective of this planner is to maximize the margin, and minimize the centroidal angular momentum during walking. Such walking pattern is generated through solving a convex optimization problem. We compare the results with the motion generated using the popular Zero Moment Point (ZMP) criteria. We will show that this planner generates similar motion as the ZMP planner on the flat ground; moreover, on an unevern terrain course with friction cone constraints, this planner can generate feasible motion, while the motion coming out from ZMP planner violates the friction cone constraints.

The second planner searches for the contact locations and the robot whole-body

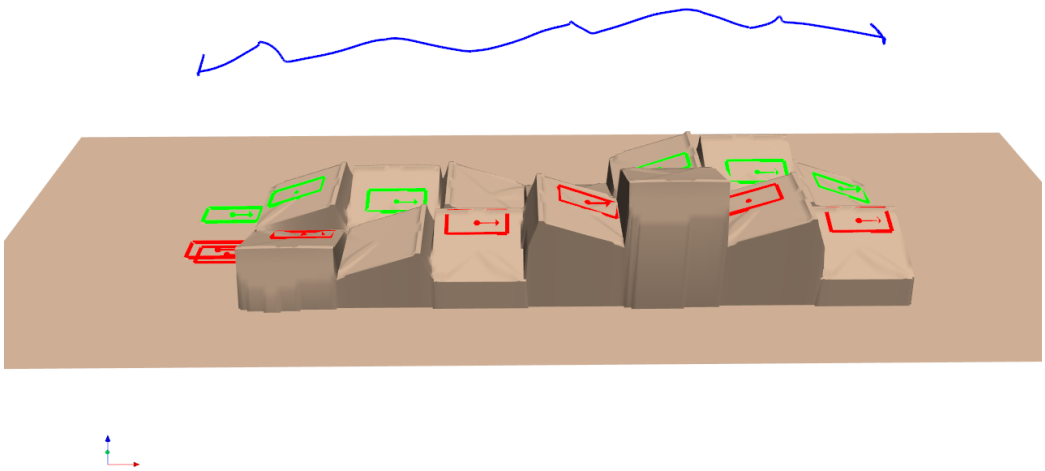


Figure 1-2: Center of Mass trajectory and pre-specified footsteps on tilted blocks

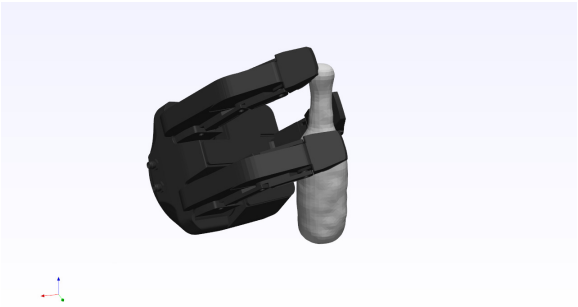
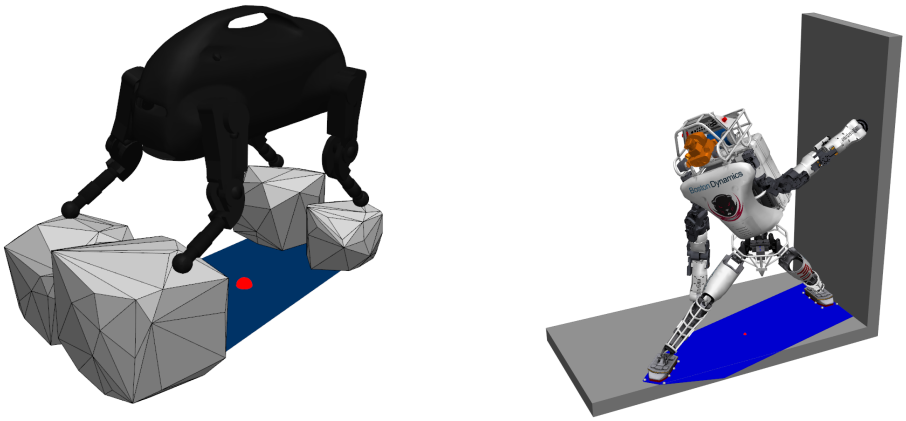


Figure 1-3: Grasp optimization

motion simultaneously, while maximizing the robustness margin. Unlike in the first planner case, in which we pre-specify the contact locations, the problem becomes non-convex when we are optimizing over the contact locations. We show that we can either solve this non-convex problem as a nonlinear optimization problem, or through sequential semidefinite programming. We highlight three applications of this planner:

- To optimize a force closure grasp (Fig.1-3).
- To find a robust posture in static equilibrium (Fig.1-4).
- To optimize a trajectory with with non-planar multi-contact locations (Fig.1-5).



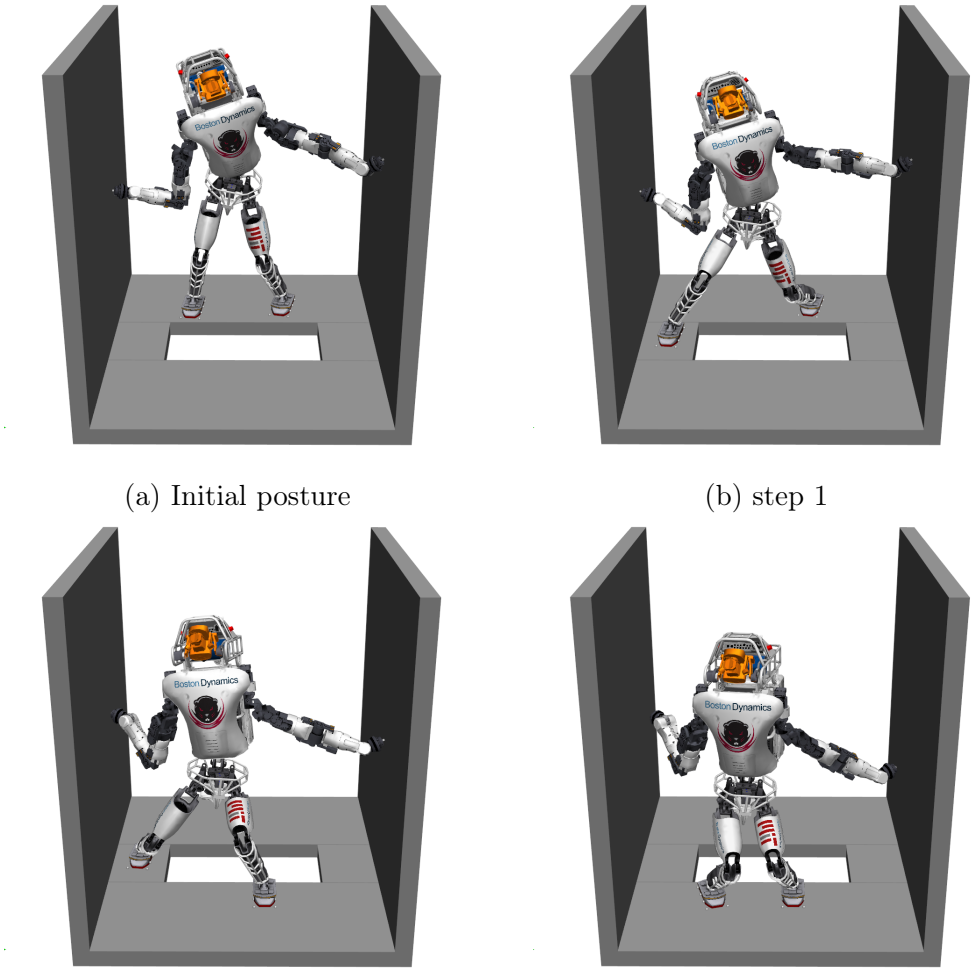
(a) Static Posture for Little Dog on stepping stones. The blue region is the support region for Center of Mass projection (red dot).

(b) Static Posture for Atlas with foot and left hand contact. The robot stretches its legs and arms to maximize the support region (blue).

Figure 1-4: Static posture optimization to maximize the support region

1.2 Outline

In Chapter 2, we review the related work on planning robust motion with contact, in walking and grasping community. In Chapter 3, we explain the Contact Wrench Set margin idea developed by previous researchers. We will use this margin as the robustness metric in the thesis. In Chapter 4, we describe a motion planner that can optimize this robustness metric when the contact locations are pre-specified. We show that our planner generates similar Center of Mass trajectory as the ZMP planner on the flat ground, and it can successfully plan the CoM motion on uneven ground, when the ZMP planner fails. In Chapter 5, we present a planner that can search over the contact locations and robot motion simultaneously, so as to optimize the robustness margin. We first demonstrate how to compute such metric when the contact locations are not given, then we adopt two approaches to optimize such metric, one through nonlinear optimization, and the other formulation as a sequential semidefinite programming. We demonstrate the application of this planner on three problems, the force closure grasp optimization, the static posture optimization, and trajectory optimization. We conclude the thesis, and discuss the possible future



(a) Initial posture

(b) step 1

(c) step 2

(d) Final posture

Figure 1-5: A snapshot of a trajectory that Atlas walks over a trap with hands and foot contact

extensions in Chapter 7.

Chapter 2

Related Work

There has been a lot of research in planning motion with multiple contacts, for a variety of robots; including humanoids [80, 78, 89, 59, 60, 115, 36], multi-leg robots [139, 17, 152, 139, 46, 127, 81, 171], and robot hands [118, 119, 140] . We will introduce some stability/robustness metrics used in these applications, including Zero Moment Point (ZMP) for humanoid walking, ϵ-ball for hand grasping, support region for multi-leg robot maintaining static posture, and finally the contact wrench set margin that extends the previous three metrics. We will then briefly introduce some motion planning techniques, including trajectory optimization and sample-based motion planning.

2.1 Robustness metric

2.1.1 Zero Moment Point

Proposed by Vukobratovic more than four decades before [166, 165], Zero Moment Point (ZMP) criteria is the most widely used stability/robustness metric for humanoid robots [82, 22, 65, 42]. It guarantees that for a robot walking on flat ground with large enough friction, the foot will not tip over if the Center of Pressure from the ground reaction force lies within the convex hull of the foot contact region. The Center of Pressure coincides with the Zero Moment Point (ZMP), namely the point at which

the moments around the horizontal axes are zero [148]. In order to keep the foot on the ground without tipping over, the robot can either control its Center of Mass motion, or use some angular momentum, to make sure the Zero Moment Point lies within the convex hull of foot region.

Zero Moment Point can also be used to measure the robustness of walking motion on flat ground. When the ZMP is far away from the boundary of the convex hull of foot contact region, the robot can resist large wrench disturbance without tipping over; or equivalently the robot can withstand large variation in its acceleration. Thus it is desirable to control the robot acceleration, so as to keep its ZMP close to the center of the convex hull of foot contact region, and the distance to the center of the foot region is used as a robustness metric to external wrench disturbance [78, 156, 95, 94].

There exist abundant research to plan robot motion based on ZMP [166, 61]. In their seminal work [80], Kajita et al. proposed a linear inverted pendulum model, such that the Zero Moment Point can be computed as a linear function from the Center of Mass (CoM) dynamics. This clever formulation opens the gate to fast planning of CoM motion using ZMP as the stability criteria [78, 170, 35, 34, 156]. In Chapter 4 we will present a planner that can also quickly compute the CoM motion using a different robustness metric, on situations where ZMP planner fails.

Though conceptually simple, ZMP has some major limitations. It is ideal for flat ground walking with unconstrained horizontal friction forces. However, when the robot walks on uneven ground, the two feet are in different surfaces. It does not hold any more that the ZMP should lie within the convex hull of the foot region. Moreover, without bounds on the tangential friction forces, ZMP ignores the friction cone constraints. So on a slippery ground, even if the ZMP lies within the convex hull of foot contact region on the flat ground, the feet can still slide on the ground if the robot requires too large horizontal tangential force to accelerate. Thus ZMP criteria is broken for walking on uneven terrain, subject to friction cone constraints. Also ZMP is not suitable for motion requiring hand and foot contact coordination, such as climbing a ladder with hand rails [89, 104, 91], or pushing against a wall [142]. There has been some work to extend ZMP to non-coplanar contacts [149], or with friction

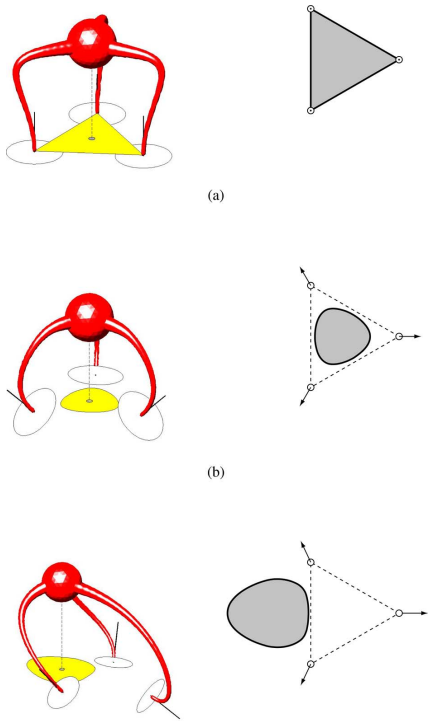


Figure 2-1: Support region of a 3-leg robot, by Bretl [17]

coefficient taken into consideration [79], but inherently ZMP lacks the ability to fully capture the dynamic effect of non-coplanar contacts with friction cone constraints. So we seek for other stability criteria that can readily tackle more general cases than ZMP can.

2.1.2 Support region for static equilibrium

For robot in static posture, it is widely known in order to maintain static equilibrium, the projection of Center of Mass should lie within the foot support region [108, 17, 59]. The support region is a two-dimensional region on the ground; it coincides with the convex hull of the foot contact region if the contacts are all on the flat ground, but can be significantly deviant from the foot contact region, when the contacts are on uneven ground, as shown in Fig. 2-1.

While support region can incorporate non-coplanar contacts with friction cone constraints, it assumes that the robot is in static equilibrium with no linear or angular

acceleration, thus it is not suitable for dynamic motion. We will show later that the support region is a special case for the contact wrench set criteria that we will introduce in Chapter 3.

When the contact locations are pre-specified, computing the support region is formulated as a convex optimization problem, which could be solved efficiently. On the other hand, if we need to search for the contact locations, the problem becomes non-convex, thus a lot harder than the pre-specified contact case. In this thesis, we will demonstrate how we can search for the contact locations through numerical optimization.

2.1.3 ϵ-ball in force closure grasp

Grasping inherently involves multiple non-coplanar contacts, with friction cone constraints at contact locations. Among the various type of grasps, force closure grasp measures the robustness to external wrench disturbance. A grasp achieves force closure if it can resist arbitrary external wrench disturbances, with contact forces within the friction cone at each contact location [43, 119, 129, 118].

A commonly used metric to measure the quality of a force closure grasp is called ϵ-ball, which measures the magnitude of the smallest wrench disturbance that the grasp cannot resist, with bounded total contact forces [43, 85, 112, 14, 141].

The ϵ-ball metric handles multiple contact with friction cone constraints. Although initially used for grasping in static equilibrium, it could also handle non-static case, by adding a "task wrench" to the analysis [102, 58]. We will see that the contact wrench set margin metric, which we will use primarily in this thesis, is equivalent to the ϵ-ball for grasping with motion.

When the contact locations are given, this metric can be readily computed through a convex optimization problem [54, 15, 56, 55]. However if we need to search for the contact locations in order to maximize the ϵ-ball, the problem becomes non-convex, which was solved previously through gradient-based nonlinear optimization approach, with carefully defined sub-gradient [103]. We will show that our approach can search over the contact locations with smooth gradient, instead of sub-gradient.

2.1.4 Contact wrench set margin

Contact Wrench Set (CWS) criteria is the conceptual generalization of the aforementioned three criteria. It was proposed by the walking community, to determine feasibility of dynamic motion, with multiple non-coplanar contact and friction cone constraints [169, 67]. The contact wrench set is the allowable set of total wrench aggregated from contact forces at each individual contact location. The CWS criteria states that the total contact wrench, computed from the robot motion, should lie with the contact wrench set [169]. Hirukawa et al. proved that the Contact Wrench Set criteria is equivalent to Zero Moment Point criteria on the flat ground with infinite friction forces, thus they call CWS as *Adios-ZMP* [67]. We will give a mathematical formulation of the contact wrench set in Section 3.2.

There has been a lot of work to plan a feasible motion using the contact wrench set criteria, to determine whether or not the contact wrench lies within the contact wrench set. Herzog et al. planned the robot total linear and angular momentum trajectory with pre-defined contact locations [64]. Escande et al. presented a planner that generates the robot whole-body motion with pre-defined contact locations [38]. Dai et al. used nonlinear optimization to plan the whole-body motion together with the contact location at the same time, by considering the contact wrench set at the Center of Mass coordinate [28]. In [20]. Caron et al. showed they can plan a time-optimal trajectory with pre-defined contact locations and fixed whole-body postures, through convex optimization.

Apart from using contact wrench set as the feasibility criteria, we can also leverage it to measure the robustness of the motion, by defining the margin of the set. The contact wrench set margin metric is directly inspired by the ϵ-ball idea in grasping literature, defined as the smallest wrench disturbance that the robot cannot resist, without losing contact [5]; thus the CWS margin measures the robustness of a motion to the external wrench disturbance. Barthelemy et al. showed how to compute the contact wrench set margin for pre-specified linearized friction cone of a given motion. Caron et al. [20] demonstrated how to test if a static posture has a desired margin

in the contact wrench set. We will present the formulation on contact wrench set margin in Section 3.4.

The aforementioned previous works either use contact wrench set to plan a feasible motion, or to compute the contact wrench set margin for a given motion, without optimizing the margin. In this thesis, we will plan robust motion and/or the contact locations, to increase this margin. To the best of our knowledge, this is the first work that attempts to optimize the motion using this robustness metric.

It is worth mentioning that the contact wrench set criteria is only a necessary condition on the feasibility of a motion[169, 142, 28]. Apart from the friction cone constraints, the robot dynamics is also constrained by the actuator torque limits. When computing the contact wrench set, we assume that any contact force within the friction cone can be applied on the robot. On the other hand, the robot cannot sustain large contact force without exceeding its actuator torque limits. Thus by ignoring the joint torque limits, the contact wrench set criteria is not a sufficient condition for dynamic feasibility. Fortunately, we notice that for many robots such as Atlas, the motor are strong enough, such that it is reasonably to ignore the torque limits to simplify the planning problem.

2.2 Motion Planning

There are two dominating approaches in motion planning, the sample-based approach, and the optimization-based approach. We give a brief overview of the two approaches here.

2.2.1 Sample-based motion planning

In sample-based motion planning, the algorithm first draws samples in the robot sample space, then efficiently determines if the sample is feasible, and connects the feasible samples to generate a path for the robot [100]. Usually a secondary smoothing algorithm should be applied to achieve continuous motion [172]. Some popular sample-based motion planning techniques, such as Probabilistic Roadmap

(PRM) [3, 83, 68] or Rapidly-exploring Random Tree (RRT) [98, 97, 99], guarantee to find a path through narrow passages with probabilistic completeness. The sample-based approach has been applied to a variety of robots, such as arm manipulators [154, 31, 9, 123], humanoids [92, 93, 160], quadruped robots [152, 150] or hands [8, 32]. Although widely used, the motion generated from sample-based approach is usually not smooth enough. In this thesis, we will adopt the optimized-based approach.

2.2.2 Optimization-based motion planning

In optimization-based motion planning, we cast the planning problem as an optimization problem. In this optimization problem, the decision variables are the finite number of parameters of the robot motion (for example, the velocity and postures at some discretized knots); the constraints in the optimization include the physical requirement on the robot, such as kinematics and/or dynamics equations, collision-free condition; the objective of the optimization encodes the robot desired behaviors, such as minimum energy or maximum robustness. The optimization problem is then solved through a numerical procedure [12, 13, 125, 164, 33]. It is possible to synthesize smooth dynamical motions for an arsenal of robots, thanks to the fast progress of numerical solvers [47, 77, 19, 167]. Examples include one-link spacecrafts [57, 37], muti-joint robots such as mobile manipulators [126], humanoids [142, 86, 28, 73, 173, 151, 64, 62], running robots [132, 131, 113, 26, 27, 51, 69, 163, 143] and multi-finger hands [131, 114, 96, 25].

For a simple robot model, such as a point-mass model, or just the feet of the robot [29], the optimization problem can be convex, and can be solved very efficiently [16]. For a more complicated robot model, the optimization problem generally becomes non-convex and nonlinear, thus computationally more difficult than the convex one. For example, when the whole-body kinematics model of a multi-link robot is considered, the constraints are generally non-convex, due to the nonlinear structure of the kinematics, especially the link orientation [39, 28, 151]. The non-convex optimization procedure can end up with local minima or infeasibility [10]. To overcome the local infeasibilty problem, in Chapter 5, we will use a sequential semidefinite programming

approach to solve the non-convex nonlinear problem. This approach can generate a global infeasibility certificate. In this thesis, we will show that we can search for the motion of a simple model using convex optimization in Chapter 4, and of a whole-body model with full kinematics using non-convex optimization in Chapter 5.

Chapter 3

Contact Wrench Set

In this chapter, we give a detailed explanation on the contact wrench set notion and its margin,proposed by previous researchers [169, 67, 5, 89, 178]. The Contact Wrench Set (CWS) is introduced to determine if the motion is dynamically feasible with multiple contacts and friction cone constraints [169]. This notion is a generalization of the popular Zero Moment Point idea, and it is called “Adios-ZMP" by some researchers [67]. Apart from using Contact wrench set as a necessary condition to determine the feasibility of a motion, researchers also use its margin to measure the robustness of the motion, to external wrench disturbance [5, 20].

We will focus on the situation when the robot has multiple non-coplanar contact with the environment, as shown in Fig.3-1, with the set of allowable contact force drawn at each contact point.

3.1 Contact force model

In this paper, we consider the Coulomb friction model, that the contact force stays within a cone. Both the nonlinear friction cone and the linearized friction cone will be used in this thesis (Fig.3-2). For each friction cone, we use the unit length vector c to represent the direction of the normal force. When using the nonlinear friction

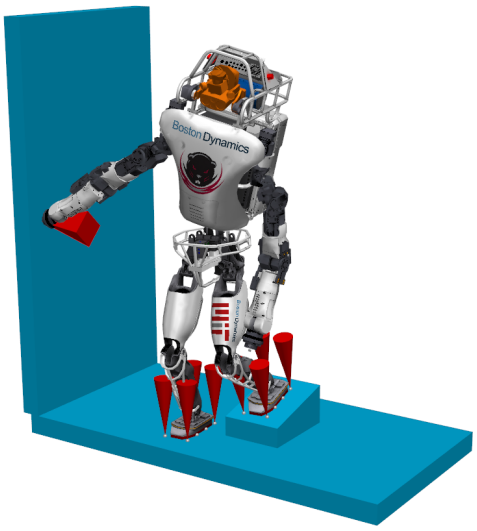


Figure 3-1: Atlas with multiple non-coplanar contacts.

cone model with friction coefficient μ, the constraint on the contact force f is that

$$f \in \mathcal{FC} = \{f : \sqrt{\mu^2 + 1} f^T c \geq |f|\} \tag{3.1}$$

When the friction cone $\mathcal{FC}$ is fixed, Eq.(3.1) is a *second-order cone* constraint on contact force f, which is a special type of convex constraint [2].

When using the linearized friction cone, with edges $e_1, \ldots, e_{n_e}$, the constraint on the contact force is

$$f \in \mathcal{FC} = \text{ConvexCone}(e_1, \ldots, e_{n_e}) \tag{3.2}$$

where $\text{ConvexCone}(x_1, \ldots, x_n)$ is the convex cone defined as $\text{ConvexCone}(x_1, \ldots, x_n) = \{\sum_{i=1}^{n} \lambda_i x_i, \lambda_i \geq 0\}$. Eq.(3.2) is a linear constraint on the contact force f.

When the robot hand or finger is in contact with the environment, to protect the brittle fingers, we constrain the magnitude of the contact force. To this end, we use a bounded convex polytope as the constrained set for hand contact force, as shown

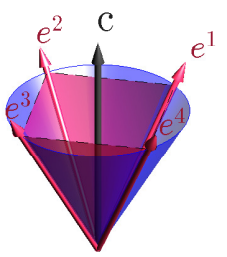


Figure 3-2: A nonlinear friction cone (blue) and the linearized friction cone (red). The unit length cone axis c is the direction of the normal contact force. $e_1, \ldots, e_4$ are the four edges of the linearized friction cone.

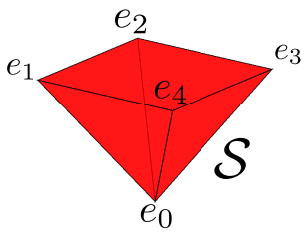


Figure 3-3: The contact force stays within the polytope $\mathcal{S}$, with vertices $e_0, \ldots, e_4$

in Fig 3-3. And the constraint on the contact force is that

$$f \in \mathcal{S} = \text{ConvexHull}(e_0, \ldots, e_{n_e}) \tag{3.3}$$

where e_i is the i'th vertex of the grasp polytope $\mathcal{S}$. $e_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. $\text{ConvexHull}(x_1, \ldots, x_n)$ is the convex hull set $\text{ConvexHull}(x_1, \ldots, x_n) = \{\sum_{i=1}^n \lambda_i x_i, \lambda_i \geq 0, \sum_{i=1}^n \lambda_i \leq 1\}$.

Although we only discuss the Coulomb friction model on contact forces in this thesis, our approach can be readily applied to other contact models, that permits torque being applied at contact. As long as the constraint set is either a cone, or a bounded polytope that contains the wrench being zero.

3.2 Computing Contact Wrench Set

To compute the Contact Wrench Set (CWS) from individual contact force constraint set, we first compute the wrench set for each individual contact. To do so, we choose some fixed coordinate system, and we compute the contact wrench w from contact

force f and contact location p

$$w = \begin{bmatrix} f \\ p \times f \end{bmatrix} \tag{3.4}$$

$\times$ is the cross product.

We can compute the set of wrenches subject to the contact force constraint in section 3.1. If we use the linearized friction cone (Eq.(3.2)), the set of contact wrench from that single contact force is a linear cone $\mathcal{K}$

$$w \in \mathcal{K} = \text{ConvexCone}\left(\begin{bmatrix} e_1 \\ p \times e_1 \end{bmatrix}, \ldots, \begin{bmatrix} e_{n_e} \\ p \times e_{n_e} \end{bmatrix}\right) \tag{3.5}$$

When we use the nonlinear friction cone (Eq.(3.1)), the set of contact wrench for that single contact force is a nonlinear cone $\mathcal{K}$

$$w \in \mathcal{K} = \left\{ w \middle| w = \begin{bmatrix} f \\ p \times f \end{bmatrix}, \sqrt{\mu^2 + 1} f^T c \geq |f| \right\} \tag{3.6}$$

When we use the polytope constraint to bound the magnitude of the contact force as in Eq.(3.3), the set of contact wrench from this single contact force is a polytope set $\mathcal{P}$ in the wrench space

$$w \in \mathcal{P} = \text{ConvexHull}\left(\begin{bmatrix} e_0 \\ p \times e_0 \end{bmatrix}, \ldots, \begin{bmatrix} e_{n_e} \\ p \times e_{n_e} \end{bmatrix}\right) \tag{3.7}$$

Suppose the robot is in contact with the environment, with n_1 of those contact locations subject to friction cone constraint, and n_2 of those locations subject to bounded polytope constraint on the contact force, we can first compute the wrench set for each individual contact locations using Eq.(3.5),(3.6),(3.7); the aggregated contact wrench should lie within the Contact Wrench Set (CWS), as the *Minkowski*

sum of each individual wrench set

$$CWS = \mathcal{K}_1 \oplus \ldots \oplus \mathcal{K}_{n_1} \oplus \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2} \tag{3.8}$$

where $\mathcal{A} \oplus \mathcal{B}$ is the Minkowski sum of sets $\mathcal{A}, \mathcal{B}$, such that $\mathcal{A} \oplus \mathcal{B} = \{a+b | a \in \mathcal{A}, b \in \mathcal{B}\}$.

It is worth mentioning here that the Contact Wrench Set is a convex set once the contact locations are fixed, since it is the Minkowski sum of several convex sets. We will use this convexity property in Chap.5.

3.3 Stability Criteria

Contact Wrench Set can be used as an approximated universal stability criteria to determine if a motion is feasible or not [67]. When interacting with the environment, the robot dynamics is only affected by the contact wrench from each contact location, and the gravitational wrench. As a result, the total wrench applied on the robot is the sum of the gravitational wrench and the contact wrench. According to Newton's law, the robot rate of momentum equals to the total wrench, thus we have the condition (3.9)

$$\dot{h} - w_g \in CWS \tag{3.9}$$

where $h \in \mathbb{R}^6$ is the robot total momentum, computed from the robot state [41, 122, 121, 168]. The momentum h includes both the linear momentum $l \in \mathbb{R}^3$ and the angular momentum $k \in \mathbb{R}^3$, $h = [l^T \ k^T]^T$. The gravitational wrench w_g is computed from the robot Center of Mass (CoM) position r

$$w_g = \begin{bmatrix} m\mathbf{g} \\ r \times m\mathbf{g} \end{bmatrix} \tag{3.10}$$

where $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration.

Notice that the condition (3.9) is just a necessary condition for the motion to be

dynamically feasible. Apart from the contact force constraints, the robot dynamics is also subject to actuator torque limits. Thus a contact wrench within the contact wrench set might not be feasible, since corresponding acceleration $\dot{h}$ could require actuator torques exceeding the torque limits. On the other hand, torque limits are the only constraints the motion might violate, when the motion satisfy the contact wrench set criteria in Eq.(3.9). The reason is that for the robot we analyze in this thesis, there is always one actuator for each joint of the robot. Since all joints are fully-actuated, the only un-actuated degrees of freedom (DOF) in the robot, is its 6-DOF floating base. The constraints on the 6 un-actuated DOF is equivalent to Eq.(3.9), that depends on external contact force, not the joint torques. Thus once the contact wrench is within the contact wrench set, we can always compute the joint torques using the inverse dynamics, although the torques might not satisfy the joint torque limits. Fortunately many robots, such as Atlas, have very powerful actuators. Thus it is reasonable to ignore the torque limits in planning for many cases. We can always add the joint torque limits back to the planning problem, after we get a first solution that ignores the torque limits.

The Contact Wrench Set is a generalization of Zero Moment Point, to non-coplanar multi-contact case. It is equivalent to Zero Moment Point when the robot walks on flat ground, with friction coefficient being infinite [67]. In such case, the Contact Wrench Set is unrestricted in the horizontal forces and vertical torque axes. We can restrict our discussion to the remaining three axes of vertical force and horizontal torques, and the constraint set becomes a three dimensional cone. It can be shown that a slice of this cone is the same as the support region of the foot, after scaling and rotation, and the conventional ZMP stability criteria that ZMP stays within the support region, is the same as saying that the desired contact wrench $\dot{h} - w_g$ stays within the Contact Wrench Set [67], as shown in Fig. 3-4.

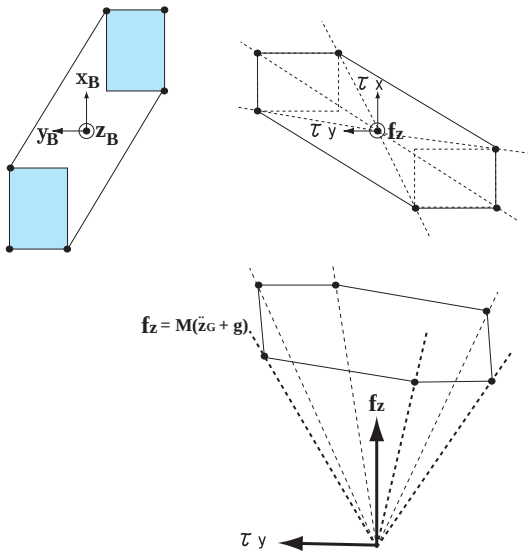


Figure 3-4: The slicing of the Contact Wrench Set is the same as the support region after scaling and rotation. ZMP lies within the support region is the same as desired contact wrench $\dot{h} - w_g$ stays within the Contact Wrench Set. [67]

3.4 Contact Wrench Set Margin

The Contact Wrench Set can be used to determine not only the feasibility of the motion, but also how robust the motion is. In the flat ground walking case, people plan the desired ZMP to be away from the boundary of the support region, so as to prevent the foot from slipping [78, 156]. In force closure grasping, the robustness of the grasp is measured by the smallest wrench disturbance that the grasp cannot resist with bounded contact forces, called ϵ-ball [85, 112]. The contact wrench set margin is directly inspired by the ϵ-ball used in the grasping literature, and it is defined as follows, as proposed by Barthelemy et al. in [5]

Definition 1. *Contact Wrench Set Margin is the smallest magnitude of the wrench disturbance, that the robot cannot resist, given the contact locations and contact force constraint.*

Algebraically, we define the contact wrench set margin as the maximum value of ϵ, such that the contact wrench plus the disturbance wrench is still within the Contact

Wrench Set, as formulated in the following condition:

$$\mathcal{B}_\epsilon = \{\dot{h} - w_g + T(p_w, I_{3\times3})w | w^T Q_w w \leq \epsilon^2\} \subset CWS \tag{3.11}$$

where $T(p_w, I_{3\times3}) \in \mathbb{R}^{6\times6}$ is a transformation matrix, that transforms a wrench disturbance being applied at a point p_w, to the equivalent wrench in the origin of the world coordinate. The parameterization of such transformation matrix, with coordinate translation $x \in \mathbb{R}^3$ and rotation matrix $R \in so(3)$ is

$$T(x, R) = \begin{bmatrix} R & 0_{3\times3} \\ \lfloor x \rfloor_\times R & R \end{bmatrix} \tag{3.12}$$

where $\lfloor x \rfloor_\times \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix representing the cross product

$$\lfloor x \rfloor_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.13}$$

Namely, for all wrench disturbance w, satisfying its weighted 2-norm $w^T Q_w w \leq \epsilon^2$, when such wrench disturbance is applied at point p_w, the robot can perfectly resist such disturbance with some appropriate contact forces.

Geometrically, the contact wrench set margin is illustrated in Fig.3-5. The contact wrench set margin is the radius of the largest ellipsoid centered at the desired wrench $\dot{h} - w_g$, and being contained in the Contact Wrench Set.

It is worth mentioning here that the value of the contact wrench margin does not depend on the choice of the coordinate frame. To see this, it is straightforward that for any wrench transformation matrix $T(x, R)$, we always have

$$\dot{h} - w_g + T(p_w, I)w \in CWS \Leftrightarrow T(x, R)(\dot{h} - w_g + T(p_w, I)w) \in T(x, R)CWS \tag{3.14}$$

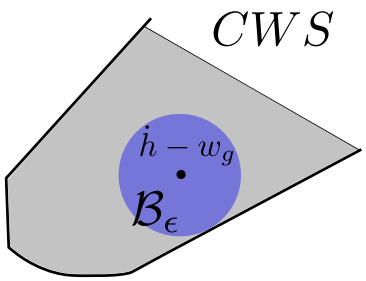


Figure 3-5: Contact Wrench Set Margin

where

$$T(x,R)CWS = \{T(x,R)w | w \in CWS\} \tag{3.15}$$

$T(x,R)CWS$ means to transform the Contact Wrench Set under linear mapping $T(x,R)$. Namely, Eq.(3.14) asserts that the transformed wrench is inside the Contact Wrench Set after transformation, if it is inside the Contact Wrench Set before transformation. On the other hand, the value of the contact wrench set will depend on the point p_w where the wrench disturbance is applied. In this thesis, we either pre-specify the location of p_w, or set it to be equal to some point on the robot, for example, the Center of Mass location.

Our goal is to find the motion of the robot, and possibly the contact locations, such that the contact wrench set margin is maximized. i.e., the robot can resist larger wrench disturbance while still perfectly tracking the desired motion.

Chapter 4

Planning with given contacts

4.1 Introduction

In this chapter, we will present our planner that generates robot Center of Mass trajectory and angular momentum trajectory, with given contact locations. Our goal is to maximize the contact wrench set margin to make the motion robust, and to minimize the centroidal angular momentum to make the motion natural. We formulate this problem as a convex optimization problem to solve it efficiently. Contrary to the prevailing Zero Moment Point approach that works best on flat ground with infinite friction coefficient, we show our approach can work on uneven terrain, and can plan feasible trajectory where the ZMP approach would fail.

4.2 Approach

4.2.1 Time integration

To plan a trajectory, we first discretize it by taking N time samples, and we denote the time interval between the $i'th$ time knot and the $i+1'th$ time knot as $dt[i]$, these time steps are given, thus not part of the decision variables. We aim to find the Center of Mass location r, the velocity $\dot{r}$, the acceleration $\ddot{r}$, the angular momentum in the world frame k_O and its derivatives $\dot{k}_O$, at every knot point. We choose the

backward Euler integration on Center of Mass acceleration $\ddot{r}$, mid-point integration on Center of Mass velocity $\dot{r}$, and backward Euler integration on the rate of angular momentum in the world frame $\dot{k}_O$.

$$\forall i = 1, \ldots, N \quad \dot{r}[i] - \dot{r}[i-1] = \ddot{r}[i]dt[i-1] \tag{4.1a}$$

$$r[i] - r[i-1] = \frac{1}{2}(\dot{r}[i] + \dot{r}[i-1])dt[i-1] \tag{4.1b}$$

$$k_O[i] - k_O[i-1] = \dot{k}_O[i]dt[i-1] \tag{4.1c}$$

Eq. (4.1a)-(4.1c) are linear constraints on decision variables $r[i], \dot{r}[i], \ddot{r}[i], k_O[i], \dot{k}_O[i]$.

We could use other integration or interpolation method for smaller discretization error, such as higher order Runge-Kutta method [18] or direct collocation method [57]. The backward-Euler integration approach is chosen here in simplicity.

In the subsequent subsections, we will impose the constraints on the robot kinematics and dynamics. Such constraints should be satisfied for every knot point, and we ignore the time indices $[i]$ when there is no ambiguity.

4.2.2 Computing Contact Wrench Set Given Contacts

In this chapter, we suppose the contact location at each time step is given, possibly from some footstep planners [30, 111, 177]. Also, we restrict to linearized friction cone in this chapter. With these assumptions, the Contact Wrench Set is fixed given as a linear cone at every time step, as shown in Fig. 4-1.

We assume for the i'th contact point p_i, its friction cone edges are $e_i^j, j = 1, \ldots, n_e$. Thus we can compute the Contact Wrench Set from contact locations and friction cone edges, and it is the linear cone spanned by the rays in the wrench space, each ray corresponds to the wrench computed from one friction cone edge.

$$CWS = \text{ConvexCone}\left(\begin{bmatrix} e_i^j \\ p_i \times e_i^j \end{bmatrix}\right), \forall\, i = 1, \ldots, n_1, j = 1, \ldots, n_e. \tag{4.2}$$

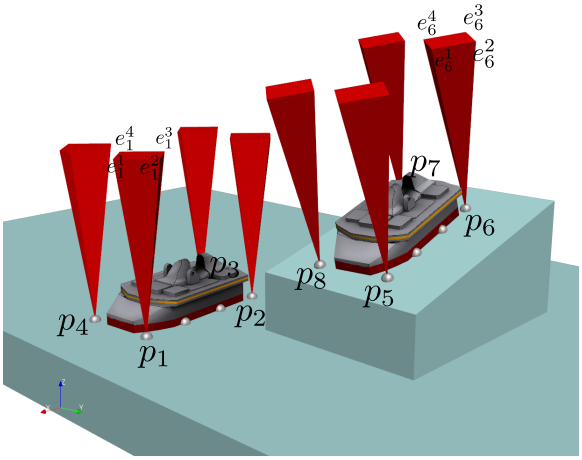


Figure 4-1: Footsteps with linearized friction cones

With the numeric procedures such as double description method [44] or Q-hull [4], we can transform the representation of the Contact Wrench Set, from a convex hull to an intersection of halfspace constraints.

$$CWS = \{w | a_i^T w \leq 0, i = 1, \ldots, n_a\} \tag{4.3}$$

where $a_i \in \mathbb{R}^6$ is the normal vector on each facet of the Contact Wrench Set, pointing outward, n_a is the number of facets of the Contact Wrench Set.

As we mentioned in Section 3.3, a necessary condition for a feasible robot motion, is that the wrench $\dot{h} - w_g$, computed from the robot momentum h and gravitational wrench w_g, lies within the contact wrench set.

$$a_i^T (\dot{h} - w_g) \leq 0, i = 1, \ldots, n_a \tag{4.4}$$

where the momentum and gravitation wrench are computed from the robot Center of Mass position r, and its rate of angular momentum around the origin of the world frame $\dot{k}_O$

$$\dot{h} = \begin{bmatrix} m\ddot{r} \\ \dot{k}_O \end{bmatrix}, \quad w_g = \begin{bmatrix} m\mathbf{g} \\ r \times m\mathbf{g} \end{bmatrix} \tag{4.5}$$

m is the mass of the robot, and $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration.

Combining Eq. (4.4)(4.5), we obtain the linear constraints on the robot states r, k_O and their derivatives, such that the motion is feasible in the sense that the contact wrench is within the Contact Wrench Set.

$$a_i^T \begin{bmatrix} m\ddot{r} - m\mathbf{g} \\ \dot{k}_O - r \times m\mathbf{g} \end{bmatrix} \leq 0, i = 1, \ldots, n_a \tag{4.6}$$

To compute the contact wrench set margin, we need to define the point where the wrench disturbance is applied at each time step. We denote that point as p_w. In the example of this thesis, we set that point to be the center of the support region, but any point near the robot should be reasonable. As we introduced in Chapter.3, the contact wrench set margin is defined as the magnitude of the disturbance, such that the wrench ball $\mathcal{B}_\epsilon$ is within the Contact Wrench Set

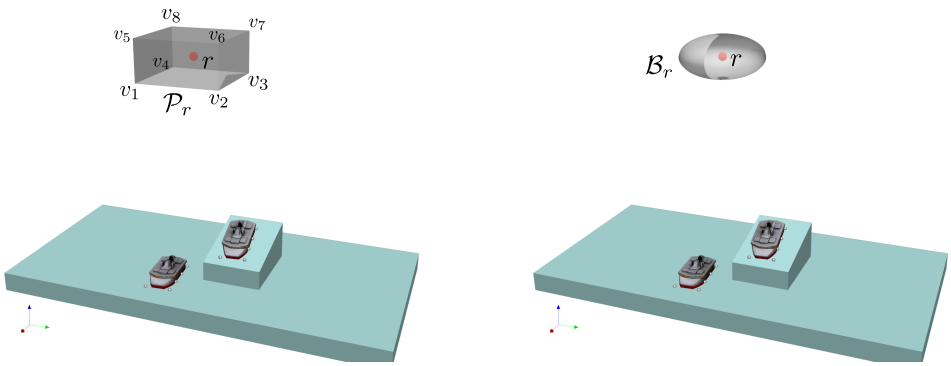
$$\mathcal{B}_\epsilon = \{\dot{h} - w_g + T(p_w, I)w | w^T Q_w w \leq \epsilon^2\} \subset CWS \tag{4.7}$$

where $T(p_w, I) \in \mathbb{R}^{6\times6}$ is the matrix that transforms a wrench at p_w to the origin of the world coordinate frame, defined in Eq. (3.12).

The contact wrench set margin can be computed as the scaled smallest distance from contact wrench $\dot{h} - w_g$ to each facet of the Contact Wrench Set, formulated as

$$\epsilon_{max} = \min_{i=1,\ldots,n_a} \bar{a}_i \begin{bmatrix} m\ddot{r} - m\mathbf{g} \\ \dot{k}_O - r \times m\mathbf{g} \end{bmatrix} \text{ where } \bar{a}_i = -\left[a_i^T T(p_w, I) Q_w^{-1} T^T(p_w, I) a_i\right]^{-\frac{1}{2}} a_i^T \tag{4.8}$$

The contact wrench set margin in Eq. (4.8) is the point-wise minimum among linear functions of $r, \ddot{r}$ and $\dot{k}_O$, and thus a concave piecewise linear function of the robot states and their time derivatives. Maximizing this concave function becomes a convex problem. We introduce a slack variable ϵ together with the following linear constraints, to represent the contact wrench set margin. We will maximize this slack



(a) Polytope region for Center of Mass (b) Ellipsoidal region for Center of Mass

Figure 4-2: Bounds on Center of Mass position.

variable ϵ

$$0 \leq \epsilon \leq \bar{a}_i \begin{bmatrix} m\ddot{r} - m\mathbf{g} \\ \dot{k}_O - r \times m\mathbf{g} \end{bmatrix} \quad \forall i = 1, \ldots, n_a \tag{4.9}$$

4.2.3 Bounds on Center of Mass Position

We want to encode some simple kinematic constraints into our planner, by constraining the region within which the Center of Mass should stay. Given the footsteps locations, we can infer the region of Center of Mass using robot leg kinematics. In this thesis, we constrain the Center of Mass position to be within either a polytope (Fig. 4-2a) or an ellipsoid (Fig. 4-2b), relative to the location of the foot; with the given foot locations, the region of the Center of Mass will also be defined.

If the polytope region $\mathcal{P}_r$ for Center of Mass is used, as in Fig 4-2a, we can represent such polytope in two ways, both as a convex hull representation using the vertices (Eq. (4.10)), and a halfspace representation using the facets (Eq. (4.11)).

$$r \in \mathcal{P}_r = \text{ConvexHull}(v_1, \ldots, v_{n_r}) \tag{4.10}$$

$$= \{r | A_r r \leq b_r\} \tag{4.11}$$

In this case, we impose linear constraints on Center of Mass r through Eq. (4.11).

The convex hull representation (Eq. (4.10)) will be used in Section 4.2.4 when we discuss the centroidal angular momentum.

If the ellipsoidal region $\mathcal{B}_r$ for Center of Mass is used, as in Fig 4-2b, we also formulate the ellipsoid in two ways, as the sub-level set of a quadratic form (Eq. (4.12)),

$$r \in \mathcal{B}_r = \{r | (r - r^*)^T Q_r (r - r^*) \leq 1\} \tag{4.12}$$

where $Q_r \in \mathbb{R}^{3\times 3}$ encodes the geometric shape of the ellipsoid. r^* is the center of the ellipsoid. The constraint (4.12) is a convex constraint on Center of Mass r, and can be formulated as a *second-order cone* constraint [16].

4.2.4 Centroidal Angular Momentum

We can control the contact forces through two ways, either by accelerating the Center of Mass, or changing the robot angular momentum. In the real application, controlling large angular momentum is generally less preferred, since the integral of the angular momentum is not meaningful, it does not corresponds to any physical orientation [124, 121]. It is also observed that in human walking, the centroidal angular momentum (angular momentum around the Center of Mass) is kept small [130, 63], the experimental results are plotted in Fig. 4-3. Thus in ZMP planning approach, the centroidal angular momentum is often set to be zero, and the robot is simplified as a point mass model with only linear momentum [78, 156]. So in this thesis, to plan a reasonable walking trajectory, we also aim to minimize the centroidal angular momentum, denoted as k_G.

The centroidal angular momentum k_G can be computed from the robot state k_O (angular momentum around origin of the world coordinate) and robot Center of Mass

$$k_G = k_O - mr \times \dot{r} \tag{4.13}$$

the centroidal angular momentum is a non-convex function in Eq. (4.13), as it con-

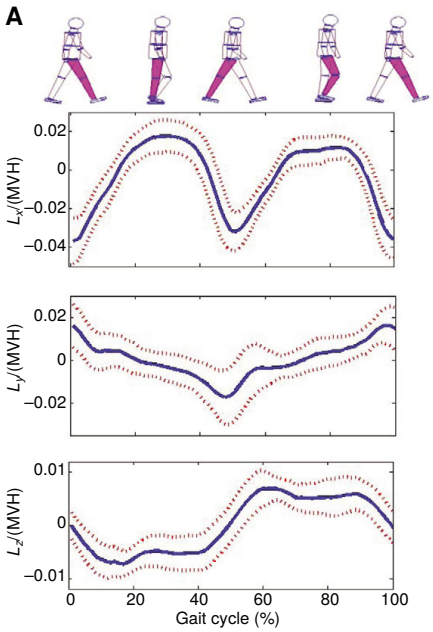


Figure 4-3: Normalized centroidal angular momentum for human walking experiments. The normalization factor is the product of the human mass (M), the walking velocity (V) and CoM height (H) [63]. The solid line is the mean value, the dashed lines are one standard deviation from the mean.

tains the product $r \times \dot{r}$.

In order to preserve the convexity of the problem, we exploit the structure in Eq. (4.13). We notice that although non-convex, $r \times \dot{r}$ is *bilinear*; namely, when fixing either r or $\dot{r}$, the function is linear. Also the Center of Mass r is bounded, as described in Section 4.2.3. With this structure bear in mind, instead of minimizing centroidal angular momentum directly, we will attempt to minimize an *upper bound* of the centroidal angular momentum in the remainder of this section.

In this thesis, we choose to minimize the L_1 norm of the centroidal angular momentum, namely

$$|k_G|_1 = |k_G^x| + |k_G^y| + |k_G^z| \tag{4.14}$$

$$= \max_{i=1,\dots,8} \omega_i^T k_G \tag{4.15}$$

where $\omega_i \in \mathbb{R}^3 = [\omega_i^x \ \omega_i^y \ \omega_i^z]^T$, each entries of ω_i is either 1 or -1. There are 8 possible combinations for ω_i. The L_1 norm of k_G is the maximum of the linear functions $\omega_i^T k_G$.

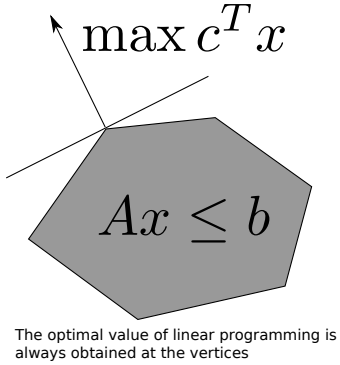


Figure 4-4: Linear programming

Eq.(4.14) is a convex piecewise linear function of k_G.

Centroidal Angular Momentum with Polytope Center of Mass Region

When the Center of Mass r is constrained to stay within the polytope $\mathcal{P}_r$, a convex upper bound of the centroidal angular momentum $|k_G|_1$ is obtained by replacing r in Eq. (4.13) by the vertices of the polytope $\mathcal{P}_r = \text{ConvexHull}(v_1, \ldots, v_{n_r})$, and take the maximum out of them. Namely

$$\begin{aligned}
|k_G|_1 =& |k_O - mr \times \dot{r}|_1 \\
&= \max_{i=1,\ldots,8} \omega_i^T (k_O - mr \times \dot{r}) \\
&\leq \max_{i=1,\ldots,8} \max_{r \in \mathcal{P}_r} \omega_i^T (k_O - mr \times \dot{r}) \qquad (4.16) \\
&= \max_{i=1,\ldots,8} \max_{j=1,\ldots,n_r} \omega_i^T (k_O - mv_j \times \dot{r}) \qquad (4.17)
\end{aligned}$$

where each entry of ω_i is either 1 or -1, as defined in (4.15).

Geometrically, the equality in Eq. (4.17) holds because for a linear programming problem (linear objectives and linear constraints), the optimal value is always obtained at one of the vertices of the constrained polytope, as shown in Fig. 4-4, thus

$$\max_{r \in \mathcal{P}_r} \omega_i^T (k_O - mr \times \dot{r}) = \max_{j=1,\ldots,n_r} \omega_i^T (k_O - mv_j \times \dot{r}) \qquad (4.18)$$

We can write Eq. (4.17) with the slack variable $s \in \mathbb{R}^3$ to represent the upper

bound of the centroidal angular momentum L_1 norm

$$s \geq \omega_i^T (k_O - m v_j \times \dot{r}) \; \forall i = 1, \ldots, 8. \; j = 1, \ldots, n_r \tag{4.19}$$

With the inequality in Eq. (4.19), s becomes an upper bound of the centroidal angular momentum L_1 norm. The constraint on this upper bound is convex on the variables $k_O, \dot{r}, s$ (Eq. (4.19)). Instead of minimizing the non-convex centroidal angular momentum norm directly, we can minimize this convex upper bound function.

This upper bound might not be tight, since it ignores other constraints on Center of Mass. For example, if we have bounds on the center of mass velocity, then the Center of Mass position cannot be too distant from each other in the adjacent time steps. Anyway, we will show by minimizing this upper bound, the actual centroidal angular momentum is effectively kept small.

Robust Optimization Perspective

The relaxation we presented in the previous subsection in Sec.4.2.4 can be understood from the *robust optimization* perspective. In order to minimize the L_1 norm of the centroidal angular momentum, we can consider to minimize an upper bound.

The L_1 norm of the centroidal angular momentum is written as

$$\mathbb{P}_1 : p_1^* = \min_{r, k_O, s, \dot{r}} \; s \tag{4.20}$$

$$\text{s.t } s \geq \omega_i^T (k_O - m r \times \dot{r}) \; \forall i = 1, \ldots, 8 \tag{4.21}$$

$$r \in \mathcal{P}_r \tag{4.22}$$

where $s \in \mathbb{R}$ represents centroidal angular momentum L_1 norm.

We can define a min-max optimization problem $\mathbb{P}_2$, in which we maximize over r in the inner loop, so as to get rid of r in the outer loop, and to make the problem

convex.

$$\mathbb{P}_2 : p_2^* = \min_{k_O, s, \dot{r}} \max_{r \in \mathcal{P}_r} s \tag{4.23}$$

$$\text{s.t } s \geq \omega_i^T (k_O - mr \times \dot{r}) \; \forall i = 1, \ldots, 8 \tag{4.24}$$

Apparently $p_1^* \leq p_2^*$, due to the maximization over r in the inner loop of $\mathbb{P}_2$. In robust optimization, problem in the form as $\mathbb{P}_2$ can be regarded as the robust counterpart of a linear programming problem, with $k_O, s, \dot{r}$ as decision variables, and r being the uncertain parameter in this linear programming problem. When the uncertain parameter is in a polytope $\mathcal{P}_r$, it is known that $\mathbb{P}_2$ is a linear optimization problem [6, 7, 11]. Section A.1 in appendix gives a brief overview on robust optimization.

Using robust optimization, it can be shown that $\mathbb{P}_2$ is equivalent to

$$\mathbb{P}_2 : p_2^* = \min_{k_O, s, \dot{r}} s \tag{4.25}$$

$$\text{s.t } s \geq \omega_i^T (k_O - mv_j \times \dot{r}) \; \forall i = 1, \ldots, 8. \; j = 1, \ldots, n_r \tag{4.26}$$

Thus we get the same constraint as Eq. (4.19).

Centroidal Angular Momentum with Ellipsoidal Center of Mass Region

When the Center of Mass is constrained within the ellipsoid $\mathcal{B}_r$, like what we present in the polytope case, we can similarly find a convex function as an upper bound of the non-convex centroidal angular momentum, and we can efficiently minimize this convex upper bound. Like we showed in the previous sub-section, we can also use robust optimization to formulate this upper bound as a special type of convex problem, specifically a second-order cone problem. We will show how we can derive such formulation in this sub-section.

Algebraically, we aim to minimize the L_1 norm of centroidal angular momentum, when the Center of Mass is within the ellipsoid $\mathcal{B}_r$, as the following optimization

problem $\mathbb{P}_1$

$$\mathbb{P}_1 : p_1^* = \min_{r,s,k_O,\dot{r}} \; s \tag{4.27}$$

$$\text{s.t } s \geq \omega_i^T (k_O - mr \times \dot{r}) \; \forall i = 1, \ldots, 8 \tag{4.28}$$

$$r \in \mathcal{B}_r = \{r | (r - r^*)^T Q_r (r - r^*) \leq 1\} \tag{4.29}$$

p_1^* is the optimal value of the non-convex optimization problem.

The optimal value p_1^* of problem $\mathbb{P}_1$ is smaller than the optimal value of problem $\mathcal{P}_2$, in which we maximize over Center of Mass r first, and then minimize over the rest of the variables

$$\mathbb{P}_2 : p_2^* = \min_{s,k_O,\dot{r}} \max_{r \in \mathcal{B}_r} s \tag{4.30}$$

$$\text{s.t } s \geq \omega_i^T (k_O - mr \times \dot{r}) \; \forall i = 1, \ldots, 8 \tag{4.31}$$

The motivation of relaxing the minimization problem in $\mathbb{P}_1$ to the min-max problem in $\mathbb{P}_2$, is to remove the variable r in the optimization problem, so as to make the problem convex. In the inner maximization step, the maximization is satisfied if and only if

$$s \geq \omega_i^T (k_O + m \lfloor \dot{r} \rfloor_\times r) \; \forall r \in \mathcal{B}_r. \; i = 1, \ldots, 8 \tag{4.32}$$

$\lfloor \dot{r} \rfloor_\times \in \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix representing the cross product with $\dot{r}$, as defined in (3.13).

We show that Eq. (4.32) imposes several *second-order cone* constraints on k_O, r and s. To see this, we take one inequality of Eq. 4.32 as an example

$$s \geq \omega_i^T (k_O + m \lfloor \dot{r} \rfloor_\times r) \; \forall r \in \mathcal{B}_r \tag{4.33a}$$

$$\Leftrightarrow s - \omega_i^T k_O \geq \max_{r \in \mathcal{B}_r} m \omega_i^T \lfloor \dot{r} \rfloor_\times r \tag{4.33b}$$

$$\Leftrightarrow s - \omega_i^T k_O \geq m \omega_i^T \lfloor \dot{r} \rfloor_\times r^* + m \sqrt{\omega_i^T \lfloor \dot{r} \rfloor_\times^T Q_r^{-1} \lfloor \dot{r} \rfloor_\times \omega_i} \tag{4.33c}$$

The condition from Eq. (4.33b) to Eq. (4.33c) holds because we can compute in close form the optimal value of maximizing a linear function subject to ellipsoidal constraints. A simple example is $\max_{x^T x \leq 1} c^T x = \sqrt{c^T c}$.

Eq. (4.33c) is a second-order cone constraint on $k_O, \dot{r}$ and s. Likewise, we can write all other inequalities in Eq. (4.32) as second order cone constraints. As a result, the upper bound p_2^* can be computed from the following second-order cone program

$$\mathbb{P}_2 : p_2^* = \min_{k_O, \dot{r}, s} \; s \tag{4.34}$$

$$\text{s.t } s - \omega_i^T k_O - m\omega_i^T \lfloor \dot{r} \rfloor_\times r^* \geq m\sqrt{\omega_i^T \lfloor \dot{r} \rfloor_\times^T Q_r^{-1} \lfloor \dot{r} \rfloor_\times \omega_i} \; \forall i = 1, \ldots, 8 \tag{4.35}$$

To summarize, we can minimize an upper bound of the centroidal angular momentum as a convex function. If the admissible region of Center of Mass is a polytope, the convex problem is a linear programming problem; if the admissible region is an ellipsoid, the convex problem is a second-order cone problem. Both can be solved efficiently.

4.2.5 Cost function

The constraints of the optimization problems are presented in the previous sections in the time integration (Eq. (4.1a)-(4.1c)), the contact wrench set margin (Eq. (4.9)), the Center of Mass region (Eq. (4.11) if using polytope region, or Eq. (4.12) if using ellipsoidal region) , and the centroidal angular momentum upper bound (Eq. (4.19) if using polytope region, or Eq. (4.35) if using an ellipsoidal region). These constraints are either linear constraints or second-order cone constraints.

The objectives for this planner include

- Maximize the contact wrench set margin.
- Minimize the upper bound of the centroidal angular momentum.
- Smooth motion.

Thus the cost can be formulated as

$$\min_{r,\dot{r},\ddot{r},s,\epsilon,k_O,\dot{k}_O} \sum_i c_s s[i] - c_\epsilon \epsilon[i] + c_{\ddot{r}} \ddot{r}[i]^T \ddot{r}[i] \tag{4.36}$$

where $c_s, c_\epsilon, c_{\ddot{r}} \in \mathbb{R}$ are positive scalars representing the weighting of cost on centroidal angular momentum, CWS margin and CoM acceleration respectively. The objective (Eq. (4.36) is a convex quadratic function of the decision variables.

To summarize, the planner is a convex Quadratic Programming problem when the Center of Mass region is a polytope, or a second-order cone prgramming problem when the Center of Mass region is an ellipsoid. Both can be solved efficiently by modern numeric solvers [120, 116].

4.3 Result

We show the results of our planner for robot walking on flat ground and uneven terrain course. We compare with the ZMP planner used in [156], which aims to minimize the distance from the ZMP to the center of the foot contact region. We will show that the Contact Wrench Set (CWS) planner generates similar result as the ZMP planner on the flat ground; and on uneven terrain course where ZMP planner fails, the CWS planner can still work.

The footsteps locations are generated by the footstep planner [30]. The time of foot contact is determined by simple heuristics on leg swing speed.

4.3.1 Flat ground walking

On the flat ground, we compute the CoM trajectory using the CWS planner, with both polytope CoM region and Ellipsoidal CoM region. The result from the Contact Wrench Set (CWS) planner is very similar to the ZMP planner, as shown in Fig. 4-5, 4-6 and 4-7. This similarity is expected because the Contact Wrench Set is a generalization of ZMP support region in the flat ground case.

The spikes in Fig. 4-6 occur during the double support phase. This makes sense

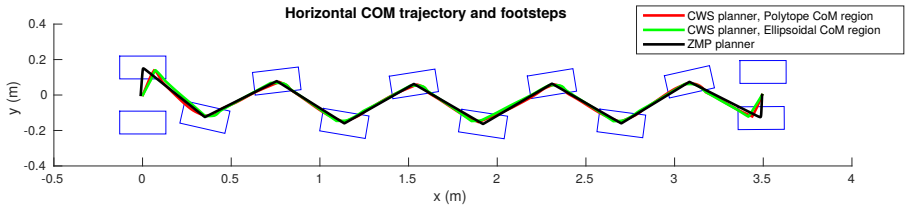
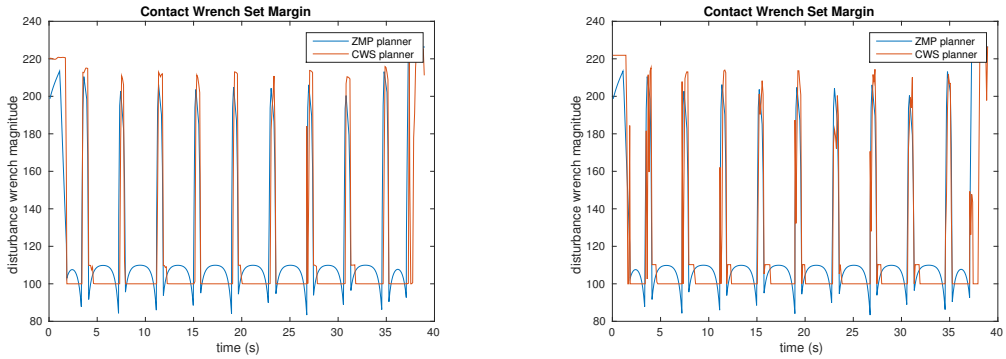


Figure 4-5



(a) CWS planner with polytope CoM region. (b) CWS planner with ellipsoidal CoM region.

Figure 4-6: Contact Wrench Set margin for flat ground walking.

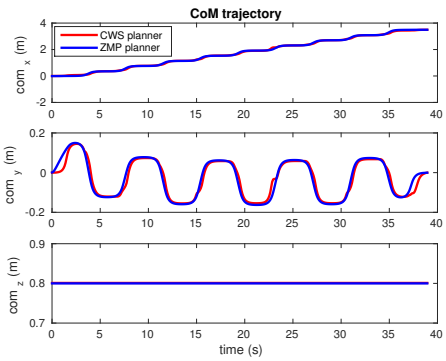
since the Contact Wrench Set is larger when the robot transit from single support phase to double support, and thus the margin increases.

We also show that the Contact Wrench Set planner keeps the centroidal angular momentum small, as shown in Fig. 4-8. The centroidal angular momentum is normalized by a constant as the product of robot mass, the walking speed and the CoM height. These values are in the same magnitude (or smaller than) the values from human experiments [63].

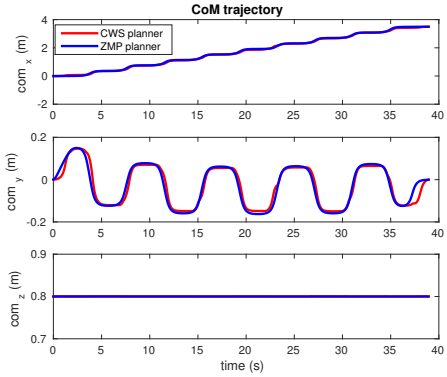
We solved the optimization problem on an Intel Core i7 machine. The computation time for this trajectory is 9 seconds when we use a polytope CoM region, and 6.2 seconds when we use an ellipsoidal region. The trajectory has 397 knot points. The optimization problem has 7543 decision variables.

4.3.2 Walking on uneven terrain course

We also test our planner on an uneven ground, as shown in Fig. 4-9. This is the terrain course used in the DARPA Robotics Challenge [135]. We set the friction cone

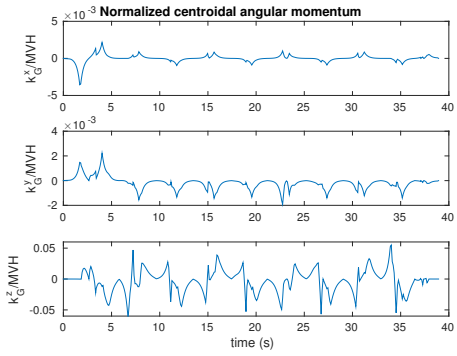


(a) CWS planner with polytope CoM region.

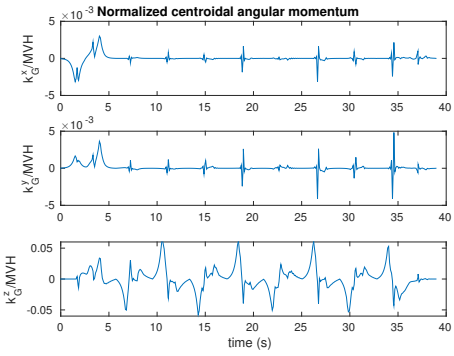


(b) CWS planner with ellipsoidal CoM region.

Figure 4-7: CoM trajectory for flat ground walking.



(a) CWS planner with polytope CoM region



(b) CWS planner with Ellipsoidal CoM region

Figure 4-8: The normalized centroidal angular momentum. The normalizing factor is the robot mass (M) times velocity (V) and com height (H), as used in [63].

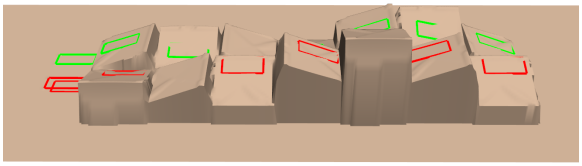
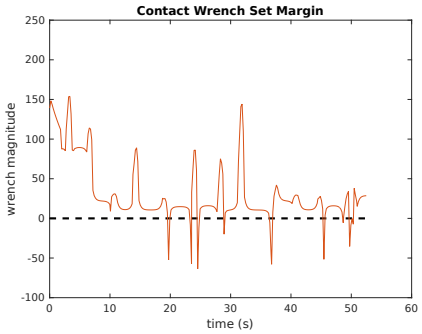
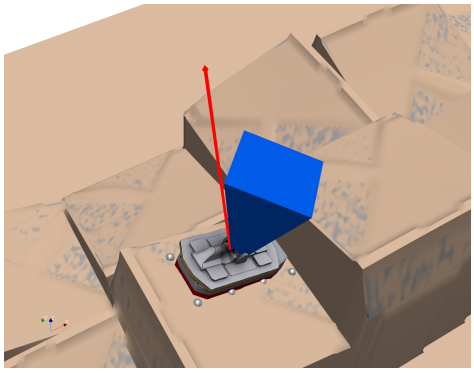


Figure 4-9: Tilted cinderblocks terrain with given footsteps



(a) Contact wrench set margin



(b) Friction cone (blue region) and contact force (red arrow)

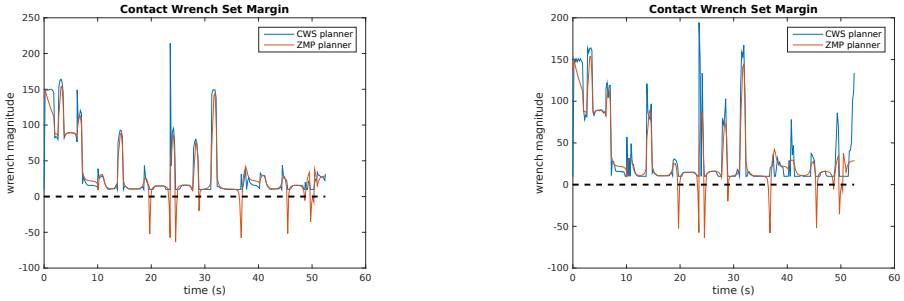
Figure 4-10: Result from ZMP planner on terrain course

coefficient to 0.4 in this test.

The motion coming out of the ZMP planner is infeasible. We draw the contact wrench set margin from the ZMP planner in Fig. 4-10a, some of the margins are negative, indicating that the friction cone constraints are violated. Also, we draw the friction force and friction cone around time 29s; the friction force falls outside of the friction cone, as shown in Fig. 4-10b. This would cause the robot to tip over when executing the planned motion. The failure occurs since the ZMP planner does not take friction cones or non-coplanar contacts into consideration.

As a comparison, the CWS planner generates feasible motion that satisfies the friction cone constraints. We draw the contact wrench set margin in Fig. 4-11. The margin is always positive. In Fig. 4-12, we also show the friction cones and the contact forces, at the same time sample as in Fig. 4-10b. The friction force lie within the friction cone, using the CWS planner.

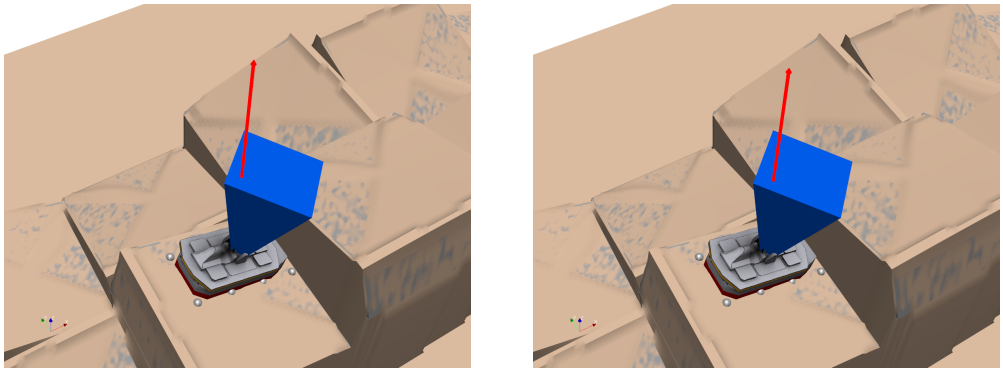
We also compare the CoM trajectories from ZMP and CWS planners. In Fig. 4-13, we draw the CoM horizontal trajectories, together with the projection of footsteps.



(a) CWS planner with polytope CoM region

(b) CWS planner with ellipsoidal CoM region

Figure 4-11: Contact Wrench Set Margin on the terrain course



(a) CWS planner with polytope CoM region

(b) CWS planner with ellipsoid CoM region

Figure 4-12: Friction cone (blue region) and contact force (red arrow) on the terrain course

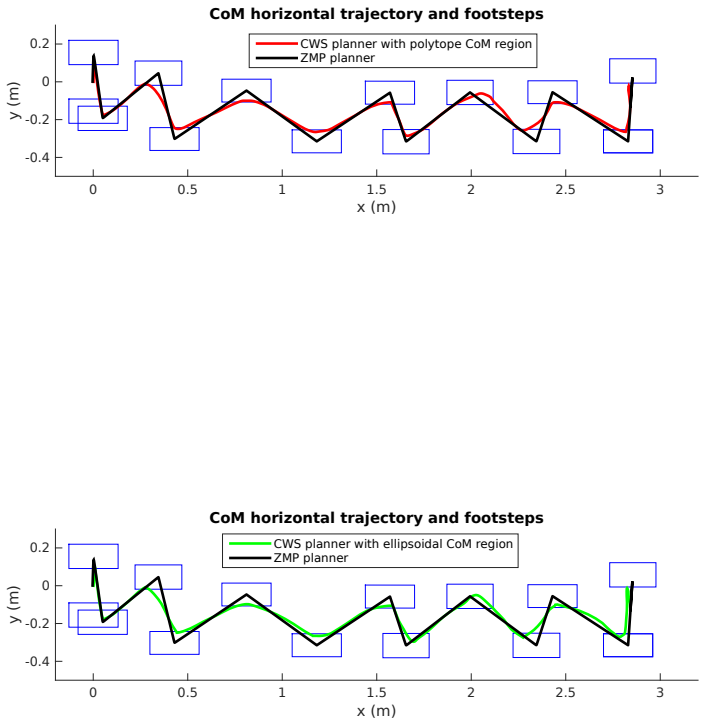
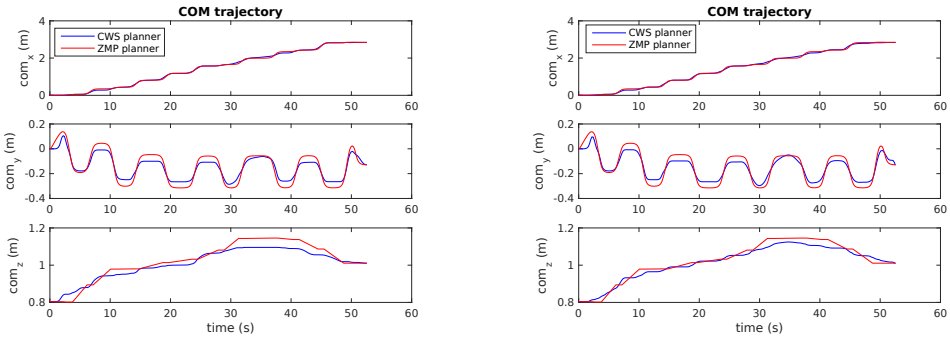


Figure 4-13: Horizontal CoM trajectories on the terrain course

We also draw the CoM trajectory in each xyz axes in Fig. 4-14. The CoM motion from the CWS planner involves less swaying in the y axes, along which the robot sways left and right. The terrain course together with the CoM trajectories are shown in Fig. 4-15.

We show that the CWS planner also generates a centroidal angular momentum trajectory with small magnitude, as plotted in Fig. 4-16. We normalize it with robot mass (M), robot velocity (V) and CoM height (H), as in Herr's paper [63]. The normalized centroidal angular momentum on the terrain course has larger magnitude than the results from human experiments on the flat ground, but still in the same scale.



(a) CWS planner with polytope CoM region

(b) CWS planner with ellipsoidal CoM region

Figure 4-14: CoM trajectory in xyz axes on the terrain course

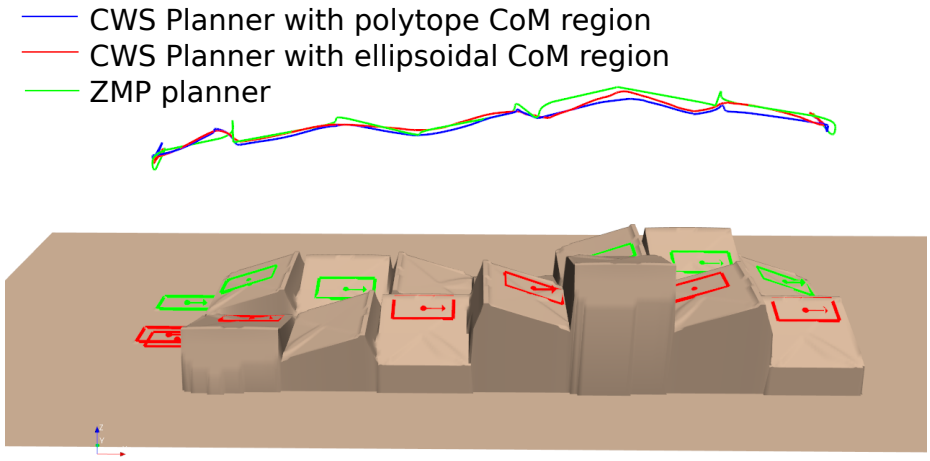
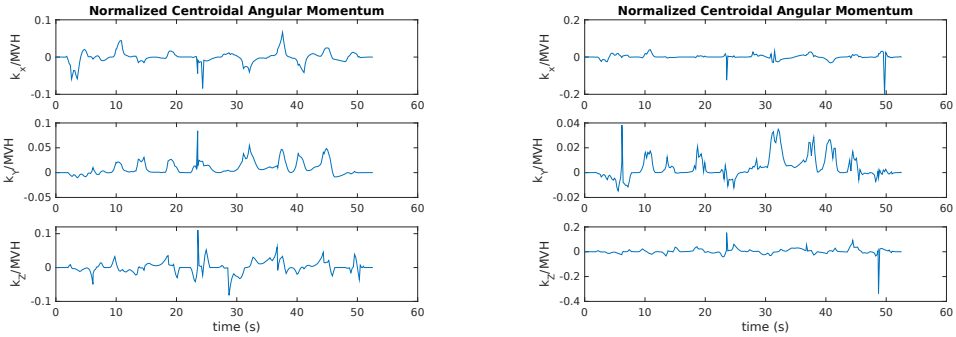


Figure 4-15: CoM trajectories on the terrain course



(a) CWS planner with polytope CoM region

(b) CWS planner with ellipsoidal CoM region

Figure 4-16: Normalized centroidal angular momentum on the terrain course

We solve the optimization problem on an Intel Core i7 machine for the Contact Wrench Set Planner. The planner with the polytope CoM region takes 9 seconds, while that with the ellipsoidal CoM region takes 5.6 seconds. The trajectories have 272 sample points, with 5168 decision variables.

Chapter 5

Planning while searching for contacts

In this chapter, we will present our planner to search for robust robot whole-body motion and the contact locations, such that the contact wrench set margin is maximized through optimization. We will show that the structure of this optimization problem allows us to solve it with two different approaches, either to solve it through sequential semidefinite programming, or as a general nonlinear optimization problem. We then apply our planner to three problems 1) force closure grasp optimization. 2) static posture optimization, and 3) trajectory optimization, and improve the robustness of the motion through optimization.

5.1 Computing Contact Wrench Set Margin

In the previous section 4.2.2, we can compute the contact wrench set margin using the facets of the contact wrench set (Eq.(4.8)), when the contact locations are prespecified, and the contact force is subject to *linearized* friction cone constraint. When the contact locations are not given, or when we want to use the nonlinear friction cone, we cannot use this formulation to compute the margin, for two reasons

- When using the linearized friction cone and search for the contact locations, in order to compute the distance from the contact wrench $\dot{h} - w_g$ to the facet of the linear wrench cone, we need to transform the cone from representation using its rays (V-representation), to the representation using the facet normals

(H-representation). Such transformation does not have a closed form representation, since we do not know which ray would be the extreme ray, i.e., the ray on the boundary of the cone, when we are changing those rays by searching over contact locations. In fact, the transformation from V-representation of a polyhedron to its H-representation is not always differentiable, and the gradient-based optimization solver would be profoundly unhappy if we had to use such non-differentiable transformation within optimization.

- When using nonlinear friction cones, the Contact Wrench Set is not a linear cone, thus we cannot compute the margin as the distance to the facet in the closed form.

Our goal in this section is to compute a *lower bound* of the contact wrench set margin as a differentiable function, and we will maximize such lower bound through optimization.

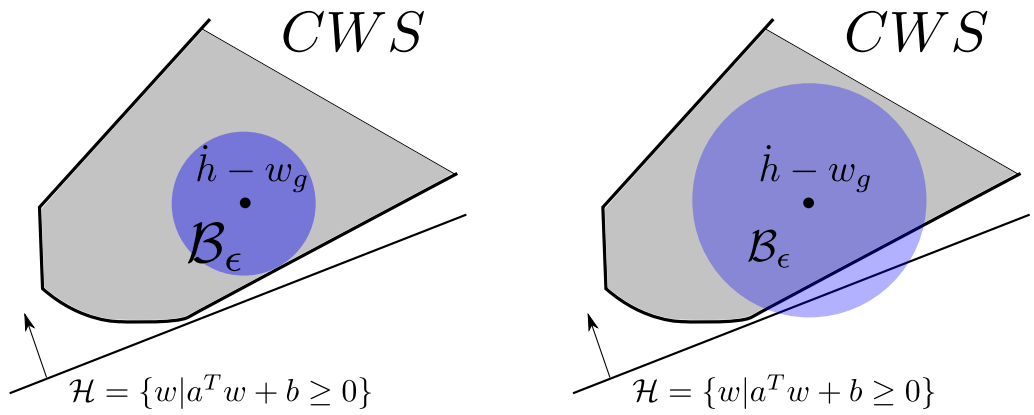
As shown in Chapter 3, the Contact Wrench Set (CWS) is always a convex set for every fixed contact locations. The contact wrench $\dot{h} - w_g$ has a margin ϵ, if and only if an ellipsoid $\mathcal{B}_\epsilon$ is contained in the Contact Wrench Set, as defined previously in Eq.(3.11)

$$\mathcal{B}_\epsilon = \{\dot{h} - w_g + T(p_w, I)w | w^T Q_w w \leq \epsilon^2\} \subset CWS \tag{5.1}$$

Geometrically, we can interpret the set inclusion (5.1) using halfspace. As illustrated in Fig.5-1. We can think of the interpretation conversely, on the condition when the ellipsoid $\mathcal{B}_\epsilon$ is *not* a subset of Contact Wrench Set.

Corollary 1. *The ellipsoid $\mathcal{B}_\epsilon$ being not a subset of the Contact Wrench Set, is equivalent to the existence of a halfspace $\mathcal{H}$, such that the halfspace contains the Contact Wrench Set, but does not contain the ellipsoid.*

$$\begin{aligned} &\mathcal{B}_\epsilon \not\subset CWS \\ \Leftrightarrow &\exists \mathcal{H} = \{w | a^T w + b \geq 0\},\ s.t\ \mathcal{H} \supset CWS, \mathcal{H} \not\supset \mathcal{B}_\epsilon \end{aligned}$$



(a) $\mathcal{B}_\epsilon \subset CWS$ if and only if for all halfspace $\mathcal{H}$ contains CWS, the same halfspace contains $\mathcal{B}_\epsilon$.

(b) $\mathcal{B}_\epsilon \not\subset CWS$ if and only if there exists a halfspace $\mathcal{H}$ that contains CWS, but does not contain $\mathcal{B}_\epsilon$.

Figure 5-1: Contact Wrench Set margin, interpreted using halfspace

Corollary 1 is illustrated in Fig.5-1b. The corollary is true because a convex set is the intersection of all halfspaces that contains the set [16], so we consider the halfspace that contains the convex Contact Wrench Set in Fig.5-1b.

Conversely, Corollary 1 is equivalent to the following corollary

Corollary 2. *The ellipsoid $\mathcal{B}_\epsilon$ is a subset of Contact Wrench Set, if and only if for any halfspace $\mathcal{H}$ that contains the Contact Wrench Set, the same halfspace $\mathcal{H}$ also contains the ellipsoid $\mathcal{B}_\epsilon$.*

$$\mathcal{B}_\epsilon \subset CWS$$
$$\Leftrightarrow (\forall \mathcal{H} \supset CWS \Rightarrow \mathcal{H} \supset \mathcal{B}_\epsilon)$$

The symbol $\Rightarrow$ in Corollary 2 means that if the condition on the left-hand side of $\Rightarrow$ is true ($\forall \mathcal{H} \supset CWS$), then the condition on the right-hand side of $\Rightarrow$ will also be true ($\mathcal{H} \supset \mathcal{B}_\epsilon$). Thus $\Rightarrow$ is read as "implies". Corollary 2 is illustrated in Fig.5-1a.

The structure of the Contact Wrench Set permits us to study the algebraic condition on the halfspace. As explained in Chapter 3 (Eq.(3.8)), the Contact Wrench Set is the Minkowski sum $CWS = \mathcal{K}_1 \oplus \ldots \oplus \mathcal{K}_{n_1} \oplus \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$, where $\mathcal{K}_i$ is a cone in the wrench space, computed from the friction cone at the i'th contact point; and

$\mathcal{P}_j$ is a polytope in the wrench space, computed from the bounded linearized friction cone the j'th contact point.

We first decompose the Contact Wrench Set as the Minkowski sum of a cone $\mathcal{K}_1 \oplus \ldots \oplus \mathcal{K}_{n_1}$, and a polytope $\mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$, and we consider the following lemma:

Lemma 1. *A halfspace $\mathcal{H} = \{w | a^T w + b \geq 0\}$ contains the Minkowski sum of a cone $\mathcal{K}$ and a set $\mathcal{P}$, where $0 \in \mathcal{P}$, if and only if the halfspace contains the cone and the set separately. i.e.,*

$$\mathcal{H} \supset \mathcal{K} \oplus \mathcal{P} \Leftrightarrow (\mathcal{H} \supset \mathcal{K}, \mathcal{H} \supset \mathcal{P}) \tag{5.2}$$

Proof. The $\Rightarrow$ direction is obvious, just replace either $\mathcal{K}$ or $\mathcal{P}$ with element 0. For the $\Leftarrow$ direction,

$$\mathcal{H} \supset \mathcal{K} \Leftrightarrow \begin{cases} a^T u \geq 0 \ \forall u \in \mathcal{K} \\ b \geq 0 \end{cases} \tag{5.3}$$

$$\mathcal{H} \supset \mathcal{P} \Leftrightarrow a^T v + b \geq 0 \ \forall v \in \mathcal{P} \tag{5.4}$$

Eq.(5.3) and (5.4) together means

$$a^T (u + v) + b \geq 0 \ \forall u \in \mathcal{K}, v \in \mathcal{P} \tag{5.5}$$

namely, $\mathcal{H} \supset \mathcal{K} \oplus \mathcal{P}$. □

Using Lemma 1, we can assert that the Contact Wrench Set is contained in the halfspace $\mathcal{H}$, if and only if $\mathcal{H} \supset (\mathcal{K}_1 \oplus \ldots \oplus \mathcal{K}_{n_1})$, and $\mathcal{H} \supset (\mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2})$. We can further investigate the condition that a halfspace contains the Minkowski sum of cones

Lemma 2. *A halfspace contains the Minkowski sum of two cones, if and only if the*

halfspace contains each cone individually. Namely

$$\mathcal{H} \supset (\mathcal{K}_1 \oplus \mathcal{K}_2) \Leftrightarrow \begin{cases} \mathcal{H} \supset \mathcal{K}_1 \\ \mathcal{H} \supset \mathcal{H}_2 \end{cases} \tag{5.6}$$

The proof is straightforward and we will ignore it.

With both Corollary 2, Lemma 1 and 2, we have the following theorem on the contact wrench set margin.

Theorem 1. *A wrench $\dot{h} - w_g$ has a margin no smaller than ϵ in the Contact Wrench Set $CWS = \mathcal{K}_1 \oplus \ldots \oplus \mathcal{K}_{n_1} \oplus \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$, if and only if for any halfspace $\mathcal{H} = \{w | a^T w + b \geq 0\}$ that satisfies*

$$\begin{cases} \mathcal{H} \supset \mathcal{K}_i, i = 1, \ldots, n_1 \\ \mathcal{H} \supset \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2} \end{cases} \tag{5.7}$$

that same halfsapce also contains the ellipsoid $\mathcal{B}_\epsilon = \{\dot{h} - w_g + T(p_w, I)w | w^T Q_w w \leq \epsilon^2\}$

$$\mathcal{H} \supset \mathcal{B}_\epsilon \tag{5.8}$$

We will analyze the algebraic condition for the set inclusions in Theorem 1. The algebraic condition of $\mathcal{H} \supset \mathcal{B}_\epsilon$ can be readily formulated as

$$\mathcal{H} \supset \mathcal{B} \Leftrightarrow a^T(\dot{h} - w_g) + b \geq \epsilon \sqrt{a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a} \tag{5.9}$$

As we already explained in Chapter 3 (Eq.(3.7)), the polytope $\mathcal{P}_i$ is parameterized using the vertices of the polytope

$$\mathcal{P}_i = \text{ConvexHull}(v_i^1, \ldots, v_i^{n_e}) \tag{5.10}$$

where the vertex v_i^j is computed from the contact location p_i and the edge of the friction cone e_i^j as

$$v_i^j = \begin{bmatrix} e_i^j \\ p_i \times e_i^j \end{bmatrix} \tag{5.11}$$

The Minkowski sum $\mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$ is a new polytope, parameterized as a convex hull

$$\mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2} = \text{ConvexHull}(\underbrace{\sum_i (v_i^{l_i})}_{\bar{v}_j}), l_i \in \{1, 2, \ldots, n_e\} \tag{5.12}$$

namely, we compute the Minkowski sum of the polytopes $\mathcal{P}_i$, where the vertices of the summed polytope is the sum of one vertex from each polytope, and there are $n_e^{n_2}$ possible vertices in the summed polytope $\mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$. For notation simplicity, we denote the vertices of this summed polytope as $\bar{v}_j, j = 1, \ldots, n_e^{n_2}$. The algebraic condition of $\mathcal{H} \supset \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2}$ is

$$\mathcal{H} \supset \mathcal{P}_1 \oplus \ldots \oplus \mathcal{P}_{n_2} \Leftrightarrow a^T \bar{v}_j + b \geq 0, \forall j = 1, \ldots, n_e^{n_2} \tag{5.13}$$

For the cone $\mathcal{K}_i$, when we use the linearized friction cone $\mathcal{K}_i = \text{ConvexCone}(v_i^1, \ldots, v_i^{n_e})$ where v_i is defined in Eq.(5.11), the condition that halfspace $\mathcal{H}$ contains the cone $\mathcal{K}_i$ is that

$$\mathcal{H} \supset \mathcal{K}_i \Leftrightarrow \begin{cases} a^T v_i^j \geq 0, \forall i = 1, \ldots, n_1, j = 1, \ldots, n_e \\ b \geq 0 \end{cases} \tag{5.14}$$

When we use the nonlinear friction cone $\mathcal{K}_i = \left\{ \begin{bmatrix} f \\ p_i \times f \end{bmatrix} \middle| \sqrt{1+\mu^2} f^T c_i \geq \sqrt{f^T f} \right\}$, the condition that the halfspace $\mathcal{H}$ contains the cone $\mathcal{K}_i$ is that

$$\mathcal{H} \supset \mathcal{K}_i \Leftrightarrow \begin{cases} a^T G(p_i) c_i \geq \mu \sqrt{a^T G(p_i)(I_{3\times 3} - c_i c_i^T) G(p_i)^T a} \\ b \geq 0 \end{cases} \tag{5.15}$$

where $G(p_i) \in \mathbb{R}^{6\times 3}$ is the matrix that transforms a force into a wrench at the origin of the world frame

$$G(x) = \begin{bmatrix} I_{3\times 3} \\ \lfloor x \rfloor_\times \end{bmatrix} \tag{5.16}$$

An alternative interpretation of the condition (5.14) and (5.15) is that $a \in \mathcal{K}_i^*, b \geq 0$, where $\mathcal{K}_i^*$ is the *dual cone* of $\mathcal{K}_i$ [16].

With the algebraic formulation on set inclusions (Eq.(5.9),(5.13),(5.14)), when we use the linearized friction cone, Theorem 1 is formulated algebraically as saying when the for all a, b satisfying the condition on the left-hand side of $\Rightarrow$, it implies that a, b satisfy the condition on the right-hand side of $\Rightarrow$ in Eq.(5.17)

$$\left.\begin{array}{l} a^T v_i^j \geq 0, \forall i = 1, \ldots, n_1, j = 1, \ldots, n_e \\ b \geq 0 \\ a^T \bar{v}_j + b \geq 0, \forall j = 1, \ldots, n_e^{n_2} \end{array}\right\} \Rightarrow a^T(\dot{h} - w_g) + b \geq \epsilon \sqrt{a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a} \tag{5.17}$$

Another way to think of the implication in Eq.(5.17) is that if we fix the values of $v_i^j, \bar{v}_i^j, \dot{h}, w_g$ and p_w, then the following optimization problem is infeasible

$$\begin{aligned}
&\min_{a,b} 0 \\
&\text{s.t } a^T(\dot{h} - w_g) + b \geq \epsilon\sqrt{a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a} \\
&\quad a^T v_i^j \geq 0, \forall i = 1, \ldots, n_1, j = 1, \ldots, n_e \\
&\quad b \geq 0 \\
&\quad a^T \bar{v}_j + b \geq 0, \forall j = 1, \ldots, n_e^{n_2}
\end{aligned}$$

From Putinar's Positivstellensatz [137], it is advantageous to write the left-hand side of $\Rightarrow$ in Eq.(5.17) as a compact set on a, b. Since the condition in Eq.(5.17) is homogeneous in a, b, thus we can scale a, b uniformly. It can be easily seen that the condition (5.17) is equivalent to the following condition, where the left-hand side of $\Rightarrow$ becomes compact, by constraining that $a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a = 1$

$$\left.\begin{aligned}
&a^T v_i^j \geq 0, \forall i = 1, \ldots, n_1, j = 1, \ldots, n_e \\
&b \geq 0 \\
&a^T \bar{v}_j + b \geq 0, \forall j = 1, \ldots, n_e^{n_2} \\
&a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a = 1
\end{aligned}\right\} \Rightarrow a^T(\dot{h} - w_g) + b \geq \epsilon \tag{5.18}$$

All the expressions in (5.18) are polynomials with a, b as indeterminates, and $v_i^j, \bar{v}_i^j, p_w, \dot{h}, w_g$ are coefficients of the polynomials. A sufficient condition for condition (5.18) is the existence of some polynomials $l_i(a, b)$, satisfying the following conditions.

$$\begin{aligned}
l_0(a,b) * (a^T(\dot{h} - w_g) + b - \epsilon) - l_1(a,b)(a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a - 1) & \\
-l_2^{ij}(a,b) * (a^T v_i^j) - l_3(a,b) * b - l_4^j(a^T \bar{v}_i^j + b) \text{ is sos} & \quad (5.19a) \\
l_0(a,b) - 1 \text{ is sos} & \quad (5.19b) \\
l_2^{ij}(a,b), l_3(a,b), l_4^j(a,b) \text{ is sos} & \quad (5.19c)
\end{aligned}$$

where a polynomial $p(x)$ is sos is abbreviated of $p(x)$ is a Sum of Squares polynomial (sos). A polynomial being sos is a sufficient condition of $p(x) \geq 0 \ \forall x$. It can be easily seen that the existence of the polynomials $l_0(a,b), l_1(a,b), l_2^{ij}(a,b), l_3(a,b), l_4^j(a,b)$ certifies the implication in Eq.(5.18), and thus shows that the contact wrench margin is no less than ϵ. The polynomials $l_i(a,b)$ are called *Lagrangian multipliers*, and this is a common trick in Sum of Squares verification [128, 157].

It is well known that a polynomial being Sum of Squares, is equivalent to a certain matrix being positive semidefinite (A brief explanation is given in Appendix A.2). The entries of this matrix and the coefficients of the polynomial should satisfy some linear constraints. Thus the sos conditions in Eq.(5.19a)-(5.19c) indicate that a sufficient condition for a contact wrench $\dot{h} - w_g$ having a margin no smaller than ϵ, is that there exist some polynomials $l_i(a,b)$, such that the polynomials of Eq.(5.19a)-(5.19b) are Sum of Squares. The entries of these polynomials are functions of the contact position p_i, the contact wrench $\dot{h} - w_g$, and coefficients of the polynomials l_i. So Eq.(5.19a)-(5.19c) will be interpreted as several matrix inequalities on decision variable $p_i, \dot{h}, w_g$ and coefficients of l_i. We will search for these variables to satisfy these matrix inequalities.

When we use nonlinear friction cone instead of linearized friction cone, we just need to replace Eq.(5.14) with (5.15) in (5.19a)-(5.19c), and we end up with the following sos conditions

$$\begin{aligned}
&l_0(a,b)(a^T(\dot{h} - w_g) + b - \epsilon) - l_1(a,b)(a^T T(p_w, I) Q_w^{-1} T(p_w, I)^T a - 1) \\
&- l_2^i(a,b)(a^T G(p_i) c_i) - l_3^i(a,b)\left((a^T G(p_i) c_i)^2 - \mu^2 (a^T G(p_i)(I_{3\times 3} - c_i c_i^T) G(p_i)^T a)\right) \\
&- l_4(a,b) b - l_5^j(a^T \bar{v}_i^j + b) \text{ is sos} \qquad (5.20a) \\
&l_0(a,b) - 1 \text{ is sos} \qquad (5.20b) \\
&l_2^i(a), l_3^i(a,b), l_4(a,b), l_5^j(a,b) \text{ is sos} \qquad (5.20c)
\end{aligned}$$

Again the constraints (5.20a)-(5.20c) is equivalent to the existence of some positive semidefinite matrices, such that the matrix inequalities corresponding to (5.20a)-(5.20c) are satisfied.

We will see in Section 5.3.1 and 5.4.1 that we have more variations on the conditions when computing the contact wrench set margin, but they all share the general form that some matrix inequalities have to be satisfied, and these matrices are functions of contact position p_i, the contact wrench $\dot{h} - w_g$, and coefficients of polynomials $l_i(a, b)$.

5.2 Solving kinematic/dynamic constraints with contact wrench set margin

To find the robot motion such that its contact wrench set margin is maximized, we will formulate an optimization problem with two types of constraints

1. Contact wrench set margin constraint as in Eq.(5.19a)-(5.19c). Such constraints are matrix inequalities, typically solved as linear matrix inequalities (LMI) if the matrix inequalities are also linear in decision variables.

2. Kinematic and dynamic constraints of robot. These are vector-valued nonconvex differentiable constraints, typically solved by gradient-based nonlinear optimization.

We need to solve these two types of constraints together, while conventionally they are solved separately as different type of optimization problems. Here we propose two approaches to unify these two types of constraints:

1. To solve the matrix inequalities as vector-valued nonlinear differentiable constraints.

2. To solve the vector-valued nonlinear constraints as matrix inequalities.

We will show each approach in the subsequent two sections.

5.2.1 Solve matrix inequality as nonlinear differentiable constraints

A simple way to enforce the matrix inequality $M \succeq 0$, is to write it as the product of a matrix and its transpose

$$M \succeq 0 \Leftrightarrow M = NN^T \tag{5.21}$$

where $M \in \mathbb{R}^{n\times n}$. Without lose of generality, we can assume that $N \in \mathbb{R}^{n\times n}$ is an upper triangular matrix, namely $N_{ij} = 0$ if $i > j$. There are $\frac{n(n+1)}{2}$ non-zero entries in the upper triangular matrix N, so for each psd matrix $M \in \mathbb{R}^{n\times n}$, we introduce $\frac{n(n+1)}{2}$ new variables, and we replace the psd matrix M by NN^T as in Eq.(5.21). This parameterization of psd constraint has smooth gradient over entries of N, and thus can be solved by gradient-based nonlinear optimization solvers. This trick allows us to combine the matrix inequalities as in contact wrench margin computation, with the kinematic and dynamic constraints of the robot, and to solve them together as nonlinear optimization problem.

5.2.2 Solve nonlinear constraints as matrix inequalities

There exists a technique to solve certain type of nonlinear optimization, as matrix inequalities, called *Bilinear Matrix Inequality* (BMI). A general form of BMI is the following matrix inequality on variables x

$$F_0 + \sum_i x_i F_i + \sum_{i,j} x_i x_j F_{ij} \succeq 0 \tag{5.22}$$

where $F_i, F_{ij} \in \mathbb{R}^{n\times n}$ are all given matrices. Unlike the linear matrix inequality $F_0 + \sum_i x_i F_i \succeq 0$ that is a convex function of x, the bilinear matrix inequality is non-convex, but it can be solved iteratively by a sequence of semidefinite programming problems [71]. We give a brief introduction on BMI and the algorithm in the Appendix A.3.

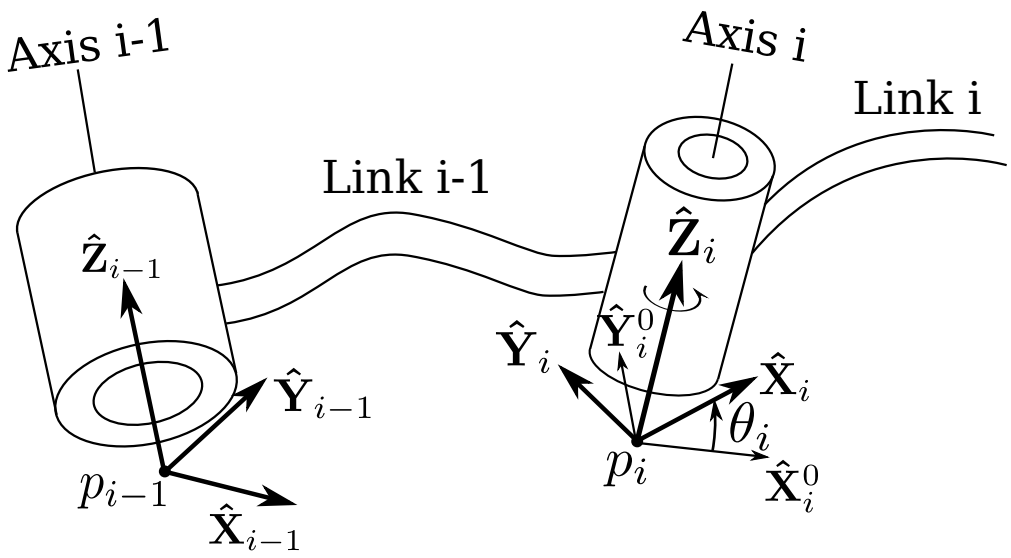


Figure 5-2: [24] A link frame $\hat{\mathbf{X}}_{i-1}, \hat{\mathbf{Y}}_{i-1}, \hat{\mathbf{Z}}_{i-1}$ is attached to link $i-1$, link frame $\hat{\mathbf{X}}_i, \hat{\mathbf{Y}}_i, \hat{\mathbf{Z}}_i$ is attached to link i. $\hat{\mathbf{Z}}_{i-1}, \hat{\mathbf{Z}}_i$ are the revolute axes of the joints. The axis frame $\hat{\mathbf{X}}_i^0, \hat{\mathbf{Y}}_i^0, \hat{\mathbf{Z}}_0$ is attached to the joint i, and is fixed in the link frame $i-1$. The axis frame i and the link frame i share the frame origin and $\hat{\mathbf{Z}}_i$ axis, the latter frame is obtained by rotating the former by angle θ_i around the $\hat{\mathbf{Z}}_i$ axis.

A rich class of vector-valued nonlinear constraints can be formulated as bilinear matrix inequality. For example, when F_i, F_{ij} are all scalars, Eq.(5.22) represents the general non-convex quadratic constraints. We will show that the kinematic constraints can be formulated as BMIs.

BMI is used extensively in control literatures [144, 159, 53]. We will use this tool in motion planning with kinematic constraints.

Kinematic constraints as bilinear matrix inequalities

We will show in this section, such kinematic constraints can be formulated as BMIs, using robot maximal coordinates.

We illustrate the kinematic chain between two links, welded by a revolute joint as in Fig.5-2. The orientation of the link frame $i-1, i$ are represented by unit quaternions q_{i-1}, q_i, and the position of frame origins are $p_{i-1}, p_i \in \mathbb{R}^3$ respectively. The transformation from the axis frame i to the link frame $i-1$ is fixed, with a given unit quaternion $z_{i-1,i}$ for the rotation, and a given vector $p_{i-1,i}$ for the translation,

both expressed in link frame $i-1$. We introduce two additional variables $\hat{c}_i, \hat{s}_i$, to represent $\cos\frac{\theta_i}{2}, \sin\frac{\theta_i}{2}$ respectively. The rotation of the axis i is thus expressed by the unit quaternion $z_{\theta_i} = \hat{c}_i + 0\mathbf{i} + 0\mathbf{j} + \hat{s}_i\mathbf{k}$. The relationship between link frame $i-1$ and i are

$$p_i = R(q_{i-1})p_{i-1,i} + p_{i-1} \tag{5.23a}$$

$$q_i = q_{i-1} \otimes z_{i-1,i} \otimes z_{\theta_i} \tag{5.23b}$$

$$q_i \otimes q_i^* = 1, q_{i-1} \otimes q_{i-1}^* = 1 \tag{5.23c}$$

where $R(q_{i-1})$ is the rotation matrix for unit quaternion q_{i-1}.

$$R(z) = \begin{bmatrix} z_2^2 + z_1^2 - z_4^2 - z_3^2 & 2z_2z_3 - 2z_1z_4 & 2z_2z_4 + 2z_1z_3 \\ 2z_2z_3 + 2z_1z_4 & z_3^2 + z_1^2 - z_2^2 - z_4^2 & 2z_3z_4 - 2z_1z_2 \\ 2z_2z_4 - 2z_1z_3 & 2z_3z_4 + 2z_1z_2 & z_4^2 + z_1^2 - z_2^2 - z_3^2 \end{bmatrix} \tag{5.24}$$

$\otimes$ is the Hamilton product for quaternions, defined as

$$q \otimes p = \begin{pmatrix} q_1p_1 - q_2p_2 - q_3p_3 - q_4p_4 \\ +(q_1p_2 + q_2p_1 + q_3p_4 - q_4p_3)\mathbf{i} \\ +(q_1p_3 - q_2p_4 + q_3p_1 + q_4p_2)\mathbf{j} \\ +(q_1p_4 + q_2p_3 - q_3p_2 + q_4p_1)\mathbf{k} \end{pmatrix} \tag{5.25}$$

In Eq.(5.23c), we denote q^* as the conjugate of quaternion q. Eq.(5.23c) means that the quaternion has unit length, so they can represent orientation in 3D.

The constraints on $\hat{c}_i, \hat{s}_i$ are

$$\hat{c}_i^2 + \hat{s}_i^2 = 1 \tag{5.26a}$$

$$\hat{c}_i \in \text{range}\left(\cos\frac{\theta_i}{2}\right), \hat{s}_i \in \text{range}\left(\sin\frac{\theta_i}{2}\right),\ \theta_i \in [\underline{\theta}_i,\ \bar{\theta}_i] \tag{5.26b}$$

where constraint (5.26a) guarantees that $\hat{c}_i, \hat{s}_i$ are the values of cosine and sine functions of a certain angle, and constraint (5.26b) encodes the joint limits $[\underline{\theta}_i,\ \bar{\theta}_i]$ for axis

i. Constraints (5.23a)-(5.26b) encode kinematics chain that welds the adjacent links.

Constraint (5.23a)-(5.23c) and (5.26a),(5.26b) are all bilinear matrix inequalities. So we can combine the kinematics problem and the contact wrench set margin computation, and solve them together using a sequential semidefinite programming approach described in Appendix A.3.

In the remaining sections of this chapter, we will see that more motion planning problems can be formulated as BMIs, and thus can be solved by the sequential semidefinite programming approach.

In the next sections, we will see we can apply the approaches described in this chapter to different motion planning problems, including 1) Force closure grasping, 2) Static posture optimization and 3) Trajectory optimization. We will formulate the nonlinear constraints as BMIs in force closure grasping, while for the other two problems, we will solve the matrix inequalities using nonlinear optimization.

5.3 Force Closure Grasp Optimization

We have been talking about Contact Wrench Set margin. In grasping, we can achieve infinite margin if the contact locations achieve force closure, with *unbounded* contact forces within the friction cone. The quality of a force closure grasp is measured by the contact wrench set margin with *bounded* total contact forces. We will apply our optimization approach to search for an optimal force closure grasp in this section.

5.3.1 Force closure grasp

Force closure, which measures the ability of a grasp to resist wrench disturbances, is an important property in grasping and has an extensive literature [119, 118]. A commonly observed fact is that synthesis of force closure grasps is a non-convex optimization problem, mostly due to the fact that computing the torque on an object involves a bilinear product between contact locations and contact forces. As a result, most approaches resort to gradient-based non-convex nonlinear optimization to synthesize a force closure grasp [21]. On the other hand, when fixing the contact lo-

cations, checking if the given contact locations achieve force closure becomes a convex optimization problem [15, 54]. Moreover, several grasp metrics have been introduced to measure the quality of a force closure grasp. These involve computing the smallest wrench that the grasp cannot resist with bounded contact forces [85, 43, 112]. Liu et al. optimized the contact locations based on this metric by solving a min-max problem through nonlinear optimization, using certain sub-gradient function [103].

The force closure property for n grasp points $p_i \in \mathbb{R}^3, i = 1, ..., n$, is achieved when these grasp points can resist arbitrary external wrenches with contact forces f_i at point p_i lying within the friction cone. Mathematically, force closure is formulated as the existence of p_i and f_i satisfying the following constraints:

$$GG' \succeq \alpha I_{6\times 6} \tag{5.27a}$$

$$Gf = 0 \tag{5.27b}$$

$$f_i \in \text{int}(\mathcal{FC}_i) \tag{5.27c}$$

$$p_i \in \mathcal{S}_i \tag{5.27d}$$

where

$$G = \begin{bmatrix} I_{3\times 3} & I_{3\times 3} & \dots & I_{3\times 3} \\ \lfloor p_1 \rfloor_\times & \lfloor p_2 \rfloor_\times & \dots & \lfloor p_n \rfloor_\times \end{bmatrix}, \tag{5.28}$$

$\lfloor p_i \rfloor_\times$ is the skew-symmetric matrix representing the cross product $\lfloor p_i \rfloor_\times f_i = p_i \times f_i$, α is a small given positive scalar, constraint (5.27a) is the same as G being full rank; $f = [f_1^T \, f_2^T \, \dots \, f_n^T]^T \in \mathbb{R}^{3n}$; $\text{int}(\mathcal{FC}_i)$ is the interior of the friction cone $\mathcal{FC}_i$ at grasp point p_i, and $\mathcal{S}_i$ is the admissible contact region of grasp point p_i (for example, the surface of the object being grasped).

We note that condition (5.27a) is quadratic on p_i, and (5.27b) is bilinear on p_i and f_i. Unlike some existing approaches that fix the contact points p_i and search only contact force f_i through convex optimization, we can search both p_i and f_i simultaneously by solving these BMIs through sequential SDP, as introduced in Section A.3.1. In the following two subsections, we will show that friction cone constraint (5.27c)

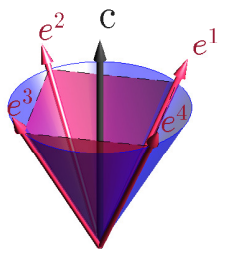


Figure 5-3: A nonlinear friction cone (blue) and the linearized friction cone (red).

and contact region constraint (5.27d) can also be formulated as BMIs.

5.3.2 Friction cones

The Coulomb friction cone constraints introduced in 3.1 (Fig.5-3) can also be formulated as BMIs. For the i^{th} friction cone $\mathcal{FC}_i$, we will constrain c_i to have unit length:

$$c_i^T c_i = 1. \tag{5.29}$$

If we use the nonlinear friction cone, then $f_i \in \text{int}(\mathcal{FC}_i)$ is equivalent to

$$f_i^T c_i > \frac{1}{\sqrt{\mu^2+1}}|f_i|, \tag{5.30a}$$

where μ is the fixed friction coefficient.

If c_i was fixed, constraint (5.30a) would be a *Second-order cone constraint* on f_i, which is a special type of psd constraint [2]. By searching both c_i and f_i, constraints (5.29)(5.30a) are both BMIs on variables c_i and f_i.

If $\mathcal{FC}_i$ is a linearized friction cone with n_e edges, to compute its edges, we can first construct a cone $\mathcal{FC}_0$ that has unit axis $c_0 = [0\ 0\ 1]^T$, with edges $e_0^1, e_0^2, \ldots, e_0^{n_e}$. Without loss of generality we suppose the projection of the edge onto the normal direction has unit length, $c_0^T e_0^j = 1, j = 1, \ldots, n_e$. The edge e_0^j can be computed using the friction coefficient and c_0, thus they are fixed. The linearized friction cone at x_i with cone axis c_i , can be obtained by appropriately rotating cone $\mathcal{FC}_0$ such that cone axis c_0 is aligned with c_i. A rotation can be parameterized with a unit quaternion z_i,

satisfying constraints:

$$z_i \otimes z_i^* = 1 \tag{5.31a}$$

$$c_i = R(z_i)c_0 \tag{5.31b}$$

where $\otimes$ is the Hamiltonian product between quaternions. z_i^* is the conjugate of z_i, and $R(z_i) \in \mathbb{R}^{3\times3}$ is the rotation matrix corresponding to z_i, each entry in $R(z_i)$ is a second-order polynomial of z_i (Eq.(5.24)) [146]. Applying the same rotation to the friction cone edges e_0^j generates the friction cone edges e_i^j at x_i.

$$e_i^j = R(z_i)e_0^j, \quad j = 1, \ldots, n_e. \tag{5.32}$$

The contact force f_i is a positive weighted sum of the edges of the friction cone:

$$f_i = \sum_{j=1}^{n_e} w_i^j e_i^j, \quad w_i^j > 0 \tag{5.33}$$

Constraints (5.31a)-(5.33) involve only bilinear terms of the decision variables z_i, w_i^j, e_i^j, and thus can be posed as a BMI.

5.3.3 Contact geometries

In this section, we consider four types of objects to be grasped, including convex polyhedra (Fig.5-4,5-5), spheres, ellipsoids and cylinders (Fig.5-6). The constraints on contact point p_i and contact normal c_i are straight-forward for the sphere, ellipsoid and cylinder, since the contact surfaces for these geometries are all parameterized by quadratic functions. Thus the constraints on p_i and c_i are also quadratic, and can be solved as BMIs. When the object is a polyhedron, and the grasp is free to choose any facets, the problem becomes trickier to handle, and we will discuss it below.

For a convex polyhedron $\mathcal{P} = \text{ConvexHull}(v_p^1, \ldots, v_p^{N_p})$ (The red box in Fig.5-4), where v_p^i is the i^{th} vertex of the polyhedron, we want to avoid contacts lying at edges or corners of the polyhedron, since these contact locations be unstable and the object

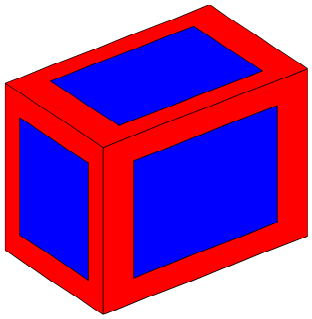


Figure 5-4: The polyhedron $\mathcal{P}$ to be grasped. The admissible contact regions are the shrunk regions on each facets (blue region).

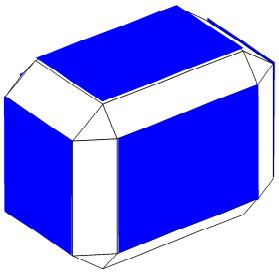


Figure 5-5: The shrunk polyhedron $\mathcal{P}_s$ obtained as the convex hull of the blue regions, which are the shrunk regions on each facets as in Fig.5-4.

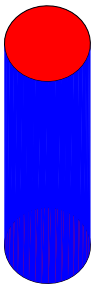


Figure 5-6: The cylinder to be grasped, the blue surface is the grasp region.

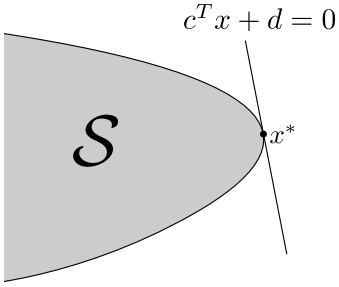


Figure 5-7: A point is on the boundary of a convex set, if and only if a hyperplane supports the set at that point.

can slide out of the grasp. Thus the admissible contact regions are given as the shrunk surface regions (blue shades). We then construct a shrunk polyhedron (Fig.5-5) as the convex hull of the shrunk surface regions (blue shades). The shrunk polyhedron is given as $\mathcal{P}_s = \{x | A_s x \leq b_s\}$; this *H-representation* of a polyhedron can be readily computed from its vertices [181]. To constrain p_i lying on one of the shrunk surface regions, we use the fact that a point is on the surface of a convex object, if and only if a supporting hyperplane intersects the object at that point (Fig.5-7). Thus we introduce a supporting hyperplane $\{x | c_i^T x + d_i = 0\}$, where c_i is the axis of the friction cone, and the constraints:

- The grasp point p_i is on the hyperplane

$$c_i^T p_i + d_i = 0 \tag{5.34}$$

- All vertices of the original polyhedron $\mathcal{P}$ lie on one side of the hyperplane, and the normal vector c_i points outward from the polyhedron

$$c_i^T v_p^j + d_i \leq 0 \quad \forall j = 1, \ldots, N_p \tag{5.35}$$

- The grasp point lies within the shrunk polyhedron $\mathcal{P}_s$

$$A_s p_i \leq b_s \tag{5.36}$$

Geometrically, constraints (5.34)-(5.36) state that $c_i^T x + d = 0$ is a supporting hyperplane of the polyhedron $\mathcal{P}$, and the supporting point p_i is not at edges or corners of the polyhedron, so c_i has to coincide with one of the face normals. We want to highlight that we do not specify on which facet the contact lies; by searching over c_i, p_i and d_i, the optimization program will determine the contact facets by itself. Constraints (5.35)(5.36) are linear on p_i, c_i and d_i. Constraint (5.34) is a BMI on p_i and c_i.

5.3.4 ϵ-ball

The ϵ-ball metric proposed by Kirkpatrick [85] measures the smallest magnitude of wrench disturbance that cannot be resisted, given an upper bound on the total contact forces. We will derive the conditions to compute the ϵ-ball for grasping under total contact force, similar to what we have shown in Sec.5.1.

For contact point $p_i, i = 1, \ldots, n$, and linearized friction cone, whose edges are $e_i^j, j = 1, \ldots, n_e$, we define the wrench set $\mathcal{W}$ as the set of wrench that can be resisted by those contact points, when the total normal contact forces on all contact points are bounded by 1.

$$CWS = \text{ConvexHull}(v_i^j),\ i = 1, \ldots, n, j = 1, \ldots, n_e, \text{ where } v_i^j = \begin{bmatrix} e_i^j \\ p_i \times e_i^j \end{bmatrix}. \tag{5.37}$$

If nonlinear fiction cone is used, then the Contact Wrench Set is computed as follows

$$CWS = \left\{ \sum_{i=1}^{n} \lambda_i \begin{bmatrix} f_i \\ p_i \times f_i \end{bmatrix} \middle| (1+\mu^2) \geq f_i^T f_i, f_i^T c_i = 1, \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i \leq 1 \right\} \quad (5.38)$$

If CWS contains the origin in the wrench space, then force closure is achieved. We will suppose that the disturbance wrench is applied at the origin of the world coordinate.

Using the same halfspace interpretation as we explained in Sec.5.1, the contact wrench margin is no smaller than ϵ, if and only if the following implication holds, that if a, b satisfy the condition on the left-hand side of $\Rightarrow$, then a, b would satisfy the condition on the right-hand side of $\Rightarrow$.

- If using the linearized friction cone,

$$\left.\begin{array}{l} (v_i^j)^T a + b \geq 0, \forall i, j \\ a^T Q_w^{-1} a = 1 \end{array}\right\} \Rightarrow b \geq \epsilon. \quad (5.39)$$

- If using the nonlinear friction cone,

$$\left.\begin{array}{l} a^T G(p_i) c_i + b \geq \mu \sqrt{a^T G(p_i)(I_{3\times 3} - c_i c_i^T) G(p_i)^T a} \\ a^T Q_w^{-1} a = 1 \end{array}\right\} \Rightarrow b \geq \epsilon. \quad (5.40)$$

Like we present in Sec.5.1, the implication in (5.39) and (5.40) holds, if there exists some polynomials $l_i(a, b)$, that satisfy the following Sum of Squares (sos) condition

- If using linearized friction cone

$$\begin{aligned} & l_0(a,b)(b-\epsilon) - l_1(a,b)(a^T Q_w^{-1} a - 1) \\ & - \sum_{1\leq i\leq n} \sum_{1\leq j\leq n_e} l_2^{i,j}(a,b)((v_i^j)^T a + b) \text{ is sos} \quad (5.41a) \\ & l_0(a,b) - 1 \text{ is sos} \quad (5.41b) \\ & l_2^{i,j}(a,b) \text{ is sos } \forall i, j \quad (5.41c) \end{aligned}$$

- If using nonlinear friction cone

$$l_0(a,b)(b-\epsilon) - l_1(a,b)(a^T Q_w^{-1} a - 1) - \sum_{i=1}^{n} l_2^i(a,b)(a^T G(p_i)c + b)$$
$$- \sum_{i=1}^{n} l_3^i(a,b)\left((a^T G(p_i)c_i + b)^2 - \mu^2(a^T G(p_i)(I_{3\times 3} - c_i c_i^T)G(p_i)^T a)\right) \text{ is sos} \tag{5.42a}$$

$$l_0(a,b) - 1 \text{ is sos} \tag{5.42b}$$

$$l_3^i(a,b), l_2^i(a,b) \text{ is sos } \forall i \tag{5.42c}$$

where $l_i(a,b)$ are polynomials on indeterminates a, b.

Since constraints (5.41a)-(5.41c) and (5.42a)-(5.42c) are sufficient conditions of (5.37) and (5.38), for p_i and e_i^j satisfying constraints (5.41a)-(5.42c), ϵ is a lower bound of its ϵ-ball metric. To maximize ϵ, we can use either bilinear alternation (Algorithm 1) or binary search (Algorithm 2).

Algorithm 1 Bilinear alternation

Start with a force closure grasp p_i, e_i^j, c_i found using approach described in sections 5.3.1-5.3.3, set $\epsilon = 0$, search for $l_i(a,b)$ that satisfy the sos conditions (5.41a)-(5.41c) and (5.42a)-(5.42c). These are linear matrix inequalities, and can be solved by convex semidefinite programming.

while $r \neg$ converged **do**

1. At iteration k, fix $p_i, c_i, l_0(a,b)$ in constraint (5.41a) or (5.42a), search for $l_i(a,b), i \neq 0$ and ϵ to maximize ϵ, subject to constraints (5.41a),(5.41c) or (5.42a),(5.42c). This optimization is a semi-definite programming problem. It finds an ellipsoid contained in the Contact Wrench Set.

2. Fix ϵ and $l_i(a,b), i > 1$ to the solution in step 1, find feasible $p_i, c_i, l_0(a,b), l_1(a,b)$ that satisfy (5.41a),(5.41b),(5.42a),(5.42b) and the kinematics and contact location constraints on p_i, c_i, as described in in section 5.3.3, 5.2.2. This is a BMI problem. It finds grasp points p_i and friction cone axis c_i, such that the grasp quality is no worse than that in the previous iteration. The solution $p_i, c_i, l_0(a,b)$ will be used in step 1 in the next iteration.

end while

In the bilinear alternation, the k^{th} iteration is guaranteed to obtain an objective ϵ that is at least as good as the previous iteration, since a solution to step 2 in iteration

k is also feasible for step 1 in both iteration k and $k+1$; also ϵ cannot increase to infinity. Hence the bilinear alternation will terminate with convergence to an optimal value, possibly just a local optimum. It is a common strategy in the SDP literature [128, 157, 106].

Algorithm 2 Binary search

Start with $\underline{\epsilon} = 0$, and $\bar{\epsilon}$ to be some big value, $\epsilon = \frac{\bar{\epsilon}+\underline{\epsilon}}{2}$.
while $\epsilon \neg$ converged **do**

1. Fix ϵ, search for the coefficients of $l_i(a,b)$, p_i, c_i together, subject to constraints (5.41a)-(5.41c) or(5.42a)-(5.42c) and the kinematics and contact location constraints on p_i, c_i in section 5.2.2,5.3.3. This is a BMI problem. If the problem converges, set the lower-bound $\underline{\epsilon} = \epsilon$; otherwise set the upper-bound $\bar{\epsilon} = \epsilon$.
2. $\epsilon = \frac{\bar{\epsilon}+\underline{\epsilon}}{2}$, go to step 1.

end while

The binary search algorithm needs to deal with psd constraints of larger size than that in bilinear alternation, since it involves the product of $l_i(a,b)$ and c_i, p_i. Thus the binary search algorithm takes longer time to solve each SDP. Experimentally, we find that the binary search algorithm is less susceptible to local minima than the bilinear alternation alone.

5.3.5 Results

Force closure contact

We first show that we can synthesize a feasible solution to achieve force closure in this sub-section, we will later demonstrate the results to optimize the force closure grasp in the next sub-section.

We show the results of finding force closure contact locations on different geometries in Fig.5-8. We also show the time scalability w.r.t number of contacts in Fig.5-9, 5-10, and number of polyhedron facets in Fig.5-11. When we increase the number of contacts (Fig.5-9, 5-10), the size of the largest psd constraints remains the same, and the number of psd constraints increases linearly. As expected, the computation time in each SDP scales linearly (Fig.5-10); and empirically we observe that the number

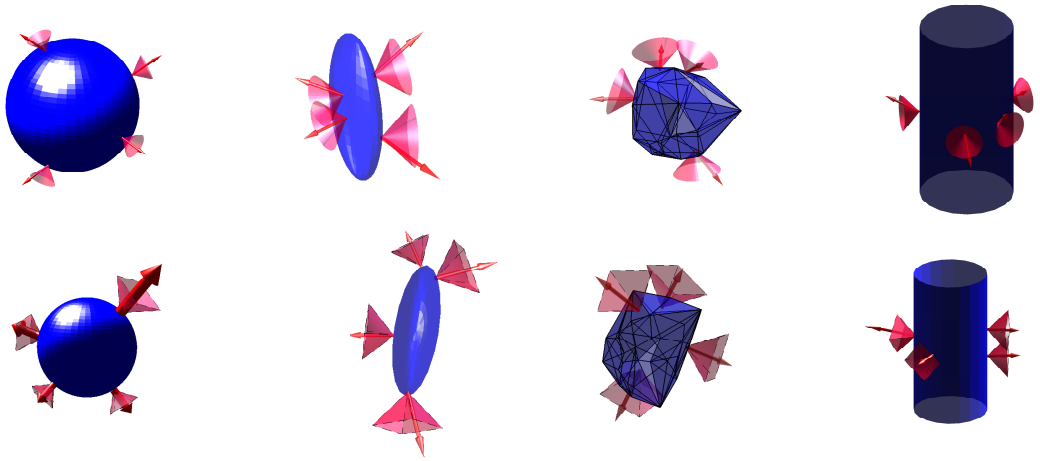


Figure 5-8: Force closure contacts on different geometries. The upper row uses nonlinear friction cone, the lower row uses linearized friction cone. For the polyhedron (column 3), the contact facets are not specified by the user beforehand.

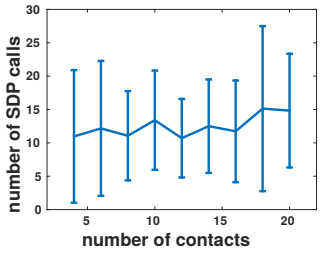


Figure 5-9: Scalability w.r.t number of contacts on a 30 facets polyhedron.

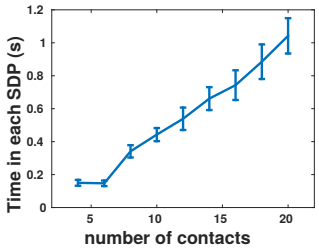


Figure 5-10: Scalability w.r.t number of contacts on a 30 facets polyhedron.

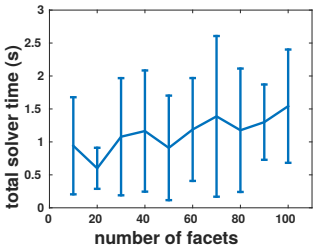


Figure 5-11: Scalability w.r.t number of facets, test with 4 contacts.

of SDP calls remains almost constant (Fig.5-9). As a result, the total computation time scales linearly w.r.t number of contacts. On the other hand, the number of polyhedron facets does not affect the size or the number of the psd constraints, so the total computation time remains almost constant (Fig.5-11).

Kinematics

We will first show that we can solve the inverse kinematics (IK) problem by formulating it as BMIs, and the sequential semidefinite programming can generate certificate for global infeasibity in some cases. Here we focus on the IK problem only, and do not include the force closure grasping. The inverse kinematics problem is solved for

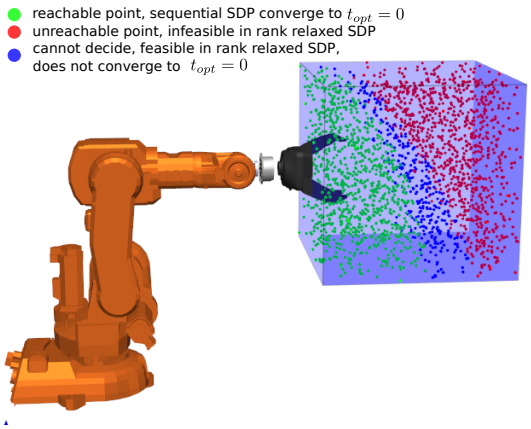


Figure 5-12: Robot arm reachability.

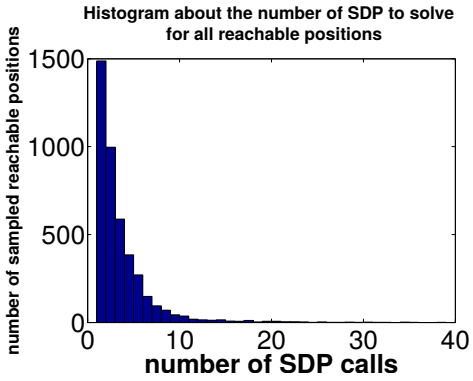


Figure 5-13: Histogram on number of SDP calls.

an ABB IRB140 arm with a Robotiq hand, with 15 joints in total. To evaluate how effective the algorithm is as solving this inverse kinematics problem, in Fig.5-12 we take 10000 samples within the 0.6m x 0.6m x 0.6m box in the shaded region, and require the center of the palm to reach the sample point. There are three possible outcomes from the sequential SDP.

- Green dot: sequential SDP converges to $t_{opt} = 0 \Rightarrow$ feasible BMIs, thus reachable.
- Red dot: the rank-relaxed SDP reports infeasibility, thus proved unreachable.
- Blue dot: the rank-relaxed SDP is feasible, but the sequential SDP does not converge to $t_{opt} = 0$.

As shown in Fig.5-12, the blue dot layer is thin, showing that in most cases the sequential SDP algorithm either solves the problem or proves that the problem is infeasible. The histogram in Fig.5-13 shows that when the sequential SDP can solve the problem, in most (81.36%) cases it converges within 5 SDP calls. The average time to solve the BMI is 0.25 seconds using MOSEK [116] on an Intel i7 machine.

Grasp optimization

In this sub-section, we demonstrate the result on optimizing the force closure contacts location and grasping postures.

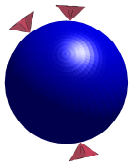


Figure 5-14: Initial contacts and linearized friction cones.

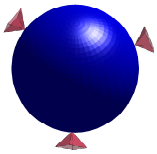


Figure 5-15: Optimized contacts and linearized friction cones.

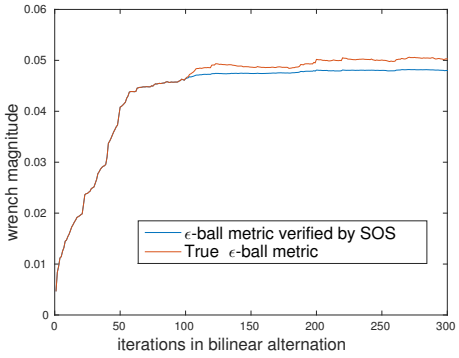
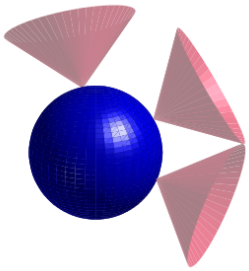
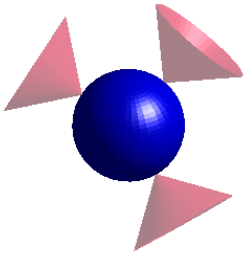


Figure 5-16: The change of ϵ-ball metric in each iteration of bilinear alternation.



(a) Initial contacts and the nonlinear friction cones.



(b) Optimized contacts and the nonlinear friction cones.

Figure 5-17: Force closure contacts with nonlinear friction cone

We first show the result of using bilinear alternation to optimize a 3-point force closure grasp on a sphere. The initial contacts and linearized friction cones are plotted in Fig.5-14, the optimized contacts become more evenly distributed (Fig.5-15), as is known to be the better 3-point grasp on the sphere [103]. In Fig.5-16 we draw the ϵ-ball metric in each iteration. The SOS programs (5.41a)-(5.41c),(5.42a)-(5.42c) find a lower bound of the ϵ-ball metric. The true ϵ-ball metric is computed as in Appendix. We can see that the gap between the SOS verified lower bound and true ϵ-ball metric is small. The computation time is 172 seconds using MOSEK solver [116] on an Intel i7 machine.

We also show that we can optimize the force closure contact points using nonlinear friction cone, as shown in Fig.5-17

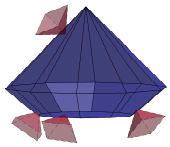


Figure 5-18: Initial contacts and linearized friction cones.

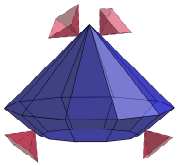


Figure 5-19: Optimized contacts and linearized friction cones.

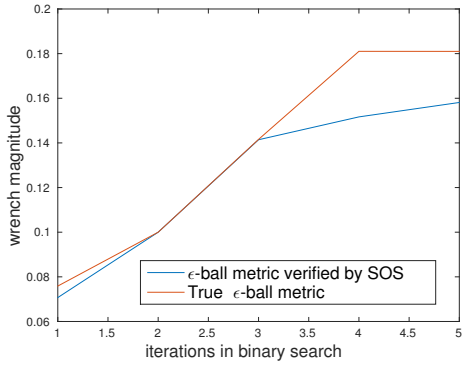


Figure 5-20: The change of ϵ-ball metric in each iteration of binary search.

We also show the result of optimizing force closure contacts on a diamond shaped polyhedron, through binary search. The optimized contacts (Fig.5-19) are more evenly distributed than the initial contacts (Fig.5-18). Also we want to highlight that the facets on which the contacts lie are changed through optimization, this again demonstrates that the optimization program can search over *all* facets by itself. The computation time is around an hour using MOSEK solver on an Intel i7 machine.

We show the result of optimizing the force closure grasp with Robotiq hand and ABB arm on a cylinder, with linearized friction cones. The initial posture grasps the tip of the cylinder (Fig.5-21), the optimized posture gets improved by grasping the center of the cylinder (Fig.5-22). The computation time is around 20 minutes using MOSEK on an Intel i7 machine.

5.4 Static Posture Optimization

We will show that we can optimize the robot posture, such that it can maintain static equilibrium under large Center of Mass position error. We will maximize the margin between the Center of Mass projection to the boundary of the support region. As we will see shortly, this margin is a special case of the Contact Wrench Set margin, thus we can optimize the robot posture to enlarge this margin. We will solve this optimization problem through nonlinear optimization.

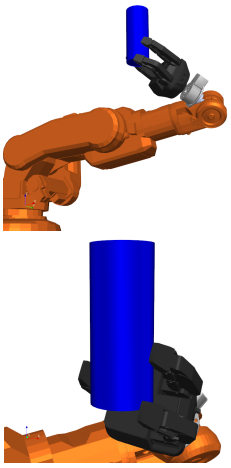


Figure 5-21: initial force closure grasp from two views.

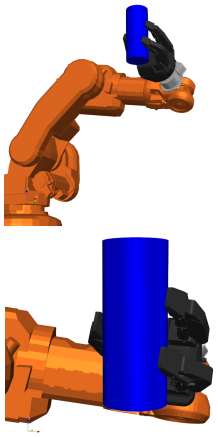


Figure 5-22: optimized force closure grasp from two views.

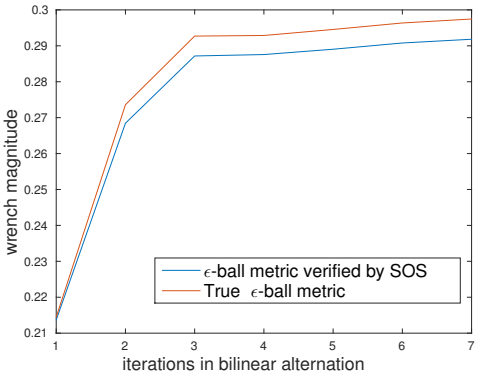


Figure 5-23: The change of ϵ-ball metric in each iteration of bilinear alternation, for robotiq hand grasping the cylinder.

5.4.1 Static Equilibrium

The robot can maintain static equilibrium, if the contact wrench can balance the gravitational wrench, namely, the negation of gravitational wrench is within the Contact Wrench Set.

$$-w_g(r) = \begin{bmatrix} 0 & 0 & mg & r_y mg & -r_x mg & 0 \end{bmatrix}^T \in CWS \tag{5.43}$$

here we write the gravitational wrench w_g as a function of CoM position r.

The static equilibrium is achieved if the Center of Mass horizontal position is within a region, called the support region, as illustrated in Fig.5-24.

We want the robot to main static equilibrium *robustly*, i.e, the CoM horizontal position not close to the boundary of the support region. To this end, we want to find the robot posture, together with the contact locations, such that the robot keeps a large distance from the Center of Mass to the boundary of the support region.

A shift of Center of Mass position is equivalent to applying a wrench disturbance in some particular form (Eq.(5.45)), due to the linearity of gravitational wrench $w_g(r)$ in Eq.(5.43). We can define the support region margin as the radius of the largest ball, centered at current CoM position r, such that when the robot CoM position is

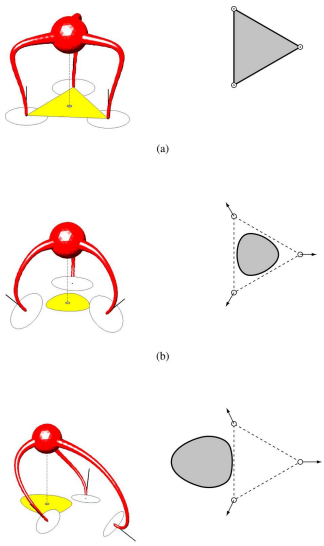


Figure 5-24: Support region for CoM [17]

within this ball, the gravitational wrench is still inside the Contact Wrench Set.

$$\mathcal{B}_\epsilon = \left\{ -w_g(r+\delta) \middle|\, \delta_x^2 + \delta_y^2 \leq \epsilon^2 \right\} \in CWS \tag{5.44}$$

$$\text{where } -w_g(r+\delta) = -w_g(r) + \begin{bmatrix} 0 & 0 & 0 & \delta_y mg & -\delta_x mg & 0 \end{bmatrix}^T \tag{5.45}$$

When the ellipsoid $\mathcal{B}_\epsilon$ lies within the Contact Wrench Set, it is guaranteed that the support region margin is no smaller than ϵ.

To see how this support region margin is connected to the Contact Wrench Set margin, we can slice the Contact Wrench Set with a plane passing the contact wrench $-w_g(r)$, and spanned by vectors $[0\ 0\ 0\ 1\ 0\ 0]^T$ and $[0\ 0\ 0\ 0\ 1\ 0]^T$, the contact wrench set margin on this plane is the support region margin.

Sum of Squares condition

As we derived the Sum of Squares condition for contact wrench margin in Sec 5.1 using halfspaces, we can similarly derive the sos condition for static equilibrium margin. We can show that when we use the linearized friction cones, the static equilibrium margin

is no less than ϵ, if and only if the following implication holds.

$$\left.\begin{array}{l} a^T v_i^j \geq 0, i = 1, \ldots, n_1, j = 1, \ldots, n_e \\ b \geq 0 \\ a^T \bar{v}_j + b \geq 0, j = 1 \ldots, n_e^{n_2} \\ a_4^2 + a_5^2 = 1 \end{array}\right\} \Rightarrow (a_3 + a_4 r_y - a_5 r_x) mg + b \geq \epsilon mg \quad (5.46)$$

where $v_i^j = \begin{bmatrix} (e_i^j)^T & (p_i \times e_i^j)^T \end{bmatrix}^T$ is the wrench corresponding to the friction cone edge e_i^j at contact position p_i, as described in Eq.(5.11). $\bar{v}_j$ is the wrench corresponding to the friction cone with *bounded* magnitude.

It is worth mentioning that if all the friction cones are unbounded, then we do not need to consider the condition on $\bar{v}_j$ on the left-hand side of $\Rightarrow$ in Eq.(5.46). Then we can get rid of the indeterminates b, and the implication is

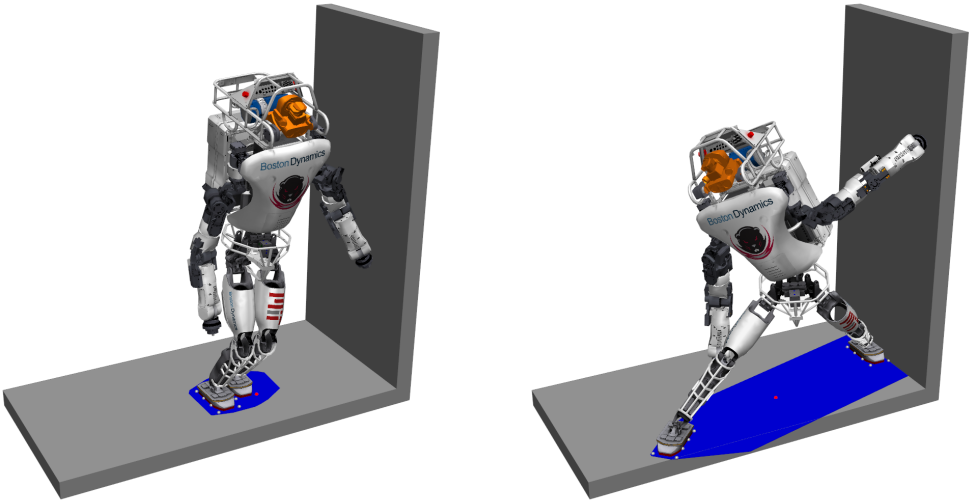
$$\left.\begin{array}{l} a^T v_i^j \geq 0, i = 1, \ldots, n_1, j = 1, \ldots, n_e \\ a_4^2 + a_5^2 = 1 \end{array}\right\} \Rightarrow a_3 + a_4 r_y - a_5 r_x \geq \epsilon \quad (5.47)$$

As we explained in Sec.5.1, the implication in Eq.(5.46) holds if there exists polynomials $l_i(a, b)$, such that the following sos conditions hold

$$\begin{aligned} & l_0(a,b)\left((a_3 + a_4 r_y - a_5 r_x) mg + b - \epsilon mg\right) - l_1(a,b)(a_4^2 + a_5^2 - 1) \\ & - \sum_{i=1}^{n} \sum_{j=1}^{n_e} l_2^{i,j}(a,b)(a^T v_i^j) - l_3(a,b) b - \sum_{j=1}^{n_e^{n_2}} l_4^j(a,b)(a^T \bar{v}_j + b) \text{ is sos} \quad (5.48) \\ & l_0(a,b) - 1 \text{ is sos} \quad (5.49) \\ & l_2^{i,j}(a,b), l_3(a,b), l_4^j(a,b) \text{ is sos} \quad (5.50) \end{aligned}$$

We will formulate these sos conditions as nonlinear differentiable constraints, using the trick $M \succeq 0 \Leftrightarrow M = NN^T$ as described in 5.2.1.

The robot static posture should satisfy the kinematic constraints also. Here we treat the kinematic constraints as general nonlinear differentiable constraints, com-



(a) Unoptimized posture (b) Optimized posture

Figure 5-25: Atlas postures, its CoM projection (red dots), and support region (blue regions)

monly solved by nonlinear optimization [39, 28]. So the static posture optimization problem can be solved by NLP solvers.

5.4.2 Results

Atlas

We show the postures of Atlas robot with foot and left hand in contact with the environment. The bound on the hand contact force is 200N. In Fig.5-25 we show the unoptimized posture and the optimized one. We deliberately set objective as maximizing the support region margin only, to highlight that our optimization problem can really increase this margin. We could add some other cost to the objective to find a more comfort robot posture.

The support region margin increases from 8.29cm to 27.26cm through optimization. The optimization takes 35 seconds on an Intel Core i7 machine with SNOPT [48]. The code is written in MATLAB.

Little Dog

If the terrain has some complicated geometry, like a stepping stone, we would want our optimizer to search over all facets on the stepping stone, as we did in grasp optimization on a polytope. We use the same support hyperplane idea presented in Sec.5.3.3, so that we are searching for a hyperplane that supports the contact point at the boundary of the stepping stone. The constraints in Sec.5.3.3 are also nonlinear constraints, so we can solve them through nonlinear optimization.

We show the results of posture optimization on four stepping stones in Fig.5-26. The support margin increases from 4.2cm to 8.07cm through optimization. After optimization, the robot not only sprawls its legs, but also switches the facets on which to place its foot. This shows that our optimization can search over all facets of the convex polytope simultaneously.

The optimization time is about 3 minutes on a Intel Core i7 machine with SNOPT. The code is written in MATLAB.

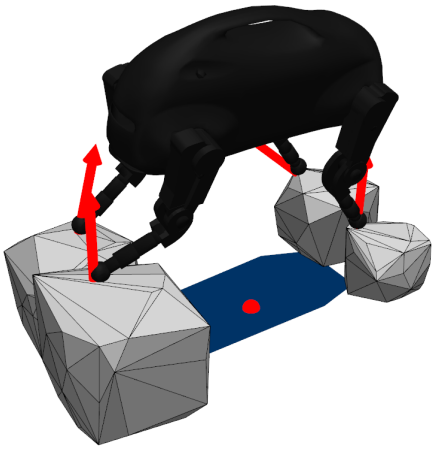
5.5 Trajectory Optimization

Finally, we want to find a trajectory of robot interacting with the environment with multiple contacts, and maximize its contact wrench set margin along the trajectory. We will solve this optimization problem through nonlinear optimization. The computation of the contact wrench set margin has been explained in Sec. 5.1, we will describe the time integration and cost here.

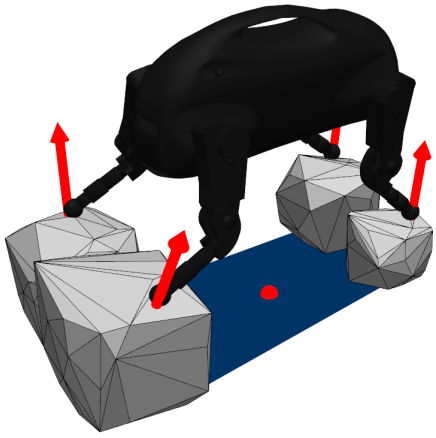
5.5.1 Nonlinear optimization formulation

Time integration

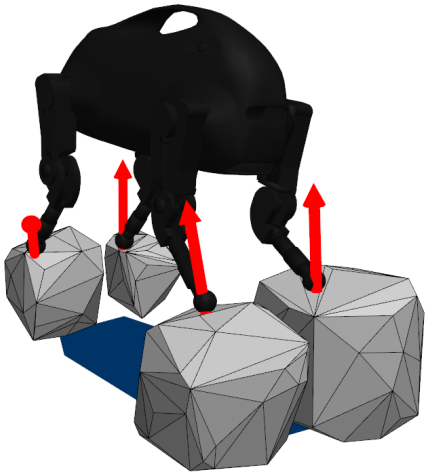
We take N time samples along the trajectory, and parameterize the trajectory using the sampled states at those knot points. We denote the time step from knot i to knot $i+1$ as $dt[i]$. We are going to search for the time steps in the trajectory optimization.



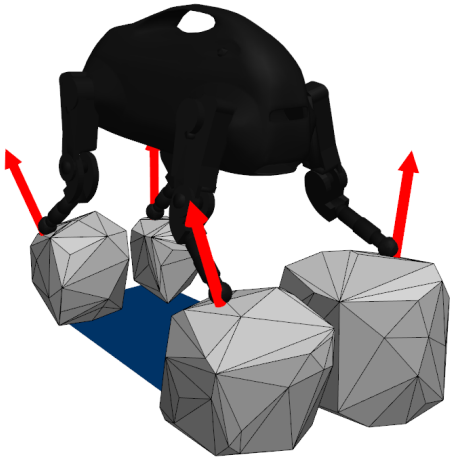
(a) Unoptimized posture, perspective a



(b) Optimized posture, perspective a



(c) Unoptimized posture, perspective b



(d) Optimized posture, perspective b

Figure 5-26: Little dog on stepping stones, with support regions (blue), CoM projection (red dots), and the normal of the contact facet (red arrows).

The robot posture q is integrated from robot velocity v using backward Euler step.

$$q[i] = q[i-1] + v[i]dt[i] \; \forall i = 1, \ldots, N \tag{5.51}$$

The robot linear and angular momentum is integrated using backward Euler

$$h[i] = h[i-1] + \dot{h}[i]dt[i] \tag{5.52}$$

Kinematics and dynamics constraints

We will list the kinematics and dynamics constraints on the robot posture and velocity. Since these constraints should be satisfied at every knot point, we will ignore the time indices i when there is no ambiguity.

The robot center of mass position r is computed from robot whole-body posture using robot kinematics

$$r = com(q) \tag{5.53}$$

The robot linear momentum $m\dot{r}$ and angular momentum k_O are computed from the robot posture and velocity [168, 122].

$$h_G = \begin{bmatrix} m\dot{r} \\ k_G \end{bmatrix} = A(q)v \tag{5.54}$$

$$k_O = k_G + mr \times \dot{r} \tag{5.55}$$

where $A(q)$ is the centroidal momentum matrix.

The robot contact positions p_i are computed from the forward kinematics function FK_i

$$p_i = FK_i(q) \tag{5.56}$$

Objective function

Our goal is to find a reasonable motion, while maximizing the contact wrench set ϵ. We use a general objective function as

$$\min_{\substack{q,v,r \\ dt,\epsilon \\ l_0,l_1,l_2,l_3,l_4}} \sum_{i=1}^{N} -c_\epsilon \epsilon[i] + f(q[i], v[i], r[i], dt[i]) \tag{5.57}$$

where f is a general differential function. It penalizes undesired behavior of the robot, such as large joint velocity, large centroidal angular momentum, etc.

5.5.2 Results

We tested our planner on Atlas robot, with 36 degrees of freedom. Boston Dynamics Inc demonstrated a video of Atlas walking between two walls on `https://www.youtube.com/watch?v=FFGfq0pRczY`, and we plan a similar motion here.

The nonlinear optimization problem has 13,000 decision variables. We first plan a trajectory without maximizing the contact wrench set margin, using the planner introduced in [28]. Then we can solve a convex optimization problem, to find the polynomials $l_i(a, b)$ in Eq. (5.19a)-(5.19c). We then take this feasible trajectories as the initial guess to the optimization problem, to maximize the contact wrench set margin.

The result of the contact wrench set margin is shown in Fig. 5-27. As the unoptimized trajectory does not attempt to maximize the contact wrench set margin, the margins are zero at several sample points, indicating that the motion is marginally stable, and the robot can easily lose balance due to a small disturbance. The optimized trajectory has CWS margin no smaller than 70 at all time samples. The disturbance points are close to the CoM location of the robot. This margin means that the robot can withstand 70 Newton force disturbance. The sampled postures of the initial trajectory and the optimized trajectory are shown in Fig. 5-28. An obvious difference between the un-optimized trajectory, and the optimized one, is that the robot feet sprawl wider in the initial posture of the optimized trajectory, thus

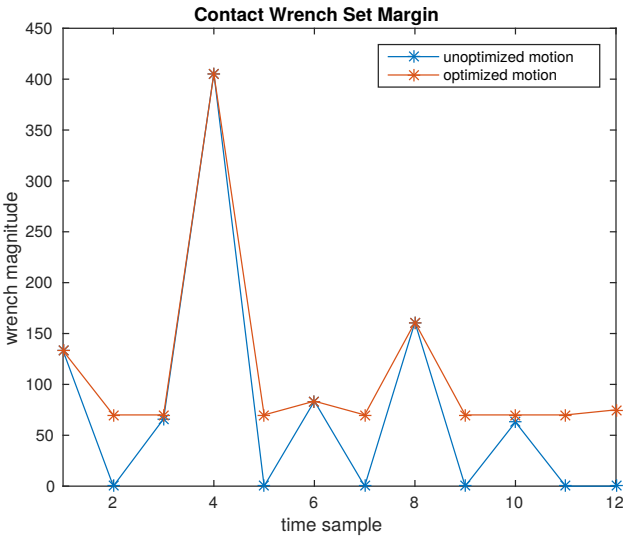
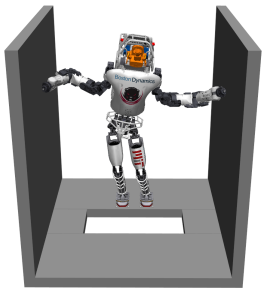
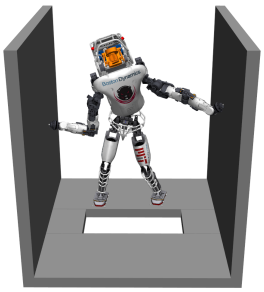


Figure 5-27: Contact Wrench Set Margin

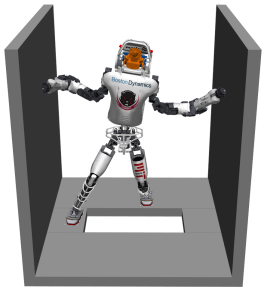
the contact wrench set is larger. Also at the end of the motion, the un-optimized trajectory still has larger acceleration, and its contact wrench is at the boundary of the set; while in the optimized trajectory, the final acceleration is a lot smaller, and thus the contact wrench is in the middle of the contact wrench set.



(a) Un-optimized initial posture



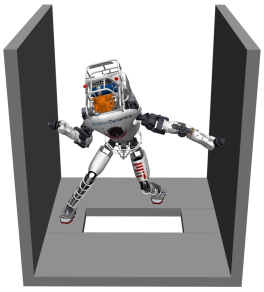
(b) Optimized initial posture



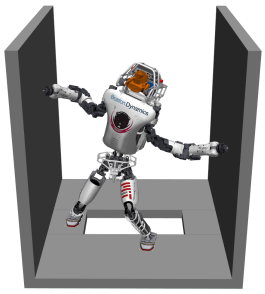
(c) Un-optimized step 1

(d) Optimized step 1

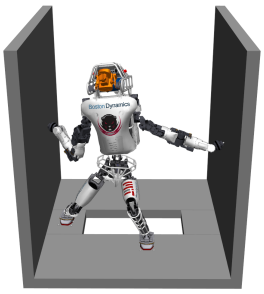
(e) Un-optimized left hand move



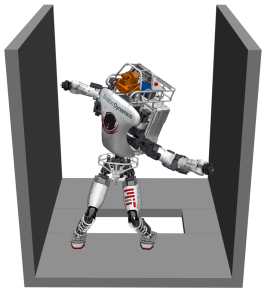
(f) Optimized left hand move



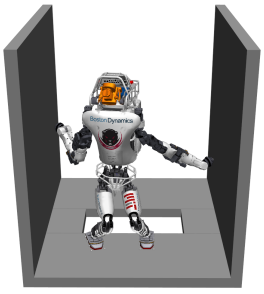
(g) Un-optimized step 2



(h) Optimized step 2



(i) Un-optimized final posture



(j) Optimized step 2

Figure 5-28: Atlas walking between walls

Chapter 6

Feedback controller for grasping

In Section 5.3.1, we described how to synthesize a grasping posture. Unfortunately if we blindly command the robot hand to achieve the desired grasping posture, without considering the pose of the object during grasping, in many cases this strategy will not generate a good grasp. The problem is that when the robot fingers start to touch the object, the object can move. If the grasping strategy ignores the motion of the object, then even if the robot fingers perfectly achieve the desired posture, the contact locations on the object can end up being totally different from the desired ones, and thus the grasp can miss its goal. To this end, we want to design a feedback control policy, that incorporates the dynamics of both the object and the hand. This control policy will adjust the finger motion based on the object motion, and we expect this control policy to be robust, even under uncertainties of the object motion.

We attempted to synthesize a feedback controller for grasping, to track the dynamics of the object while closing the grasp. This controller should be robust to state estimation error on the grasped object. The state estimation error is ubiquitous during grasping, especially when the hand approaches the grasped object, that the fingers and palm occlude the visual features. Occlusion prevents the camera or Lidar from obtaining a good view of the grasped object, when the robot need the accurate position/velocity estimation of the object most. Our proposed solution is to use tactile sensors to measure contact forces, and then feed this contact force measurement directly to the finger controller. We thought this approach is more robust to position

measurement error, compared to using the estimated state of the object from error-prone cameras. We attempted to minimize the set of the final states for the grasped object, by designing this feedback controller through Sum of Squares programming. We encountered numerical difficulties when solving the Sum of Squares problem. We will present our approach and the negative results in this chapter.

6.1 Related Work

There has been a lot of research to use visual feedback to control the robot, such approach is called visual servoing [1, 147, 70, 90] . It typically involves two subsystems, a visual tracking system to estimate the state of the object using visual sensors, and a control system to move the robot manipulator based on the sensory input. Although widely used, visual servoing has its drawbacks, that it needs to keep a good view on the features of the object for the whole time, to get a good estimate of the object motion. Unfortunately when the hands approach the object, those features can get occluded. Also due to the small estimation errors, the visual sensor does not always generate accurate estimation on whether the hand is in contact with the object or not.

Tactile sensors can be used as a complement to visual sensors, to provide local contact information [84, 75, 76]. It is immune to occlusion problems, and also gives a good estimate on whether the contact is active or not. There has been a lot of work to use contact sensors to help estimate the motion of the object [74, 52, 176]. This approach typically involves two stages: first to estimate the state, including hand and object poses and velocities; and then the controller computes the hand joint action based on the estimated state. In this setting, the contact measurement is usually treated as a binary signal, to determine whether the finger touches the object or not. The control action is typically different before and after triggering the contact, thus the control law that maps the robot state to the control action does not share the same formulation [88]. This switch of control policy complicates the problem, as it requires detecting the accurate timing of contact event. In this chapter, we will

present a controller that uses the same control policy before and after the contact event, by using the contact force measurement as a continuous signal, rather than a binary signal. This control policy formulation will enable us to analyze the stability of the closed-loop system using only one Lyapunov function for different contact modes.

Apart from using the two-stage approach to compute the control action from the estimated pose and velocities, people also try to control the robot motion directly. The contact force measurement can be used directly as an input to the controller. In hybrid force/position control [138], the control action directly depends on the error from the desired force. In [72], the robot hand can spin the pen in very high speed, using tactile sensor feedback to control the joint motors. Before and after the impact, although the change on the robot pose is small, the contact force can jump significantly, and a controller depending on the contact force measurement can output different joint command based on the contact force measurement, thus remove the necessity to use a separate control formulation for each individual contact mode. We will use this force feedback approach in this chapter.

To verify the stability of a dynamical system in the Lyapunov sense, people resort to searching for a Lyapunov function using Sum of Squares programming [128, 155]. Such Lyapunov function can verify a region of attraction [155, 161, 157] or an invariant region [158] for a closed-loop system with given controllers. Majumdar et al. exploited the property that a manipulator system is affine in the control actions, and searched the control policy to enlarge the region of attractions through Sum of Squares programming [106]; they demonstrated their approach on systems with continuous dynamics, such as Acrobot. Posa et al. exploited the superposition property of the manipulator dynamics with external force, and used one Lyapunov function to verify convergence or invariance across different hybrid modes [133, 134], we will explain this approach in greater detail in Section 6.2. We will apply Posa's approach to grasping, to verify the stability of the closed-loop hybrid system using one single Lyapunov function.

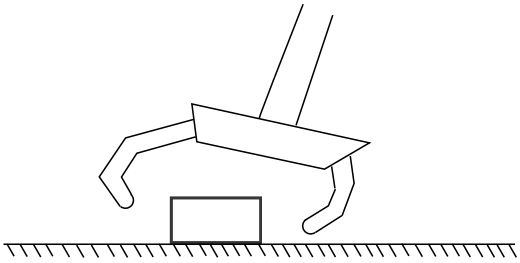


Figure 6-1: Fingers grasping a block

6.2 Approach

6.2.1 Problem formulation

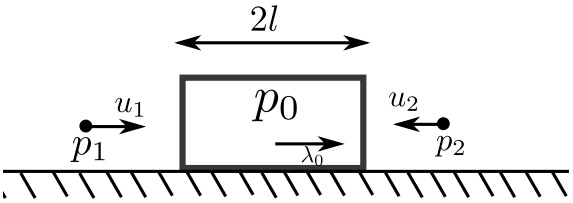
We want to study how to control the robot fingers, so as to move an object on the table, as shown in Fig.6-1. We suppose the robot finger motors accept torque commands, and at the finger tip, the tactile sensor can measure the contact force. Also the finger position and velocity are known from kinematics. For the object, we suppose that its mass and geometry are known. The object can slide on the table, with friction coefficient μ. We suppose we do not know the block position and velocity. Our goal is to synthesize a controller, that takes the finger position, velocity, and tactile sensor measurement as inputs, and output the finger joint torques, such that we can move the object to a desired position. We aim to verify that such controller can achieve this goal, by searching for a Lyapunov function as a certificate.

Here for simplicity, we start with the object moving in one dimension, and the fingers are modeled as two point-mass pushers, on both sides of the block object, as shown in Fig. 6-2a,6-2b. The mass of the pushers are m_1, m_2 respectively. We can command the thrust u_1, u_2 on the pushers, and measure pushers' positions p_1, p_2 and velocities $\dot{p}_1, \dot{p}_2$. The dynamics of the pushers are formulated as the following

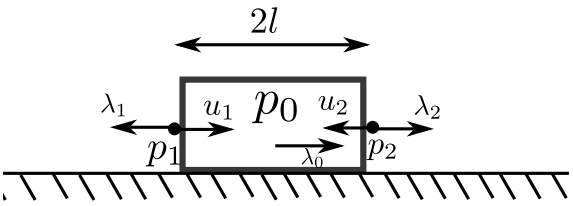
$$m_1 \ddot{p}_1 = u_1 - \lambda_1 \tag{6.1}$$

$$m_2 \ddot{p}_2 = u_2 + \lambda_2 \tag{6.2}$$

where $\lambda_1, \lambda_2 \geq 0$. When the pusher is not incontact with the block, the contact force



(a) Pushers not in contact with the block



(b) Pushers in contact with the block

Figure 6-2: Point-mass pushers and the block. The pushers positions are p_1, p_2 respectively. The position of the center of the block is p_0. The table friction force on the block is λ_0. The pushers are controlled by the thrust u_1, u_2 respectively. When making contact with the block, the contact force between the pushers and the block are λ_1, λ_2.

λ_1 or λ_2 is zero.

The block is under the pusher contact forces λ_1, λ_2 and the table friction force λ_0. From Newton's law, the dyamics of the block is

$$m\ddot{p}_0 = \lambda_1 - \lambda_2 + \lambda_0 \tag{6.3}$$

The table friction force λ_0 is subject to the Coulomb friction constraint, namely

$$-\mu mg \leq \lambda_0 \leq \mu mg \tag{6.4}$$

And the energy should dissipate due to the friction

$$\lambda_0 \dot{p}_0 \leq 0 \tag{6.5}$$

When the block is sliding, the friction force should be at the edge of the friction cone

$$\dot{p} > 0 \Rightarrow \lambda_0 = -\mu mg \tag{6.6}$$

$$\text{or } \dot{p} < 0 \Rightarrow \lambda_0 = \mu mg \tag{6.7}$$

We want to command the pusher thrust based on the contact force. Unfortunately, the thrust command cannot depend on contact force λ_1, λ_2 directly, otherwise it will bring an *algebraic loop* in the feedback system, that the output λ_1, λ_2 in pusher dynamics (6.1),(6.2) depend on the input commanded thrust directly. Thus the controller cannot take the contact force λ_1, λ_2 directly as the input, but instead we need to introduce a state for the contact force sensor, and use this state as the input to the controller.

We consider to use a first order system to model the contact force sensor dynamics, with the sensor state ζ_1, ζ_2. This time constant $\frac{1}{k}$ of this first order system can represent the delay on the sensor measurement.

$$\dot{\zeta}_1 = k(\lambda_1 - \zeta_1) \tag{6.8}$$

$$\dot{\zeta}_2 = k(\lambda_2 - \zeta_2) \tag{6.9}$$

and the sensor outputs its state ζ_1, ζ_2 as the measurement. The choice of sensor model is not restricted to this linear first order system, here we choose this model for simplicity.

The state of the overall system is denoted as $x = [p_0, p_1, p_2, \zeta_1, \zeta_2, \dot{p}_0, \dot{p}_1, \dot{p}_2]^T$. The dynamics of the state is formulated in Eq.(6.1),(6.2),(6.3),(6.8),(6.9). The output of the system is $y = [p_1, p_2, \zeta_1, \zeta_2, \dot{p}_1, \dot{p}_2]^T$.

Our goal is to design a control law $[u_1, u_2]^T = \pi(y)$ that will move the block to some desired region. Notice that our control law does not depend on the block state $p_0, \dot{p}_0$, but on the finger state $p_1, p_2, \dot{p}_1, \dot{p}_2$ and contact force measurement ζ_1, ζ_2.

6.2.2 Lyapunov function

Our goal is to design a controller, such that the block will converge to an invariant set under this control law. To verify the invariance set, we consider to search for a Lyapunov-like function $V(x)$, satisfying the following condition

$$V(x) \geq \rho \Rightarrow dV \leq 0 \tag{6.10}$$

$$V(x) \geq \alpha(|x|) \tag{6.11}$$

where $\alpha(|x|)$ is a function that is monotonically increasing, and $\alpha(0) = 0$. (6.10) proves that the state x will converge to the sub-level set $\mathcal{G} = \{x|V(x) \leq \rho\}$.

The size of this invariant set $\mathcal{G}$ can be measured by its outer ellipsoidal region $\{x|x^T G_O x \leq 1\}$. To guarantee this ellipsoid contains the invariant set $\mathcal{G}$, we impose the following condition

$$V(x) \leq \rho \Rightarrow x^T G_O x \leq 1 \tag{6.12}$$

Our goal is to minimize the invariant set, so that the terminal set for the state is as small as possible. To this end, we will minimize the size of the outer ellipsoidal approximation, by maximizing the trace of matrix G_O.

We now analyze the condition that $dV \leq 0$. We adopt the same approach as in Posa's work [134]. Since dV is a function of the dynamics of the overall system, which has different form depending on whether the pusher is making contact with the block. We will analyze the formulatoin of dV depending on whether the contact is active or not. For notation simplicity, we will use function $\phi_i(x)$ to represent whether the pusher is in contact with the block, defined as follows

$$\phi_1(x) = p_0 - p_1 - l \tag{6.13}$$

$$\phi_2(x) = p_2 - p_0 - l \tag{6.14}$$

We can analyze the contact mode based on the guard function $\phi(x)$ and its derivative.

- When $\phi_i(x) > 0$, pusher i is not in contact with the block.
- When $\phi_i(x) = 0$, $\dot{\phi}_i(x) < 0$, pusher i is hitting the block with an instantaneous impact.
- When $\phi_i(x) = 0, \dot{\phi}_i(x) = 0$, pusher i is maintaing contact with the block.
- When $\phi_i(x) = 0, \dot{\phi}_i(x) > 0$, pusher i is breaking contact with the block.

Based on the value of $\phi_i(x)$ and its derivatives, we partition the state space into three sets

1. set $\mathcal{F}$. Both pushers are either not in contact with the block, or breaking contact with the block.

$$\mathcal{F} = \{x | \phi_i(x) = 0 \Rightarrow \dot{\phi}_i(x) > 0\} \tag{6.15}$$

2. set $\mathcal{I}$. A pusher is hitting the block with instantaneous impact

$$\mathcal{I} = \{x | \exists i \in \{1, 2\}, \phi_i(x) = 0, \dot{\phi}_i(x) < 0\} \tag{6.16}$$

3. set $\mathcal{U}$. Pushers are maintaining contact with the block

$$\mathcal{U} = \{x | \exists i \in \{1, 2\}, \phi_i(x) = 0, \dot{\phi}_i(x) = 0\} \tag{6.17}$$

Intuitively, on set $\mathcal{F}$, the contact force λ_1, λ_2 vanish. On set $\mathcal{I}$, the contact force is an impulse. On set $\mathcal{U}$, the contact force is finite and non-zero.

For notation simplicity, we write $p = [p_0,\ p_1,\ p_2]^T, \dot{p} = [\dot{p}_0,\ \dot{p}_1,\ \dot{p}_2]^T, \zeta = [\zeta_1,\ \zeta_2]^T$. On set $\mathcal{F}$, since the contact forces λ_1, λ_2 vanish, we can write dV as

$$dV = \frac{\partial V}{\partial p}\dot{p} + \frac{\partial V}{\partial \zeta}(-k\zeta) + \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{\lambda_0}{m} \\ \frac{u_1}{m_1} \\ \frac{u_2}{m_2} \end{bmatrix} \leq 0 \tag{6.18}$$

On set $\mathcal{I}$, the contact force is an instantaneous impulse, so V will jump before and after the impulsive contact, due to the abrupt change on the velocity $\dot{p}$, and the contact sensor state ζ. dV is written in the following form

- If pusher 1 hits the block

$$dV = \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{\lambda_1}{m} \\ \frac{-\lambda_1}{m_1} \\ 0 \end{bmatrix} + \frac{\partial V}{\partial \zeta_1} k\lambda_1 \leq 0 \tag{6.19}$$

- If pusher 2 hits the block

$$dV = \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{-\lambda_2}{m} \\ 0 \\ \frac{\lambda_2}{m_2} \end{bmatrix} + \frac{\partial V}{\partial \zeta_2} k\lambda_2 \leq 0 \tag{6.20}$$

On set $\mathcal{U}$, we want to verify when the contact force may be non-zero. The condition for dV is that

$$dV = \frac{\partial V}{\partial p}\dot{p} + \frac{\partial V}{\partial \zeta} \begin{bmatrix} k(\zeta_1 - \lambda_1) \\ k(\zeta_2 - \lambda_2) \end{bmatrix} + \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{\lambda_0+\lambda_1-\lambda_2}{m} \\ \frac{u_1-\lambda_1}{m_1} \\ \frac{u_2+\lambda_2}{m_2} \end{bmatrix} \leq 0 \tag{6.21}$$

As Posa et.al demonstrated in [134], the condition on the left-hand side of Eq.(6.21) is the summation of the condition on the left-hand side of Eq.(6.18)-(6.20). Thus if dV satisfy the conditions (6.18)-(6.20), it automatically verifies that dV is non-positive in set $\mathcal{U}$. This superposition property on dV allows us to analyze the situation either when the contact forcese are all vanishing, or when the impulsive contact occurs.

We further notice that the complementary condition on the table friction force λ_0 has a special structure. We can analyze the three distinct cases for the table friction force, and the condition on dV on set $\mathcal{F}$ for each case. Here for notation simplicity, we denote all the states except $\dot{p}_0$ as z.

1. When the block is sliding to the right, $\dot{p}_0 > 0, \lambda_0 = -\mu mg$.

$$dV = -\frac{\partial V}{\partial \dot{p}_0}\mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \tag{6.22}$$

2. When the block is sliding to the left, $\dot{p}_0 < 0, \lambda_0 = \mu mg$.

$$dV = \frac{\partial V}{\partial \dot{p}_0}\mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \tag{6.23}$$

3. When the block is static on the table, $\dot{p}_0 = 0, \lambda_0 = \beta\mu mg$, where $-1 \leq \beta \leq 1$.

$$dV = \left.\frac{\partial V}{\partial \dot{p}_0}\right|_{\dot{p}_0=0} \beta\mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \tag{6.24}$$

We know that when the block velocity approaches zero from both postive and negative direction, both condition (6.22) and (6.23) still hold. By continuity argument, condition (6.22) and (6.23) prove that when the block velocity vanishes, dV is still negative, for table friction force $\lambda_0 = \pm\mu mg$. Namely

$$\begin{cases} -\left.\frac{\partial V}{\partial \dot{p}_0}\right|_{\dot{p}_0=0^+} \mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \\ \left.\frac{\partial V}{\partial \dot{p}_0}\right|_{\dot{p}_0=0^-} \mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \end{cases} \tag{6.25}$$

$$\Rightarrow \left.\frac{\partial V}{\partial \dot{p}_0}\right|_{\dot{p}_0=0} \beta\mu g + \frac{\partial V}{\partial z}\dot{z} \leq 0 \tag{6.26}$$

The implication holds because Eq.(6.26) is a convex combination of the two conditions in Eq.(6.25). Thus out of the three conditions (Eq.(6.22)-(6.24)) when the pusher contact forces vanish, we only need to verify the two conditions (6.22),(6.23). The third condition holds automatically if the other two conditions are satisfied.

To summarize this section, in order to verify dV is non-positive for all contact modes, we only need to verify dV is non-positive for the following four situations.

1. When the pusher contact forces vanish, and the block is sliding to the right, dV

is non-positive

$$dV = \frac{\partial V}{\partial p}\dot{p} + \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} -\mu g \\ \frac{u_1}{m_1} \\ \frac{u_2}{m_2} \end{bmatrix} - \frac{\partial V}{\partial \zeta} k\zeta \leq 0 \tag{6.27}$$

2. When the pusher contact forces vanish, and the block is sliding to the left, dV is non-positive

$$dV = \frac{\partial V}{\partial p}\dot{p} + \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \mu g \\ \frac{u_1}{m_1} \\ \frac{u_2}{m_2} \end{bmatrix} - \frac{\partial V}{\partial \zeta} k\zeta \leq 0 \tag{6.28}$$

3. When the pusher 1 hits the block with an instantaneous impulsive impact, dV is non-positive

$$dV = \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{1}{m} \\ -\frac{1}{m_1} \\ 0 \end{bmatrix} + \frac{\partial V}{\partial \zeta} \begin{bmatrix} k \\ 0 \end{bmatrix} \leq 0 \tag{6.29}$$

4. When the pusher 2 hits the block with an instantaneous impulsive impact, dV is non-positive

$$dV = \frac{\partial V}{\partial \dot{p}} \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{1}{m_2} \end{bmatrix} + \frac{\partial V}{\partial \zeta} \begin{bmatrix} 0 \\ k \end{bmatrix} \leq 0 \tag{6.30}$$

6.2.3 Tri-linear alternation

As we mentioned in the previous sub-section, our goal is to design a controller $[u_1,\ u_2]^T = \pi(y)$, such that we can minimize the set of the terminal state. Our

approach is to find a function V, satisfying the condition

$$\max_{V,G_O} \text{trace}(G_O) \tag{6.31}$$

$$V \geq \rho \Rightarrow dV \leq 0 \tag{6.32}$$

$$V \geq \alpha(|x|) \tag{6.33}$$

$$V \leq \rho \Rightarrow x^T G_O x \leq 1 \tag{6.34}$$

We will use the tri-linear alternation algorithm below to solve this problem iteratively. In each step of the *while* loop in Algorithm 3, we solve a Sum-of-Squares problem.

Algorithm 3 Tri-linear alternation

Start with an initial control policy π, and V, ρ. Maximize the trace of G_O by searching over Lagrangian multipliers in the *S-procedure*.

while trace(G_O) not converged **do**

1. fix V and Lagrangian multipliers, maximize trace of G_O by searching over control policy π, and G_O, ρ, subject to (6.32)-(6.34).
2. fix π and Lagrangian multipliers, maximize trace of G_O by searching over V, ρ, subject to constraints (6.32)-(6.34).
3. fix G_O, V, ρ, solve a feasibility problem by searching over controller π and Lagrangian multipliers, subject to constraints (6.32),(6.34)

end while

Initial guess

We need an initial guess on control policy and Lyapunov function to start the trilinear alternation. Here we designed a simple LQR controller for the pusher. The dynamics of the pusher in Eq.(6.1),(6.2) are linear, if the contact force vanish. We can design an LQR controller for each pusher separately, with the goal state as $[p_1^*, \dot{p}_1^*, p_2^*, \dot{p}_2^*]^T = [-l, 0, l, 0]^T$. The gain for this LQR controller is K, we then superimpose the contact

force measurement to this LQR controller, and our control policy is

$$u_1 = K \begin{bmatrix} p_1 + l \\ \dot{p}_1 \end{bmatrix} + \gamma\zeta_1 \tag{6.35}$$

$$u_2 = K \begin{bmatrix} p_2 - l \\ \dot{p}_2 \end{bmatrix} - \gamma\zeta_2 \tag{6.36}$$

where γ is a constant, falling strictly between 0 to 1. This LQR controller also generates a cost-to-go function, which we will use as the initial Lyapunov function V. We can then find an initial guess of ρ through binary search.

6.3 Results

We tested this trilinear alternation approach on with $k = 50$ in the contact sensor model (6.8),(6.9), to introduce a 20ms delay in the contact sensor. Unfortunately we encounter severe numerical problems in the trilinear alternation.

We can successfully find an initial guess of control policy and the Lyapunov function. Using the cost-to-go function from the LQR controller, and the control policy, we can do binary search to find an appropriate ρ, such that the Lagrangian multipliers exist for the *S-procedure*. We then proceed to the trilinear alternation with this initial guess. We first fix the Lyapunov function V and the Lagrangian multipliers, and search for G_O, ρ and a linear control policy $u = Ky + b$, where K and b are matrices of the correct size, and they are the parameters of this control policy to be searched for. The outputs from the solvers are dubious. SeDuMi reports running into numerical problems, while Mosek says the problem is infeasible. We also tried to bypass step 1 in the trilinear alternation, and proceed to step 2 directly. In step 2, with fixed control policy and Lagrangian multipliers from the initial guess, we search over ρ and the Lyapunov function $V = x^T S_1 x + s_2^T x$, where S_1, s_2 are the parameters of the Lyapunov function to be searched for. Again Sedumi runs into numerical problems, and Mosek reports infeasibility. Same thing happens if we take the initial guess and proceed to step 3 directly. If we take the SeDuMi solution as the initial guess,

and proceed to the next step in the trilinear alternation, then SeDuMi solver again runs into numerical problems. It is known that Mosek 7 has a bug that would report infeasibility, even if the problem is feasible in some cases. We are still attempting to find out the cause of such numerical difficulty.

Chapter 7

Conclusion

In this thesis, we present planners that can generate robust motion for a robot with multiple non-planar contact. The dynamics of the motion is constrained due to the limit on the contact forces. We plan the motion that can robustly satisfy those limits with some preserved margin, so the motion can withstand certain disturbance in the online motion execution.

We adopt the Contact Wrench Set margin as the robustness metric, which measures the ability of the robot to perfectly resist external wrench disturbance, and to track the desired motion exactly. This robustness metric is a generalization of the prevailing Zero Moment Point notion on the flat ground with unbounded friction, and it extends to the case of non-planar contact with friction limits.

In Chapter 4, we formulate a convex optimization problem, to generate the Center of Mass motion, with pre-specified contact locations and time. The planner finds a robust center of mass trajectory and an angular momentum trajectory, so as to maximize the contact wrench set margin, and to minimize the centroidal angular momentum. We show that this planner outputs similar results as the ZMP planner on the flat ground, and it can successfully find feasible motion on uneven terrain with bounded friction, where the ZMP planner would fail.

In Chapter 5, we search for the contact locations and the robot whole-body motion simultaneously, so as to improve the robustness metric. We present some matrix inequality constraints to compute the contact wrench margin, and we show that we can

solve the matrix inequalities together with non-convex nonlinear kinematic/dynamic constraints, either by using a nonlinear optimization approach, or by sequentially solving a series of semidefinite programming problems. We apply our planner to three problems, the force closure grasp optimization, the static posture optimization and trajectory optimization. We improve the robustness of the motion after optimization.

7.1 Discussion

7.1.1 Joint torque limit

As we mentioned in Section.3.3, the Contact Wrench Set criteria is just a necessary condition to determine whether the motion is feasible. This criteria ignores the actuator torque limits, by supposing the actuator can generate any arbitrary torque to accelerate the robot. In reality, we cannot always generate a large enough motor torque, as some motors are not power enough. Moreover, some of the joints are not actuated on some robot, such as the under-actuated hands [109, 45]. Thus it is important to consider the torque limits, to determine the feasibility of a motion.

We can introduce the torque limits to our problem, but the problem size gets a lot larger. We are currently working on a simpler formulation with less decision variables.

7.1.2 Sliding contact

In this thesis, we want to avoid sliding contact, and keep the contact link static. Some of the motion may prefer sliding contact, for example, to rotate a pen with in-hand manipulation. Unlike in static contact case, the friction force in sliding contact would depend on the velocity of the contact. According to Coulomb friction model, the contact force is at the boundary of the friction cone, and in the opposite direction of the sliding velocity, and we have to take velocity into consideration when computing the contact wrench set.

There are some work to compute the contact wrench set when a patch is sliding on the surface [49, 50, 101]. We can possibly extend our work using the limit surface

idea to compute the wrench set, and plan the wrench set which depends on both contact location and velocity.

7.1.3 Disturbance wrench

The wrench space does not have a well-defined norm. Some researchers argue that since there is not a natural unit in the wrench space (like Newton for force, and Newton×meter for torque), it is better to think about a force disturbance only, being applied at the surface of the robot. To do so, we can sample many points on the robot surface, and compute the minimal force disturbance that the robot cannot resist, among all these disturbance locations. This approach would add a lot more constraints into the optimization, as we need to compute the contact wrench margin for each disturbance location separately.

7.1.4 Contact position error

We presented that if a motion has a positive contact wrench margin, then it can withstand certain wrench disturbance. Apart from the wrench disturbance, the robot motion can suffer from another type of error, the error on contact location, during online motion execution. Since the contact wrench set margin is a continuous function of the contact location, if the normal direction does not change when the contact location moves, we can also show that a positive contact wrench margin guarantees robustness to contact position error. We should be able to formulate an optimization problem, that can search over the contact locations, such that the contact wrench always lies within the contact wrench set, even with small perturbation on the contact locations.

Appendix A

A.1 Robust Optimization

The goal of the robust optimization is to find the optimal value to a problem when the parameters of the constraints have uncertainty. The general form of robust optimization is

$$\min_{x} f_0(x) \tag{A.1}$$

$$\text{s.t } f(x,u) \leq 0 \quad \forall u \in \mathcal{U} \tag{A.2}$$

where

- $x \in \mathbb{R}^n$, the decision variable.
- $u \in \mathbb{R}^l$, the uncertain parameters of the problem.
- $\mathcal{U}$, the uncertain parameter set.
- $f \in \mathbb{R}^m$, the constraint function.

Here we focus on the case when the functions f_0, f_i are all linear, and the uncertainty set $\mathcal{U}$ is either a polytope or an ellipsoid. For more complicated functions and sets, the reader can refer to [6, 11] for more details.

With linear functions, the optimization problem is written as

$$\min_x \ c^T x \tag{A.3}$$

$$\text{s.t } A(u)x \leq b \quad \forall u \in \mathcal{U} \tag{A.4}$$

where $A(u) \in \mathbb{R}^{m \times n}$ is a linear expression of the uncertain parameters u.

When the uncertainty set $\mathcal{U}$ is a polytope,

$$\mathcal{U} = \text{ConvexHull}(v_1, \ldots, v_s) \tag{A.5}$$

the constraint (A.4) is satisfied for any u inside the polytope, if and only if it is satisfied at the vertices of the polytope. Thus the *robust counterpart* of this linear optimization problem under polytope uncertainty set, is the following linear optimization problem.

$$\min_x \ c^T x \tag{A.6}$$

$$\text{s.t } A(v_i)x \leq b \quad \forall i = 1, \ldots, s \tag{A.7}$$

When the uncertainty set $\mathcal{U}$ is an ellipsoid $\mathcal{B} = \{u | u^T Q u \leq 1\}$, we denote the rows of $A(u)$

$$A(u) = \begin{bmatrix} u^T A_1 + a_1^0 \\ \vdots \\ u^T A_m + a_m^0 \end{bmatrix} \tag{A.8}$$

where $A_i \in \mathbb{R}^{l \times n}, a_i^0 \in \mathbb{R}^{1 \times n}$. The constraint (A.4) is satisfied for any u inside the ellipsoid, if and only if the following second-order cone constraint is satisfied

$$\max_{u \in \mathcal{B}} u^T A_i x \leq b_i \Leftrightarrow \sqrt{x^T A_i^T Q^{-1} A_i x} \leq b_i - a_i^0 x \tag{A.9}$$

Thus the robust counterpart of the linear optimization problem under ellipsoidal uncertainty set, is the following second-order cone problem

$$\min_x \ c^T x \tag{A.10}$$

$$\text{s.t } b_i - a_i^0 x \geq \sqrt{x^T A_i^T Q^{-1} A_i x} \quad \forall i = 1, \ldots, m \tag{A.11}$$

Both linear problems and second order cone problems can be solved efficiently by modern optimization solvers [116, 120]. Thus the robust counter part of linear programming problems under either polytope or ellipsoidal uncertainty set, are tractable optimization problems.

A.2 Sum of Squares Polynomial

A polynomial $p(x)$ being Sum of Squares (sos) is a sufficient condition such that the polynomial is non-negative for any x. Determining if a polynomial is Sum of Squares is equivalent to finding a matrix being positive semidefinite (psd). A sos polynomial of order $2d$ on indeterminate $x \in \mathbb{R}^n$ can be written in the form

$$p(x) \text{ is sos} \Leftrightarrow \left(\exists M \succeq 0, \text{s.t } p(x) = v(x)^T M v(x)\right) \tag{A.12}$$

where

$$v(x) = \left[1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_{n-1} x_n, x_n^2, \ldots, x_1^d, \ldots, x_n^d\right]^T \tag{A.13}$$

each entry of $v(x)$ is of form $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \alpha_i \geq 0, \alpha_i \in \mathbb{N}, \sum \alpha_i \leq d$. Namely $v(x) \in \mathbb{R}^{\binom{n+d}{d}}$ contains all the *monomials* of x up to order d. And $M \in \mathbb{R}^{\binom{n+d}{d} \times \binom{n+d}{d}}$ is called *Gramian matrix*, whose entries are linear functions of coefficients in polynomial $p(x)$. So saying a polynomial is sos is equivalent to finding the positive semidefinite matrix M in Eq.(A.12).

One example of a sos polynomial is that

$$p(x) = ax^2 + 2bx + c \text{ is sos} \Leftrightarrow \begin{bmatrix} a & b \\ b & c \end{bmatrix} \succeq 0 \tag{A.14}$$

A more detailed explanation on Sum of Squares can be found in [128, 157].

A.3 Bilinear Matrix Inequality

Bilinear matrix inequalities (BMIs) are problems of the following form:

$$\text{Find} \qquad x \in \mathbb{R}^n \tag{A.15}$$

$$\text{s.t.} \qquad F_0 + \sum_{i=1}^{N} x_i F_i + \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j F_{ij} \succeq 0, \tag{A.16}$$

where F_0, F_i, F_{ij} are constant $m \times m$ symmetric matrices. $\succeq 0$ means the matrix on the left hand-side is positive semidefinite (psd), i.e. all the eigenvalues are non-negative; the special case is when the matrix is just a scalar, then $\succeq 0$ is the same as ≥ 0. We also note that BMIs include constraints that are both bilinear ($i \neq j$) as well as quadratic ($i = j$).

There has been a lot of approaches to solve the BMI problem [40, 162, 110]. Here we will briefly introduce the approach proposed by Ibaraki [71], where the BMI problem is solved through a sequence of semidefinite programming problems.

A.3.1 Finding Feasible Solutions to BMIs

While it is well known that BMIs are NP-hard in general [71], there exist very good heuristic methods based on semidefinite programming (SDP) for solving them. Here we review the method presented in [71] for finding feasible solutions to BMIs.

The first step is to write the BMI (A.15) as a rank-constrained *Linear Matrix*

Inequality (LMI) with an additional variable $X \in \mathbb{R}^{N \times N}$:

$$\text{Find:} \qquad x \in \mathbb{R}^N, X \in \mathbb{R}^{N \times N} \tag{A.17}$$

$$\text{s.t.} \qquad F_0 + \sum_{i=1}^{N} x_i F_i + \sum_{i=1}^{N} \sum_{j=1}^{N} X_{ij} F_{ij} \succeq 0, \tag{A.18}$$

$$M := \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \tag{A.19}$$

$$\text{rank}(M) = 1. \tag{A.20}$$

Here, each occurrence of bilinear terms $x_i x_j$ in (A.15) has been replaced by the (i, j) element of the decision matrix X. Constraints (A.19)(A.20) have been introduced to ensure that $X = xx^T$, resulting in the problems (A.17) and (A.15) having equivalent constraints. We note that without the rank constraint (A.20), problem (A.17) is a semidefinite program, which is a particular kind of *convex* optimization problem and can be solved efficiently (e.g., using interior point methods) [16].

The key idea in [71] is to drop the rank constraint in (A.17) and solve a sequence of SDPs that attempt to minimize the rank of M, as shown in Algorithm 4.

Algorithm 4 Finding feasible solutions to BMIs

Minimize trace(X) subject to constraints (A.18) and (A.19). If problem is infeasible, then problem (A.15) is infeasible. If problem is feasible, initialize $x^{(0)}$ and $X^{(0)}$ with the solution. Initialize $k = 1$.

while ¬converged **do**

1. Minimize trace($X^{(k)}$) - $2x^{(k-1)T}x^{(k)}$ subject to the constraints (A.18) and (A.19).
2. Set $k \leftarrow k + 1$

end while

Note that the first step in Algorithm 4 is the standard trace heuristic for minimizing the rank of a positive semidefinite matrix [16, 40]. The justification for the proceeding steps in the algorithm is based on the observation that the constraint (A.19) implies (by the Schur complement lemma) that $X \succeq 0$ and $X - xx^T \succeq 0$. This in turn implies that $\text{trace}(X) - x^T x \geq 0$ with equality holding if and only if $X = xx^T$ (i.e. when we have a feasible solution to (A.15)). Thus, Algorithm 4 proceeds by

linearizing the function $\text{trace}(X) - x^T x$ and minimizing this linearization at every iteration. A termination criterion for Algorithm 4 is provided by the following Lemma in [71].

Lemma 3. *[71] The following sequence is bounded below by 0 and non-increasing for* $k = 1, 2, \ldots$*:*

$$t_k := trace(X^{(k)}) - 2x^{(k-1)T}x^{(k)} + x^{(k-1)T}x^{(k-1)}.$$

Hence, this sequence converges to a value $t_{opt} \geq 0$*. Equality holds if and only if* $X^{(k)} = x^{(k)}x^{(k)T}$ *as* $k \to \infty$*.*

Lemma 3 provides us with a convergence criterion for Algorithm 4. Assuming that the first step in Algorithm 4 is feasible (if this is not the case, the original BMI is infeasible), then convergence of the value of t_k to 0 implies that we have found a feasible solution to the BMI. In the case where t_{opt} is not 0, nothing can be inferred.

A.3.2 Implementation Details

An important detail in implementing Algorithm 4 is that the SDP constraint (A.19) can be quite large if one has many decision variables x. However, it is typically the case that a large number of variables do not multiply with each other as bilinear products. Formally, consider a graph whose vertices are the variables in x. Two vertices are connected by an edge if the corresponding variables appear in a bilinear product in some constraint. Then we can partition the variables x into subsets $x_{I_1}, x_{I_2}, \ldots, x_{I_k}, \ldots, x_{I_K}$ corresponding to the connected components of the graph. We can then replace the constraints (A.19) and (A.20) by the following constraints:

$$M_k := \begin{bmatrix} X_{I_k,I_k} & x_{I_k} \\ x_{I_k}^T & 1 \end{bmatrix} \succeq 0, \quad \text{rank}(M_k) = 1, \ \forall k = 1, \ldots, K. \tag{A.21}$$

The cost function in Algorithm 4 is then replaced by the sum of the traces of the matrices X_{I_k,I_k}. While we end up with more psd constraints in general, each constraint involves a smaller matrix. Since SDP solve times typically scale poorly with the size of the largest psd constraint, we observe large computational gains in practice.

Another important implementation detail is to employ a randomization step in Algorithm 4, as described in [71]. In each iteration k of the algorithm, we sample a point $x_{rand}^{(k)}$ from the Gaussian distribution with mean $x^{(k)}$ and covariance $X^{(k)} - x^{(k)}x^{(k)T}$, where $(x^{(k)}, X^{(k)})$ is a solution to the SDP at the k-th iteration, and use cost function $\text{trace}(X^{(k+1)}) - 2x_{rand}^{(k)T}x^{(k+1)}$ in $k + 1^{th}$ iteration. In practice, the randomization step prevents the algorithm from getting stuck in local minima.

Bibliography

[1] Gerald J Agin. *Real time control of a robot with a mobile camera*. SRI International, 1979.

[2] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[3] N.M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 113 – 120. IEEE, 1996.

[4] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[5] Sebastien Barthelemy and Philippe Bidaud. Stability measure of postural dynamic equilibrium based on residual radius. In *Advances in Robot Kinematics: Analysis and Design*, pages 399–407. Springer, 2008.

[6] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of operations research*, 23(4):769–805, 1998.

[7] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.

[8] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *International Conference on Humanoid Robots*, pages 42–48. IEEE, 2007.

[9] Dmitry Berenson, Siddhartha S Srinivasa, Dave Ferguson, and James J Kuffner. Manipulation planning on constraint manifolds. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 625–632. IEEE, 2009.

[10] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[11] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[12] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[13] J.T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[14] Ch Borst, Max Fischer, and Gerd Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 319–325. IEEE, 2004.

[15] Boyd, S.P., Wegbreit, and B. Fast computation of optimal contact forces. *IEEE Transactions on Robotics*, 23(6):1117–1132, 2007.

[16] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] Timothy Bretl and Sanjay Lall. Testing static equilibrium for legged robots. *Robotics, IEEE Transactions on*, 24(4):794–807, 2008.

[18] John C Butcher. Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 3(02):185–201, 1963.

[19] Richard H Byrd, Jorge Nocedal, and Richard A Waltz. Knitro: An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization*, pages 35–59. Springer, 2006.

[20] Stephane Caron, Quang-Cuong Pham, and Yoshihiko Nakamura. Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics. *Robotics: Science and System*, 2015.

[21] I-Ming Chen and Joel W Burdick. Finding antipodal point grasps on irregularly shaped objects. *Robotics and Automation, IEEE Transactions on*, 9(4):507–512, 1993.

[22] Christine Chevallereau, Dalila Djoudi, and Jessy W Grizzle. Stable bipedal walking with foot rotation through direct regulation of the zero moment point. *Robotics, IEEE Transactions on*, 24(2):390–401, 2008.

[23] Cyrille Collette, Alain Micaelli, Claude Andriot, and Pierre Lemerle. Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3187–3193. IEEE, 2008.

[24] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Education, Inc, third edition, 2005.

[25] Hongkai Dai, Anirudha Majumdar, and Russ Tedrake. Synthesis and optimization of force closure grasps via sequential semidefinite programming. In *International Symposium on Robotics Research*, 2015.

[26] Hongkai Dai and Russ Tedrake. Optimizing robust limit cycles for legged locomotion on unknown terrain. In *Proceedings of the IEEE Conference on Decision and Control*, page 8, Maui, Hawaii, December 2012.

[27] Hongkai Dai and Russ Tedrake. L2-gain optimization for robust bipedal walking on unknown terrain. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[28] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[29] Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. *IEEE-RAS International Conference on Humanoid Robots*, Nov 2014.

[30] Robin L H Deits and Russ Tedrake. Computing large convex regions of obstacle-free space through semidefinite programming. In *Proceedings of the Eleventh International Workshop on the Algorithmic Foundations of Robotics (WAFR 2014)*, Istanbul, 2014.

[31] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, August 2010.

[32] Rosen Diankov, Nathan Ratliff, Dave Ferguson, Siddhartha Srinivasa, and James Kuffner. Bispace planning: Concurrent multi-space exploration. In *Proceedings of Robotics: Science and Systems IV*, June 2008.

[33] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.

[34] Dimitar Dimitrov, Alexander Sherikov, and Pierre-Brice Wieber. A sparse model predictive control formulation for walking motion generation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[35] Dimitar Dimitrov, Pierre-Brice Wieber, Hans Joachim Ferreau, and Moritz Diehl. On the implementation of model predictive control for on-line walking pattern generation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2685–2690. IEEE, 2008.

[36] Johannes Englsberger, Twan Koolen, Sylvain Bertrand, Jerry Pratt, Christian Ott, and Alin Albu-Schaffer. Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4022–4029. IEEE, 2014.

[37] Paul J. Enright and Bruce A. Conway. Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming. *J. Guidance*, 14(5):981–985, Sept-Oct 1991.

[38] Adrien Escande, Abderrahmane Kheddar, and Sylvain Miossec. Planning contact points for humanoid robots. *Robotics and Autonomous Systems*, 61(5):428–442, 2013.

[39] Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, Toby Schneider, Hongkai Dai, Claudia Pérez D'Arpino, Robin Deits, Matt DiCicco, Dehann Fourie, Twan Koolen, Pat Marion, Michael Posa, Andrés Valenzuela, Kuan-Ting Yu, Julie Shah, Karl Iagnemma, Russ Tedrake, and Seth Teller. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 32(2):229–254, September 2014.

[40] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, 2002.

[41] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2007.

[42] Siyuan Feng, X. Xinjilefu, Weiwei Huang, and Christopher G. Atkeson. 3d walking based on online optimization. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robotics*, Atlanta, GA, August 2013.

[43] Carlo Ferrari and John Canny. Planning optimal grasps. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2290–2295. IEEE, 1992.

[44] Komei Fukuda and Alain Prodon. Double description method revisited. In *Combinatorics and computer science*, pages 91–111. Springer, 1996.

[45] Marco Gabiccini, Edoardo Farnioli, and Antonio Bicchi. Grasp analysis tools for synergistic underactuated robotic hands. *The International Journal of Robotics Research*, page 0278364913504473, 2013.

[46] Elena Garcia, Joaquin Estremera, and Pablo Gonzalez De Santos. A comparative study of stability margins for walking machines. *Robotica*, 20(6):595–606, 2002.

[47] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[48] Philip E. Gill, Walter Murray, and Michael A. Saunders. *User's Guide for SNOPT Version 7: Software for Large -Scale Nonlinear Programming*, February 12 2006.

[49] Suresh Goyal, Andy Ruina, and Jim Papadopoulos. Planar sliding with dry friction part 1. limit surface and moment function. *Wear*, 143(2):307–330, 1991.

[50] Suresh Goyal, Andy Ruina, and Jim Papadopoulos. Planar sliding with dry friction part 2. dynamics of motion. *Wear*, 143(2):331–352, 1991.

[51] Brent Griffin and Jessy Grizzle. Walking gait optimization for accommodation of unknown terrain height variations. In *American Control Conference (ACC), 2015*, pages 4810–4817. IEEE, 2015.

[52] Steffen Haidacher and Gerd Hirzinger. Estimating finger contact location and object pose from contact measurements in 3d grasping. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1805–1810. IEEE, 2003.

[53] Kaveh Akbari Hamed, Brian G Buss, and Jessy W Grizzle. Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. *The International Journal of Robotics Research*, page 0278364915593400, 2015.

[54] Li Han, Jeffrey C Trinkle, and Zexiang X Li. Grasp analysis as linear matrix inequality problems. *Robotics and Automation, IEEE Transactions on*, 16(6):663–674, 2000.

[55] Kaiyu Hang, Miao Li, Johannes A. Stork, Yasemin Bekiroglu, Florian T. Pokorny, Aude Billard, and Danica Kragic. Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transaction on Robotics*, accepted.

[56] Kaiyu Hang, Johannes A Stork, Florian T Pokorny, and Danica Kragic. Combinatorial optimization for hierarchical contact-level grasping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 381–388. IEEE, 2014.

[57] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.

[58] Robert Haschke, Jochen J Steil, Ingo Steuwer, and Helge Ritter. Task-oriented quality measures for dextrous grasping. In *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pages 689–694. IEEE, 2005.

[59] Kris Hauser, Timothy Bretl, Kensuke Harada, and Jean-Claude Latombe. Using motion primitives in probabilistic sample-based planning for humanoid robots. In *Algorithmic foundation of robotics VII*, pages 507–522. Springer, 2008.

[60] Kris Hauser, Timothy Bretl, Jean-Claude Latombe, Kensuke Harada, and Brian Wilcox. Motion planning for legged robots on varied terrain. *The International Journal of Robotics Research,*, 27(11-12):1325–1349, 2008.

[61] A. Herdt, H. Diedam, P.B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl. Online walking motion generation with automatic footstep placement. *Advanced Robotics*, 24(5-6):719–737, 2010.

[62] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2016.

[63] Hugh Herr and Marko Popovic. Angular momentum in human walking. *Journal of Experimental Biology*, 211(4):467–481, 2008.

[64] Alexander Herzog, Nicholas Rotella, Stefan Schaal, and Ludovic Righetti. Trajectory generation for multi-contact momentum control. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 874–880. IEEE, 2015.

[65] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1321–1326, 1998.

[66] H Hirukawa, S Hattori, K Harada, S Kajita, K Kaneko, F Kanehiro, K Fujiwara, and M Morisawa. A universal stability criterion of the foot contact of legged robots - Adios ZMP. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1976–1983, May 2006.

[67] Hirohisa Hirukawa. Humanoid robotics projects in Japan. In *Proceedings of the Understanding Humanoid Robots Workshop at IEEE ICRA 2006*, 2006.

[68] David Hsu, Robert Kindel, Jean-Claude Latombe, and Stephen Rock. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 21(3):233–255, 2002.

[69] Christian Hubicki, Jesse Grimes, Mikhail Jones, Daniel Renjewski, Alexander Sproewitz, y Abate, and Jonathan Hurst. Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot. *International Journal of Robotics Research*, accepted.

[70] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[71] Soichi Ibaraki and Masayoshi Tomizuka. Rank minimization approach for solving bmi problems with random search. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 1870–1875. IEEE, 2001.

[72] Tatsuya Ishihara, Akio Namiki, Masatoshi Ishikawa, and Makoto Shimojo. Dynamic pen spinning using a high-speed multifingered hand with high-speed tactile sensor. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 258–263. IEEE, 2006.

[73] Sumit Jain, Yuting Ye, and C Karen Liu. Optimization-based interactive motion synthesis. *ACM Transactions on Graphics (TOG)*, 28(1):10, 2009.

[74] Yan-Bin Jia and Michael Erdmann. Pose and motion from contact. *The International Journal of Robotics Research*, 18(5):466–487, 1999.

[75] Micah K Johnson and Edward H Adelson. Retrographic sensing for the measurement of surface texture and shape. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1070–1077. IEEE, 2009.

[76] Micah K Johnson, Forrester Cole, Alvin Raj, and Edward H Adelson. Microgeometry capture using an elastomeric sensor. In *ACM Transactions on Graphics (TOG)*, volume 30, page 46. ACM, 2011.

[77] Steven G Johnson. The nlopt nonlinear-optimization package, 2014.

[78] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiware, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA IEEE International Conference on Robotics and Automation*, pages 1620–1626. IEEE, Sep 2003.

[79] S Kajita, K Kaneko, K Harada, F Kanehiro, and K Fujiwara. Biped walking on a low friction floor. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. Sendai, Japan*, pages 3546–3552, 2004.

[80] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. pages 239 – 246. IEEE International Conference on Intelligent Robots and Systems (IROS), 2001.

[81] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry, and Stefan Schaal. Learning, planning, and control for quadruped locomotion over challenging terrain. *I. J. Robotic Res.*, 30(2):236–258, 2011.

[82] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Hirohisa Hirukawa, Toshikazu Kawasaki, Masaru Hirata, Kazuhiko Akachi, and Takakatsu Isozumi. Humanoid robot HRP-2. In *ICRA*, pages 1083–1090. IEEE, 2004.

[83] L.E. Kavraki, P. Svestka, JC Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.

[84] Haruhisa Kawasaki, Tsuneo Komatsu, and Kazunao Uchiyama. Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. *IEEE/ASME transactions on mechatronics*, 7(3):296–303, 2002.

[85] David Kirkpatrick, Bhubaneswar Mishra, and Chee-Keng Yap. Quantitative steinitz's theorems with applications to multifingered grasping. *Discrete & Computational Geometry*, 7(1):295–318, 1992.

[86] J Koenemann, Andrea Del Prete, Yuval Tassa, E Todorov, Olivier Stasse, M Bennewitz, and Nicolas Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3346–3351. IEEE, 2015.

[87] Twan Koolen, Tomas de Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, 2012.

[88] Michael C Koval, Nancy S Pollard, and Siddhartha S Srinivasa. Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research*, 35(1-3):244–264, 2016.

[89] K. Koyanagi, H. Hirukawa, S. Hattori, M. Morisawa, S. Nakaoka, K. Harada, and S. Kajita. A pattern generator of humanoid robots walking on a rough terrain using a handrail. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2617–2622, Sept 2008.

[90] Danica Kragic and Henrik I Christensen. Robust visual servoing. *The international journal of robotics research*, 22(10-11):923–939, 2003.

[91] M Kudruss, Maximilien Naveau, Olivier Stasse, Nicolas Mansard, C Kirches, Philippe Soueres, and K Mombaur. Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 684–689. IEEE, 2015.

[92] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Motion planning for humanoid robots under obstacle and dynamic balance constraints. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 1:692–698 vol.1, 2001.

[93] James J. Kuffner, Koichi Nishiwaki, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Motion planning for humanoid robots. In Paolo Dario and Raja Chatila, editors, *ISRR*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 365–374. Springer, 2003.

[94] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake.

Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

[95] Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Proceedings of the International Conference on Robotics and Automation*, Hong Kong, China, May 2014. IEEE.

[96] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation.

[97] LaValle, S.M., Kuffner, J.J., and Jr. Randomized kinodynamic planning. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1:473–479, 1999.

[98] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98–11, Iowa State University, Dept. of Computer Science, 1998.

[99] S. LaValle and J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

[100] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[101] Soo Hong Lee and MR Cutkosky. Fixture planning with friction. *Journal of Engineering for Industry*, 113(3):320–327, 1991.

[102] Zexiang Li and S Shankar Sastry. Task-oriented optimal grasping by multifingered robot hands. *Robotics and Automation, IEEE Journal of*, 4(1):32–44, 1988.

[103] Guanfeng Liu, Jijie Xu, Xin Wang, and Zexiang Li. On quality functions for grasp synthesis, fixture planning, and coordinated manipulation. *Automation Science and Engineering, IEEE Transactions on*, 1(2):146–162, 2004.

[104] Jingru Luo, Yajia Zhang, Kris Hauser, H Andy Park, Manas Paldhe, CS Lee, Michael Grey, Mike Stilman, Jun Ho Oh, Jungho Lee, et al. Robust ladder-climbing with a humanoid robot with application to the darpa robotics challenge. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2792–2798. IEEE, 2014.

[105] Anirudha Majumdar. *Funnel Libraries for Real-Time Robust Feedback Motion Planning*. PhD thesis, Massachusetts Institute of Technology, Jun 2016.

[106] Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control design along trajectories with sums of squares programming. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4054–4061, 2013.

[107] Anirudha Majumdar and Russ Tedrake. Robust online motion planning with regions of finite time invariance. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, page 16, Cambridge, MA, June 2012.

[108] Richard Mason, Elon Rimon, and Joel Burdick. Stable poses of 3-dimensional objects. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 391–398. IEEE, 1997.

[109] Bruno Massa, Stefano Roccella, Maria Chiara Carrozza, and Paolo Dario. Design and development of an underactuated prosthetic hand. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3374–3379. IEEE, 2002.

[110] Mehran Mesbahi and George P Papavassilopoulos. On the rank minimization problem over a positive semidefinite linear matrix inequality. *Automatic Control, IEEE Transactions on*, 42(2):239–243, 1997.

[111] Philipp Michel, Joel Chestnutt, James Kuffner, and Takeo Kanade. Vision-guided humanoid footstep planning for dynamic environments. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 13–18. IEEE, 2005.

[112] Bud Mishra. Grasp metrics: Optimality and complexity. In *Proceedings of the workshop on Algorithmic foundations of robotics*, pages 137–165. AK Peters, Ltd., 1995.

[113] Katja D. Mombaur. Using optimization to create self-stable human-like running. *Robotica*, 27(3):321–330, 2009.

[114] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–144. Eurographics Association, 2012.

[115] Federico L Moro, Nikos G Tsagarakis, and Darwin G Caldwell. A human-like walking for the compliant humanoid coman based on com trajectory reconstruction from kinematic motion primitives. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 364–370. IEEE, 2011.

[116] APS Mosek. The mosek optimization software. *Online at http://www.mosek.com*, 54, 2010.

[117] Beipeng Mu, Girish Chowdhary, and Jonathan P How. Efficient distributed sensing using adaptive censoring-based inference. *Automatica*, 50(6):1590–1602, 2014.

[118] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.

[119] Van-Duc Nguyen. Constructing force-closure grasps. *The International Journal of Robotics Research*, 7(3):3–16, 1988.

[120] Gurobi Optimization et al. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, 2012.

[121] David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, (September 2012):1–16, jun 2013.

[122] D.E. Orin and A Goswami. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pages 653–659, sep 2008.

[123] Jia Pan, Liangjun Zhang, and Dinesh Manocha. Collision-free and smooth trajectory computation in cluttered environments. *The International Journal of Robotics Research*, page 0278364912453186, 2012.

[124] Evangelos George Papadopoulos. *On the dynamics and control of space manipulators*. PhD thesis, Massachusetts Institute of Technology, 1990.

[125] Diego Pardo, Lukas Moller, Michael Neunert, Alexander W Winkler, and Jonas Buchli. Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *arXiv preprint arXiv:1504.05803*, 2015.

[126] Chonhyon Park, Jia Pan, and Dinesh Manocha. Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments. 2012.

[127] Hae-Won Park, Patrick M Wensing, Sangbae Kim, et al. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. 2015.

[128] Pablo A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 18 2000.

[129] Nancy S Pollard. Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research*, 23(6):595–613, 2004.

[130] M. B. Popovic, A. Hofmann, and H. Herr. Zero spin angular momentum control: definition and applicability. *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, pages 478–493, 2004.

[131] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, January 2014.

[132] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, page 16, Cambridge, MA, June 2012.

[133] Michael Posa, Mark Tobenkin, and Russ Tedrake. Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control (HSCC 2013)*, pages 63–72. ACM, April 8-11 2013.

[134] Michael Posa, Mark Tobenkin, and Russ Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control (TAC)*, 61(6):1423 – 1437, June 2016.

[135] Gill Pratt and Justin Manzo. The darpa robotics challenge [competitions]. *IEEE Robotics & Automation Magazine*, 20(2):10–12, 2013.

[136] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *2006 6th IEEE-RAS international conference on humanoid robots*, pages 200–207. IEEE, 2006.

[137] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[138] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[139] C. David Remy. *Optimal Exploitation of Natural Dynamics in Legged Locomotion*. PhD thesis, ETH ZURICH, 2011.

[140] Maximo A Roa, Katharina Hertkorn, Christoph Borst, and Gerd Hirzinger. Reachable independent contact regions for precision grasps. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5337–5343. IEEE, 2011.

[141] Máximo A Roa and Raúl Suárez. Grasp quality measures: review and performance. *Autonomous robots*, 38(1):65–88, 2015.

[142] Layale Saab, Oscar E Ramos, Francois Keith, Nicolas Mansard, Philippe Soueres, and J Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *Robotics, IEEE Transactions on*, 29(2):346–362, 2013.

[143] Nasser Sadati, Guy A Dumont, Kaveh Akabri Hamed, and William A Gruver. *Hybrid Control and Motion Planning of Dynamical Legged Locomotion*, volume 2. John Wiley & Sons, 2012.

[144] Michael G Safonov, Keat-Choon Goh, and JH Ly. Control system synthesis via bilinear matrix inequalities. In *American Control Conference, 1994*, volume 1, pages 45–49. IEEE, 1994.

[145] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, and N. Higakiand K. Fujimura. The intelligent ASIMO: system overview and integration. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 3, pages 2478 – 2483. IEEE, 2002.

[146] Eugene Salamin. Application of quaternions to computation with rotations. Technical report, Working Paper, 1979.

[147] AC Sanderson and LE Weiss. Image-based visual servo control using relational graph error signals. *Proc. ieee*, 1074, 1980.

[148] Philippe Sardain and Guy Bessonnet. Forces acting on a biped robot. Center of Pressure – Zero Moment Point. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(5):630–637, Sep 2004.

[149] Tomoya Sato, Sho Sakaino, Eijiro Ohashi, and Kouhei Ohnishi. Walking trajectory planning on stairs using virtual slope for biped robots. *Industrial Electronics, IEEE Transactions on*, 58(4):1385–1396, 2011.

[150] Brian W Satzinger, Chelsea Lau, Marten Byl, and Katie Byl. Tractable locomotion planning for robosimian. *The International Journal of Robotics Research*, page 0278364915584947, 2015.

[151] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[152] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research (IJRR)*, 30(2):192–215, Feb 2011.

[153] Benjamin Stephens. Humanoid push recovery. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 589–595. IEEE, 2007.

[154] Mike Stilman. Global manipulation planning in robot joint space with task constraints. *Robotics, IEEE Transactions on*, 26(3):576–584, 2010.

[155] Weehong Tan and Andrew Packard. Stability region analysis using composite lyapunov functions and sum of squares programming. *submitted to IEEE Trans. Automat. Contr.*, 2006.

[156] Russ Tedrake, Scott Kuindersma, Robin Deits, and Kanako Miura. A closed-form solution for real-time ZMP gait generation and feedback stabilization. In *Proceedings of the International Conference on Humanoid Robotics*, Seoul, Korea, November 2015.

[157] Russ Tedrake, Ian R. Manchester, Mark M. Tobenkin, and John W. Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.

[158] Mark M. Tobenkin, Ian R. Manchester, and Russ Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *Proceedings of the 18th IFAC World Congress, extended version available online: arXiv:1010.3013 [math.DS]*, 2011.

[159] Onur Toker and Hitay Ozbay. On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *American Control Conference, Proceedings of the 1995*, volume 4, pages 2525–2526. IEEE, 1995.

[160] Steve Tonneau, Nicolas Mansard, Chonhyon Park, Dinesh Manocha, Franck Multon, and Julien Pettré. A reachability-based planner for sequences of acyclic contacts in cluttered environments. In *International Symposium on Robotics Research (ISSR 2015)*, 2015.

[161] U. Topcu, A.K. Packard, P. Seiler, and G.J. Balas. Robust region-of-attraction estimation. *IEEE Transactions on Automatic Control*, 55(1):137 –142, Jan 2010.

[162] HD Tuan, P Apkarian, and Y Nakashima. A new lagrangian dual global optimization algorithm for solving bilinear matrix inequalities. In *American Control Conference, 1999. Proceedings of the 1999*, volume 3, pages 1851–1855. IEEE, 1999.

[163] Andrés K. Valenzuela. *Mixed-Integer Convex Optimization for Planning Aggressive Motions of Legged Robots Over Rough Terrain*. PhD thesis, Massachusetts Institute of Technology, Feb 2016.

[164] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

[165] Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[166] Miomir Vukobratovic and Davor Juricic. Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, (1):1–6, 1969.

[167] Andreas Wachter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[168] Patrick M Wensing and David E Orin. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *International Journal of Humanoid Robotics*, page 1550039, 2015.

[169] Pierre-Brice Wieber. On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robots*, 2002.

[170] Pierre-Brice Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 137–142. IEEE, 2006.

[171] Alexander W Winkler, Carlos Mastalli, Ioannis Havoutis, Michele Focchi, Darwin G Caldwell, and Claudio Semini. Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5148–5154. IEEE, 2015.

[172] Kwangjin Yang and Salah Sukkarieh. An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics*, 26(3):561–568, 2010.

[173] Yuting Ye and C Karen Liu. Optimal feedback control for character animation using an abstract model. In *ACM Transactions on Graphics (TOG)*, volume 29, page 74. ACM, 2010.

[174] Li Yu, Wen-Yao Chang, Kewei Zuo, Jean Wang, Douglas Yu, and Duane Boning. Methodology for analysis of tsv stress induced transistor variation and circuit performance. In *Thirteenth International Symposium on Quality Electronic Design (ISQED)*, pages 216–222. IEEE, 2012.

[175] Li Yu, Sharad Saxena, Christopher Hess, Ibrahim Abe M Elfadel, Dimitri Antoniadis, and Duane Boning. Efficient performance estimation with very small sample size via physical subspace projection and maximum a posteriori estimation. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 226. European Design and Automation Association, 2014.

[176] Li Zhang and Jeffrey C Trinkle. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3805–3812. IEEE, 2012.

[177] Ye Zhao and Luis Sentis. A three dimensional foot placement planner for locomotion in very rough terrains. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 726–733. IEEE, 2012.

[178] Yu Zheng, Ming C Lin, Dinesh Manocha, Albertus Hendrawan Adiwahono, and Chee-Meng Chew. A walking pattern generator for biped robots on uneven terrains. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4483–4488. IEEE, 2010.

[179] Zhou, Kemin, Doyle, John C., Glover, and Keith. *Robust and optimal control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[180] Kemin Zhou and John C. Doyle. *Essentials of Robust Control*. Prentice Hall, 1997.

[181] Gunther Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 1995.