

博士論文

Computational Foundation for Planner-in-the-Loop Multi-Contact Whole-Body Control of Humanoid Robots

(運動計画をフィードバックループに含むヒューマノイド
ロボットの多点接触全身制御のための計算基盤)

カロン ステファン



Department of Mechano-Informatics
Nakamura-Takano Laboratory

Computational Foundation for Planner-in-the-Loop Multi-Contact Whole-Body Control of Humanoid Robots

運動計画をフィードバックループに含むヒューマノイド
ロボットの多点接触全身制御のための計算基盤

Stéphane Caron

Thesis Committee

Yoshihiko	NAKAMURA	Professor at the University of Tokyo	Chair
Isao	SHIMOYAMA	Professor at the University of Tokyo	Examiner
Masayuki	INABA	Professor at the University of Tokyo	Examiner
Yasuo	KUNIYOSHI	Professor at the University of Tokyo	Examiner
Wataru	TAKANO	Professor at the University of Tokyo	Examiner
Jean-Paul	LAUMOND	Directeur de recherche at LAAS-CNRS	Examiner

Stéphane Caron

Computational Foundation for Planner-in-the-Loop Multi-Contact Whole-Body Control of Humanoid Robots (運動計画をフィードバックループに含むヒューマノイドロボットの多点接触全身制御のための計算基盤)

Submitted: December 11th, 2015

Revised: February 1st, 2016, and March 10th, 2017

The University of Tokyo

Nakamura-Takano Laboratory

Department of Mechano-Informatics

7-3-1 Hongo, Bunkyo-ku

Tokyo, 113-8656

Abstract

In this thesis, we explore the questions of motion planning and control for humanoid robots with the aim to integrate motion planning in a fast control loop. Our contributions towards this goal revolve around three axes: kinodynamic decoupling, force-space curtailment, and dimensional reduction of the control space. In the first one, we decouple the kinematic and dynamic components of the planning problem by an original integration with time-optimal control methods. This approach allows us to keep planning in a geometric space, the benefits of which we demonstrate both empirically and through theoretical proofs. In the second axis, we focus on the contact aspects of planning. To avoid slippage or other contact losses, planners usually consider a large number of contact forces and their associated Coulomb friction cones. We show how this redundant representation can be reduced to contact wrenches, unique to each contacting articulation, and propose the first analytical derivation of the associated frictional wrench cone for rectangular contact surfaces. We then connect these developments to the gravito-inertial wrench for whole-body motion planning. However, we note that using wrenches for planning leads to challenging open questions such as the interpolation of the non-holonomic angular momentum. We attack this problem with a paradigm shift: rather than controlling wrenches, we generalize the notion of ZMP (point where the tangential component of the gravito-inertial moment vanishes) to that of “ZMP of a wrench”. We then propose efficient algorithms to compute the associated support areas, and show how to use these tools to generate locomoting trajectories from simplified dynamics model such as the Linear Pendulum, even in arbitrary multi-contact scenarios. This reduction of the control space rounds the third and last axis of the computational foundations advanced by this thesis. We demonstrate the applicability of each by simulations and empirical experiments on the HRP-4 humanoid robot.

Résumé

Cette thèse explore les questions de la planification et du contrôle du mouvement pour les robots humanoïdes, en se donnant pour objectif de jeter les bases calculatoires qui permettront d'inclure cette première dans une boucle de contrôle haute-fréquence. Ces bases se répartissent en trois axes : découplage kინodynamique, réduction de l'espace des forces, et réduction de l'espace de contrôle. Dans le premier, nous séparons les composantes géométrique et dynamique du problème de planification par des méthodes issues de la paramétrisation en temps optimal. Cette approche permet de maintenir un espace de planification purement géométrique, dont nous démontrons l'intérêt à la fois empiriquement et par des preuves théoriques. Nous nous concentrons ensuite sur l'aspect *continger*¹ de la planification humanoïde. La nécessité d'éviter les pertes de contact oblige généralement les planificateurs à considérer un grand nombre de forces de contact associées à des cônes de frottements de Coulomb. Nous montrons comment réduire cette représentation redondante à des torseurs de contact, uniques pour chaque articulation, et proposons la première dérivation analytique du cônes de frottement torsoriel pour les surfaces de contact rectangulaires. Nous connectons ensuite ce développement au cône du torseur gravito-inertiel pour la planification corps complet. Nous remarquons toutefois que l'utilisation de torseurs pour cette tâche pose des questions difficiles telles que l'interpolation du moment angulaire, qui est une grandeur non-holonome. Nous y répondons par un changement de paradigme : plutôt que d'utiliser ces torseurs directement, nous introduisons une généralisation de la notion du ZMP (point d'annulation de la composante tangentielle du moment gravito-inertial) à des torseurs généraux. Nous proposons ensuite des méthodes efficaces pour calculer la zone de support du ZMP généralisé, et montrons comment celles-ci peuvent être utilisées générer des trajectoires locomotrices à partir de modèles dynamiques simplifiés tels que le pendule linéaire. Cette réduction de l'espace de contrôle à l'aide de modèle simplifiés constitue le troisième et dernier axe des bases calculatoires avancées par cette thèse. Nous démontrons l'applicabilité de l'ensemble de ces propositions par des simulations et expériences sur le robot humanoïde HRP-4.

¹ Relatif au contact ; du latin *contingo* (saisir, toucher).

Acknowledgments

My first thoughts, as I write the last lines of this manuscript, go to the two persons who have watched over its growth, Yoshihiko Nakamura and Quang-Cường Phạm. I was aware from the start of the difficulties that could arise by placing my research under the advice of two purposeful researchers, and it is with great delight that I see how we have been working together in *bonne entente*.

中 I want to thank Nakamura-*sensei* for all the time and thorough feedback that he gave me. I have always been impressed how, in the midst of professoral business, he would still find ways to set aside time for deep discussion. It has been a privilege for me to work under his relentless curiosity and high scientific standards.

范 I want to thank equally Quang-Cường Phạm for these three years. Not only did Dr Phạm kick-off my PhD by taking me directly as co-author in his ongoing work, but he has kept the momentum going with never-ending energy and enthusiasm, making me discover the excitement of research *sicut nemo nisi per amicitiam cognoscitur*. Looking at the path behind us leaves me eager to ask: *et après?*

守 Jean-Paul Laumond is the man who made this PhD possible, not only by connecting the right people together, but also by being always so attentive and open to discussion. Knowing at the back of my mind that I could always reach out to him was of great support.

神 I would like to address a special thank to Kaminaga-*sensei* as well. They say one can really appreciate the value of people in times of stress, to which I will only add that I was very glad to be with Kaminaga-*sensei* as our team was preparing for the DARPA Robotics Challenge.

This thesis has also benefited from open and friendly technical discussions with Hervé Audren, Andrea Del Prete, Adrien Escande, Abderrahmane Kheddar, Enrico Mingo and Francesco Nori. Thank you all for your time. I am looking forward to our next encounters.

I would like to thank the old guard of YNL students with whom I spent most of my time in Japan: Rohan Budhiraja, Tianyi Kang, Thomas Nierhoff, Tanut Treratanakulwong, and last but far from least Tianwei Zhang. I would also like to address a special thank to Ueno-san, who has always been so nice and supportive. Thank you all for making this place more than a workplace.

I would be torn apart to thank my family and close friends at home only in the end, and such a limited amount of space would be nothing but representative. So, à toi qui lis ceci : on va manger des chips, t'entends ? Des chips !² And for you reading this, I did not forget what's essential: 相伴主義 !

²C'est donc tout ce que ça te fait quand je te dis qu'on va manger des chips ?

Contents

1	Introduction	13
1.1	Motivation	13
1.2	Related works and contributions	14
1.3	Computational foundation	16
1.4	Outline of the thesis	17
2	Kinodynamic Planning in the State Space	21
2.1	Background	21
2.1.1	Motion planning problem	21
2.1.2	Structure of roadmap planners	23
2.1.3	Steering methods	25
2.2	Probabilistic completeness	26
2.2.1	Existing completeness results	26
2.2.2	System and interpolation assumptions	28
2.2.3	Completeness theorem	32
2.3	Completeness and interpolation in practice	36
2.3.1	Bezier interpolation	36
2.3.2	Second-order continuous interpolation	37
2.3.3	Comparison in simulations	38
3	Kinodynamic Planning in the Configuration Space	41
3.1	Time-Optimal Path Parameterization	41
3.2	Admissible Velocity Propagation	45
3.2.1	Limiting curves	46
3.2.2	Maximum final velocity	47
3.2.3	Minimum final velocity	49
3.2.4	Implementation	51
3.3	Combining AVP with randomized planners	51
3.3.1	AVP-RRT	52

3.3.2	Comparison to state-space planning	54
3.3.3	Implementation	56
3.4	Probabilistic completeness of AVP-based planners	60
4	Contact and friction	65
4.1	Contact	65
4.1.1	Contact modes	66
4.1.2	Friction model	66
4.2	Cone duality	69
4.2.1	Weyl-Minkowski theorem	69
4.2.2	Double-description method	70
4.2.3	Polyhedron projection	71
4.2.4	Fourier-Motzkin elimination	71
4.3	Surface contacts	73
4.3.1	Continuous shear and pressure distributions	73
4.3.2	Wrench cone for rectangular surfaces	74
4.3.3	Yaw moment at the center of pressure	81
4.4	Application to single-support contact phases	82
4.4.1	Expression of the CWC as ZMP support polygon	82
4.4.2	Experimental validation	87
5	Multi-contact stability	91
5.1	Whole-body dynamics	91
5.1.1	Dynamic equilibrium	93
5.1.2	Dynamic wrenches	94
5.1.3	Contact stability	95
5.2	Motivation for wrench reduction	96
5.2.1	TOPP for ZMP constraints	97
5.2.2	TOPP for frictional contacts	98
5.2.3	Preliminary experiments and discussion	99
5.3	Gravito-inertial wrench cone	104
5.3.1	Static multi-contact stability	105
5.3.2	Robust multi-contact static stability	108
5.3.3	Dynamic multi-contact stability	111
6	Multi-contact ZMP	119
6.1	Previous work	120

6.2	ZMP support areas under frictional constraints	124
6.2.1	Linearized wrench cones	124
6.2.2	Zero-tilting Moment Point	125
6.2.3	Construction of the support area	126
6.2.4	Geometric properties of the support area	130
6.2.5	Generality of the concept	133
6.3	Multi-contact locomotion with pendulum control	133
6.3.1	Relationship between ZMP and COM	133
6.3.2	Linear pendulum models: LP and LIP	135
6.3.3	ZMP support areas under LP regulation	136
6.3.4	Trajectory generation for the LP model	137
6.3.5	Validation in simulations	139
6.3.6	Performance	141
6.4	Discussion	142
6.4.1	Plane coordinate selection	142
6.4.2	CHCP and infinite friction	143
6.4.3	Can we do better than ZMP?	144
7	Conclusion	147
A	Elements of Contact planning	151
A.1	Problem statement	152
A.2	Contact planning in COM space	153
A.3	Experiment	156
B	Components developed for the DARPA Robotics Challenge	159
B.1	Perception	160
B.2	Motion generation and control	163
B.3	Valve turning task	165
	Bibliography	167

Introduction

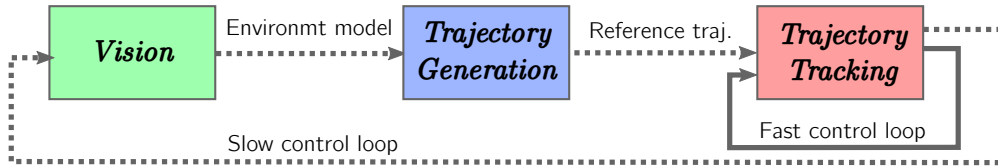
1.1 Motivation

Robots are the synthesis of mechanisms and computers, made possible by actuators and sensors. They implement a *perception, planning and action* loop:



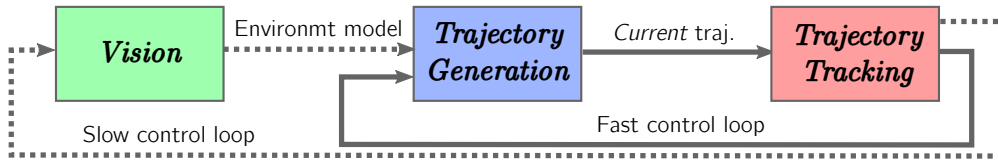
Internally, a robot maintains a model of itself, the environment, and the interactions between them. *Perception* is the passive update of model states from sensor readings. *Action* is the active output of actuator commands, which will affect the robot's state and the environment through the interactions between them. Finally, *planning* is the computation of actions that bring the system to a desired state. It implies by essence prediction, *i.e.*, the anticipation of actions' consequences: how will the robot or the environment react to the application of a given command? Providing a complete and accurate answer is a vastly complex task, fortunately the prediction capabilities of a system need not be perfect thanks to the *feedback loop*: after executing the first actions of a plan, the robot can perceive its new state, evaluate the difference with its prediction, and update its plan consequently.

For humanoid robots, interactions with the environment require a fast feedback loop, with frequencies typically on the scale of the kilohertz. Yet, in the current state of the technology, perception and motion planning are computationally intensive and do not typically run at this such frequencies. This fact has led roboticists to develop systems with open-loop planning and closed-loop control laws:



Teleoperated systems, such as those demonstrated recently at the DARPA Robotics Challenge, are instances of this approach. Perception and motion planning are executed at slow rates involving a human in the loop (for example, to specify goals from 3D visualization data). A reference command is then sent to the robot, which tracks the reference at best via a fast control loop. Teams using this paradigm could successfully complete the challenge. Yet, for future research, one can note that the constructed systems are not autonomous, *i.e.*, they cannot cope with unexpected situations without intervention from a human operator.

In this thesis, we focus on developing the planning capabilities of humanoid robots. As a guiding principle, we argue that, for humanoids to gain more autonomy, motion planning should fit into the fast feedback loop:¹



The challenge in this endeavor is that the constraint space of a humanoid robot has a complex structure, both kinematic (collision avoidance, joint limits, etc.) and dynamic (friction, torque limits, etc.), while fast motion planning can only be achieved in spaces with simple structures. A great deal of our study will therefore focus on the reduction of humanoid constraints to smaller spaces.

1.2 Related works and contributions

Motion planning and whole-body control have been fields of extensive research in the robotics literature. This thesis makes a contribution in each, before advancing a paradigm in which they can be combined.

¹ Ideally, perception would also take part in this loop, however this part is not within the scope of the present thesis.

Motion planning is the problem of generating a long-term command driving the system from its initial to a goal state. At first, this problem was considered from the perspective of *geometric* constraints, such as obstacle avoidance, or those imposed by the kinematic structures of manipulators (Lozano-Perez, 1983). The two emblematic planners of this epoch are Probabilistic Roadmap (PRM) (Kavraki et al., 1996) and glsRRT (LaValle and Kuffner, 2000), both of which have been successfully demonstrated empirically and theoretically through proofs of *probabilistic completeness* (Hsu et al., 1997; LaValle and Kuffner, 2001). More recently, *kinodynamic* constraints (Donald et al., 1993), which stem from differential equations of dynamic systems, have also been taken into account (LaValle and Kuffner, 2001; Hsu et al., 2002). These constraints are crucial for humanoid robots. However, the performance of PRM or RRT, which is well established for geometric systems, has been questioned and is still an open question for kinodynamic systems at large (see Chapter 2). **The first contribution of this thesis is to provide a precise proof of probabilistic completeness for a class of kinodynamic planners that applies to humanoids**, which we will subsequently use in the rest of our work.

Kuffner et al. (2001) were among the first ones to apply RRT to humanoid motion planning. To keep motion planning as a geometric problem, they only considered the regime of “quasi-static” movements, *i.e.*, motions of extremely slow velocities. This choice seriously restricts the range of motions that the system can realize. To overcome this limitation, we developed a new family of planners based on Time-Optimal Path Parameterization (TOPP) (see Chapter 2) that can explore both the static and dynamic range of motion of dynamical systems. Applying these new techniques to humanoids naturally led us to the question of balance, or *contact stability*. The difficulty here is that humanoids do not directly control the contact forces applied on them by the environment. Rather, they actuate their joint motors, which generates *in response* a combination of contact forces, contact motions or centroidal motions, according to the laws of physics (Wieber, 2006). The motion planner needs to ensure *a priori* that the motion it sends to the whole-body controller is feasible, *i.e.*, that it can be achieved by the controller, given an anticipation of the set of contact forces and centroidal motions that can be exerted by the environment on the robot. This condition is summarized at the whole-body level by the Gravito-Inertial Wrench Cone (GIWC) (Saida et al., 2003; Hirukawa et al., 2006; Qiu et al., 2011). **This thesis advances the algorithmic aspects of wrench-based contact stability at two levels:**

- at the local level, we provide an *analytical formula* for the wrench cone of a contacting link. With this development, it becomes unnecessary to consider sets of a contact points for contact stability, which has both theoretical and computational benefits (see Chapter 4).
- at the whole-body level, we show how to integrate the GIWC condition within TOPP to create a dynamic motion generator, which we demonstrate on challenging motions (see Chapter 5).

Wrench-based conditions are suitable for testing *e.g.*, postural stability (Bretl and Lall, 2008; Escande et al., 2013), but they pose conceptual challenges in planning: how to interpolate trajectories in the 6D wrench space? How to select contacts to maximize the volume of a wrench cone? Interestingly, these questions have been readily answered in the simplified setting where the humanoid walks on horizontal floors. In this case, a well-known (though incomplete) dynamic stability condition is that the Zero-tilting Moment Point (ZMP) (Vukobratović and Stepanenko, 1972) lies within the convex hull of ground contact points, the so-called *support area*. Yet, the ZMP has been historically defined as a point on a unique ground surface, which prevented its application to more general environments. **The third contribution of this thesis is to generalize the notions of ZMP and support areas to arbitrary contact conditions.** Based on this advance, we demonstrate in Chapter 6 a whole-body controller with which a humanoid can walk in challenging environments using a simple linear pendulum rooted at the generalized ZMP.

1.3 Computational foundation

Bringing motion planning into the fast feedback loop is the thread behind the apparent variety of contributions of this thesis. Figure 1.1 shows how they combine together into a framework, which is what we designate by *computational foundation* for planner-in-the-loop control of humanoids. It consists in three components, working on increasingly high-dimensional variables:

Low-dimensional planning: for fast-rate outputs, high-level planning is done on a low-dimensional model of the full robot dynamics, which we obtain by reduction of the force space (Chapters 4 and 5) and a proper generalization of

the ZMP to multi-contact scenarios (Chapter 6). Contact planning (in gray) is a complementary component for which we provide technical elements in Appendix A, however the main focus of this thesis is on the high-level planner.

Kinodynamically-decoupled path planning: low-dimensional references are treated as objectives to be fulfilled by a whole-body path tracker. Thanks to the integration with Time-Optimal Path Parameterization (TOPP, Chapter 3), we decouple kinematic and dynamic constraints, which limits dimensional expansion by solving consecutively two simpler problems: inverse kinematics and time-optimal control. This part is demonstrated in Chapter 5.

Joint-level control: the tracker computes a whole-body reference joint-angle trajectory, which is independently tracked at best by each joint controller.

At the end of this pipeline, up-to-date joint states are fed back to the motion planner and a new update cycle begins.

1.4 Outline of the thesis

The rest of this thesis is organized as follows.

In Chapter 2, we review classical motion planning concepts and results before making our contribution to the problem of kinodynamic planning: proving probabilistic completeness for the class of planners that we will use within the rest of the thesis. We also provide simulation experiments that corroborate our theoretical finding.

Next, we introduce in Chapter 3 a combination of kinodynamic planning with time-optimal control that allows a new form of kinodynamic planning in the *configuration* space, as opposed to the bigger *state* space (including velocities) that is commonly used.

Narrowing down on humanoids, we review in Chapter 4 the physics of contact and fundamental calculation techniques that allow us to derive a novel analytical formula of the contact wrench cone of rectangular surfaces.

We then broaden our scope to multi-contact stability in Chapter 5 with the Gravitoinertial Wrench Cone, and provide an original time-optimal formulation of the path

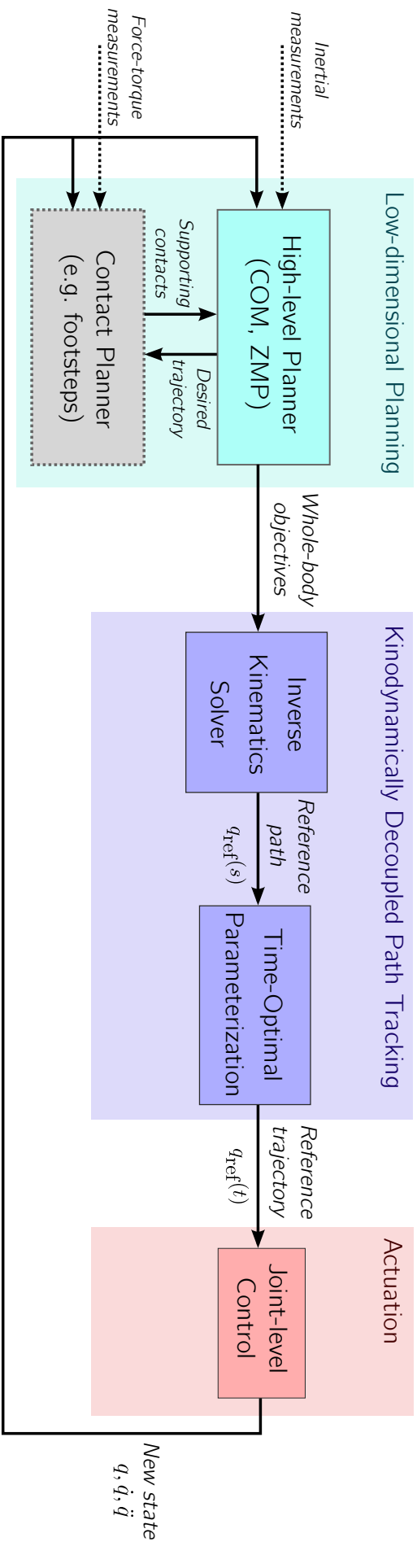


Fig. 1.1.: Computational framework proposed in this thesis. For fast-rate outputs, high-level motion planning is done on low-dimensional models of the full robot dynamics, which we obtain by reduction of the force space (Chapters 4 and 5) and a proper generalization of the ZMP to multi-contact scenarios (Chapter 6). The output of the planner is then conveyed as objectives to a whole-body path tracker. We limit dimensional expansion by a decoupling of kinematic and dynamic constraints presented in Chapter 3. Theoretical guarantees are provided for the high-level planner (Chapter 2) and its integration with time-optimal control (Chapter 3). The tracker computes a reference joint-angle trajectory, which is independently tracked at best by each joint controller. Up-to-date joint states are fed back to the motion planner for plan updates. Contact planning (in gray) is another constitutive part of motion planning. Appendix A provides technical elements on this component, however the main focus of this thesis is on the high-level planner.

tracking dynamics under GIWC constraints. We demonstrate the complete framework by retiming dynamic, non-quasi-static humanoid motions.

Finally, in Chapter 6, we generalize the concept of Zero-tilting Moment Point to arbitrary contact wrenches. This development allows us to create a whole-body controller based on a simple linear-pendulum model, yet general enough to apply to locomotion over arbitrary terrains.

Kinodynamic Planning in the State Space

” 凡事豫则立，不豫则废。
言前定则不跌，事前定则不困，
行前定则不疚，道前定则不穷。

— Variation on 《礼记·中庸》

This chapter provides background knowledge on kinodynamic motion planning, as well as the contribution of this thesis to the field: the proof of probabilistic completeness for the class of planners that we will use later on.

2.1 Background

2.1.1 Motion planning problem

In what follows, a *robot* defines a system of articulated rigid bodies with n degrees of freedom. We will denote by $\mathcal{C} \subset \mathbb{R}^n$ the *configuration space* of the robot, where a configuration is an n -dimensional vector \mathbf{q} of generalized coordinates. The *state space* $\mathcal{X} \subset \mathbb{R}^{2n}$ denotes the differential manifold of positions and velocities¹ $x = (\mathbf{q}, \dot{\mathbf{q}})$, satisfying $d\mathbf{q} = \dot{\mathbf{q}} dt$ where t denotes the time variable and $\dot{\mathbf{q}}$ is the n -dimensional vector of joint-angle velocities. The *equations of motion* of the robot can be described by a time-invariant differential equation in the state space:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad (2.1)$$

¹ In all generality, state spaces may also include higher-order derivatives. However, all our systems will be at most second-order ones from Newtonian mechanics, so that small variations (differentials) within a first-order state space will suffice to describe them fully for our purpose.

where $\mathbf{u} \in \mathcal{U}$ denotes a vector of control inputs, *e.g.*, joint torques $\boldsymbol{\tau}$ for a humanoid or robotic arm. We denote by $\mathcal{U}_{\text{adm}} \subset \mathcal{U}$ denote the subset of admissible controls. For example, $\mathcal{U}_{\text{adm}} = \{\boldsymbol{\tau} \in \mathbb{R}^n | \forall i, \tau_{\min} \leq \tau_i \leq \tau_{\max}\}$ denotes a set of admissible torques with lower and upper torque limits. All vector inequalities thereafter are to be understood component-wise, *i.e.*, $\boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}$ if and only if, for each coordinate i , $\tau_i \leq \tau_{\max,i}$.

Motion planning is the problem of finding a long-term sequence of controls $\mathbf{u}(t)$ that drive the system from its initial to a goal configuration. All configurations may not be reachable, depending on the constraints exerted on the robot and the environment. Accounting for this fact, a desirable property for a motion planner is to be *complete*, that is to say, to return a solution whenever one exists, and a negative response otherwise (Latombe, 1991). However, this guarantee has been shown to be PSPACE-hard (LaValle, 2006).

Randomized planners (Kavraki et al., 1996; LaValle and Kuffner, 2000) brought a paradigm shift in the motion planning literature. They are grounded on the observation that common planning problems do not have the pathological features of theoretical edge-cases. Rather, they can be efficiently explored by making use of random samples and natural geometrical biases (like the Voronoi bias resulting from uniform sampling, see *e.g.*, (Yershova et al., 2005)). A randomized planner is said to be *probabilistically complete* if the probability of returning a solution, when there is one, tends to one as execution time goes to infinity (LaValle, 2006).

Theoretical as they may seem, the notions of completeness are of notable practical interest, as proving completeness requires one to formalize the problem by hypotheses on the robot, the environment, etc. While experiments can show that a planner works for a given robot, in a given environment, for a given query, etc., a proof of completeness is a certificate that the planner works for a precise *set* of problems. The size of this set depends on how strong the assumptions required to make the proof are: the weaker the assumptions, the larger the set of solvable problems.

Kinodynamic motion planning (Donald et al., 1993) refers to the motion planning problem for systems under kinodynamic constraints. Such constraints are more difficult to deal with than geometric constraints because they cannot in general be expressed using only *configuration-space variables* – such as the joint angles of a

manipulator, the position and the orientation of a mobile robot, etc. Rather, they involve higher-order derivatives such as velocities and accelerations. There are two types of kinodynamic constraints:

Non-holonomic constraints: non-integrable *equality* constraints on higher-order derivatives, such as found in wheeled vehicles (Laumond, 1998), under-actuated manipulators (Bullo and Lynch, 2001) or space robots. Systems under such constraints have the property that their state depends not only on configuration variables, but also on the path taken to reach it. When a non-holonomic systems executes a loop trajectory in its configuration space, it may end in a state different from the one it started from (Wieber, 2006).

Hard bounds: *inequality* constraints on higher-order derivatives such as torque bounds for manipulators (Bobrow et al., 1985), support areas (Wieber, 2002) or wrench cones for humanoid stability (Caron et al., 2015a), etc.

Some authors have considered systems that are subject to both types of constraints, such as under-actuated manipulators with torque bounds (Bullo and Lynch, 2001). Humanoid robots are non-holonomic systems due to the asymmetry (in joint-torque) of the Newton-Euler equations of motion, which means that a motion that is feasible forward in time may require different torques, and thus be unfeasible backward in time (see Chapter 4 or Wieber, 2006).

2.1.2 Structure of roadmap planners

By *randomized planners*, we essentially refer to the two emblematic instances of PRM (Kavraki et al., 1996) or Rapidly-exploring Random Trees (RRT) (LaValle and Kuffner, 2001). Interestingly, their structures are very similar, so that our theoretical developments apply equally to one or the other. Both PRM and RRT build a roadmap on the state space and rely on repeated random sampling of the free state space, *i.e.*, states with non-colliding configurations and velocities within the system bounds. New states are connected to the roadmap using a *steering* function, which is a method used to drive the system from an initial to a goal configuration. The steering method may be imperfect, *e.g.*, it may not reach the goal exactly, not take environment collisions into account, only apply to states that are sufficiently close,

Algorithm 1 Extension step of randomized planners (PRM or RRT)

Require: initial node x_{init} , number of iterations N

```
1:  $(V, E) \leftarrow (\{x_{init}\}, \emptyset)$ 
2: for  $N$  steps do
3:    $x_{rand} \leftarrow \text{SAMPLE}(\mathcal{X}_{free})$ 
4:    $X_{parents} \leftarrow \text{PARENTS}(x_{rand}, V)$ 
5:   for  $x_{parent}$  in  $X_{parents}$  do
6:      $x_{steer} \leftarrow \text{STEER}(x_{parent}, x_{rand})$ 
7:     if  $x_{steer}$  is a valid state then
8:        $V \leftarrow V \cup \{x_{steer}\}$ 
9:        $E \leftarrow E \cup \{(x_{parent}, x_{steer})\}$ 
10:    end if
11:  end for
12: end for
13: return  $(V, E)$ 
```

etc. The objective of the motion planner is to overcome these limitations, turning a local steering function into a global planning method.

PRM builds a roadmap that is later used to generate motions between many initial and final states (many-to-many queries). When new samples are drawn, they are connected to *all* neighboring states in the roadmap using the steering function, resulting in a connected graph. Meanwhile, RRT focuses on driving the system from *one* initial state x_{init} towards a goal area (one-to-one queries). It grows a tree by connecting new samples to *one* neighboring state, usually their closest neighbor.

Both PRM's and RRT's *extension* step are represented by Algorithm 1, which relies on the following sub-routines (see Fig. 2.1 for an illustration):

- $\text{SAMPLE}(S)$: randomly sample an element from a set S ;
- $\text{PARENTS}(x, V)$: return a set of states in the roadmap V from which steering towards x will be attempted;
- $\text{STEER}(x, x')$: generate a system trajectory from x towards x' . If successful, return a new node x_{steer} ready to be added to the roadmap. Depending on the planner, the success criterion may be “reach x' exactly” or “reach a vicinity of x' ”.

The design of each sub-routine greatly impacts the quality and even the completeness of the resulting planner. In the literature, $\text{SAMPLE}(S)$ is usually implemented as uniform random sampling over S , but some authors have suggested adaptive

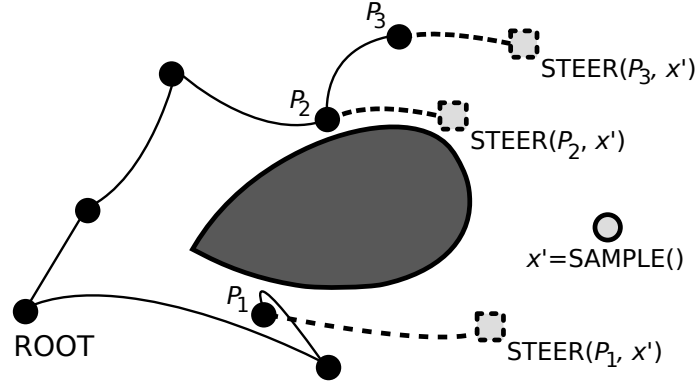


Fig. 2.1.: Illustration of the extension routine of randomized planners. To grow the roadmap toward the sample x' , the planner selects a number of parents $\text{PARENTS}(x') = \{P_1, P_2, P_3\}$ from which it applies the $\text{STEER}(P_i, x')$ method.

sampling as a way to improve planner performance (Bialkowski et al., 2013). In geometric planners, $\text{PARENTS}(q, V)$ is usually implemented from the Euclidean norm over \mathcal{C} as

$$\text{PARENTS}(q, V) := \arg \min_{q' \in V} \|q' - q\|.$$

This choice results in the so-called Voronoi bias of RRTs (LaValle and Kuffner, 2001). Both experiments and theoretical analysis support this choice for geometric planning, however it becomes inefficient for kinodynamic planning, as was showed by Shkolnik et al. (Shkolnik et al., 2009) on systems as simple as the torque-limited pendulum.

2.1.3 Steering methods

We focus on steering functions, which can be classified into three categories: analytical, state-based and control-based steering.

Analytical steering. This category corresponds to the ideal case when one can compute analytical trajectories respecting the system's differential constraints, which are usually called (perfect) *steering functions* in the literature (LaValle and Kuffner, 2001; Papadopoulos et al., 2014). Unfortunately, it only applies to a handful of systems. Reeds and Shepp curves for cars are a notorious example of this (Laumond, 1998).

Control-based steering. Generate a control $u : [0, \Delta t] \rightarrow \mathcal{U}_{\text{adm}}$, where \mathcal{U}_{adm} denotes the set of *admissible* controls, and compute the corresponding trajectory

by *forward dynamics*. This approach has been called *incremental simulation* (Kunz and Stilman, 2014), *control application* (LaValle and Kuffner, 2001) or *control-space sampling* (Papadopoulos et al., 2014) in the literature. It is widely applicable, as it only requires forward-dynamic calculations, but usually results in weak steering functions as the user has no or limited control over the destination state. In works such as (LaValle and Kuffner, 2001; Hsu et al., 2002), random functions u are sampled from a family of primitives (e.g., piecewise-constant functions), a number of them are tried and only the one bringing the system closest to the target is retained. Linear-Quadratic Regulation (LQR) (Perez et al., 2012; Tedrake, 2009) also qualifies as control-based steering: in this case, u is computed as the optimal policy for a linear approximation of the system given a quadratic cost function.

State-based steering. Interpolate a trajectory $\gamma_{\text{int}} : [0, \Delta t] \rightarrow \mathcal{C}$, for instance a Bezier curve matching the initial and target configurations and velocities, and compute a control that makes the system track that trajectory. For fully-actuated system, this is typically done using *inverse dynamics*. An interpolated trajectory is rejected if no suitable control can be found. Compared to control-based steering, this approach applies to a more limited range of systems, but delivers more control over the destination state. Algorithm 2 gives the prototype of state-based steering functions.

Algorithm 2 Prototype of state-based steering functions $\text{STEER}(x, x')$

```

1:  $\gamma_{\text{int}} \leftarrow \text{INTERPOLATE}(x, x')$ 
2:  $u_{\text{int}} := \text{INVERSE\_DYNAMICS}(\gamma_{\text{int}}(t), \dot{\gamma}_{\text{int}}(t), \ddot{\gamma}_{\text{int}}(t))$ 
3: if  $(\forall t \in [0, \Delta t], u_{\text{int}}(t) \subset \mathcal{U}_{\text{adm}})$  then
4:   return the last state of  $\gamma_{\text{int}}$ 
5: end if
6: return failure

```

2.2 Probabilistic completeness

2.2.1 Existing completeness results

Randomized planners such as RRT and PRM are both simple to implement yet efficient for geometric planning. The completeness of these planners has been established for geometric planning in (LaValle and Kuffner, 2001; Karaman and Frazzoli, 2011; Hsu et al., 1997). In their proof, Hsu et al. (Hsu et al., 1997) quantified

the problem of narrow passages in configuration space with the notion of (α, β) -expansiveness. The two constants α and β express a geometric lower bound on the rate of expansion of reachability areas.

There is, however, a gap between geometric and *kinodynamic* planning (Donald et al., 1993) in terms of proving probabilistic completeness. When Hsu et al. extended their solution to kinodynamic planning (Hsu et al., 2002), they applied the same notion of expansiveness, but this time in the $\mathcal{X} \times \mathcal{T}$ (state and time) space with control-based steering. Their proof states that, when $\alpha > 0$ and $\beta > 0$, their planner is probabilistically complete. However, whether $\alpha > 0$ or $\alpha = 0$ in the non-geometric space $\mathcal{X} \times \mathcal{T}$ remains an open question. As a matter of fact, the problem of evaluating (α, β) has been deemed as difficult as the initial planning problem (Hsu et al., 1997). In a parallel line of work, LaValle et al. (LaValle and Kuffner, 2001) provided a completeness argument for kinodynamic planning, based on the hypothesis of an *attraction sequence*, i.e., a covering of the state space where two major problems of kinodynamic planning are already solved: steering and antecedent selection. Unfortunately, the existence of such a sequence was not established.

In the two previous examples, completeness is established under assumptions whose verification is at least as difficult as the motion planning problem itself. Arguably, too much of the complexity of kinodynamic planning has been abstracted into hypotheses, and these results are not strong enough to hold the claim that their planners are probabilistically complete in general. This was exemplified recently when Kunz and Stilman (Kunz and Stilman, 2015) showed that RRTs with control-based steering were *not* probabilistically complete for a family of control inputs (namely, those with fixed time step and best-input extension). At the same time, Papadopoulos et al. (Papadopoulos et al., 2014) established probabilistic completeness for the same planner using a different family of control inputs (randomly sampled piecewise-constant functions). The picture of completeness for kinodynamic planners therefore seems to be a nuanced one.

Karaman et al. (Karaman and Frazzoli, 2011) introduced the RRT* path planner and extended it to kinodynamic planning with differential constraints in (Karaman and Frazzoli, 2010), providing a sketch of proof for the completeness of their solution. However, they assumed that their planner had access to the optimal cost metric

and optimal local steering, which restricts their analysis to systems for which these ideal solutions are known. The same authors tackled the problem from a slightly different perspective in (Karaman and Frazzoli, 2013) where they supposed that the PARENTS function had access to w -weighted boxes, an abstraction of the system's local controllability. However, they did not show how these boxes can be computed in practice² and did not prove their theorem, arguing that the reasoning was similar to the one in (Karaman and Frazzoli, 2011) for kinematic systems.

2.2.2 System and interpolation assumptions

Terminology. A function is *smooth* when all its derivatives exist and are continuous. Let $\|\cdot\|$ denote the Euclidean norm. A function $f : A \rightarrow B$ between metric spaces is Lipschitz when there exists a constant K_f such that

$$\forall (x, y) \in A, \|f(x) - f(y)\| \leq K_f \|x - y\|.$$

The (smallest) constant K_f is called the Lipschitz constant of the function f . A trajectory is a continuous function $\gamma : [0, \Delta t] \rightarrow \mathcal{C}$, and the distance of a state $x \in \mathcal{X}$ to a trajectory γ is

$$\text{dist}_\gamma(x) := \min_{t \in [0, \Delta t]} \|(\gamma, \dot{\gamma})(t) - x\|.$$

A control function $\mathbf{u} : [0, \Delta t] \rightarrow \mathcal{U}$ has δ -clearance when its image is in the δ -interior of \mathcal{U}_{adm} , i.e., for any time t , $\mathcal{B}(\mathbf{u}(t), \delta) \subset \mathcal{U}_{\text{adm}}$. A trajectory $\mathbf{q}(t)$ that is solution to the differential system (2.1) using only controls $\mathbf{u}(t) \in \mathcal{U}_{\text{adm}}$ is called an *admissible* trajectory. The kinodynamic motion planning problem is to find an admissible trajectory from \mathbf{q}_{init} to \mathbf{q}_{goal} .

System assumptions. Our model for an \mathcal{X} -state randomized planner is given by Algorithm 1 using state-based steering. Our first assumption is about actuation:

Assumption 1. *The system is fully actuated.*

Full actuation allows us to write the equations of motion of the system in generalized coordinates in a more convenient form.

² The definition of w -weighted boxes is quite involved: it depends on the joint flow of vector fields spanning the tangent space of the system's manifold.

Property 1. *The equations of motion for a fully-actuated robot (i.e., articulated system of rigid bodies) with n degrees of freedom can be written as:*

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (2.2)$$

with $\ddot{\mathbf{q}}$ the vector of joint-angle accelerations, $\mathbf{M}(\mathbf{q})$ the $n \times n$ joint inertia matrix, $\mathbf{C}(\mathbf{q})$ the $n \times n \times n$ Coriolis tensor, $\mathbf{g}(\mathbf{q})$ the n -dimensional gravity vector, and $\boldsymbol{\tau}$ the n -dimensional vector of actuated torques.

The linear maps \mathbf{M} , \mathbf{C} and \mathbf{g} can be readily computed by Inverse Dynamics algorithms, e.g., the method from Walker and Orin (1982). In the present context, our controls $\mathbf{u} \in \mathcal{U}_{\text{adm}}$ will denote actuated torques, so that the admissible controls \mathcal{U}_{adm} form the compact set³ defined by:

$$\mathcal{U}_{\text{adm}} := \{\mathbf{u} \in \mathcal{U}, |\mathbf{u}| \leq \tau_{\max}\}, \quad (2.3)$$

Finally, we suppose that forward and inverse dynamics mappings have Lipschitz smoothness:

Assumption 2. *The forward dynamics function f is Lipschitz continuous in both of its arguments, and its inverse f^{-1} (the inverse dynamics function $\mathbf{u} = f^{-1}(\mathbf{x}, \dot{\mathbf{x}})$) is Lipschitz in both of its arguments.*

These two assumptions are satisfied when f is given by (2.2) as long as the matrices $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q})$ are bounded and the gravity term $\mathbf{g}(\mathbf{q})$ is Lipschitz. Indeed, for a small displacement between \mathbf{x} and \mathbf{x}' ,

$$\|\mathbf{u}' - \mathbf{u}\| \leq \|\mathbf{M}\| \|\ddot{\mathbf{q}}' - \ddot{\mathbf{q}}\| + \|\mathbf{C}(\mathbf{q})\| \|\dot{\mathbf{q}}' - \dot{\mathbf{q}}\|^2 + K_g \|\mathbf{q}' - \mathbf{q}\| \quad (2.4)$$

Let us illustrate this on the double pendulum depicted in Figure 3.9. When both links have mass m and length l , the gravity term

$$\mathbf{g}(\theta_1, \theta_2) = \frac{mgl}{2} [\sin \theta_1 + \sin(\theta_1 + \theta_2) \quad \sin(\theta_1 + \theta_2)]$$

³ The application of our proof of completeness to an arbitrary compact set presents no technical difficulty: one can just replace $|\mathbf{u}| \leq \tau_{\max}$ with $d(\mathbf{u}, \partial\mathcal{U}_{\text{adm}})$, with $\partial\mathcal{U}_{\text{adm}}$ the boundary of \mathcal{U}_{adm} . Using Equation (2.3) avoids this level of verbosity.

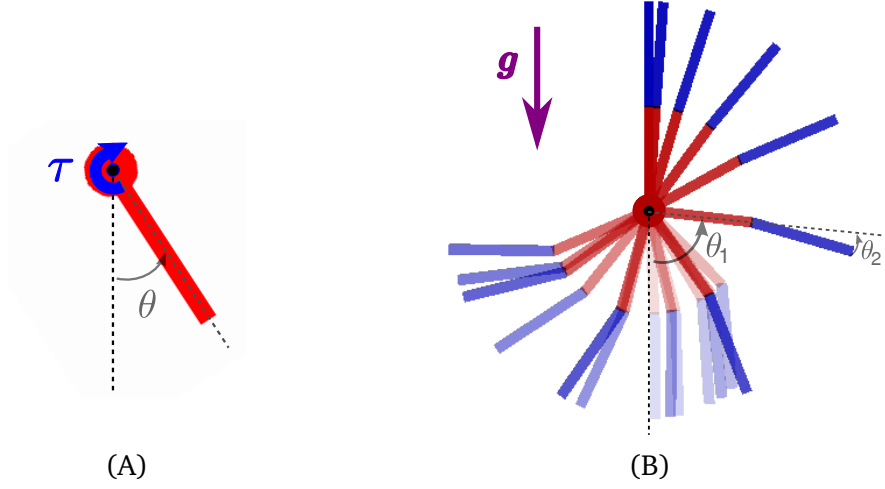


Fig. 2.2.: Single (A) and double (B) pendulums. Under torque bounds, these systems must swing back and forth several times before they can reach for the upright position, as depicted in (B) (lighter images represent earlier times).

is Lipschitz with constant $K_g = 2mgl$, while the inertial term is bounded by $\|M\| \leq 3ml^2$. When joint angular velocities are bounded by ω , the norm of the Coriolis tensor is bounded by $2\omega ml^2$. Using (2.4), one can therefore derive the Lipschitz constant $K_{f^{-1}}$ of the inverse dynamics function.

Interpolation assumptions. Our first assumption is a mere smoothness requirement:

Assumption 3. *Interpolated trajectories γ_{int} are smooth Lipschitz functions, and their time-derivatives $\dot{\gamma}_{\text{int}}$ (i.e., interpolated velocities) are also Lipschitz.*

The following two assumptions ensure a continuous behavior of the interpolation procedure:

Assumption 4 (Local boundedness). *Interpolated trajectories stay within a neighborhood of their start and end states, i.e., there exists a constant η such that, for any $(x, x') \in \mathcal{X}^2$, the interpolated trajectory $\gamma_{\text{int}} : [0, \Delta t] \rightarrow \mathcal{C}$ resulting from $\text{INTERPOLATE}(x, x')$ is included in a ball of center x and radius $\eta \|x' - x\|$.*

Assumption 5 (Discrete-acceleration convergence). *When start and end states become close, accelerations of interpolated trajectories uniformly converge to the discrete acceleration between them, i.e., there exists some $\nu > 0$ such that, if $\gamma_{\text{int}} : [0, \Delta t] \rightarrow \mathcal{C}$ results from $\text{INTERPOLATE}(x, x')$, then*

$$\forall \tau \in [0, \Delta t], \left\| \ddot{\gamma}_{\text{int}}(\tau) - \frac{\Delta \dot{q}}{\Delta t_{\text{disc}}} \right\| \leq \nu \|\Delta x\|,$$

where $\Delta t_{\text{disc}} := \|\Delta q\| / \|\dot{q}\|$.

Note that the expression $\frac{\Delta \dot{q}}{\Delta t_{\text{disc}}}$ above represents the discrete acceleration between x and x' . Its continuous analog would be $\frac{\|\dot{q}\| d\dot{q}}{\|\dot{q}\|} = \frac{\|\dot{q}\| d\dot{q}}{\|\dot{q}\| dt} = \frac{d\dot{q}}{dt}$.

These three assumptions ensure that the planner interpolates trajectories locally and “continuously” when x and x' are close. We will call them altogether *second-order continuity*, where “second-order” refers to the discrete acceleration encoded in small variations $(\Delta q, \Delta \dot{q})$. This continuous behavior plays a key role in our proof of completeness, as it ensures that denser sampling will allow finding arbitrarily narrow state-space passages.

Let us consider again the example the double pendulum, for the interpolation function $\gamma = \text{INTERPOLATE}(x, x')$ given by

$$\begin{aligned} \gamma : [0, \Delta t] &\rightarrow \mathcal{C} \\ t &\mapsto \frac{\Delta \dot{q}}{2\Delta t} t^2 + \left(\frac{\Delta q}{\Delta t} - \frac{\Delta \dot{q}}{2} \right) t + q. \end{aligned} \tag{2.5}$$

The duration Δt is taken as Δt_{disc} , so that $\gamma(0) = q$, $\gamma(\Delta t) = q'$ and $\ddot{\gamma}$ is the discrete acceleration. This interpolation, like any polynomial function, is Lipschitz smooth; Assumption 5 is verified by construction, and Assumption 4 can be checked as follows:

$$\begin{aligned} \|\gamma(t) - \gamma(0)\| &\leq t \left\| \frac{1}{2} \frac{\Delta \dot{q}}{\Delta t} t + \frac{\Delta q}{\Delta t} - \frac{\Delta \dot{q}}{2} \right\| \\ &\leq \Delta t \left\| \frac{\Delta \dot{q}}{2\Delta t} t + \frac{\Delta q}{\Delta t} - \frac{\Delta \dot{q}}{2} \right\| \\ &\leq \frac{3}{2} \|\Delta \dot{q}\| \Delta t + \|\Delta q\| \\ &\leq \|\Delta q\| \left(1 + \frac{\|\Delta \dot{q}\|}{\|\dot{q}\|} \right) \\ &\leq \|\Delta q\| \left(1 + o_{\|\Delta x\|}(1) \right). \end{aligned}$$

2.2.3 Completeness theorem

We will use the following two lemmas in the proof of our the theorem:

Lemma 1. *Let $g : [0, \Delta t] \rightarrow \mathbb{R}^k$ denote a smooth Lipschitz function. Then, for any $(t, t') \in [0, \Delta t]^2$,*

$$\left\| \dot{g}(t) - \frac{g(t') - g(t)}{|t' - t|} \right\| \leq \frac{K_g}{2} |t' - t|.$$

Proof. For $t' > t$, $\left\| \dot{g}(t) - \frac{g(t') - g(t)}{t' - t} \right\| \leq \frac{1}{t' - t} \left\| \int_t^{t'} (\dot{g}(t) - \dot{g}(w)) dw \right\|$, which is itself $\leq \frac{1}{t' - t} \int_t^{t'} \|\dot{g}(t) - \dot{g}(w)\| dw \leq \frac{K_g}{t' - t} \int_t^{t'} |t - w| dw \leq \frac{K_g}{2} (t' - t)$. \square

Lemma 2. *If there exists an admissible trajectory γ with δ -clearance in control space, then there exists $\delta' < \delta$ and a neighboring admissible trajectory γ' with δ' -clearance in control space whose acceleration never vanishes, i.e., such that $\|\ddot{\gamma}'\|$ is always greater than some constant $\ddot{m} > 0$.*

Proof. If there is a time interval $[t, t']$ on which $\ddot{\gamma} \equiv 0$, suffices to add a wavelet function $\delta\ddot{\gamma}_i$ of arbitrary small amplitude $\delta\ddot{q}_i$ and zero integral over $[t, t']$ to generate a new trajectory $\ddot{\gamma} + \delta\ddot{\gamma}$ where the acceleration cancels on at most a discrete number of time instants. Adding accelerations $\delta\ddot{\gamma}_i$ directly is possible thanks to full actuation, while δ' -clearance can be achieved for $\delta' \leq \delta$ by taking sufficiently small amplitudes $\delta\ddot{q}_i$.

Suppose now that the roots of $\ddot{\gamma}$ form a discrete set $\{t_0, t_1, \dots, t_m\}$. Let t_0 be one of these roots, and let $[t, t']$ denote a neighborhood of t_0 . Repeat the process of adding wavelet functions $\delta\ddot{\gamma}_i$ and $\delta\ddot{\gamma}_j$ of zero integral over $[t, t']$ and arbitrary small amplitude to two coordinates i and j , but this time enforcing that the sum of the two wavelets satisfies $|\delta\ddot{\gamma}_i + \delta\ddot{\gamma}_j| \geq \epsilon_{ij} > 0$. This method ensures that the root t_0 is eliminated (either $\ddot{\gamma}_i(t_0) \neq 0$ or $\ddot{\gamma}_j(t_0) \neq 0$) without introducing new roots. We conclude by iterating the process on the finite set of roots. \square

We can now state our theorem:

Theorem 1. Consider a time-invariant differential system (2.1) with smooth Lipschitz dynamics f and full actuation over a compact set of admissible controls \mathcal{U}_{adm} . Suppose that the kinodynamic planning problem between two states \mathbf{x}_{init} and \mathbf{x}_{goal} admits a smooth Lipschitz solution $\gamma : [0, T] \rightarrow \mathcal{C}$ with δ -clearance in control space. A randomized motion planner (Algorithm 1) using a second-order continuous interpolation is probabilistically complete.

Proof. Let $\gamma : [0, \Delta t] \rightarrow \mathcal{C}, t \mapsto \gamma(t)$ denote a smooth Lipschitz admissible trajectory from \mathbf{x}_{init} to \mathbf{x}_{goal} , and $u : [0, \Delta t] \rightarrow \mathcal{U}_{\text{adm}}$ its associated control trajectory with δ -clearance in control space. Consider two states $x = (\mathbf{q}, \dot{\mathbf{q}})$ and $x' = (\mathbf{q}', \dot{\mathbf{q}}')$, as well as their corresponding time instants on the trajectory

$$\begin{aligned} t &:= \arg \min_t \|(\gamma(t), \dot{\gamma}(t)) - x\|, \\ t' &:= \arg \min_t \|(\gamma(t), \dot{\gamma}(t)) - x'\|. \end{aligned}$$

Supposing without loss of generality that $t' > t$, we denote by $\Delta t = t' - t$ and $\Delta t_{\text{disc}} = \|\dot{\mathbf{q}}\| / \|\Delta \mathbf{q}\|$. Given a sufficiently dense sampling of the state space, we suppose that $\text{dist}_\gamma(x) \leq \rho$ and $\text{dist}_\gamma(x') \leq \rho$ for a radius ρ such that $\rho/\Delta t = O(\Delta t)$ and $\rho/\Delta t_{\text{disc}} = O(\Delta t)$; i.e., the radius ρ is quadratic in the time difference.

Let $\gamma_{\text{int}} : [0, \Delta t] \rightarrow \mathcal{C}$ denote the result of the interpolation between x and x' . For $\tau \in [0, \Delta t]$, the torque required to follow the trajectory γ_{int} is $\mathbf{u}_{\text{int}}(\tau) := f(\gamma_{\text{int}}(\tau), \dot{\gamma}_{\text{int}}(\tau), \ddot{\gamma}_{\text{int}}(\tau))$. Since u has δ -clearance in control space,

$$\begin{aligned} |\mathbf{u}_{\text{int}}(\tau)| &\leq |\mathbf{u}_{\text{int}}(\tau) - \mathbf{u}(t)| + |\mathbf{u}(t)| \\ &\leq |f(\gamma_{\text{int}}(\tau), \dot{\gamma}_{\text{int}}(\tau), \ddot{\gamma}_{\text{int}}(\tau)) - f(\gamma(t), \dot{\gamma}(t), \ddot{\gamma}(t))| + (1 - \delta) \tau_{\text{max}}, \end{aligned}$$

(As previously, vector inequalities are component-wise.) Let us denote by $|\widetilde{\mathbf{u}_{\text{int}}}|$ the first term of this inequality. We will now show that $|\widetilde{\mathbf{u}_{\text{int}}}| = O(\Delta t) \rightarrow 0$ when $\Delta t \rightarrow 0$, and therefore that $|\mathbf{u}_{\text{int}}(\tau)| \leq \tau_{\text{max}}$ for a small enough Δt (i.e., when sampling density is high enough). Let us first rewrite it as follows:

$$\begin{aligned} |\widetilde{\mathbf{u}_{\text{int}}}| &= |f(\gamma_{\text{int}}(\tau), \dot{\gamma}_{\text{int}}(\tau), \ddot{\gamma}_{\text{int}}(\tau)) - f(\gamma(t), \dot{\gamma}(t), \ddot{\gamma}(t))| \\ &\leq \|f(\gamma_{\text{int}}(\tau), \dot{\gamma}_{\text{int}}(\tau), \ddot{\gamma}_{\text{int}}(\tau)) - f(\gamma(t), \dot{\gamma}(t), \ddot{\gamma}(t))\|_\infty \\ &\leq K_f \|(\gamma_{\text{int}}(\tau), \dot{\gamma}_{\text{int}}(\tau)) - (\gamma(t), \dot{\gamma}(t))\| + K_f \|\ddot{\gamma}_{\text{int}}(\tau) - \ddot{\gamma}(t)\| \end{aligned}$$

$$\leq \underbrace{K_f [(\eta + \nu) \|\Delta \mathbf{x}\| + \text{dist}_\gamma(x)]}_{\text{position-velocity term (PV)}} + \underbrace{K_f \left\| \frac{\|\dot{\mathbf{q}}\|}{\|\Delta \mathbf{q}\|} \Delta \dot{\mathbf{q}} - \ddot{\gamma}(t) \right\|}_{\text{acceleration term (A)}}.$$

The replacement of the norm $\|\cdot\|$ by $\|\cdot\|_\infty$ is possible because all norms of \mathbb{R}^n are equivalent (a change in norm will be reflected by a different constant K_f). The transition from the second to the third row uses Lipschitz smoothness of f , as well as the triangular inequality to separate position-velocity and acceleration coordinates. The transition from the third to the fourth row relies on the two interpolation assumptions: local boundedness (yields the η factor in the distance term) and convergence to the discrete-acceleration (yields the ν factor in the distance term, as well as the acceleration term).

The position-velocity term (PV) satisfies:

$$(D) \leq (2\rho + \|\Delta \gamma\|)(\eta + \nu) + \rho \leq \frac{1}{2} K_\gamma (\eta + \nu) \Delta t + (1 + 2(\eta + \nu))\rho.$$

Since $\rho = O(\Delta t)$, we have (PV) = $O(\Delta t)$ and thus $|\tilde{u}| \leq (A) + O(\Delta t)$. Next, the difference (A) can be bounded as:

$$\begin{aligned} (A) &\leq \underbrace{\left\| \frac{\Delta \dot{\mathbf{q}} \|\dot{\mathbf{q}}\|}{\|\Delta \mathbf{q}\|} - \Delta \dot{\gamma} \frac{\|\dot{\gamma}(t)\|}{\|\Delta \gamma\|} \right\|}_{(\Delta)} + \underbrace{\left\| \frac{\Delta \dot{\gamma}}{\|\Delta \gamma\|} \left\| \dot{\gamma}(t) \right\| - \frac{\|\Delta \gamma\|}{\Delta t} \right\|}_{(A')} \\ &\quad + \underbrace{\left\| \frac{\Delta \dot{\gamma}}{\Delta t} - \ddot{\gamma}(t) \right\|}_{(A'')}. \end{aligned}$$

From Lemma 1, the two terms (A') and (A'') satisfy:

$$\begin{aligned} (A') &\leq \frac{K_{\dot{\gamma}} \|\Delta \dot{\gamma}\|}{2 \|\Delta \gamma\|} \Delta t = O(\Delta t), \\ (A'') &\leq \frac{K_{\dot{\gamma}}}{2} \Delta t = O(\Delta t), \end{aligned}$$

where the first upper bound $O(\Delta t)$ comes from the fact that $\frac{\|\Delta \dot{\gamma}\|}{\|\Delta \gamma\|} \underset{\Delta t \rightarrow 0}{\sim} \Delta t$. We now have $|\tilde{u}| \leq (\Delta) + O(\Delta t)$. The term (Δ) can be seen as the deviation between the discrete accelerations of γ_{int} and γ . Let us decompose it in terms of norm and angular deviations:

$$(\Delta) \leq \left\| \left(\frac{\Delta \dot{\gamma}}{\|\Delta \dot{\gamma}\|} - \frac{\Delta \dot{\mathbf{q}}}{\|\Delta \dot{\mathbf{q}}\|} \right) \frac{\|\dot{\gamma}\| \|\Delta \dot{\gamma}\|}{\|\Delta \gamma\|} + \frac{\Delta \dot{\mathbf{q}}}{\|\Delta \dot{\mathbf{q}}\|} \left(\frac{\|\Delta \dot{\gamma}\| \|\dot{\gamma}\|}{\|\Delta \gamma\|} - \frac{\|\Delta \dot{\mathbf{q}}\| \|\dot{\mathbf{q}}\|}{\|\Delta \mathbf{q}\|} \right) \right\|$$

$$\leq \underbrace{2 \frac{\|\dot{\gamma}\| \|\Delta\dot{\gamma}\|}{\|\Delta\gamma\|} \left(1 - \cos(\widehat{\Delta\dot{q}}, \Delta\dot{\gamma})\right)}_{\text{angular deviation term } (\theta)} + \underbrace{\left| \frac{\|\dot{\gamma}\| \|\Delta\dot{\gamma}\|}{\|\Delta\gamma\|} - \frac{\|\Delta\dot{q}\| \|\dot{q}\|}{\|\Delta q\|} \right|}_{\text{norm deviation term (N)}}$$

The factor $\frac{2\|\dot{\gamma}\| \|\Delta\dot{\gamma}\|}{\|\Delta\gamma\|}$ before (θ) is $O(1)$ when $\Delta t \rightarrow 0$, while simple vector geometry then shows that

$$\sin(\widehat{\Delta\dot{q}}, \Delta\dot{\gamma}) \leq \frac{\text{dist}_\gamma(x) + \text{dist}_\gamma(x')}{\|\Delta\dot{\gamma}\|} \leq \frac{\rho}{\ddot{m}\Delta t},$$

where $\ddot{m} := \min_t \|\ddot{\gamma}(t)\|$. From Lemma 2, we can assume this minimum acceleration to be strictly positive. Then, it follows from $\rho = O(\Delta t^2)$ that the sine above is $O(\Delta t)$. Recalling the fact that $1 - \cos \theta < \sin \theta$ for any $\theta \in [0, \pi/2]$, we have $(\theta) = O(\Delta t)$.

Finally,

$$\begin{aligned} \text{(N)} &\leq \frac{\|\Delta\dot{\gamma}\|}{\|\Delta\gamma\|} \|\dot{\gamma}\| - \|\dot{q}\| + \|\dot{q}\| \left| \frac{\|\Delta\dot{\gamma}\|}{\|\Delta\gamma\|} - \frac{\|\Delta\dot{q}\|}{\|\Delta q\|} \right| \\ &\leq O(\Delta t \cdot \rho) + \|\dot{q}\| \frac{(\|\Delta q\| + \|\Delta\dot{q}\|) O(\rho)}{\|\Delta q\| (\|\Delta q\| + O(\rho))} \\ &\leq O(\Delta t \cdot \rho) + \frac{\|\dot{q}\| \rho}{\|\Delta q\| + O(\rho)} + \frac{\|\dot{q}\| \|\Delta\dot{q}\|}{\|\Delta q\|} \frac{O(\rho)}{\|\Delta q\| + O(\rho)} \end{aligned}$$

Where we used the fact that $\|\Delta\gamma\| \leq \text{dist}_\gamma(x) + \|\Delta q\| + \text{dist}_\gamma(x') = \|\Delta q\| + O(\rho)$, and similarly for $\|\Delta\dot{\gamma}\|$. Because $\|\Delta q\| = \|\dot{q}\| \Delta t_{\text{disc}} + O(\Delta t_{\text{disc}}^2)$ and $\rho / \Delta t_{\text{disc}} = O(\Delta t)$, the last two fractions are $O(\Delta t)$, so our last term (N) = $O(\Delta t)$.

Overall, we have derived an upper bound $|u(\tau)| \leq (1 - \delta)\tau_{\max} + O(\Delta t)$. As a consequence, there exists a constant $\delta t > 0$ such that, whenever $\Delta t \leq \delta t$, interpolated torques satisfy $|u| \leq \tau_{\max}$ and the interpolated trajectory $\gamma_{\text{int}} = \text{INTERPOLATE}(x, x')$ is admissible. Note that the constant δt is uniform, in the sense that it does not depend on the index t on the trajectory.

Conclusion of the proof. We have effectively constructed the attraction sequence conjectured in (LaValle and Kuffner, 2001). We can now conclude the proof similarly to the strategy sketched in this previous work. Let us denote by $\mathcal{B}_t := \mathcal{B}((\gamma, \dot{\gamma})(t), \delta\rho)$, the ball of radius $\delta\rho$ centered on $(\gamma, \dot{\gamma})(t) \in \mathcal{X}$, where $\delta\rho = O(\delta t^2)$ as before. Suppose that the roadmap contains a state $x \in \mathcal{B}_t$, and let $t' := t + \delta t$. If the planner samples a state $x' \in \mathcal{B}_{t'}$, the interpolation between x and x' will be

successful and x' will be added to the roadmap. Since the volume of $\mathcal{B}_{t'}$ is non-zero, the event $\{\text{SAMPLE}(\mathcal{X}_{\text{free}}) \in \mathcal{B}_{t'}\}$ will happen with probability one as the number of extensions goes to infinity. At the initialization of the planner, the roadmap is reduced to $x_{\text{init}} = (\gamma(0), \dot{\gamma}(0))$. Therefore, using the property above, by induction on the number of time steps δt , the last state $x_{\text{goal}} = (\gamma(T), \dot{\gamma}(T))$ will be eventually added to the roadmap with probability one, and the planner will find an admissible trajectory connecting x_{init} to x_{goal} . ■

2.3 Completeness and interpolation in practice

Shkolnik et al. (2009) showed how RRTs could not be directly applied to kinodynamic planning due to their poor expansion rate at the boundaries of the roadmap. They illustrated this phenomenon on the planning problem of swinging up a (single) pendulum vertically against gravity. Let us consider the same system, *i.e.*, the one degree-of-freedom (DOF) pendulum depicted in Figure 3.9 (A), with length $l = 20$ cm and mass $m = 8$ kg. It satisfies the system assumptions of Theorem 1 *a fortiori*, as we saw that they apply to the double pendulum.

We assume that the single actuator of the pendulum, corresponding to the joint angle θ in Figure 3.9, has limited actuation power: $|\tau| \leq \tau_{\text{max}}$. The static equilibrium of the system requiring the most torque is given at $\theta = \pm\pi/2$ with $\tau = \frac{1}{2}lmg \approx 7.84$ Nm. Therefore, when $\tau_{\text{max}} < 7.84$ Nm, it is impossible for the system to raise up-right directly, and the pendulum rather needs to swing back and forth to accumulate kinetic energy before it can swing up. For any $\tau_{\text{max}} > 0$, the pendulum can achieve the swingup in a finite number of swings N , with $N \rightarrow \infty$ as $\tau_{\text{max}} \rightarrow 0$.

2.3.1 Bezier interpolation

A common solution (Jolly et al., 2009; Škrjanc and Klančar, 2010; Hauser, 2013) to connect two states (q, \dot{q}) and (q', \dot{q}') is the cubic Bezier curve (also called “Hermit curve”) which is the quadratic function $B(t)$ such that $B(0) = q$, $\dot{B}(0) = \dot{q}$, $B(T) = q'$ and $\dot{B}(T) = \dot{q}'$, where T is the fixed duration of the interpolated trajectory. Its expression is given by:

$$B(t) = \frac{-2\Delta q + T(\dot{q} + \dot{q}')}{T^3} t^3 + \frac{3\Delta q - 2\dot{q} - \dot{q}'}{T^2} t^2 + \dot{q}t + q$$

This interpolation is straightforward to implement, however it does not verify our Assumption 5, as for instance

$$\ddot{B}(0) = \frac{6\Delta\mathbf{q} - 4\dot{\mathbf{q}} - 2\dot{\mathbf{q}}'}{T^2} \xrightarrow{\Delta\mathbf{x} \rightarrow 0} \frac{-6\dot{\mathbf{q}}}{T^2} \neq 0. \quad (2.6)$$

Our proof of completeness does not apply to such interpolators: even though a feasible trajectory is sampled as closely as possible ($\Delta\mathbf{x} \rightarrow 0$), the interpolated acceleration will *not* approximate the smooth acceleration underlying the feasible trajectory.

Proposition 1. *A randomized motion planner interpolating pendulum trajectories by Bezier curves with a fixed duration T cannot find non-quasi static solutions by increasing sampling density.*

Proof. When actuation power decreases, the pendulum needs to store kinetic energy in order to swing up, which implies that all swingup trajectories go through velocities $|\dot{\theta}| > \dot{\theta}_{\text{swingup}}(\tau_{\text{max}})$. The function $\dot{\theta}_{\text{swingup}}$ increases to a positive limit $\dot{\theta}_{\text{swingup}}^{\text{lim}}$ as $\tau_{\text{max}} \rightarrow 0$, where $\dot{\theta}_{\text{swingup}}^{\text{lim}} > \sqrt{8g/l}$ from energetic considerations.⁴ Yet, feasible accelerations are also bound by $|\ddot{\theta}| \leq K\tau_{\text{max}}$ for some constant $K > 0$. Combining both observations in (2.6) yields:

$$K\tau_{\text{max}} \geq 6\frac{|\dot{\theta}|}{T^2} > 6\frac{\dot{\theta}_{\text{swingup}}(\tau_{\text{max}})}{T^2} \Rightarrow \dot{\theta}_{\text{swingup}}(\tau_{\text{max}}) \leq \frac{KT^2}{6}\tau_{\text{max}}.$$

Since the planner uses a constant T and $\dot{\theta}_{\text{swingup}}$ increases to $\dot{\theta}_{\text{swingup}}^{\text{lim}} > \sqrt{8g/l}$ when τ_{max} decreases to 0, this inequality cannot be satisfied for arbitrary small actuation power τ_{max} . Hence, even with an arbitrarily high sampling density around a feasible trajectory $\gamma(t)$, the planner will not be able to reconstruct a feasible approximation $\gamma_{\text{int}}(t)$. \square

2.3.2 Second-order continuous interpolation

Let $\dot{\mathbf{q}}_{\text{avg}} := \frac{1}{2}(\dot{\mathbf{q}} + \dot{\mathbf{q}}')$ denote the average velocity between $(\mathbf{q}, \dot{\mathbf{q}})$ and $(\mathbf{q}', \dot{\mathbf{q}}')$. Since the system has only one degree of freedom, one can interpolate trajectories that

⁴ The expression $\dot{\theta} = \sqrt{8g/l}$ corresponds to the kinetic energy $\frac{1}{4}ml\dot{\theta}^2 = mgl$, the latter being the (potential) energy of the system at rest in the upward equilibrium. During a successful last swing, the kinetic energy at $\theta = 0$ is $\frac{1}{4}ml\dot{\theta}_{\text{swingup}}^2 + \mathcal{W}_g + \mathcal{W}_\tau = mgl$, with $\mathcal{W}_g < 0$ the work of gravity and \mathcal{W}_τ the work of actuation forces between $\theta = 0$ and $\theta = \pi$. The work \mathcal{W}_τ vanishes when $\tau_{\text{max}} \rightarrow 0$.

comply with our Assumption 5 using constant accelerations with a suitable trajectory duration:

$$\begin{aligned} C : [0, \Delta t_C] &\rightarrow]-\pi, \pi] \\ t &\mapsto C(t) = \mathbf{q} + t\dot{\mathbf{q}} + \frac{t^2}{2}(\Delta\dot{\mathbf{q}}/\Delta t_C). \end{aligned}$$

One can check that choosing $\Delta t_C = (\Delta\mathbf{q}/\dot{\mathbf{q}}_{\text{avg}})$ results in $\dot{C}(0) = \dot{\mathbf{q}}$, $\dot{C}(\Delta t_C) = \dot{\mathbf{q}}'$, $C(0) = \mathbf{q}$ and $C(\Delta t_C) = \mathbf{q}'$. This duration is similar to the term Δt_{disc} in Assumption 5, with both expressions converging to the same value as $\Delta\mathbf{x} \rightarrow 0$. We call $C(t)$ the *second-order continuous 1-DOF (SOC1) interpolation*.

Note that this interpolation function only applies to single-DOF systems. For multi-DOF systems, the correct duration Δt_C used to transfer from one state to another is different for each DOF, hence constant accelerations cannot be used. One can then apply optimization techniques (Perez et al., 2012; Pham et al., 2013a) or use a richer family of curves such as piecewise linear-quadratic segments (Hauser and Ng-Thow-Hing, 2010).

2.3.3 Comparison in simulations

According to Theorem 1 and our previous discussion, a randomized planner based on Bezier interpolation is not expected to be probabilistically complete as $\tau_{\text{max}} \rightarrow 0$, while the same planner using the SOC1 interpolation will be complete at any rate. We asserted this statement in simulations of the pendulum with RRT (LaValle and Kuffner, 2000).

Our implementation of RRT is that described in Algorithm 1, with the addition of the *steer-to-goal* heuristic: every $m = 100$ steps, the planner tries to steer to \mathbf{x}_{goal} rather than \mathbf{x}_{rand} . This extra step speeds up convergence when the system reaches the vicinity of the goal area. We use uniform random sampling for $\text{SAMPLE}(S)$, while for $\text{PARENTS}(\mathbf{x}', V)$ returns the $k = 10$ nearest neighbors of \mathbf{x}' in the roadmap V .

We compared the performance of RRT with the Bezier and SOC1 interpolations, all other parameters being the same, on a single pendulum with $\tau_{\text{max}} = 5$ Nm. The RRT-SOC1 combo found a four-swing solution after 26,300 RRT extensions, building a roadmap with 6434 nodes (Figure 2.3).

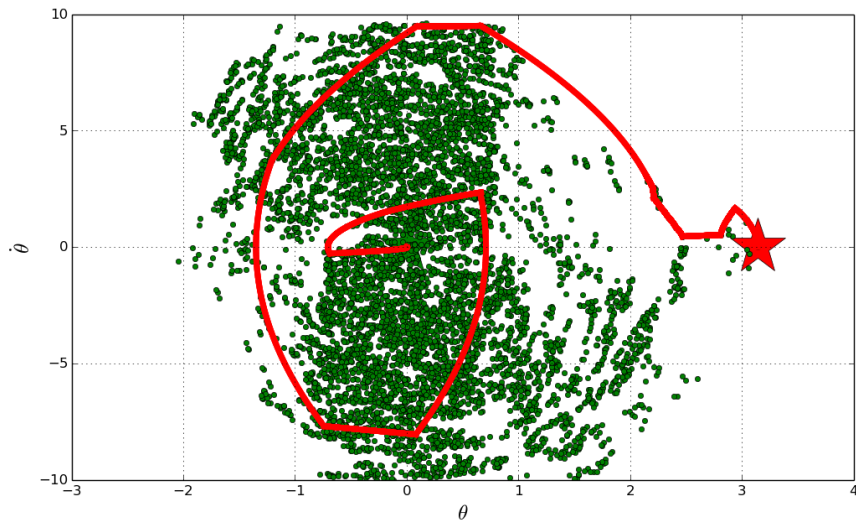


Fig. 2.3.: Phase-space portrait of the roadmap constructed by RRT using the second-order continuous (SOC1) interpolation. The planner found a successful trajectory (red line) after 26,300 extensions. This planner is probabilistically complete (Theorem 1) thanks to the fact that SOC1 curves satisfy Assumption 5.

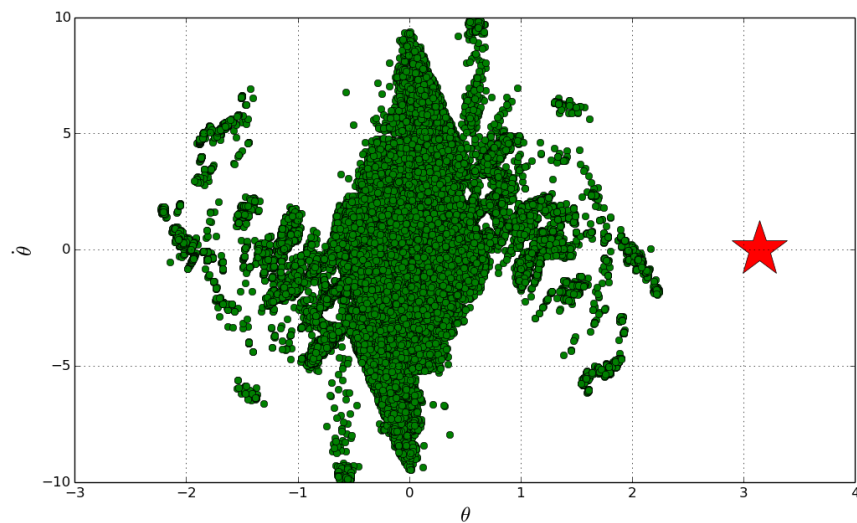


Fig. 2.4.: Roadmap constructed by RRT after 100,000 extensions using the Bezier interpolation. Reachable states are distributed in two major areas: a central, diamond shape corresponding to the states that the planner can connect at any rate, and two cones directed towards the goal ($\theta = \pi$ or $\theta = -\pi$). Even after several days of computations, this planner could not find a successful motion plan. Our completeness theorem does not apply to this planner because Bezier curves do not satisfy Assumption 5.

Meanwhile, even after one day of computations and more than 200,000 RRT extensions, the RRT-Bezier combo could not find any solution. Figure 2.4 shows the roadmap at 100,000 extensions (26,663 nodes). Interestingly, we can distinguish two zones in this roadmap. The first one is a dense, diamond-shape area near the downward equilibrium $\theta = 0$. It corresponds to states that are straightforward to connect by Bezier interpolation, and as expected from Proposition 1, velocities $\dot{\theta}$ in this area decrease sharply with θ . The second one consists of two cones directed towards the goal. Both areas exhibit a higher density near the axis $\dot{\theta} = 0$, which is also consistent with Proposition 1.

The comparison of the two roadmaps is clear: with a second-order continuous interpolation, the RRT-SOC1 planner leverages additional sampling into exploration of the state space. Conversely, RRT-Bezier lacks this property (Proposition 1), and its roadmap stays confined to a subset of the pendulum’s reachable space.

Conclusion

This chapter aimed at furthering the understanding of completeness results for kinodynamic motion planning. To the best of our knowledge, this is **the first time that a proof of probabilistic completeness is given with assumptions that can be straightforwardly checked on the robot** for kinodynamic planners that rely on trajectory interpolation.

From a practical standpoint, we have identified in our analysis the key role played by *second-order continuity* of the interpolated trajectories. While the SOC1 interpolation function that we provided here is second-order continuous, its formula does not generalize to multi-DOF systems. We will introduce in the following chapter a principled procedure that enables generalization to such systems.

Kinodynamic Planning in the Configuration Space

Kinodynamic planning usually takes places in the *state* space \mathcal{X} , a differential manifold that includes both positions and their higher-order derivatives. Kuffner et al. (2002) avoided the complexity resulting from high-dimension and differential constraints by restricting themselves to quasi-static configurations, yet at the cost of a severe restriction of the robot's motion range. In this chapter, we propose an alternative approach that preserves planning in the configuration space but is still able to explore the dynamic, non-quasi-static range of robot motions.

3.1 Time-Optimal Path Parameterization

TOPP is the method at the core of our approach. Its implementation admits two main variants that have both been explored in the literature: numerical integration (Bobrow et al., 1985; Shin and McKay, 1985), which we will now review, and convex optimization (Verschuere et al., 2009; Hauser, 2013). The reader is referred to Pham (2014) for a deeper review of implementation variants.

Let \mathbf{q} denote the n -dimensional vector of generalized coordinates of the robot. Consider second-order inequality constraints of the form

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{B}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{f}(\mathbf{q}) \leq 0, \quad (3.1)$$

where $\mathbf{A}(\mathbf{q})$, $\mathbf{B}(\mathbf{q})$ and $\mathbf{f}(\mathbf{q})$ are respectively an $M \times n$ matrix, an $n \times M \times n$ tensor and an M -dimensional vector. Inequality (3.1) is general and may represent a large variety of second-order systems and constraints. Applications include fully-actuated manipulators¹ subject to velocity, acceleration or torque limits (see e.g. Bobrow et al., 1985; Shin and McKay, 1986), wheeled vehicles subject to sliding and tip-

¹ When dry Coulomb friction or viscous damping are not negligible, one may consider adding an extra term $\mathbf{C}(\mathbf{q})\dot{\mathbf{q}}$. Such a term would simply change the computation of the fields α and β (see infra), but all the rest of the development would remain the same (Slotine and Yang, 1989).

over constraints (Shiller and Gwo, 1991), and more importantly in the scope of the present thesis, humanoid robots in multi-contact configurations (see Chapters 4 and 6). Note however that the list does not include *under-actuated* systems.

Consider a path \mathcal{P} in the configuration space, represented as the underlying path of a trajectory $\mathbf{q}(s)_{s \in [0, s_{\text{end}}]}$. Assume that $\mathbf{q}(s)_{s \in [0, s_{\text{end}}]}$ is \mathcal{C}^1 and piecewise \mathcal{C}^2 continuous.

Definition 1. A time-parameterization of \mathcal{P} , or time-reparameterization of the trajectory $\mathbf{q}(s)_{s \in [0, s_{\text{end}}]}$, is an increasing scalar function $s : [0, T'] \rightarrow [0, s_{\text{end}}]$. A time-parameterization can be seen alternatively as a velocity profile, which is the curve $\dot{s}(s)_{s \in [0, s_{\text{end}}]}$ in the s – \dot{s} plane. We say that a time-parameterization or, equivalently, a velocity profile, is valid if $s(t)_{t \in [0, T']}$ is continuous, \dot{s} is always strictly positive, and the retimed trajectory $\mathbf{q}(s(t))_{t \in [0, T]}$ satisfies the constraints of the system.

To check whether the retimed trajectory satisfies the system constraints, one may differentiate $\mathbf{q}(s(t))$ with respect to t :

$$\dot{\mathbf{q}} = \mathbf{q}_s \dot{s}, \quad \ddot{\mathbf{q}} = \mathbf{q}_s \ddot{s} + \mathbf{q}_{ss} \dot{s}^2, \quad (3.2)$$

where dots denote differentiation with respect to the time parameter t and $\mathbf{q}_s = \frac{d\mathbf{q}}{ds}$ and $\mathbf{q}_{ss} = \frac{d^2\mathbf{q}}{ds^2}$. Substituting (3.2) into (3.1) then leads to

$$\ddot{s} \mathbf{A}(\mathbf{q}) \mathbf{q}_s + \dot{s}^2 \mathbf{A}(\mathbf{q}) \mathbf{q}_{ss} + \dot{s}^2 \mathbf{q}_s^\top \mathbf{B}(\mathbf{q}) \mathbf{q}_s + \mathbf{f}(\mathbf{q}) \leq 0,$$

which can be rewritten as

$$\ddot{s} \mathbf{a}(s) + \dot{s}^2 \mathbf{b}(s) + \mathbf{c}(s) \leq 0, \quad (3.3)$$

where

$$\begin{aligned} \mathbf{a}(s) &:= \mathbf{A}(\mathbf{q}(s)) \mathbf{q}_s(s), \\ \mathbf{b}(s) &:= \mathbf{A}(\mathbf{q}(s)) \mathbf{q}_{ss}(s) + \mathbf{q}_s(s)^\top \mathbf{B}(\mathbf{q}(s)) \mathbf{q}_s(s), \\ \mathbf{c}(s) &:= \mathbf{f}(\mathbf{q}(s)). \end{aligned} \quad (3.4)$$

Each row i of equation (3.3) is of the form $a_i(s) \ddot{s} + b_i(s) \dot{s}^2 + c_i(s) \leq 0$.

- If $a_i(s) > 0$, then one has $\ddot{s} \leq \frac{-c_i(s)-b_i(s)\dot{s}^2}{a_i(s)}$. Define the acceleration *upper bound* $\beta_i(s, \dot{s}) := \frac{-c_i(s)-b_i(s)\dot{s}^2}{a_i(s)}$;
- If $a_i(s) < 0$, then one has $\ddot{s} \geq \frac{-c_i(s)-b_i(s)\dot{s}^2}{a_i(s)}$. Define the acceleration *lower bound* $\alpha_i(s, \dot{s}) := \frac{-c_i(s)-b_i(s)\dot{s}^2}{a_i(s)}$.

One can then define, for each (s, \dot{s}) ,

$$\alpha(s, \dot{s}) := \max_i \alpha_i(s, \dot{s}), \quad \beta(s, \dot{s}) := \min_i \beta_i(s, \dot{s}).$$

From the above transformations, one can conclude that $\mathbf{q}(s(t))_{t \in [0, T']}$ satisfies the constraints (3.1) if and only if

$$\forall t \in [0, T] \quad \alpha(s(t), \dot{s}(t)) \leq \ddot{s}(t) \leq \beta(s(t), \dot{s}(t)). \quad (3.5)$$

Note that $(s, \dot{s}) \mapsto (\dot{s}, \alpha(s, \dot{s}))$ and $(s, \dot{s}) \mapsto (\dot{s}, \beta(s, \dot{s}))$ can be viewed as two vector fields in the s - \dot{s} plane. One can integrate velocity profiles following the field $(\dot{s}, \alpha(s, \dot{s}))$ (from now on, α in short) to obtain *minimum acceleration* profiles (or α -profiles), or following the field β to obtain *maximum acceleration* profiles (or β -profiles).

Next, observe that if $\alpha(s, \dot{s}) > \beta(s, \dot{s})$ then, from (3.5), there is no possible value for \ddot{s} . Thus, to be valid, every velocity profile must stay below the Maximum Velocity Curve (MVC) defined by²

$$\text{MVC}(s) := \begin{cases} \min\{\dot{s} \geq 0 : \alpha(s, \dot{s}) = \beta(s, \dot{s})\} & \text{if } \alpha(s, 0) \leq \beta(s, 0), \\ 0 & \text{if } \alpha(s, 0) > \beta(s, 0). \end{cases} \quad (3.6)$$

It was shown (see e.g. Shiller and Lu, 1992) that the time-minimal velocity profile is obtained by a *bang-bang*-type control, *i.e.*, one where the optimal profile follows alternatively the β and α fields while always staying below the MVC. A method to find the optimal profile then consists in (see Figure 3.1A):

² Setting $\text{MVC}(s) = 0$ whenever $\alpha(s, 0) > \beta(s, 0)$ as in (3.6) precludes multiple-valued MVCs (see Shiller and Dubowsky, 1985). We made this choice for clarity of exposition. However, in the implementation, we did consider multiple-valued MVCs.

- find all the possible $\alpha \rightarrow \beta$ switch points. There are three types of such switch points: “discontinuous”, “singular” or “tangent” and they must all be on the MVC. The procedure to find these switch points is detailed in Pham (2014);
- from each of these switch points, integrate backward following α and forward following β to obtain the Limiting Curve (LC) (Slotine and Yang, 1989);
- construct the Concatenated Limiting Curve (CLC) by considering, for each s , the value of the lowest LC at s ;
- integrate forward from $(0, \dot{s}_{\text{beg}})$ following β and backward from $(s_{\text{end}}, \dot{s}_{\text{end}})$ following α , and consider the intersection of these profiles with each other or with the CLC. Note that the path velocities \dot{s}_{beg} and \dot{s}_{end} are computed from the desired initial and final velocities v_{beg} and v_{end} by

$$\dot{s}_{\text{beg}} := v_{\text{beg}} / \|\mathbf{q}_s(0)\|, \quad \dot{s}_{\text{end}} := v_{\text{end}} / \|\mathbf{q}_s(s_{\text{end}})\|. \quad (3.7)$$

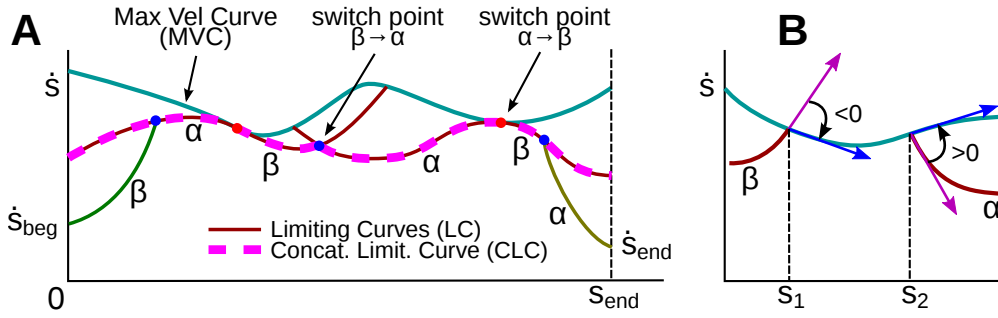


Fig. 3.1.: A: Illustration for Maximum Velocity Curve (MVC) and Concatenated Limiting Curve (CLC). The optimal velocity profile follows the green β -profile, then a portion of the CLC, and finally the yellow α -profile. B: Illustration for the Switch Point Lemma.

We now prove two lemmas that will be important later on.

Lemma 3 (Switch Point Lemma). *Assume that a forward β -profile hits the MVC at $s = s_1$ and a backward α -profile hits the MVC at $s = s_2$, with $s_1 < s_2$, then there exists at least one $\alpha \rightarrow \beta$ switch point on the MVC at some position $s_3 \in [s_1, s_2]$.*

Proof. At $(s_1, \text{MVC}(s_1))$, the angle from the vector β to the tangent to the MVC is negative (see Figure 3.1B). In addition, since we are on the MVC, we have $\alpha = \beta$, thus the angle from α to the tangent is negative too. Next, at $(s_2, \text{MVC}(s_2))$, the

angle of α to the tangent to the MVC is positive (see Figure 3.1B). Thus, since the vector field α is continuous, there exists, between s_1 and s_2

1. either a point where the angle between α and the tangent to the MVC is 0 – in which case we have a *tangent* switch point;
2. or a point where the MVC is discontinuous – in which case we have a *discontinuous* switch point;
3. or a point where the MVC is continuous but non differentiable – in which case we have a *singular* switch point.

For more details, the reader is referred to Pham (2014). □

Lemma 4 (Continuity of the CLC). *Either one of the LC's reaches $\dot{s} = 0$, or the CLC is continuous.*

Proof. Assume by contradiction that no LC reaches $\dot{s} = 0$ and that there exists a “hole” in the CLC. The left border s_1 of the hole must then be defined by the intersection of the MVC with a forward β -LC (coming from the previous $\alpha \rightarrow \beta$ switch point), and the right border s_2 of the hole must be defined by the intersection of the MVC with a backward α -LC (coming from the following $\alpha \rightarrow \beta$ switch point). By Lemma 3 above, there must then exist a switch point between s_1 and s_2 , which contradicts the definition of the hole. □

3.2 Admissible Velocity Propagation

This section presents the Admissible Velocity Propagation (AVP) algorithm, which is the key development thanks to which we will connect TOPP to kinodynamic planning. This algorithm takes as inputs:

- a path \mathcal{P} in the configuration space, and
- an interval $[\dot{s}_{\text{beg}}^{\min}, \dot{s}_{\text{beg}}^{\max}]$ of initial path velocities;

and returns the *interval* (cf. Theorem 2) $[\dot{s}_{\text{end}}^{\min}, \dot{s}_{\text{end}}^{\max}]$ of *all* path velocities that the system can reach *at the end* of \mathcal{P} after traversing \mathcal{P} while respecting the system constraints.³ The algorithm comprises the following three steps:

- A** Compute the limiting curves;
- B** Determine the *maximum* final velocity $\dot{s}_{\text{end}}^{\max}$ by integrating *forward* from $s = 0$;
- C** Determine the *minimum* final velocity $\dot{s}_{\text{end}}^{\min}$ by bisection search and by integrating *backward* from $s = s_{\text{end}}$.

We now detail each of these steps.

3.2.1 Limiting curves

We first compute the Concatenated Limiting Curve (CLC) as shown in Section 3.1. From Lemma 4, either one of the LC's reaches 0 or the CLC is continuous. The former case is covered by A1 below, while the latter is covered by A2–5.

- A1** One of the LC's hits the line $\dot{s} = 0$. In this case, the path cannot be traversed by the system without violating the kinodynamic constraints: AVP returns *Failure*. Indeed, assume that a backward (α) profile hits $\dot{s} = 0$. Then any profile that goes from $s = 0$ to $s = s_{\text{end}}$ must cross that profile somewhere and *from above*, which violates the α bound (see Figure 3.2A). Similarly, if a forward (β) profile hits $\dot{s} = 0$, then that profile must be crossed somewhere and *from below*, which violates the β bound. Thus, no valid profile can go from $s = 0$ to $s = s_{\text{end}}$;

The CLC is now assumed to be continuous and strictly positive. Since it is bounded by $s = 0$ from the left, $s = s_{\text{end}}$ from the right, $\dot{s} = 0$ from the bottom and the MVC from the top, there are only four exclusive and exhaustive cases, listed below.

- A2** The CLC hits the MVC while integrating backward and while integrating forward. In this case, let $\dot{s}_{\text{beg}}^* := \text{MVC}(0)$ and go to **B**. The situation where there is no switch point is assimilated to this case;

³ Johnson and Hauser (2012) also introduced a velocity interval propagation algorithm along a path but for pure kinematic constraints and moving obstacles.

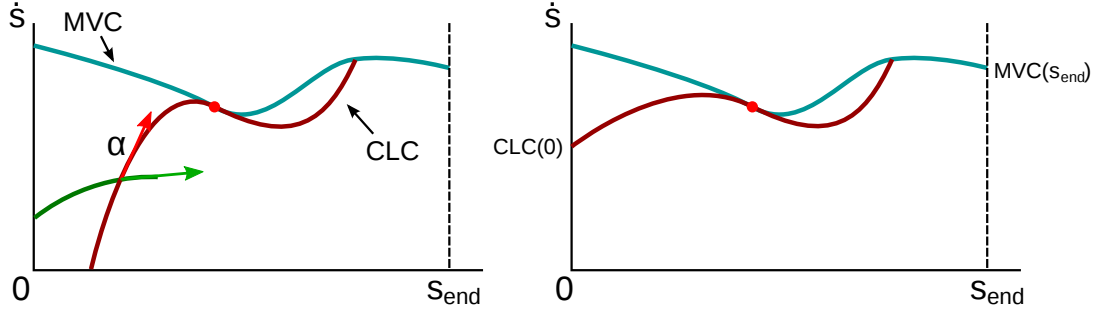


Fig. 3.2.: Illustration for step A (computation of the LC's). **Left:** illustration for case A1. A profile that crosses an α -CLC violates the α bound. **Right:** illustration for case A3.

- A3** The CLC hits $s = 0$ while integrating backward, and the MVC while integrating forward (see Figure 3.2B). In this case, let $\dot{s}_{\text{beg}}^* := \text{CLC}(0)$ and go to **B**;
- A4** The CLC hits the MVC while integrating backward, and $s = s_{\text{end}}$ while integrating forward. In this case, let $\dot{s}_{\text{beg}}^* := \text{MVC}(0)$ and go to **B**;
- A5** The CLC hits $s = 0$ while integrating backward, and $s = s_{\text{end}}$ while integrating forward. In this case, let $\dot{s}_{\text{beg}}^* := \text{CLC}(0)$ and go to **B**.

3.2.2 Maximum final velocity

Note that, in any of the cases A2–5, \dot{s}_{beg}^* was defined so that no valid profile can start above it. Thus, if $\dot{s}_{\text{beg}}^{\min} > \dot{s}_{\text{beg}}^*$, the path is not traversable: AVP returns Failure. Otherwise, the interval of valid initial velocities is $[\dot{s}_{\text{beg}}^{\min}, \dot{s}_{\text{beg}}^{\max*}]$ where $\dot{s}_{\text{beg}}^{\max*} := \min(\dot{s}_{\text{beg}}^{\max}, \dot{s}_{\text{beg}}^*)$.

Definition 2. Under the nomenclature introduced in Definition 1, we say that a velocity \dot{s}_{end} is a valid final velocity if there exists a valid profile that starts at $(0, \dot{s}_0)$ for some $\dot{s}_0 \in [\dot{s}_{\text{beg}}^{\min}, \dot{s}_{\text{beg}}^{\max}]$ and ends at $(s_{\text{end}}, \dot{s}_{\text{end}})$.

We argue that the maximum valid final velocity can be obtained by integrating forward from $\dot{s}_{\text{beg}}^{\max*}$ following β . Let's call Φ the velocity profile obtained by doing so. Since Φ is continuous and bounded by $s = s_{\text{end}}$ from the right, $\dot{s} = 0$ from the bottom, and either the MVC or the CLC from the top, there are four exclusive and exhaustive cases, listed below (see Figure 3.3 for illustration).

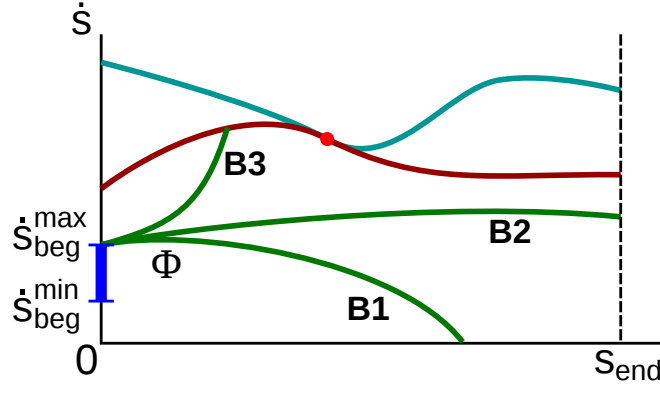


Fig. 3.3.: Illustration for step B: one can determine the maximum final velocity by integrating forward from $(0, \dot{s}_{\text{beg}}^*)$.

B1 Φ hits $\dot{s} = 0$ (cf. profile B1 in Figure 3.3). Here, as in the case A1, the path is not traversable: AVP returns *Failure*. Indeed, any profile that starts below $\dot{s}_{\text{beg}}^{\max}$ and tries to reach $s = s_{\text{end}}$ must cross Φ somewhere and *from below*, thus violating the β bound;

B2 Φ hits $s = s_{\text{end}}$ (cf. profile B1 in Figure 3.3). Then $\Phi(s_{\text{end}})$ corresponds to the $\dot{s}_{\text{end}}^{\max}$ we are looking for. Indeed, $\Phi(s_{\text{end}})$ is reachable – precisely by $\Phi -$, and to reach any value above $\Phi(s_{\text{end}})$, the corresponding profile would have to cross Φ somewhere and from below;

B3 Φ hits the CLC. There are two sub-cases:

1. If we proceed from cases A4 or A5 (in which the CLC reaches $s = s_{\text{end}}$, cf. profile B3 in Figure 3.3), then $\text{CLC}(s_{\text{end}})$ corresponds to the $\dot{s}_{\text{end}}^{\max}$ we are looking for. Indeed, $\text{CLC}(s_{\text{end}})$ is reachable – precisely by the concatenation of Φ and the CLC –, and no value above $\text{CLC}(s_{\text{end}})$ can be valid by the definition of the CLC;
2. If we proceed from cases A2 or A3, then the CLC hits the MVC while integrating forward, say at $s = s_1$; we then proceed as in case B4 below;

B4 Φ hits the MVC, say at $s = s_1$. It is clear that $\text{MVC}(s_{\text{end}})$ is an upper bound of the valid final velocities, but we have to ascertain whether this value is reachable. For this, we use the predicate `IS_VALID` defined in Algorithm 3 of C:

- if `IS_VALID(MVC(s_{end}))`, then $\text{MVC}(s_{\text{end}})$ is the $\dot{s}_{\text{end}}^{\max}$ we are looking for;

- else, the path is not traversable: AVP returns Failure. Indeed, as we shall see, if for a certain \dot{s}_{test} , the predicate $\text{IS_VALID}(\dot{s}_{\text{test}})$ is False, then no velocity below \dot{s}_{test} can be valid either.

3.2.3 Minimum final velocity

Assume that we proceed from the cases B2–4. Consider a final velocity \dot{s}_{test} where

- $\dot{s}_{\text{test}} < \Phi(s_{\text{end}})$ if we proceed from B2;
- $\dot{s}_{\text{test}} < \text{CLC}(s_{\text{end}})$ if we proceed from B3a;
- $\dot{s}_{\text{test}} < \text{MVC}(s_{\text{end}})$ if we proceed from B3b or B4.

Let us integrate backward from $(s_{\text{end}}, \dot{s}_{\text{test}})$ following α and call the resulting profile Ψ . We have the following lemma.

Lemma 5. *Ψ cannot hit the MVC before hitting either Φ or the CLC.*

Proof. If we proceed from B2 or B3a, then it is clear that Ψ must first hit Φ (case B2) or the CLC (case B3a) before hitting the MVC. If we proceed from B3b or B4, assume by contradiction that Ψ hits the MVC first at a position $s = s_2$. Then by Lemma 3, there must exist a switch point between s_2 and the end of the CLC (in case B3b) or the end of Φ (in case B4). In both cases, there is a contradiction with the fact that the CLC is continuous. \square

We can now detail in Algorithm 3 the predicate IS_VALID which assesses whether a final velocity \dot{s}_{test} is valid.

At this point, we have that, either the path is not traversable, or we have determined $\dot{s}_{\text{end}}^{\text{max}}$ in B. Remark from C3–5 that, if some \dot{s}_0 is a valid final velocity, then any $\dot{s} \in [\dot{s}_0, \dot{s}_{\text{end}}^{\text{max}}]$ is also valid. Similarly, from C1 and C2, if some \dot{s}_0 is *not* a valid final velocity, then *no* $\dot{s} \leq \dot{s}_0$ can be valid. We have thus established the following result:

Theorem 2. *The set of valid final velocities is an interval.*

This interval property enables one to efficiently search for the minimum final velocity as follows. First, test whether 0 is a valid final velocity: if $\text{IS_VALID}(0)$, then

Algorithm 3 IS_VALID

Input: candidate final velocity \dot{s}_{test} **Output:** True iff there exists a valid velocity profile with final velocity \dot{s}_{test}

Consider the profile Ψ constructed above. Since it must hit Φ or the CLC before hitting the MVC, the following five cases are exclusive and exhaustive (see Figure 3.4 for illustrations):

- C1** Ψ hits $\dot{s} = 0$ (Figure 3.4, profile C1). Then, as in cases A1 or B1, no velocity profile can reach \dot{s}_{test} : return False;
 - C2** Ψ hits $s = 0$ for some $\dot{s}_0 < \dot{s}_{\text{beg}}^{\min}$ (see Figure 3.4, profile C2). Then any profile that ends at \dot{s}_{test} would have to hit Ψ from above, which is impossible: return False;
 - C3** Ψ hits $s = 0$ at a point $\dot{s}_0 \in [\dot{s}_{\text{beg}}^{\min}, \dot{s}_{\text{beg}}^{\max*}]$ (Figure 3.4, profile C3). Then \dot{s}_{test} can be reached following the valid velocity profile Ψ : return True. (Note that, if $\dot{s}_0 > \dot{s}_{\text{beg}}^{\max*}$ then Ψ must have crossed Φ somewhere before arriving at $s = 0$, which is covered by case C4 below);
 - C4** Ψ hits Φ (Figure 3.4, profile C4). Then \dot{s}_{test} can be reached, precisely by the concatenation of a part of Φ and Ψ : return True;
 - C5** Ψ hits the CLC (Figure 3.4, profile C5). Then \dot{s}_{test} can be reached, precisely by the concatenation of Φ , a part of the CLC and Ψ : return True.
-

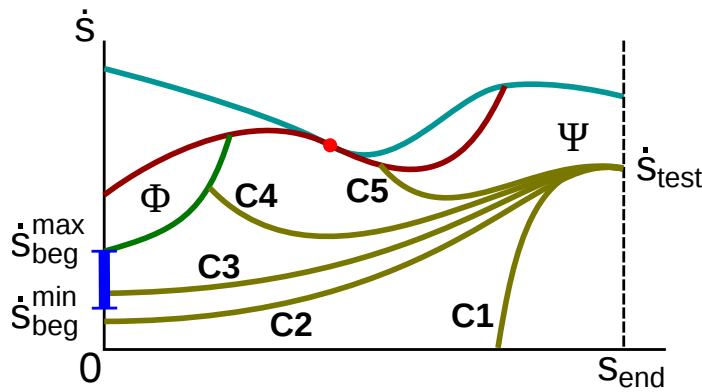


Fig. 3.4.: Illustration for the predicate IS_VALID: one can assess whether a final velocity \dot{s}_{test} is valid by integrating backward from $(s_{\text{end}}, \dot{s}_{\text{test}})$.

the sought-after $\dot{s}_{\text{end}}^{\min}$ is 0. Else, run a standard bisection search with initial bounds $(0, \dot{s}_{\text{end}}^{\max}]$ where 0 is not valid and $\dot{s}_{\text{end}}^{\max}$ is valid. Thus, after executing $\log_2(1/\epsilon)$ times the routine IS_VALID, one can determine $\dot{s}_{\text{end}}^{\min}$ with a precision ϵ .

3.2.4 Implementation

AVP can be readily adapted from the numerical integration approach to TOPP. Pham et al. (2013b) implemented AVP in about 100 lines of C++ code based on the TOPP library (Pham, 2014). The latter now implements both features in a single package, which has been publicly released online:

 <https://github.com/quangounet/TOPP>

The library includes the following extensions. We will not cover them in here, and refer the reader to suitable references for further investigation:

First-order velocity bounds: The derivation present here encounters a singularity when $\mathbf{A}(\mathbf{q}) = \mathbf{0}$ in Equation (3.1). Zlajpah (1996) showed how to further take into account velocity bounds of the form $\dot{\mathbf{q}}^\top \mathbf{B}_v(\mathbf{q}) \dot{\mathbf{q}} + \mathbf{f}_v(\mathbf{q}) \leq 0$.

Backward integration: the “AVP-backward” problem is: given an interval of final velocities $[\dot{s}_{\text{end}}^{\min}, \dot{s}_{\text{end}}^{\max}]$, compute the interval $[\dot{s}_{\text{beg}}^{\min}, \dot{s}_{\text{beg}}^{\max}]$ of all possible initial velocities. Lertkultanon and Pham (2014) showed how this problem can be solved similarly to AVP via backward integration and bisection search of the initial (instead of final) velocities. Solving the backward version of the problem enables the integration with the *bi-directional* variant of RRT.

3.3 Combining AVP with randomized planners

The AVP algorithm can be combined with various iterative path planners. We now detail a planner, which we call *AVP-RRT*, that integrates AVP within the original RRT algorithm described by LaValle and Kuffner (2000).

3.3.1 AVP-RRT

The AVP-RRT algorithm is described in Algorithm 7 and illustrated in Figure 3.5. It inherits from RRT the iterative extension of a tree \mathcal{T} in the configuration space. However, in contrast with the standard RRT, a vertex V here consists of a triple $(V.\text{config}, V.\text{inpath}, V.\text{interval})$ where $V.\text{config}$ is an element of the configuration space \mathcal{C} , $V.\text{inpath}$ is a path $\mathcal{P} \subset \mathcal{C}$ that connects the configuration of V 's parent to $V.\text{config}$, and $V.\text{interval}$ is the interval of reachable velocities at $V.\text{config}$, that is, at the end of $V.\text{inpath}$.

At each iteration, a random configuration q_{rand} is generated. The EXTEND routine (see Algorithm 5) then tries to extend the tree \mathcal{T} towards q_{rand} from the closest – in a certain metric d – vertex in \mathcal{T} . The algorithm terminates when either

- A newly-found vertex can be connected to the goal configuration (line 10 of Algorithm 7). In this case, AVP guarantees by recursion that there exists a path from q_{start} to q_{goal} and that this path is time-parameterizable;
- After N_{maxrep} repetitions, no vertex could be connected to q_{goal} . In this case, the algorithm returns Failure.

The other routines are defined as follows:

- $\text{CONNECT}(V, q_{\text{goal}})$ attempts at connecting directly V to the goal configuration q_{goal} , using the same algorithm as in lines 2 to 10 of Algorithm 5, but with the further requirement that the goal velocity is included in the final velocity interval.
- $\text{COMPUTE_TRAJECTORY}(\mathcal{T}, q_{\text{goal}})$ reconstructs the entire path $\mathcal{P}_{\text{total}}$ from q_{start} to q_{goal} by recursively concatenating the $V.\text{inpath}$. Next, $\mathcal{P}_{\text{total}}$ is time-parameterized by applying TOPP. The existence of a valid time-parameterization is guaranteed by recursion by AVP.
- $\text{NEAREST_NEIGHBOR}(\mathcal{T}, q)$ returns the vertex of \mathcal{T} whose configuration is closest to configuration q in the metric d .
- $\text{INTERPOLATE}(V, q)$ returns a path \mathcal{P}_{new} connecting $V.\text{config}$ to q . This path needs to be *first-order continuous*.

By *first-order continuous*, we mean that the interpolated path $\mathcal{P}_{\text{new}} : q : [0, s_{\text{end}}] \rightarrow \mathcal{C}$ is

Algorithm 4 AVP-RRT

Input: q_{start} , q_{goal} **Output:** A valid trajectory connecting q_{start} to q_{goal} or Failure

```
1:  $\mathcal{T} \leftarrow \text{NEW\_TREE}()$ 
2:  $V_{\text{start}} \leftarrow \text{NEW\_VERTEX}()$ 
3:  $V_{\text{start}}.\text{config} \leftarrow q_{\text{start}}$ ;  $V_{\text{start}}.\text{inpath} \leftarrow \text{Null}$ ;  $V_{\text{start}}.\text{interval} \leftarrow [0, 0]$ 
4:  $\text{INITIALIZE}(\mathcal{T}, V_{\text{start}})$ 
5: for  $\text{rep} = 1$  to  $N_{\text{maxrep}}$  do
6:    $q_{\text{rand}} \leftarrow \text{RANDOM\_CONFIG}()$ 
7:    $V_{\text{new}} \leftarrow \text{EXTEND}(\mathcal{T}, q_{\text{rand}})$ 
8:   if  $\text{EXTEND}$  succeeds then
9:      $\text{ADD\_VERTEX}(\mathcal{T}, V_{\text{new}})$ 
10:    if  $\text{CONNECT}(V_{\text{new}}, q_{\text{goal}})$  succeeds then
11:      return  $\text{COMPUTE\_TRAJECTORY}(\mathcal{T}, q_{\text{goal}})$ 
12:    end if
13:  end if
14: end for
15: return Failure
```

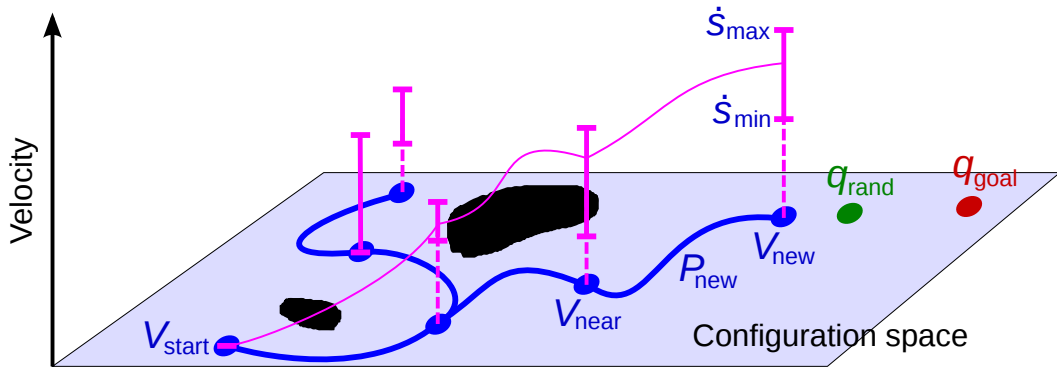


Fig. 3.5.: Illustration for AVP-RRT. The horizontal plane represents the configuration space while the vertical axis represents the path velocity space. Black areas represent configuration space obstacles. A vertex in the tree is composed of a configuration (blue disks), the incoming path from the parent (blue curve), and the interval of admissible velocities (bold magenta segments). At each tree extension step, one interpolates a smooth, collision-free path in the configuration space and propagates the interval of admissible velocities along that path using AVP. The fine magenta line shows one possible valid velocity profile (which is guaranteed to exist by AVP) “above” the path connecting q_{start} and q_{new} .

Algorithm 5 EXTEND

Input: $\mathcal{T}, \mathbf{q}_{\text{rand}}$ **Output:** A new vertex V_{new} or Null

```
1:  $V_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOR}(\mathcal{T}, \mathbf{q}_{\text{rand}})$ 
2:  $(\mathcal{P}_{\text{new}}, \mathbf{q}_{\text{new}}) \leftarrow \text{INTERPOLATE}(V_{\text{near}}, \mathbf{q}_{\text{rand}})$ 
3: if  $\mathcal{P}$  is collision-free then
4:    $[\dot{s}_{\min}, \dot{s}_{\max}] \leftarrow \text{AVP}(\mathcal{P}_{\text{new}}, V_{\text{near}}.\text{interval})$ 
5:   if AVP succeeds then
6:      $V_{\text{new}} \leftarrow \text{NEW\_VERTEX}()$ 
7:      $V_{\text{new}}.\text{config} \leftarrow \mathbf{q}_{\text{new}}$ 
8:      $V_{\text{new}}.\text{inpath} \leftarrow \mathcal{P}_{\text{new}}$ 
9:      $V_{\text{new}}.\text{interval} \leftarrow [\dot{s}_{\min}, \dot{s}_{\max}]$ 
10:    return  $V_{\text{new}}$ 
11:   end if
12: end if
13: return Failure
```

1. smooth in the Lipschitz sense and locally bounded, as for *second-order continuity* (see Section 2.2.2)
2. matches the boundary conditions that $\mathbf{q}(0) = V.\text{config}$, $\mathbf{q}(s_{\text{end}}) = \mathbf{q}$ and $\dot{\mathbf{q}}(0)$ is aligned with the last velocity vector \mathbf{v}_{end} of V .

The second condition ensures that the concatenation of $V.\text{inpath}$ and the \mathcal{P}_{new} is \mathcal{C}^1 continuous at $V.\text{config}$. Note how it only involves velocity coordinates, while the property of *second-order continuity* defined in Chapter 2 binds interpolated accelerations. First-order continuity is a weaker requirement thanks to the ability of AVP to connect acceleration coordinates in a later step. The resulting planner still achieves probabilistic completeness, as we will see below.

We showed in Chapter 2 how second-order continuity of interpolation functions was crucial to state-space planners. Yet, we did not provide a second-order continuous solution for systems with more than one DOF. Now that we have integrated TOPP within AVP-RRT, we can generate such solutions for multi-DOF systems from first-order continuous interpolations. This time, Bezier curves can be utilized to produce the latter.

3.3.2 Comparison to state-space planning

We demonstrate the extension to multi-DOF systems on the torque-limited double-pendulum depicted in Figure 3.9 (B). Such a pendulum can be seen as a 2-link manipulator, so that the reduction to the form of (3.1) is straightforward. The task

is again to bring the pendulum from its initial state $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = (0, 0, 0, 0)$ to the upright state $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = (\pi, 0, 0, 0)$, while respecting the torque bounds. For simplicity, we did not consider self-collision issues.

Any trajectory that achieves the task must pass through a configuration where $\theta_1 = \pi/2$. Note that the configuration with $\theta_1 = \pi/2$ that requires the smallest torque at the first joint to stay still is $(\theta_1, \theta_2) = (\pi/2, \pi)$. Let then τ_1^{qs} be this smallest torque. It is clear that, if $\tau_1^{\text{max}} < \tau_1^{\text{qs}}$, then *no quasi-static trajectory* can achieve the task. We chose a link length $l = 0.2$ m and mass $m = 8$ kg for each link, which yields $\tau_1^{\text{qs}} = 15.68$ N·m. For information, the smallest torque at the second joint to keep the configuration $(\theta_1, \theta_2) = (0, \pi/2)$ stationary was 7.84 N·m.

We implemented AVP-RRT and a state-space RRT on the

K nearest neighbors: attempting connection from $K > 1$ nearest neighbors can improve the performance of RRT. We compared various values of the neighborhood size $K \in \{1, 10, 40, 100\}$.

Steer-to-goal: every m extension attempts, the planner tries to steer the new state directly to \mathbf{x}_{goal} . We observed that the choice of the period m of this operation did not significantly alter the performance of the algorithm, and thus only report results for the reasonable value $m = 5$.

Metric: AVP-RRT uses the configuration-space distance defined by

$$d_{\mathcal{C}}(\mathbf{q}_a, \mathbf{q}_b) = \frac{1}{2} \sum_{j=1,2} \sqrt{1 - \cos(\mathbf{q}_{aj} - \mathbf{q}_{bj})},$$

which is similar to an Euclidean metric but takes into account the periodicity of the joint values. Similarly, the state-space RRT uses the metric:

$$d_{\mathcal{X}}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{4} \sum_{j=1,2} \sqrt{1 - \cos(\mathbf{q}_{aj} - \mathbf{q}_{bj})} + \frac{1}{4V_{\text{max}}} \sum_{j=1,2} |\mathbf{v}_{aj} - \mathbf{v}_{bj}|,$$

where V_{max} denotes the maximum velocity bound set in the uniform random sampling.

We carried experiments for $(\tau_1^{\text{max}}, \tau_2^{\text{max}}) \in \{(11, 7), (13, 5), (11, 5)\}$ (N·m). Inverse Dynamics computations required by TOPP were performed using OpenRAVE (Di-


ankov, 2010), which implements the algorithm from Walker and Orin (1982). We ran 40 simulations for each value of $(\tau_1^{\max}, \tau_2^{\max})$ on a 2 GHz Intel Core Duo computer with 2 GB RAM. Figure 3.6, 3.7, 3.8 and Table 3.1 summarize the results.

	$\tau^{\max} = (11, 7)$		$\tau^{\max} = (11, 5)$	
Planner	Success rate	Search time (min)	Success rate	Search time (min)
AVP-RRT	100%	3.3 ± 2.6	100%	9.8 ± 12.1
RRT-1	40%	70.0 ± 34.1	47.5%	63.8 ± 36.6
RRT-10	82.5%	53.1 ± 59.5	85%	56.3 ± 60.1
RRT-40	92.5%	44.6 ± 42.6	87.5%	54.6 ± 52.2
RRT-100	82.5%	88.4 ± 54.0	92.5%	81.2 ± 46.7

Tab. 3.1.: Comparing the success rate and search time of the configuration-space AVP-RRT planner versus its state-space counterpart. On instances with severe torque constraints, AVP-RRT always found solutions, and did so much faster than the state-space RRT in all our experiments.

In the two problem instances, AVP-RRT was respectively 13.4 and 5.6 times faster than the best state-space RRT in terms of search time. We noted however that the search time of AVP-RRT increased significantly from instance $(\tau_{\max}^1, \tau_{\max}^2) = (11, 5)$ to instance $(\tau_{\max}^1, \tau_{\max}^2) = (11, 7)$, while that of RRT only marginally increased. This may be caused by the “superposition” phenomenon: as torque constraints become tighter, more “pumping” swings are necessary to reach the upright configuration. However, since our metric was only on the configuration-space variables, configurations with different speeds (corresponding to different pumping cycles) may become indistinguishable. This issue can be addressed in future work by including a measure of reachable velocity intervals directly into the metric d_C .

3.3.3 Implementation

In these simulations, we used the Python implementation of AVP developed in Pham et al. (2013b). See (Pham et al., 2014) for details on how the various parameters of the state-space RRT were tuned. The source code is also publicly available:  <https://github.com/stephane-caron/rss-2013>

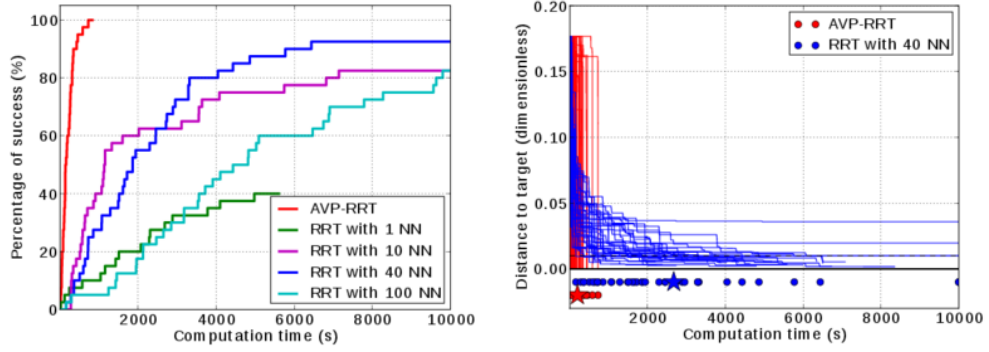


Fig. 3.6.: Performances of AVP-RRT compared to that of a state-space RRT. Left: percentage of trials that have reached the goal area at given time instants for $\tau_{\max} = (11, 7)$. Right: individual plots for each trial. Each curve shows the distance to the goal as a function of time for a given instance (red: AVP-RRT, blue: RRT-40). Dots indicate the time instants when a trial successfully terminated. Stars show the mean values of termination times.

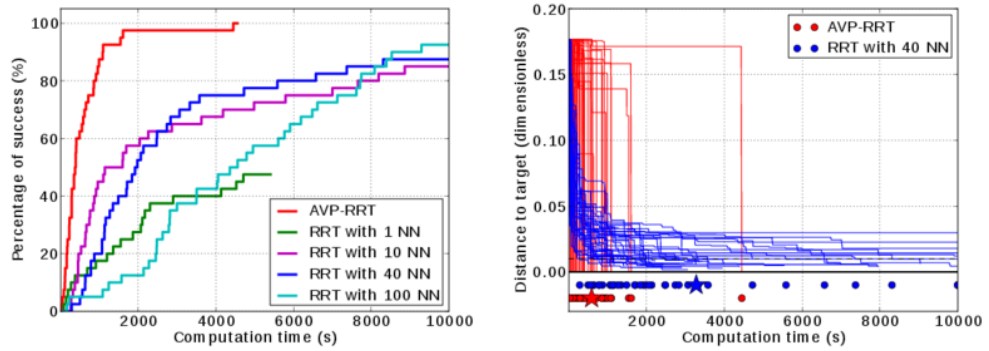


Fig. 3.7.: Performances of AVP-RRT compared to that of a state-space RRT. Left: percentage of trials that have reached the goal area at given time instants for $\tau_{\max} = (11, 5)$. Right: individual plots for each trial. Each curve shows the distance to the goal as a function of time for a given instance (red: AVP-RRT, blue: RRT-40). Dots indicate the time instants when a trial successfully terminated. Stars show the mean values of termination times.

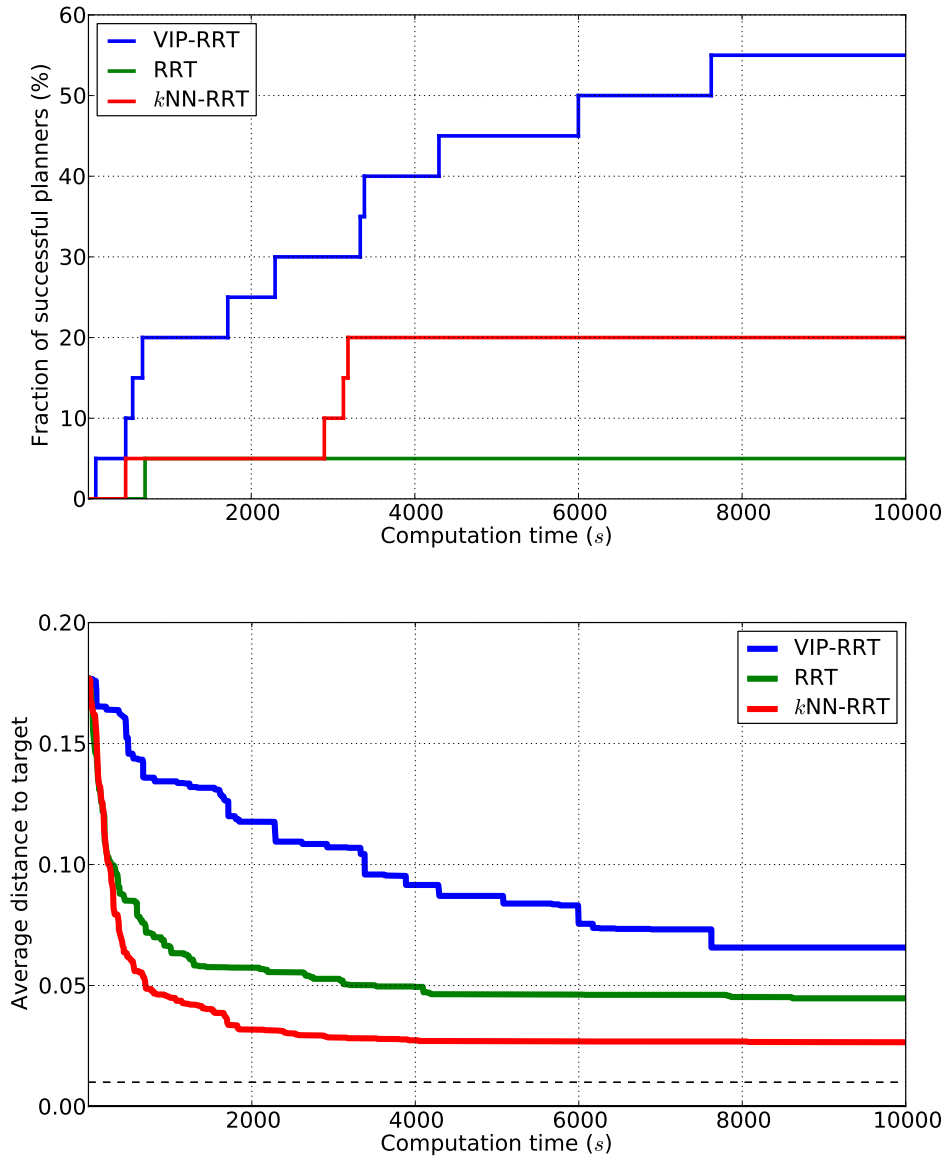


Fig. 3.8.: RRT, k NN-RRT and VIP-RRT ran over 40 instances of a double inverted pendulum with torque limits (8 Nm on the first joint and 4 Nm on the second one). The horizontal axis shows computation time, while the vertical axis represents the number of successful planners (left) and the distance to the goal area (right) averaged over all runs.

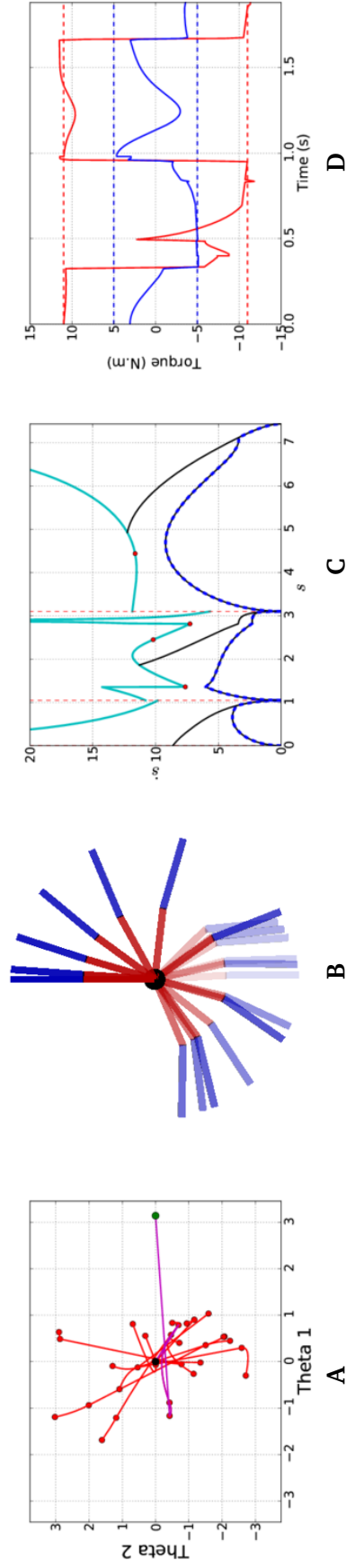


Fig. 3.9.: Swinging up a fully-actuated double pendulum. A typical solution for the case $(\tau_1^{\max}, \tau_2^{\max}) = (11, 5)$ N·m, with trajectory duration 1.88 s (see also the attached video). **A:** The tree in the (θ_1, θ_2) space. The final path is highlighted in magenta. **B:** Snapshots of the trajectory, taken every 0.1 s. Snapshots taken near the beginning of the trajectory are lighter. **C:** Velocity profiles in the (s, \dot{s}) space. The MVC is in cyan. The various velocity profiles (CLC, Φ , Ψ , cf. Section 3.2.2) are in black. The final, optimal, velocity profile is in dashed blue. The vertical dashed red lines correspond to vertices where 0 is a valid velocity, which allowed a discontinuity of the path tangent at that vertex. **D:** Torques profiles. The torques for joint 1 and 2 are respectively in red and in blue. The torque limits are in dotted line. Note that, in agreement with time-optimal control theory, at each time instant, at least one torque limit was saturated (the small overshoots were caused by discretization errors).

3.4 Probabilistic completeness of AVP-based planners

Essentially, establishing the probabilistic completeness of an AVP-based planner requires proving the following two propositions: (1) any smooth path \mathcal{P} in the configuration space will be approximated as closely as needed by the sampling process when computation time goes to infinity; (2) if a smooth path \mathcal{P} obtained by the sampling process can be time-parameterized – according to a certain velocity profile v – into a solution trajectory, then v is contained within the velocity band propagated by AVP. Indeed, assuming the above two propositions, a “high-level” proof of probabilistic completeness of an AVP-based planner can go as follows.

Let d designate the \mathcal{L}_1 distance between two trajectories or between two paths: $d(\Pi, \Pi') = \sup_{t \in [0, T]} \|\Pi(t) - \Pi'(t)\|$ and $d(\mathcal{P}, \mathcal{P}') = \sup_{s \in [0, 1]} \|\mathcal{P}(s) - \mathcal{P}'(s)\|$. Assume that there exists a smooth state-space trajectory Π that solves the query, with Δ -clearance in the state space, i.e., every smooth trajectory Π' such that $d(\Pi, \Pi') \leq \Delta$ also solves the query. Let \mathcal{P} be the underlying path of Π in the configuration space. By proposition (1), with probability 1, there exists a time instant when the sampling process will generate a smooth path \mathcal{P}' such that $d(\mathcal{P}, \mathcal{P}') \leq \Delta/3$ and that the distances between the unit tangent vectors along \mathcal{P} and \mathcal{P}' are also bounded by $\Delta/3$. One can then construct a velocity profile v' above \mathcal{P}' (see Figure 3.10), such that the time-parameterization of \mathcal{P}' according to that profile yields a trajectory Π' within a radius Δ of Π . As Π has Δ -clearance, Π' also solves the query. Thus, by proposition (2), the velocity profile (or time-parameterization) v' must be contained within the velocity band propagated by AVP, which implies finally that \mathcal{P}' can be successfully time-parameterized in the last step of the planner.

As presented in Section 3.3, AVP-RRT might not be probabilistically complete, essentially because it might not satisfy proposition (1). Note that proving proposition (1) for AVP-PRM would likely be less troublesome. We now present a modification to AVP-RRT (basically a hybridization with PRM), which enables us to prove proposition (1).

Fix a $\delta > 0$. Each time a random configuration \mathbf{q}_{rand} is sampled, after the original AVP-RRT propagation attempt, we consider the set \mathcal{S} of existing vertices within a

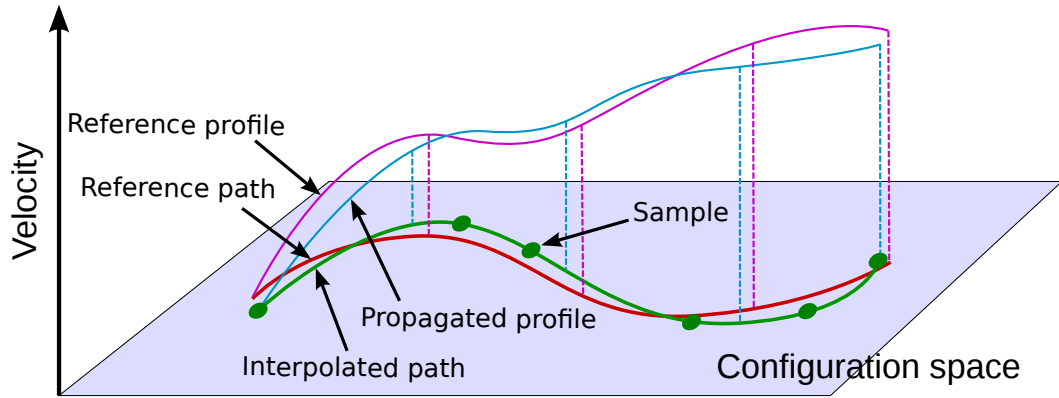


Fig. 3.10.: Illustration for the existence of an admissible velocity profile above an approximated path in the proof of completeness for AVP-RRT.

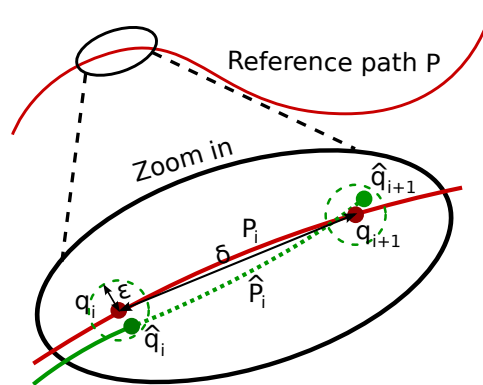


Fig. 3.11.: Approximation of a given smooth path in the proof of completeness for AVP-RRT.

distance δ of \mathbf{q}_{rand} in the configuration space. We select K random vertices within S (the value of K will be determined later), and attempt to connect these vertices to \mathbf{q}_{rand} through the usual interpolation and AVP procedures. For each successful connection, we create a new vertex V'_{new} , which has the same configuration \mathbf{q}_{rand} but a different “inpath” and a different “interval”, depending on the parent vertex in S (which could very well be the one considered in the original AVP-RRT attempt). We can now prove proposition (1) for the modified AVP-RRT.

Proposition 2. *Consider a path \mathcal{P} in the configuration space. Assume that the function $s \mapsto \mathbf{u}(s)$, where $\mathbf{u}(s)$ is the unit tangent vector at $\mathcal{P}(s)$, is uniformly continuous. Then the sampling process of the modified AVP-RRT will approximate \mathcal{P} as closely as needed in both positions and unit tangent vectors.*

Proof. By the uniform continuity of the unit tangent vector function, there exists a δ such that if $\|\mathcal{P}(s) - \mathcal{P}(s')\| \leq \delta$ then $\|\mathbf{u}(s) - \mathbf{u}(s')\| \leq \Delta/3$. Now we chop the path \mathcal{P} into n sub-paths $\mathcal{P}_1, \dots, \mathcal{P}_n$ of length at most δ . Since for each sub-paths the unit tangent vector varies at most by $\Delta/3$, they look like straight segments if Δ is small enough, see Figure 3.11. \square

Next, we prove proposition (2).

Proposition 3. *Consider a path \mathcal{P} obtained by the sampling process. If there exists a profile v such that the time-parameterization of \mathcal{P} according to v yields a valid trajectory, then v is contained within the velocity band propagated by AVP.*

Proof. Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be the n interpolated paths resulting from the sampling process, which together form \mathcal{P} by concatenation. Let v_1, \dots, v_n be the corresponding subdivisions of v . We prove by induction on $i \in [0, n]$ that the concatenated profile $[v_1, \dots, v_i]$ is contained within the velocity band propagated by AVP.

For $i = 0$, i.e., at the start vertex, $v(0) = 0$ is contained within the initial velocity band, which is $[0, 0]$. Assume that the statement holds at i . This imply in particular that the final value of v_i , which is also the initial value of v_{i+1} , belongs to $[v_{\min}, v_{\max}]$, where (v_{\min}, v_{\max}) are the values returned by AVP at step i . Next, consider the velocity band that AVP propagates at step $i + 1$ from $[v_{\min}, v_{\max}]$. Since

$v_{i+1}(0) \in [v_{\min}, v_{\max}]$ and that v_{i+1} is continuous, the whole profile v_{i+1} will be contained in the velocity band propagated by AVP by construction. \square

Conclusion

By the end of this chapter, we have designed, theoretically proved, implemented and tested—on small systems—a general framework for motion planning of multi-DOF systems subject to kinodynamic constraints. To the best of our knowledge, **it is the first time that a kinodynamic planner in the configuration space can discover truly dynamic motions**, that is to say, motions that do not enforce static stability neither in their trajectories (edges of the roadmap) nor at junction postures (nodes of the roadmap).

Contact and friction

” *L’Académie voulant un travail qui puisse diriger dans le calcul des machines exige « que les lois du frottement et l’examen des effets résultans de la roideur des cordages soient déterminés d’après des expériences nouvelles et faites en grand [...] ». Je ne me flatte pas d’avoir rempli les vues aussi vastes qu’utiles de cette illustre compagnie mais je crois avoir fait quelques pas dans la carrière qu’elle a ouverte.*

— **Charles Augustin Coulomb**
(Théorie des machines simples, 1821)

Contact interactions are one of the greatest challenges for mobile mechanisms. The ability to make and break contacts with the environment grants mobility to humanoids, at the expense of a delicate control problem and risks of high damages on failures, that is to say, when falling. Understanding contact is therefore crucial to design control laws or motion plans for these robots.

4.1 Contact

From the perspective of Lagrangian mechanics, contacts are additional mechanical constraints between the robot and the environment. Kinematically, they can be described as points whose relative positions are fixed, such as the tip of a foot hooked to a point on the ground, or a flat foot surface immobile with respect to the floor. Dynamically, this immobility translates as constraint forces (called *contact forces* in this case) exchanged between the robot and the environment, and maintaining the balance in the equations of motion.

4.1.1 Contact modes

Kinematically, a contact binds the transform between a frame on an environment surface and a frame on one of the robot's links. In general, a kinematic transform has three linear and three angular degrees of freedom (DOF), accounting in total for six DOF. Contact amounts to constraining a certain subset of these DOFs by equalities, where each new constraint yields a dual generalized force of the same nature: binding a linear DOF yields a contact force, while binding an angular DOF yields a contact torque. A combination of kinematic contact constraints is called a *contact mode* (Balkcom and Trinkle, 2002). For instance, Figure 4.1 depicts the *point* (sphere-to-plane) and *full* (surface-to-surface) contact modes.

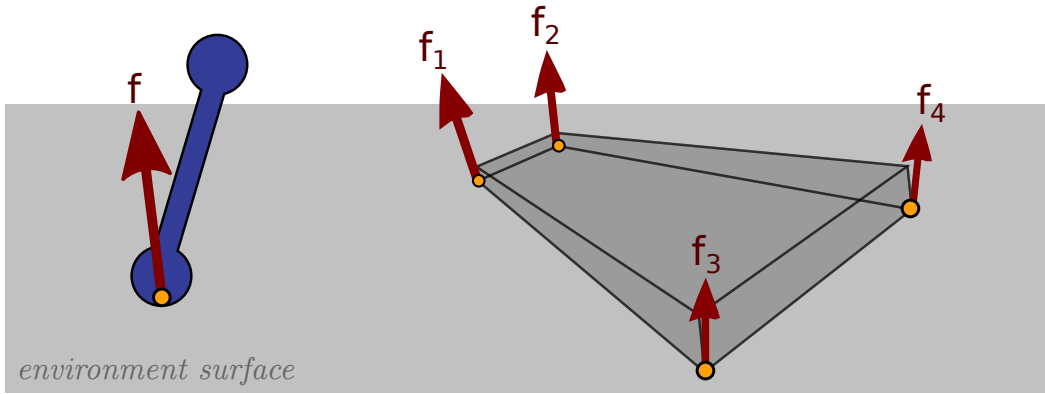


Fig. 4.1.: Two different contact modes. **Left:** a sphere-to-plane contact yields a point positional constraint: one point on the link coincides with one point on the surface. This constraint binds the three linear DOF between the link frame and the environment frame, resulting in a three-dimensional contact force. The three angular DOF are not constraint, thus there is no contact torque. **Right:** a surface-to-surface contact yields a full positional constraint: the link and environment frame coincide. All six relative DOF are constrained, resulting in a six-dimensional contact wrench (force and torque).

In what follows, we will use only two contact modes: fixed and separated, *i.e.*, either the robot makes full surface contact with the environment (*e.g.*, foot or hand contact), or it makes no contact at all and the end-effector can move freely. Moving in other contact modes such as sliding is possible, as demonstrated by humans, yet out of the scope of our present study.

4.1.2 Friction model

Switching between contact modes can occur depending on the forces exerted by or on the system. Consider the case of a block lay flat on a table: If an external operator

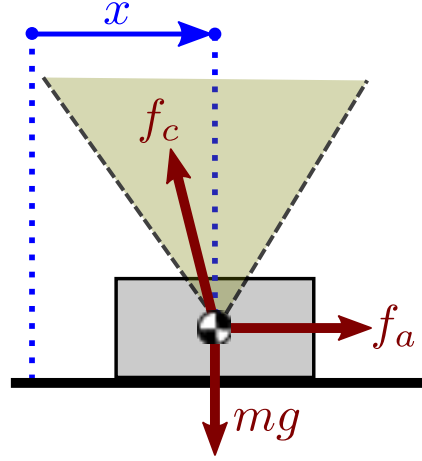


Fig. 4.2.: Simple case of a block on table pushed by an external force f_a .

applies a very small force f_a along the x -axis, this force will be compensated by friction and the block will not move. The system is in *fixed* contact mode. The limits of this mode are described by Coulomb's friction model: the fixed mode is achieved as long as the contact force $f^c (= mg - f_a$ in this example) lies inside the *friction cone* $|f_x^c| \leq \mu f_z^c$, where the static friction coefficient μ depends on the nature of the two surfaces in contact.

When f^c crosses the limits of this cone, the system switches to the *sliding* contact mode, in which $\ddot{x} > 0$ and the contact force obeys another frictional law $f_x^c = -\mu' f_z^c$, with μ' the kinetic friction coefficient.

Balkcom and Trinkle (2002) described how the Lagrangian duality between position constraints and forces is reflected in contact modes: to each acceleration variable corresponds a complementary force variable, and an equality constraint on one systematically implies inequality constraints on the other. In the example above, \ddot{x} and f_x^c are complementary variables: either $\ddot{x} = 0$ and $|f_x^c| \leq \mu f_z^c$ or $\ddot{x} > 0$ and $f_x^c = -\mu' f_z^c$.

Definition 3. A friction model is a continuous description of the constraints (equalities or inequalities) that apply to complementary variables in each contact mode.

The Coulomb model described above is a complete friction model for our 2D example. It generalizes to three dimensional bodies in contact. In what follows, we will only consider the fixed contact mode.

Definition 4 (Coulomb complementary condition). *Consider the point contact (C_i, \mathbf{n}_i) , where \mathbf{n}_i indicates the surface normal at point C_i . In the fixed contact mode, the contact force \mathbf{f}_i^c applied at C_i must lie inside the friction cone directed by \mathbf{n}_i :*

$$\begin{aligned}\mathbf{f}_i^c \cdot \mathbf{n}_i &> 0 \\ \|\mathbf{n}_i \times \mathbf{f}_i^c \times \mathbf{n}_i\|_2 &\leq \mu_i(\mathbf{f}_i^c \cdot \mathbf{n}_i),\end{aligned}$$

where μ_i is the static friction coefficient at contact C_i .

The Euclidean norm $\|\cdot\|_2$ in the definition above yields friction cones with circular sections. Although the underlying assumption of isotropy is more realistic from a physical point of view, linear approximations are usually considered in the literature as they make computations more tractable (Qiu et al., 2011; Hauser, 2014; Del Prete et al., 2015).

Definition 5 (Linearized Coulomb complementary condition). *Consider the point contact (C_i, \mathbf{n}_i) , where \mathbf{n}_i indicates the surface normal at point C_i . In the fixed contact mode, the contact force \mathbf{f}_i^c applied at C_i must lie inside the linearized friction cone directed by \mathbf{n}_i :*

$$\begin{aligned}\mathbf{f}_i^c \cdot \mathbf{n}_i &> 0 \\ |\mathbf{f}_i^c \cdot \mathbf{t}_i| &\leq \mu_i(\mathbf{f}_i^c \cdot \mathbf{n}_i), \\ |\mathbf{f}_i^c \cdot \mathbf{b}_i| &\leq \mu_i(\mathbf{f}_i^c \cdot \mathbf{n}_i),\end{aligned}$$

where μ_i is the static friction coefficient at contact C_i .

The approximation can be made conservative by taking the linearized cone inside the circular one, as depicted in Figure 4.3.

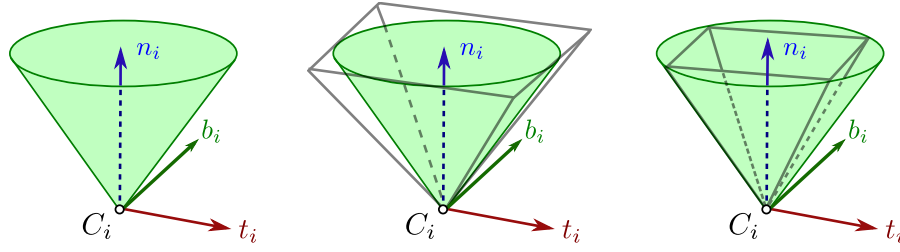


Fig. 4.3.: Friction cone at the contact point C_i with local contact frame (t_i, b_i, n_i) , represented respectively alone (left), with the outer (middle) and inner (right) linear approximations. (This figure is inspired from Fig. 1 in Trinkle et al. (1997).)

4.2 Cone duality

Linearized friction cones enable the use of a family of algorithms related to polyhedral convex cones. From now on, all the “cones” we mention will implicitly be polyhedral.¹

4.2.1 Weyl-Minkowski theorem

Let us recall a few fundamental definitions from computational geometry (Fukuda, 2000). The Minkowski sum of two sets P and Q is defined by

$$P + Q = \{p + q : p \in P \text{ and } q \in Q\}.$$

The convex hull of a set of points or vectors is $\text{conv}(v_1, \dots, v_n) = \sum_i \alpha_i v_i$ where $\forall i, \alpha_i > 0$ and $\sum_i \alpha_i = 1$. The positive span of a set of vectors is $\text{nonneg}(r_1, \dots, r_s) = \sum_i \lambda_i r_i$ where $\forall i, \lambda_i > 0$.

Theorem 3 (Minkowski-Weyl). *For a subset P of \mathbb{R}^d , the following statements are equivalent:*

1. *P is a polyhedron: $P = \{x : Ax \leq b\}$ for $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$*
2. *There are finite real vectors v_1, v_2, \dots, v_n and r_1, r_2, \dots, r_s in \mathbb{R}^d such that*

$$P = \text{conv}(v_1, v_2, \dots, v_n) + \text{nonneg}(r_1, r_2, \dots, r_s).$$

¹ A polyhedron is a geometric object with flat sides. Some authors use the word “polytope” to denote such objects in arbitrary dimensions, while reserving the word “polyhedron” for the three-dimensional case. A previous usage was to describe polyhedron as intersections of half-spaces and polytopes as convex combinations of vertices (both being equivalent in the light of the Weyl-Minkowski theorem). We make none of these distinctions in the present manuscript, and will use the word “polyhedron” in its most general sense.

Thus, every polyhedron has two representations of type (1) and (2), known as (half-space) H-representation and (vertex) V-representation, respectively. Polyhedral convex cones, such as linearized friction cones, are particular polyhedra. Note that all matrix inequalities are to be taken component-wise.

Definition 6. A polyhedral convex cone \mathcal{C} is defined by a set of inequalities $\mathcal{C} = \{\mathbf{x} : \mathbf{u}_1^\top \mathbf{x} \leq 0, \dots, \mathbf{u}_m^\top \mathbf{x} \leq 0\}$, for a family of vectors $(\mathbf{u}_1, \dots, \mathbf{u}_m)$. In matrix form, $\mathcal{C} = \{\mathbf{x} : \mathbf{U}\mathbf{x} \leq \mathbf{0}\}$.

Corollary 1. For every polyhedral convex cone \mathcal{C} , there exists a set of generators $(\mathbf{v}_1, \dots, \mathbf{v}_n)$ such that $\mathcal{C} = \{\mathbf{x} = \sum_{i=1}^n \lambda_i \mathbf{v}_i : \lambda_1 \geq 0 \dots \lambda_n \geq 0\}$. In matrix form, $\mathcal{C} = \{\mathbf{x} = \mathbf{V}\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq \mathbf{0}\}$.

Half-space and vertex representations have their own advantages. Checking whether a vector \mathbf{x} belongs to a cone is fast in H-representation (suffices to check each inequality), but it amounts to solve a Linear Program (LP) in V-representation. Meanwhile, the V-representation is compatible with linear combinations (suffices to apply the mapping to each generator), but more involved in H-representation, as we will see.

4.2.2 Double-description method

An important aspect of cone duality is the availability of an efficient algorithm to convert between the half-space and vertex representations: the *double-description method* (Fukuda and Prodon, 1996).

Given a matrix \mathbf{U} corresponding to an H-representation, we will denote by \mathbf{U}^V the associated vertex matrix, such that:

$$\forall \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{U}\mathbf{x} \leq \mathbf{0} \Leftrightarrow (\exists \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{x} = \mathbf{U}^V \boldsymbol{\lambda})$$

Similarly, given a matrix \mathbf{V} corresponding to a V-representation, we will denote by \mathbf{V}^H the associated half-space matrix, such that:

$$\forall \mathbf{x} \in \mathbb{R}^d, \quad (\exists \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{x} = \mathbf{V}\boldsymbol{\lambda}) \Leftrightarrow \mathbf{V}^H \mathbf{x} \leq \mathbf{0}$$

Both conversions \cdot^H and \cdot^V are performed by the double description method. In our implementation, we used the cdd library by Fukuda and Prodon (1996).

4.2.3 Polyhedron projection

Projection is one of the operations that will be important to reduce sets of contact forces. Suppose that we have a polyhedron \mathcal{P} defined in H-representation by $\mathcal{P} = \{\mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, where $\mathbf{x} = (x_1, x_2, \dots, x_r)$. The *projection* \mathcal{P}' of \mathcal{P} on the set of variables $\mathbf{x}' = (x_1, x_2, \dots, x_{r'})$, where $r' < r$, is the set

$$\mathcal{P}' = \left\{ \mathbf{x}' = (x_1, x_2, \dots, x_{r'}), \exists \mathbf{x}'' = (x_{r'+1}, \dots, x_r), \mathbf{A} \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \end{bmatrix} \leq \mathbf{b} \right\} \quad (4.1)$$

Informally, the two systems of linear inequalities \mathcal{P} and \mathcal{P}' have the same set of solutions over the variables $(x_1, \dots, x_{r'})$. The figure below illustrates the simplest example where a two-dimensional polygon is projected along the axes of a plane.

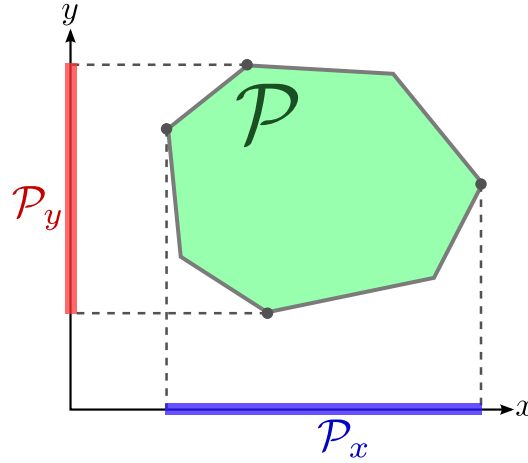


Fig. 4.4.: Projection of a 2D polygon by elimination of the x coordinate (red interval \mathcal{P}_y) or of the y coordinate (blue interval \mathcal{P}_x).

4.2.4 Fourier-Motzkin elimination

The Fourier-Motzkin algorithm² is a projection method that works exclusively in H-representation. Its purpose is to eliminate variables in systems of inequalities by a pivoting method similar to Gauss-Jordan elimination.

² Named after a first sketch of it given by Fourier (1827) and a more recent formalization by Motzkin (1952).

To compute the projection \mathcal{P}' from \mathcal{P} , the algorithm proceeds by successive elimination of individual variables. Let us consider the elimination of the variable x_r , and write $\alpha_i^\top x \leq b_i$ the inequalities of \mathcal{P} . The latter can be partitioned in three distinct sets:

- lower bounds $a_r x_r \geq b_i - \sum_{j < r} a_{ij} x_j$ with $a_r > 0$, which we will write equivalently $x_r \geq \underline{b}_j(x_1, \dots, x_{r-1})$ for $j \in \underline{B}$;
- upper bounds $a_r x_r \leq b_i - \sum_{j < r} a_{ij} x_j$ with $a_r > 0$, which we will write equivalently $x_r \leq \bar{b}_j(x_1, \dots, x_{r-1})$ for $j \in \bar{B}$;
- inequalities in which x_r does not appear ($a_r = 0$), which we will denote by $b_j(x_1, \dots, x_{r-1}) \geq 0$ for $j \in B$.

The polyhedron \mathcal{P} can then be written:

$$\begin{aligned} \forall (j_l, j_u) \in (\underline{B}, \bar{B}) \quad & \underline{b}_{j_l}(x_1, \dots, x_{r-1}) \leq x_r \leq \bar{b}_{j_u}(x_1, \dots, x_{r-1}) \\ \forall j \in B \quad & 0 \leq b_j(x_1, \dots, x_{r-1}) \end{aligned}$$

There exists a solution x_r to this system if and only if all upper bounds are greater than all lower bounds, and the projected polygon \mathcal{P}' can then be written:

$$\begin{aligned} \forall (j_l, j_u) \in (\underline{B}, \bar{B}) \quad & \underline{b}_{j_l}(x_1, \dots, x_{r-1}) \leq \bar{b}_{j_u}(x_1, \dots, x_{r-1}) \\ \forall j \in B \quad & 0 \leq b_j(x_1, \dots, x_{r-1}) \end{aligned}$$

We have thus derived an H-representation of the projected polygon. Note however that the elimination step replaced $|\underline{B}| + |\bar{B}|$ inequalities by $|\underline{B}| \times |\bar{B}|$ new ones, which points out to one of the main drawbacks of this method: memory consumption is doubly-exponential in the worst case.³

³ This case corresponds to $|B| = 0$ and $|\underline{B}| = |\bar{B}| = m/2$, where the number of inequalities representing the system after k eliminations is $O\left((m/4)^{2^k}\right)$.

4.3 Surface contacts

4.3.1 Continuous shear and pressure distributions

When the contact occurs via a surface and not through a set of points, the physical reality of contact is continuous. To account for this continuity, we model the action of the environment at the surface \mathcal{S} by two quantities: a scalar field $p(x, y)$ corresponding to normal *pressure*, and a two-dimensional vector field $\boldsymbol{\sigma}(x, y)$ for tangential mechanical *stress*. Figure 4.5-(A) illustrates these two fields for a rectangular contact area. For convenience, we also will denote by $\boldsymbol{\nu} := \boldsymbol{\sigma}(x, y) + p(x, y)\mathbf{n}$, where \mathbf{n} is the unit vector normal to the contact surface (pointing upward). The wrench resulting from $\boldsymbol{\nu}(x, y)$ is

$$\mathbf{f}^c := \int_{\mathcal{S}} \boldsymbol{\nu}(x, y) dx dy, \quad (4.2)$$

$$\boldsymbol{\tau}_O^c := \int_{\mathcal{S}} \overrightarrow{OC_{xy}} \times \boldsymbol{\nu}(x, y) dx dy, \quad (4.3)$$

where O is the origin of the link frame. Note how the resulting torque in Equation (4.3) involves infinitesimal forces $\boldsymbol{\nu}(x, y)$ but not infinitesimal torques (under bounded forces, torques vanish when application points draw infinitely closer). In other words, our model of surface contacts is a continuum of point contacts.

Accordingly, under Coulomb friction, the inequality constraints for $\boldsymbol{\nu}(x, y)$ are $p(x, y) > 0$ and $\|\boldsymbol{\sigma}(x, y)\| \leq \mu p(x, y)$. In the present literature, surface contact is often modeled using sets of contact points. In the light of a continuous model, the proposition below gives a theoretical justification for this practice.

Proposition 4. *Assume that the contact surface \mathcal{S} is a convex polygon with vertices C_1, \dots, C_N . If there exists a field $(x, y) \mapsto \boldsymbol{\nu}(x, y)$ satisfying unilaterality and non-slippage conditions, then there exists contact forces applied at C_1, \dots, C_N , summing up to the same contact wrench, and satisfying Coulomb complementary conditions (Definition 4).*

Proof. Consider pressure and stress fields summing up to \mathbf{w} . By convexity, one can find barycentric coordinates $\alpha_1(x, y), \dots, \alpha_k(x, y)$, i.e., positive functions such that

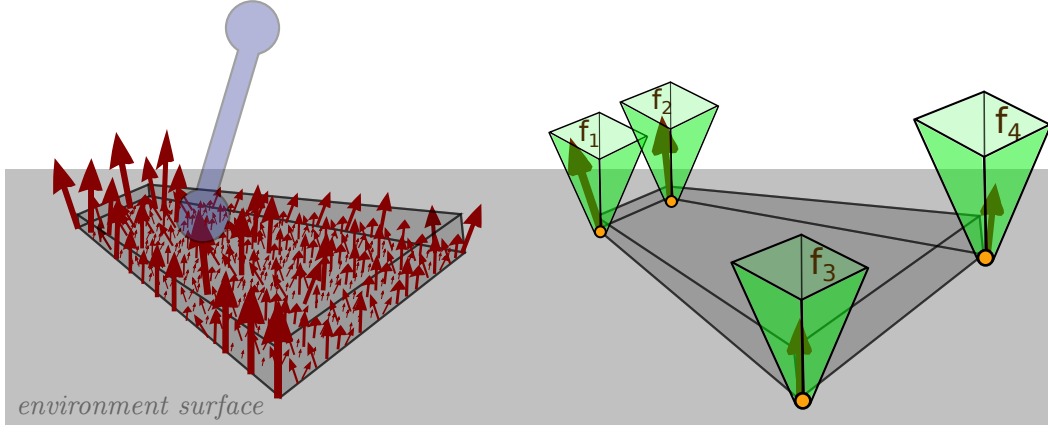


Fig. 4.5.: Contact in the surface plane. The physics of contact is a continuous distribution of stress and pressure fields (illustrated on the left). We show that, under Coulomb friction, this distribution is equivalent (in terms of the resulting wrench) to contact forces lying in frictions cones at the corners of the contact polygon (right).

$\sum_i \alpha_i(x, y) = 1$ and each point $C_{xy} \in \mathcal{S}$ can be written $C_{xy} = \sum_i \alpha_i(x, y) C_i$. Then, define for each vertex C_i a force

$$\mathbf{f}_i := \int_{\mathcal{S}} \alpha_i(x, y) \boldsymbol{\nu}(x, y) dx dy,$$

By positivity of the α_i 's, it is straightforward to check that all $f_i^n > 0$ and $\|\mathbf{f}_i^t\| \leq \mu f_i^n$. In addition, this expression of \mathbf{f}_i ensures that the resulting wrenches are equal. \square

If Dirac fields are authorized, then one can show that the converse implication of Proposition 4 is true. If not, whether this converse implication is true remains an open question. The bottom line of this argument is that the forces at the vertices of the convex hull completely render the dynamics of the surface contact. The wrench cone that we derive in the next section will share the same property.

4.3.2 Wrench cone for rectangular surfaces

While the wrench cone could be computed *numerically* for arbitrary contact surfaces using the double-description method, the *analytical* formula that we will derive using the Fourier-Motzkin-Imbert method gives even faster computations. Consider a rectangular area with corners C_1, C_2, C_3 and C_4 . We calculate the contact wrench \mathbf{w}_O^c at a reference point O in the link frame. Let us denote by (f_i^x, f_i^y, f_i^z) the three components of the contact force \mathbf{f}_i^c at the contact point C_i , expressed this time

in the link frame. The following proposition gives the analytical formulae for the wrench cone.

Proposition 5 (Contact Wrench Cone). *The contact wrench $w_O^c = (f^c, \tau_O)$ can be generated by contact forces (f_1^c, \dots, f_4^c) lying in their respective friction cones if and only if it lies in the Contact Wrench Cone (CWC) defined by:*

$$|f^x| \leq \mu f^z \quad (4.4)$$

$$|f^y| \leq \mu f^z \quad (4.5)$$

$$|\tau_O^x| \leq Y f^z \quad (4.6)$$

$$|\tau_O^y| \leq X f^z \quad (4.7)$$

$$\tau_{\min}^z \leq \tau_O^z \leq \tau_{\max}^z \quad (4.8)$$

where τ_{\min}^z and τ_{\max}^z are defined by:

$$\tau_{\min}^z := -\mu(X + Y)f^z + |Y f^x - \mu \tau_O^x| + |X f^y - \mu \tau_O^y|,$$

$$\tau_{\max}^z := +\mu(X + Y)f^z - |Y f^x + \mu \tau_O^x| - |X f^y + \mu \tau_O^y|.$$

Furthermore, (4.8) does not contain any implicit constraint on f^x, f^y, f^z, τ_O^x or τ_O^y , as (4.4)–(4.7) $\Rightarrow \tau_{\min}^z \leq \tau_{\max}^z$.

Proof. The complete equality-inequality system is:

$$f^x = f_1^x + f_2^x + f_3^x + f_4^x \quad (4.9)$$

$$f^y = f_1^y + f_2^y + f_3^y + f_4^y \quad (4.10)$$

$$f^z = f_1^z + f_2^z + f_3^z + f_4^z \quad (4.11)$$

$$\tau_O^x = Y(f_1^z - f_2^z - f_3^z + f_4^z) \quad (4.12)$$

$$\tau_O^y = -X(f_1^z + f_2^z - f_3^z - f_4^z) \quad (4.13)$$

$$\tau_O^z = X(f_1^y + f_2^y - f_3^y - f_4^y) - Y(f_1^x - f_2^x - f_3^x + f_4^x) \quad (4.14)$$

$$\forall i, |f_i^x| \leq \mu f_i^z \quad (4.15)$$

$$\forall i, |f_i^y| \leq \mu f_i^z \quad (4.16)$$

$$\forall i, f_i^z > 0 \quad (4.17)$$

The positivity of all $f_i^z > 0$ implies that $f^z > 0$ by positive combination, so we can normalize the resultant force into:

$$K_1 := \frac{f^x}{\mu f^z}, \quad K_2 := \frac{f^y}{\mu f^z}, \quad K_3 := \frac{\tau_O^z}{\mu(X+Y)f^z}.$$

And similarly for the resultant moment:

$$C_1 := \frac{\tau_O^x}{Y f^z}, \quad C_2 := \frac{\tau_O^y}{X f^z}.$$

Contact force variables are themselves normalized into:

$$\alpha_i^x := \frac{f_i^x}{\mu f_i^z}, \quad \alpha_i^y := \frac{f_i^y}{\mu f_i^z}, \quad \beta_i^z := \frac{f_i^z}{f_1^z + f_2^z + f_3^z + f_4^z}.$$

Dividing each row of the initial system by f^z , we get:

$$1 = \beta_1^z + \beta_2^z + \beta_3^z + \beta_4^z \quad (4.18)$$

$$C_1 = \beta_1^z - \beta_2^z - \beta_3^z + \beta_4^z \quad (4.19)$$

$$C_2 = -\beta_1^z - \beta_2^z + \beta_3^z + \beta_4^z \quad (4.20)$$

$$K_1 = \alpha_1^x + \alpha_2^x + \alpha_3^x + \alpha_4^x \quad (4.21)$$

$$K_2 = \alpha_1^y + \alpha_2^y + \alpha_3^y + \alpha_4^y \quad (4.22)$$

$$K_3 = p_x(\alpha_1^y \beta_1^z + \alpha_2^y \beta_2^z - \alpha_3^y \beta_3^z - \alpha_4^y \beta_4^z) \\ - p_y(\alpha_1^x \beta_1^z - \alpha_2^x \beta_2^z - \alpha_3^x \beta_3^z + \alpha_4^x \beta_4^z) \quad (4.23)$$

$$\forall i, |\alpha_i^x| \leq 1 \quad (4.24)$$

$$\forall i, |\alpha_i^y| \leq 1 \quad (4.25)$$

$$\forall i, \beta_i^z > 0 \quad (4.26)$$

where $p_x := \frac{X}{X+Y}$ and $p_y := \frac{Y}{X+Y}$.

Rewriting of α_i 's. Let us introduce the new variables:

$$\gamma_x = \alpha_1^x \beta_1^z + \alpha_4^x \beta_4^z, \quad \gamma'_x = \alpha_2^x \beta_2^z + \alpha_3^x \beta_3^z, \\ \gamma_y = \alpha_1^y \beta_1^z + \alpha_2^y \beta_2^z, \quad \gamma'_y = \alpha_3^y \beta_3^z + \alpha_4^y \beta_4^z.$$

Note that the vector β is a convex combination, i.e., $\beta_i^z > 0$ and $\sum_i \beta_i^z = 1$. As a consequence, the mapping $M(\beta)$ from α to γ is a linear surjection from $[-1, 1]^8$

to $S_\gamma = \{|\gamma_i^{x|y}| \leq \beta_j^z + \beta_k^z\}$, and $\alpha \in [-1, 1]^8$ if and only if $\gamma \in S_\gamma$. Using Equations (4.18)-(4.20), we can rewrite S_γ as:

$$\begin{aligned} 2|\gamma_x| &\leq 1 + C_1, & 2|\gamma'_x| &\leq 1 - C_1, \\ 2|\gamma_y| &\leq 1 - C_2, & 2|\gamma'_y| &\leq 1 + C_2. \end{aligned}$$

Elimination of β_i 's. Next, we reduce Equations (4.18)-(4.20) by rewriting β_1^z, β_2^z and β_3^z as functions of β_4^z . The positivity inequalities $\forall i, \beta_i^z \geq 0$ thus:

$$\begin{aligned} -1 + C_1 &\leq 2\beta_4^z \leq 1 + C_1 \\ C_1 + C_2 &\leq 2\beta_4^z \leq 2 + C_1 + C_2 \\ -1 + C_2 &\leq 2\beta_4^z \leq 1 + C_2 \\ 0 &\leq 2\beta_4^z \leq 2 \end{aligned}$$

The complete system is now:

$$K_1 = \gamma_x + \gamma'_x \tag{4.27}$$

$$K_2 = \gamma_y + \gamma'_y \tag{4.28}$$

$$K_3 = p_x(\gamma_y - \gamma'_y) - p_y(\gamma_x - \gamma'_x) \tag{4.29}$$

$$2|\gamma_x| \leq 1 + C_1 \tag{4.30}$$

$$2|\gamma'_x| \leq 1 - C_1 \tag{4.31}$$

$$2|\gamma_y| \leq 1 - C_2 \tag{4.32}$$

$$2|\gamma'_y| \leq 1 + C_2 \tag{4.33}$$

$$2\beta_4^z \leq 1 + C_1 \tag{4.34}$$

$$2\beta_4^z \leq 2 + C_1 + C_2 \tag{4.35}$$

$$2\beta_4^z \leq 1 + C_2 \tag{4.36}$$

$$2\beta_4^z \leq 2 \tag{4.37}$$

$$2\beta_4^z \geq -1 + C_1 \tag{4.38}$$

$$2\beta_4^z \geq C_1 + C_2 \tag{4.39}$$

$$2\beta_4^z \geq -1 + C_2 \tag{4.40}$$

$$2\beta_4^z \geq 0 \tag{4.41}$$

We apply Fourier-Motzkin elimination to β_4^z : matching all pairs of lower and upper bounds for this variable yields $C_1 \in [-1, 1]$ and $C_2 \in [-1, 1]$, so that the complete system is:

$$K_1 = \gamma_x + \gamma'_x \quad (4.42)$$

$$K_2 = \gamma_y + \gamma'_y \quad (4.43)$$

$$K_3 = p_x(\gamma_y - \gamma'_y) - p_y(\gamma_x - \gamma'_x) \quad (4.44)$$

$$2|\gamma_x| \leq 1 + C_1 \quad (4.45)$$

$$2|\gamma'_x| \leq 1 - C_1 \quad (4.46)$$

$$2|\gamma_y| \leq 1 - C_2 \quad (4.47)$$

$$2|\gamma'_y| \leq 1 + C_2 \quad (4.48)$$

$$C_1 \in [-1, 1] \quad (4.49)$$

$$C_2 \in [-1, 1] \quad (4.50)$$

Elimination of γ 's. We follow the same procedure as above, first expressing all γ 's as functions of γ_x , which yields:

$$2p_y\gamma_x \leq p_y(1 + C_1) \quad (4.51)$$

$$2p_y\gamma_x \leq p_y(1 - C_1) + 2p_yK_1 \quad (4.52)$$

$$2p_y\gamma_x \leq p_x(1 - C_2) - K_3 + p_yK_1 - p_xK_2 \quad (4.53)$$

$$2p_y\gamma_x \leq p_x(1 + C_2) - K_3 + p_yK_1 + p_xK_2 \quad (4.54)$$

$$2p_y\gamma_x \geq -p_y(1 + C_1) \quad (4.55)$$

$$2p_y\gamma_x \geq -p_y(1 - C_1) + 2p_yK_1 \quad (4.56)$$

$$2p_y\gamma_x \geq -p_x(1 - C_2) - K_3 + p_yK_1 - p_xK_2 \quad (4.57)$$

$$2p_y\gamma_x \geq -p_x(1 + C_2) - K_3 + p_yK_1 + p_xK_2 \quad (4.58)$$

Applying Fourier-Motzkin elimination to γ_x , one can check that matching all pairs of lower bounds (4.55)-(4.58) and upper bounds (4.51)-(4.54) yields:

- (4.55) \leq (4.51) $\Leftrightarrow C_1 \geq -1$
- (4.55) \leq (4.52) $\Leftrightarrow K_1 \geq -1$
- (4.55) \leq (4.53) $\Leftrightarrow K_3 - p_yK_1 + p_xK_2 - p_yC_1 + p_xC_2 \leq 1$
- (4.55) \leq (4.54) $\Leftrightarrow K_3 - p_yK_1 - p_xK_2 - p_yC_1 - p_xC_2 \leq 1$

- $(4.56) \leq (4.51) \Leftrightarrow K_1 \leq 1$
- $(4.56) \leq (4.52) \Leftrightarrow C_1 \leq 1$
- $(4.56) \leq (4.53) \Leftrightarrow K_3 + p_y K_1 + p_x K_2 + p_y C_1 + p_x C_2 \leq 1$
- $(4.56) \leq (4.54) \Leftrightarrow K_3 + p_y K_1 - p_x K_2 + p_y C_1 - p_x C_2 \leq 1$
- $(4.57) \leq (4.51) \Leftrightarrow -K_3 + p_y K_1 - p_x K_2 - p_y C_1 + p_x C_2 \leq 1$
- $(4.57) \leq (4.52) \Leftrightarrow -K_3 - p_y K_1 - p_x K_2 + p_y C_1 + p_x C_2 \leq 1$
- $(4.57) \leq (4.53) \Leftrightarrow C_2 \leq 1$
- $(4.57) \leq (4.54) \Leftrightarrow K_2 \geq -1$
- $(4.58) \leq (4.51) \Leftrightarrow -K_3 + p_y K_1 + p_x K_2 - p_y C_1 - p_x C_2 \leq 1$
- $(4.58) \leq (4.52) \Leftrightarrow -K_3 - p_y K_1 + p_x K_2 + p_y C_1 - p_x C_2 \leq 1$
- $(4.58) \leq (4.53) \Leftrightarrow K_2 \leq 1$
- $(4.58) \leq (4.54) \Leftrightarrow C_2 \geq -1$

Consequently, the complete system is now:

$$\begin{aligned}
K_3 &\leq 1 - p_y K_1 - p_x K_2 - p_y C_1 - p_x C_2 \\
K_3 &\leq 1 - p_y K_1 + p_x K_2 - p_y C_1 + p_x C_2 \\
K_3 &\leq 1 + p_y K_1 - p_x K_2 + p_y C_1 - p_x C_2 \\
K_3 &\leq 1 + p_y K_1 + p_x K_2 + p_y C_1 + p_x C_2 \\
K_3 &\geq -1 + p_y K_1 + p_x K_2 - p_y C_1 - p_x C_2 \\
K_3 &\geq -1 + p_y K_1 - p_x K_2 - p_y C_1 + p_x C_2 \\
K_3 &\geq -1 - p_y K_1 + p_x K_2 + p_y C_1 - p_x C_2 \\
K_3 &\geq -1 - p_y K_1 - p_x K_2 + p_y C_1 + p_x C_2 \\
C_1 &\in [-1, 1] \\
C_2 &\in [-1, 1] \\
K_1 &\in [-1, 1] \\
C_2 &\in [-1, 1]
\end{aligned}$$

In a more concise form, these inequalities can be written

$$\begin{aligned}
K_3 &\geq -1 + p_y |K_1 - C_1| + p_x |K_2 - C_2|, \\
K_3 &\leq +1 - p_y |K_1 + C_1| - p_x |K_2 + C_2|.
\end{aligned}$$

The inequalities (4.4)–(4.8) follow from de-normalizing all inequalities. The last claim on the absence of implicit constraint in (4.8) can be seen by applying Fourier-Motzkin elimination to K_3 :

$$\begin{aligned} p_y(|K_1 - C_1| + |K_1 + C_1|) + p_x(|K_2 - C_2| + |K_2 + C_2|) &\leq 2 \\ p_y \max(|K_1|, |C_1|) + p_x \max(|K_2|, |C_2|) &\leq 1, \end{aligned}$$

which is always true since $(p_x, p_y) \in [-1, 1]^2$. \square

Let us now detail the structure of the Contact Wrench Cone.

- The first two inequalities (4.4)–(4.5) correspond to the Coulomb friction model applied to the resultant contact force. (Note how 4.6 and 4.7 naturally imply that $f^z > 0$.)
- Inequalities (4.6) and (4.7) are equivalent to having the Center Of Pressure (COP) lie inside of the support polygon.
- The last inequality (4.8) provides a bound on the resultant yaw torque, which was implicitly encoded in the contact-force model.

Observe how the latter relation is more complex than “no rotation occurs while τ_O^z is small enough”, as it is coupled with all other components of the contact wrench. The yaw torque is bounded by τ_{\min}^z and τ_{\max}^z , both of which may be either positive or negative (for instance, both become negative when the COP nears the corner $\tau_O^x = Y f^z$, $\tau_O^y = X f^z$ of the support polygon). Notably, the “safest” value for τ_O^z is not zero but:

$$\begin{aligned} \tau_{\text{safe}} &:= \frac{\tau_{\min}^z + \tau_{\max}^z}{2} \\ &= \text{sgn}(-f^x \tau_O^x) \min(Y|f^x|, \mu|\tau_O^x|) + \text{sgn}(-f^y \tau_O^y) \min(X|f^y|, \mu|\tau_O^y|), \end{aligned}$$

with sgn the sign function. From Equation (4.8), τ_O^z may deviate from τ_{safe} by at most

$$\mu(X + Y)f^z - \max(Y|f^x|, \mu|\tau_O^x|) - \max(X|f^y|, \mu|\tau_O^y|).$$

We see from this rewriting that higher tangential forces or roll-pitch torques reduce the range of admissible yaw torques. In particular, when these other constraints are

saturated (e.g., when the COP reaches a corner of the support polygon), τ_{safe} is the *only solution* that prevents the contact from breaking. Therefore, $\tau_O^z = \tau_{\text{safe}}$ appears as a sensible control law to prevent undesired yaw rotations.

4.3.3 Yaw moment at the center of pressure

Additional insights appear when considering the yaw constraint (4.8) at the COP rather than the center O of the rectangular surface. The plane coordinates of the center of pressure C are given by

$$x_C = \frac{-\tau_O^y}{f^z}, \quad y_C = \frac{\tau_O^x}{f^z}.$$

The yaw moment at the COP is $\tau_C^z = \tau_O^z - x_C f^y + y_C f^x$ so that

$$\tau_O^z = \tau_C^z - \frac{\tau_O^y}{f^z} f^y - \frac{\tau_O^x}{f^z} f^x.$$

The inequality $\tau_O^z \geq \tau_{\min}^z$ becomes $\tau_C^z \geq m_{\min}^z$ with

$$\begin{aligned} m_{\min}^z &= \left(-\mu X f^z + |X f^y - \mu \tau_O^y| + \frac{\tau_O^y}{f^z} f^y \right) + \left(-\mu Y f^z + |Y f^x - \mu \tau_O^x| + \frac{\tau_O^x}{f^z} f^x \right) \\ &= \max \begin{Bmatrix} (+X - x_C)(f^y - \mu f^z) \\ (-X - x_C)(f^y + \mu f^z) \end{Bmatrix} + \max \begin{Bmatrix} -(-Y - y_C)(f^x - \mu f^z) \\ -(+Y - y_C)(f^x + \mu f^z) \end{Bmatrix} \\ &= -\mu f^z (X + Y - |x_C| - |y_C|) + \max \begin{Bmatrix} (+X - x_C)f^y - (+Y - y_C)f^x \\ (+X - x_C)f^y - (-Y - y_C)f^x \\ (-X - x_C)f^y - (+Y - y_C)f^x \\ (-X - x_C)f^y - (-Y - y_C)f^x \end{Bmatrix} \\ &= -\mu f^z (X + Y - |x_C| - |y_C|) + \max_{i=1}^4 \overrightarrow{CC_i} \times \mathbf{f}_t \end{aligned}$$

Similarly, $\tau_O^z \leq \tau_{\max}^z$ becomes $\tau_C^z \leq m_{\max}^z$ with

$$\begin{aligned} m_{\max}^z &= \max \begin{Bmatrix} (+X - x_C)(\mu f^z + f^y) \\ (-X - x_C)(f^y - \mu f^z) \end{Bmatrix} + \max \begin{Bmatrix} -(+Y - y_C)(f^x - \mu f^z) \\ -(-Y - y_C)(f^x + \mu f^z) \end{Bmatrix} \\ &= \mu f^z (X + Y - |x_C| - |y_C|) + \max \begin{Bmatrix} (+X - x_C)f^y - (+Y - y_C)f^x \\ (+X - x_C)f^y - (-Y - y_C)f^x \\ (-X - x_C)f^y - (+Y - y_C)f^x \\ (-X - x_C)f^y - (-Y - y_C)f^x \end{Bmatrix} \end{aligned}$$

$$= \mu f^z (X + Y - |x_C| - |y_C|) + \max_{i=1}^4 \overrightarrow{CC_i} \times \mathbf{f}_t$$

It appears from these two equations that the safest value for the moment exerted by contact forces at the COP is

$$\tau_{C,\text{safe}}^z = \frac{m_{\min}^z + m_{\max}^z}{2} = \max_{i=1}^4 \overrightarrow{CC_i} \times \mathbf{f}_t,$$

i.e., the moment exerted at the COP by applying the resultant friction force stress at the vertices of the contact polygon. It is yet unclear to us how this simple expression, which was naturally embedded in the expression of the wrench cone, could be derived from physical considerations.

$$\tau_{C,\text{safe}}^z = |X f^y - \mu x_C f^z| + |Y f^x - \mu y_C f^z|.$$

Meanwhile, the vertical moment τ_C^z at the COP may deviate from the reference value $\tau_{C,\text{safe}}^z$ by at most

$$\mu f^z d_{\text{vert}}(x_C, y_C),$$

where $d_{\text{vert}}(x_C, y_C) := (X + Y - |x_C| - |y_C|)$ is the (norm-1) distance⁴ from the COP to the nearest vertex of the support area. It is indeed sensible that a higher friction coefficient, more contact pressure or a COP closer to the center of the support area will reduce the chances of yaw rotations. Within our analysis, we have identified that they are the *only* factors to be accounted for under Coulomb friction.

4.4 Application to single-support contact phases

In this section, we consider the case of a humanoid making a single foot contact with a horizontal floor.

4.4.1 Expression of the CWC as ZMP support polygon

All coordinates being expressed within the contacting link's reference frame, the z -axis of which is vertical, we denote by $\mathbf{p}_G = (x_G, y_G, z_G)$ and $\mathbf{p}_Z = (x_Z, y_Z, 0)$ the coordinates of the robot's Center Of Mass (COM) G and ZMP Z . The ZMP and

⁴ Remember that we have linearized friction cones, which corresponds to an exchange of the ℓ^2 (Euclidean) for the ℓ^1 norm.

COP coincide when all contacts occur with the same contact surface (Sardain and Bessonnet, 2004), so that the formulae derived in the previous section apply to Z . Following common practice, we use the ZMP as a whole-body quantity and centers of pressure as local contact quantities, although both refer to the same in single contact.

In dynamic equilibrium, the resultant of contact forces “balances” (is equal to the opposite of) the resultant of gravity and inertial forces (a proper introduction to this law will be given in Chapter 5). The resultant of contact forces is then

$$f^x = -m\ddot{x}_G \quad (4.59)$$

$$f^y = -m\ddot{y}_G \quad (4.60)$$

$$f^z = m(g - \ddot{z}_G), \quad (4.61)$$

with m the total mass of the robot and $g > 0$ the gravity constant. The Contact Wrench Cone can be rewritten as:

$$|\ddot{x}_G| \leq \mu(\ddot{z}_G + g) \quad (4.62)$$

$$|\ddot{y}_G| \leq \mu(\ddot{z}_G + g) \quad (4.63)$$

$$|x_Z| \leq X \quad (4.64)$$

$$|y_Z| \leq Y \quad (4.65)$$

$$\tau_{\min}^z \leq \tau_O^z \leq \tau_{\max}^z \quad (4.66)$$

We make the assumption that there is no angular momentum acting on the system—or, equivalently, that the angular momentum is regulated to zero—which corresponds to the Linear Pendulum assumption (a more detailed discussion of this point will be given in Chapter 6). The ZMP is defined by the following equations (see *e.g.* Sugihara et al., 2002):

$$\ddot{x}_G = (x_G - x_Z) \frac{\ddot{z}_G + g}{z_G} \quad (4.67)$$

$$\ddot{y}_G = (y_G - y_Z) \frac{\ddot{z}_G + g}{z_G} \quad (4.68)$$

Using Equations (4.67)-(4.68), one can rewrite the non-slippage conditions (4.62)-(4.63) as:

$$\begin{aligned} |x_G - x_Z| &\leq \mu z_G \\ |y_G - y_Z| &\leq \mu z_G \\ \ddot{z}_G &> -g \end{aligned}$$

Next, consider the yaw inequality. The vertical moment at the ZMP is the same as that at the COM, as the assumption that $\dot{\mathbf{L}}_G^x = \dot{\mathbf{L}}_G^y = 0$ amounts to say that the COM is on the non-central axis of the ZMP (Sardain and Bessonnet, 2004):

$$\begin{aligned} \tau_Z^z &= \tau_G^z + (\overrightarrow{ZG} \times \mathbf{f}) \cdot \mathbf{e}_Z \\ &= 0 + (x_G - x_Z)f^y - (y_G - y_Z)f^x \\ &= -(x_G - x_Z)m\ddot{y}_G + (y_G - y_Z)m\ddot{x}_G \\ &= 0. \end{aligned}$$

The condition (4.8), which is equivalent to $m_{\min}^z \leq \tau_Z^z \leq m_{\max}^z$, can now be expressed as $m_{\min}^z \leq 0 \leq m_{\max}^z$, that is

$$\max_{i=1}^4 |\overrightarrow{ZC}_i \times \mathbf{f}_t| \leq \mu f^z d_{\text{vert}}.$$

From (4.59)-(4.61), this expands to:

$$\begin{aligned} \left| \begin{pmatrix} +X - x_Z \\ +Y - y_Z \end{pmatrix} \times \begin{pmatrix} \ddot{x}_G \\ \ddot{y}_G \end{pmatrix} \right| &\leq \mu(\ddot{z}_G + g)d_{\text{vert}} \\ \left| \begin{pmatrix} +X - x_Z \\ -Y - y_Z \end{pmatrix} \times \begin{pmatrix} \ddot{x}_G \\ \ddot{y}_G \end{pmatrix} \right| &\leq \mu(\ddot{z}_G + g)d_{\text{vert}} \\ \left| \begin{pmatrix} -X - x_Z \\ +Y - y_Z \end{pmatrix} \times \begin{pmatrix} \ddot{x}_G \\ \ddot{y}_G \end{pmatrix} \right| &\leq \mu(\ddot{z}_G + g)d_{\text{vert}} \\ \left| \begin{pmatrix} -X - x_Z \\ -Y - y_Z \end{pmatrix} \times \begin{pmatrix} \ddot{x}_G \\ \ddot{y}_G \end{pmatrix} \right| &\leq \mu(\ddot{z}_G + g)d_{\text{vert}}. \end{aligned}$$

Which is then, from (4.67)-(4.68):

$$\left| \begin{pmatrix} +X - x_Z \\ +Y - y_Z \end{pmatrix} \times \begin{pmatrix} x_G - x_Z \\ y_G - y_Z \end{pmatrix} \right| \leq \mu z_G d_{\text{vert}}$$

$$\begin{aligned}
\left| \begin{pmatrix} +X - x_Z \\ -Y - y_Z \end{pmatrix} \times \begin{pmatrix} x_G - x_Z \\ y_G - y_Z \end{pmatrix} \right| &\leq \mu z_G d_{\text{vert}} \\
\left| \begin{pmatrix} -X - x_Z \\ +Y - y_Z \end{pmatrix} \times \begin{pmatrix} x_G - x_Z \\ y_G - y_Z \end{pmatrix} \right| &\leq \mu z_G d_{\text{vert}} \\
\left| \begin{pmatrix} -X - x_Z \\ -Y - y_Z \end{pmatrix} \times \begin{pmatrix} x_G - x_Z \\ y_G - y_Z \end{pmatrix} \right| &\leq \mu z_G d_{\text{vert}}
\end{aligned}$$

Expanding all absolute values, the complete H-representation of the ZMP support polygon can finally be calculated (Figure 4.6). In a concise (hypercube) form, this system can also be written:

$$\zeta_1(\zeta_3 Y - y_G)x_Z - \zeta_1(\zeta_2 X - x_G)y_Z + \zeta_1(\zeta_2 X y_G - \zeta_3 Y x_G) \leq \mu z_G (X + Y - |x_Z| - |y_Z|)$$

where $(\zeta_1, \zeta_2, \zeta_3) \in \{-1, +1\}^3$. This polygon takes into account all of the wrench cone constraints *except* unilaterality of contact, *i.e.*, $f^z > 0 \Leftrightarrow \ddot{z}_G \geq -g$, which ought to be checked separately if needed. In ZMP controllers, the altitude z_G of the COM is usually assumed to be constant, so that the system is regulated around the zero-dynamics of $\ddot{z}_G = 0$.

Figure 4.7 shows the ZMP polygon computed when the COM height is $z_G = 0.8$ m and the friction coefficient $\mu = 0.1$ is low (this value corresponds *e.g.*, to metal-on-ice contacts). We see that the actual support polygon for the ZMP to keep a stable contact is not the complete contact surface, but a small region extending from the COM to the foot center, which shows that:

Property 2. *The support area S of the ZMP is **not** the Convex Hull of Contact Points (CHCP). Although it is true that $S \subset \text{CHCP}$, the inclusion becomes strict for *e.g.*, low COM altitudes z_G or at small friction μ .*

We will discuss this phenomenon in more detail in Chapter 6 (in particular in Section 6.4.2).

	+	x_Z		\leq		X
	-	x_Z		\leq		X
	+	x_Z		\leq		$\mu z_G + x_G$
	-	x_Z		\leq		$\mu z_G - x_G$
			+	y_Z	\leq	Y
			-	y_Z	\leq	Y
			+	y_Z	\leq	$\mu z_G + y_G$
			-	y_Z	\leq	$\mu z_G - y_G$
(+Y - y_G + 1)	x_Z	-(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G - Y x_G)$	
(+Y - y_G + 1)	x_Z	-(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G - Y x_G)$	
(-Y - y_G + 1)	x_Z	-(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G + Y x_G)$	
(-Y - y_G + 1)	x_Z	-(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G + Y x_G)$	
(+Y - y_G - 1)	x_Z	-(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G - Y x_G)$	
(+Y - y_G - 1)	x_Z	-(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G - Y x_G)$	
(-Y - y_G - 1)	x_Z	-(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G + Y x_G)$	
(-Y - y_G - 1)	x_Z	-(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G + Y x_G)$	
(+Y - y_G + 1)	x_Z	-(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G - Y x_G)$	
(+Y - y_G + 1)	x_Z	-(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G - Y x_G)$	
(-Y - y_G + 1)	x_Z	-(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G + Y x_G)$	
(-Y - y_G + 1)	x_Z	-(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G + Y x_G)$	
(+Y - y_G - 1)	x_Z	-(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G - Y x_G)$	
(+Y - y_G - 1)	x_Z	-(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G - Y x_G)$	
(-Y - y_G - 1)	x_Z	-(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (+X y_G + Y x_G)$	
(-Y - y_G - 1)	x_Z	-(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) - (-X y_G + Y x_G)$	
-(+Y - y_G + 1)	x_Z	(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G - Y x_G)$	
-(+Y - y_G + 1)	x_Z	(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G - Y x_G)$	
-(-Y - y_G + 1)	x_Z	(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G + Y x_G)$	
-(-Y - y_G + 1)	x_Z	(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G + Y x_G)$	
-(+Y - y_G - 1)	x_Z	(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G - Y x_G)$	
-(+Y - y_G - 1)	x_Z	(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G - Y x_G)$	
-(-Y - y_G - 1)	x_Z	(+X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G + Y x_G)$	
-(-Y - y_G - 1)	x_Z	(-X - x_G + 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G + Y x_G)$	
-(+Y - y_G + 1)	x_Z	(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G - Y x_G)$	
-(+Y - y_G + 1)	x_Z	(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G - Y x_G)$	
-(-Y - y_G + 1)	x_Z	(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G + Y x_G)$	
-(-Y - y_G + 1)	x_Z	(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G + Y x_G)$	
-(+Y - y_G - 1)	x_Z	(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G - Y x_G)$	
-(+Y - y_G - 1)	x_Z	(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G - Y x_G)$	
-(-Y - y_G - 1)	x_Z	(+X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (+X y_G + Y x_G)$	
-(-Y - y_G - 1)	x_Z	(-X - x_G - 1)	y_Z	\leq	$\mu z_G(X + Y) + (-X y_G + Y x_G)$	

Fig. 4.6.: Complete expression of the ZMP support polygon, accounting for the complete contact stability condition (i.e., taking into account not only roll and pitch, as is usually done, but also slippage and yaw rotations). Most of these inequalities are redundant and can be eliminated once the position of the COM is determined. However, in the general case any of them can be saturated.

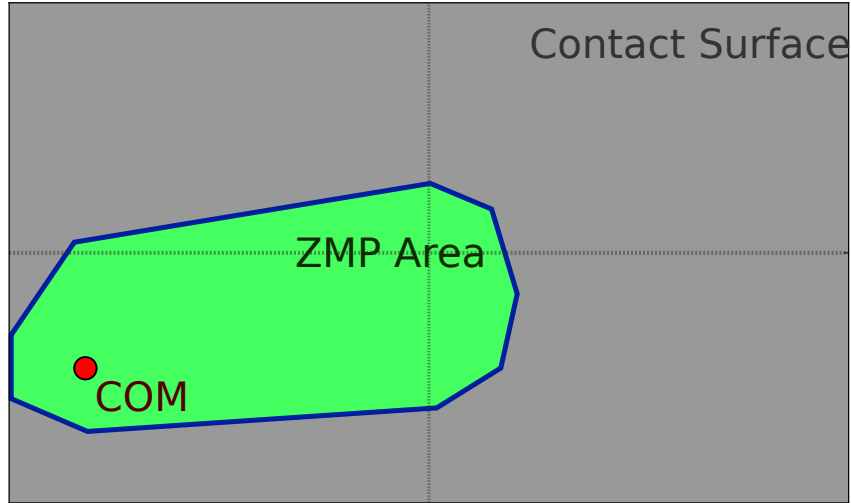



Fig. 4.7.: ZMP support area (in green) for the complete contact stability condition, *i.e.*, taking into account translational and rotational friction. The foot dimensions are that of HRP-4. The area was computed for a static friction coefficient $\mu = 0.1$ (icy surface) and a COM height of 0.8 m. The polygon has eight edges; the other thirty-two inequalities of the complete system (Figure 4.6) could be eliminated once the position of the COM (red disc) was determined.

4.4.2 Experimental validation

We tested the validity of the CWC as a stability conditions via dynamics simulations in OpenHRP⁵ with our model of the HRP4 humanoid robot (Kaneko et al., 2011). Note that OpenHRP performs its own physics simulations for contact, as described in (Kanehiro et al., 2004), *i.e.*, it has its own model for simulating what contact forces the environment will choose inside friction cones. Our criterion is independent from this choice, as stability is enforced at the cone level rather than relying on individual contact forces.


We use TOPP (see Chapter 3) to enforce the CWC condition. In this experiment, we consider a single contact made by the left foot and design a motion that challenges all six contact DOFs. Remember that the contact wrench is fully determined by the unactuated rows of the equation of motion in single contact (Wieber, 2006). First, we described by hand a set of eleven key postures. A geometric path connecting them was obtained by interpolating Bezier curves between these postures and using inverse kinematics to fix the position and orientation of the support foot on the ground. Then, we retimed the path into a suitable trajectory by TOPP. As is common with numerical TOPP implementations, we added safety margins to constraint

⁵  <https://github.com/fkanehiro/openhrp3>

inequalities: the foot dimensions X and Y were scaled by 70% and the friction coefficient set to $\mu = 0.4$ (while the actual value was set to $\mu = 0.8$ in OpenHRP).

Figure 4.8 shows a time-lapse of the final retimed motion. Because time optimality yields bang-bang control laws, the retimed motion always saturates at least one of the contact constraints. Therefore, trying to execute it faster should result in the end-effector breaking surface contact, *e.g.*, tipping on the edge of the foot. We observed this phenomenon in the experiment, as illustrated by Figure 4.9. Note that, as we used HRP4's stabilizer while executing the motion, it was still possible in practice to accelerate the motion by about 5%⁶ without observing any change in the foot contact condition.

Implementation. The source code of this experiment is publicly available at

 <https://github.com/stephane-caron/icra-2015>

Conclusion

We have reviewed the physics of contact, as well as techniques from computational geometry that are of particular interest to deal with stability conditions. We then derived the analytical formulation of the complementary condition for a contact wrench, the Contact Wrench Cone. To the best of our knowledge, **this is the first time that an analytical formula is derived for the friction wrench cone**, which is crucial to eliminate the redundancy that appears when considering multiple contact points. We also saw that this formula implies a closed-form expression of the COP-ZMP support polygon, which can be smaller than the convex hull of contact points in practice.

⁶ That is to say by reducing the total duration by 5% using uniform time scaling.

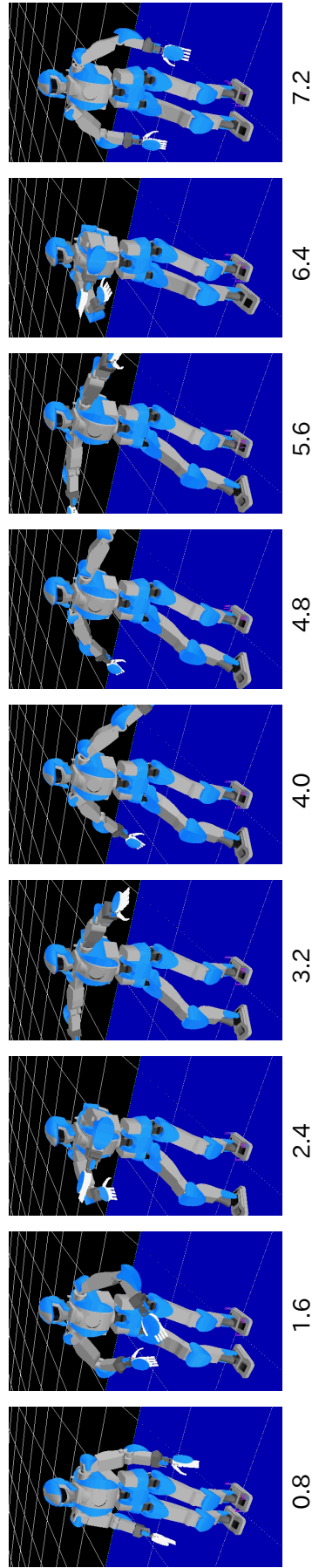


Fig. 4.8.: Snapshots of the retimed motion. Total duration is 7.3 s. Time stamps are shown below each frame. The motion stresses all components of the wrench cone. The first segment stresses the pitch by moving the COP forward. The second stresses the roll through arm motions. Meanwhile, the yaw component is stressed by chest-pitch motions. Finally, the waist performs an elliptic motion (back and forth, up and down) throughout the whole motion, thus stressing the translation of the contact foot.

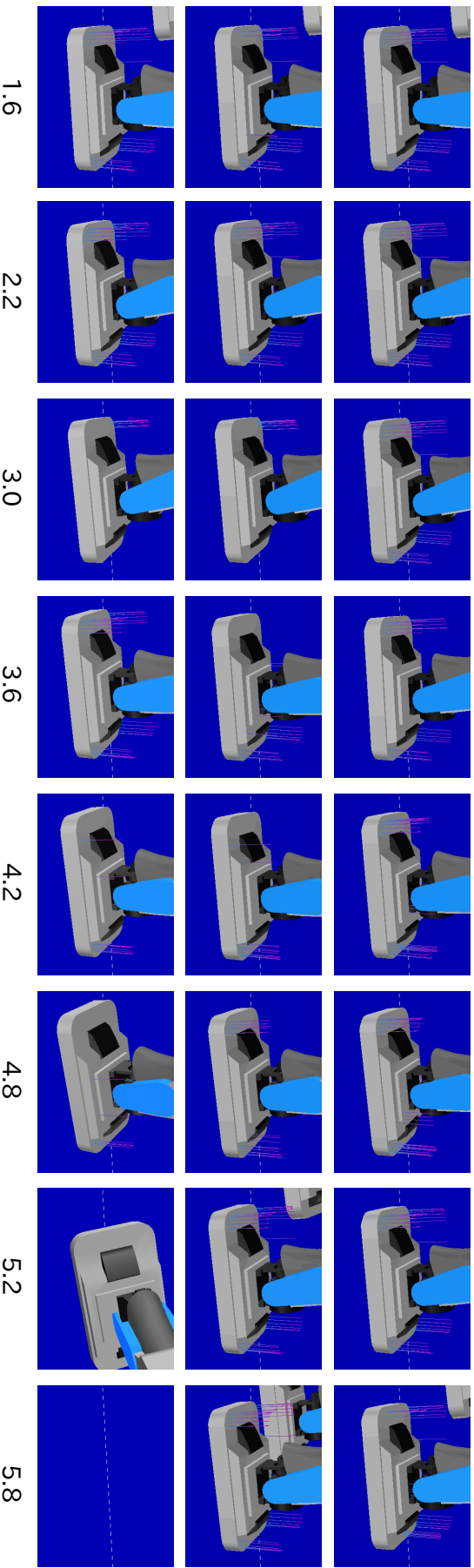


Fig. 4.9.: Zoom on the contact forces computed by OpenHRP at the left foot. Each thin purple segment corresponds to one force. Time stamps are shown below each column. **First row:** retimed motion with CWC enforced. **Second row:** 10% acceleration of the motion, CWC is not enforced. Around 4.2 s, the contact surface degenerates to a line contact on the left edge of the foot. **Third row:** 15% acceleration of the motion, CWC is not enforced. The contact surface degenerates to a line before contact is lost and the humanoid falls.

Multi-contact stability

In this chapter, we review the fundamentals of whole-body dynamics before providing our own derivation of a general multi-contact stability criterion, namely the gravito-inertial wrench cone. We then show how to apply this criterion to static and dynamic equilibrium problems. In particular, we show how to express it within the framework of TOPP, which advances our plan for integration with a fast kinodynamic motion planner.

5.1 Whole-body dynamics

Suppose that our robot makes N point contacts with the environment at C_1, \dots, C_N . The *translation* Jacobian \mathbf{J}_{C_j} calculated at the contact point C_j is defined by

$$\mathbf{J}_{C_j} := \frac{\partial \mathbf{p}_{C_j}}{\partial \mathbf{q}},$$

so that $\mathbf{J}_{C_j} \dot{\mathbf{q}} = \dot{\mathbf{p}}_{C_j}$. (Note that all our coordinate vectors \mathbf{p} are defined with respect to the inertial frame.) The equations of motion of the robot are (see *e.g.* Wieber, 2006):

Property 3 (Equations of motion under point contact constraints). *For the robot making point or surface (Proposition 4) contacts with the environment represented contact points $\{C_1, \dots, C_N\}$, the equations of motion can be written:*

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{f}^{gi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \sum_{j=1}^N \mathbf{J}_{C_j}^\top \mathbf{f}_j^c, \quad (5.1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are the n -dimensional vectors of DOF values, velocities and accelerations, $\mathbf{M}(\mathbf{q})$ is the $n \times n$ joint-inertia matrix, $\mathbf{f}^{gi}(\mathbf{q}, \dot{\mathbf{q}})$ the n -dimensional vector of gravity and Coriolis forces. In case the robot has n_a actuated joints, $\boldsymbol{\tau}$ is the n_a -dimensional vector of torques at the actuated joints and \mathbf{S} is the $n_a \times n$ joint selection matrix. Finally, for each $j \in \{1, \dots, N\}$, \mathbf{f}_j^c is a 3-dimensional vector of contact force and \mathbf{J}_{C_j} is the $3 \times n$ translation Jacobian calculated at point C_i .

As discussed in the previous Chapter, all contact forces on a link can be uniquely described by a contact wrench. This property transfers to the equations of motion under proper definition of jacobian matrices.

Let $\mathbf{J}_i^{\text{rot}}$ denote the *rotation* Jacobian for the link i , characterized by the property

$$\mathbf{J}_i^{\text{rot}} \dot{\mathbf{q}} = \boldsymbol{\omega}_i,$$

where $\boldsymbol{\omega}_i$ denotes the angular velocity of the link. Note that both translation and rotation jacobians are readily available in forward kinematics software. For instance, they correspond to the function prototypes `ComputeJacobianTranslation(link, point)` and `ComputeJacobianAxisAngle(link)` in OpenRAVE.

Property 4. *The equations of motion can be equivalently written as:*

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{f}^{gi}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \sum_{\text{contact } i} \mathbf{J}_i^w{}^\top \mathbf{w}_i^c. \quad (5.2)$$

where \mathbf{w}_i^c denotes the contact wrench taken with respect to a single contact point C_i on link i , and $\mathbf{J}_i^w = [\mathbf{J}_{C_i}^\top \mathbf{J}_i^{\text{rot}, \top}]^\top$ denotes the $6 \times n$ matrix obtained by stacking vertically the translation and rotation jacobians.

Proof. Let us denote by $\mathbf{w}_i^c = (\mathbf{f}_i^c, \boldsymbol{\tau}_i)$ the contact wrench exerted on link i and taken at a reference contact point C_i .

$$\begin{aligned}\mathbf{f}_i^c &:= \sum_j \mathbf{f}_{ij}^c, \\ \boldsymbol{\tau}_i &:= \sum_j \overrightarrow{C_i C_{ij}} \times \mathbf{f}_{ij}^c.\end{aligned}$$

The definition of the rotation Jacobian implies that, for any point C on the link and any vector \mathbf{u} , one has

$$\frac{\partial \overrightarrow{C_i C}}{\partial \mathbf{q}}^\top \mathbf{u} = \mathbf{J}_{\text{rot}}^\top \left(\overrightarrow{C_i C} \times \mathbf{u} \right).$$

The position of a contact point C_{ij} can be expressed as $\mathbf{p}_{C_{ij}} = \mathbf{p}_{C_i} + \overrightarrow{C_i C_{ij}}$. Differentiating with respect to \mathbf{q} yields:

$$\begin{aligned}\mathbf{J}_{C_{ij}} &= \mathbf{J}_{C_i} + \frac{\partial \overrightarrow{C_i C_{ij}}}{\partial \mathbf{q}} \\ \sum_i \mathbf{J}_{C_{ij}}^\top \mathbf{f}_i^c &= \sum_i \mathbf{J}_{C_i}^\top \mathbf{f}_i^c + \frac{\partial \overrightarrow{C_i C_{ij}}}{\partial \mathbf{q}}^\top \mathbf{f}_i^c \\ &= \mathbf{J}_{C_i}^\top \sum_i \mathbf{f}_i^c + \sum_j \mathbf{J}_i^{\text{rot}\top} \left(\overrightarrow{C_i C_{ij}} \times \mathbf{f}_{ij}^c \right).\end{aligned}$$

Combining the two properties above, we finally have that $\sum_i \mathbf{J}_{C_{ij}}^\top \mathbf{f}_i^c = \mathbf{J}_i^{w\top} \mathbf{w}_i^c$. \square

5.1.1 Dynamic equilibrium

For a given link i (this time not necessarily a contacting link), we write

- m_i and G_i its mass and COM, respectively;
- \mathbf{R}_i its orientation matrix in the absolute frame;
- $\boldsymbol{\omega}_i^\ell$ its angular velocity in the link frame;
- \mathbb{I}_i^ℓ its inertia matrix in the link frame.

The linear momentum \mathbf{P} and angular momentum \mathbf{L}_G of the system, taken at the COM G , are defined by:

$$\mathbf{P} := \frac{1}{m} \sum_{\text{link } k} m_i \dot{\mathbf{p}}_{G_i}, \quad (5.3)$$

$$\mathbf{L}_G := \sum_{\text{link } i} m_i \overrightarrow{GG_i} \times \dot{\mathbf{p}}_{G_i} + \mathbf{R}_i \mathbb{I}_i^\ell \boldsymbol{\omega}_i^\ell. \quad (5.4)$$

The *dynamic wrench* of the robot at G is the wrench $(\dot{\mathbf{P}}, \dot{\mathbf{L}}_G)$. It is a purely kinematic object that can be computed by forward kinematics (differentiating the two equations above) from joint-angle positions, velocities and accelerations.

The fundamental principle of dynamics states that the dynamic wrench of the robot is equal to the total wrench of forces acting on the system, that is

$$\begin{bmatrix} \dot{\mathbf{P}} \\ \dot{\mathbf{L}}_G \end{bmatrix} = \begin{bmatrix} \mathbf{f}^g \\ \mathbf{0} \end{bmatrix} + \sum_{\text{contact } i} \begin{bmatrix} \mathbf{f}_i^c \\ \overrightarrow{GC_i} \times \mathbf{f}_i^c \end{bmatrix}, \quad (5.5)$$

where \mathbf{f}^g denotes the gravity force and \mathbf{f}_i^c is the contact force exerted by the environment on the robot at C_i . This equation is also called *dynamic balance* or the *dynamic equilibrium* of the system. It can be derived from Gauss's principle of least constraint (equivalently, from d'Alembert's principle) and corresponds to the six unactuated components in the equations of motion of the system (robot + environment) (Wieber, 2006).

5.1.2 Dynamic wrenches

The Gravito-Inertial Wrench (GIW) \mathbf{w}^{gi} , considered with respect to a fixed point O (not necessarily the origin of the world frame), is defined by:

$$\mathbf{w}_O^{gi} := \begin{bmatrix} \mathbf{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} := \begin{bmatrix} \mathbf{f}^g - \dot{\mathbf{P}} \\ \overrightarrow{OG} \times (\mathbf{f}^g - \dot{\mathbf{P}}) - \dot{\mathbf{L}}_G \end{bmatrix}. \quad (5.6)$$

Meanwhile, the *contact wrench* \mathbf{w}^c is defined as:

$$\mathbf{w}_O^c := \begin{bmatrix} \mathbf{f}^c \\ \boldsymbol{\tau}_O^c \end{bmatrix} := \sum_{\text{contact } i} \begin{bmatrix} \mathbf{f}_i^c \\ \overrightarrow{OC_i} \times \mathbf{f}_i^c \end{bmatrix} \quad (5.7)$$

Dynamic equilibrium (5.5) can then be written as

$$\mathbf{w}^{gi} + \mathbf{w}^c = \mathbf{0}. \quad (5.8)$$

The rationale behind this wrench-space formulation is that the gravito-inertial describes the motion of the overall system while the contact wrench describes its interactions with the environment.

5.1.3 Contact stability

We saw in Chapter 4 how accelerations and forces are in a complementary relationship. However, complementary variables are not equivalent from a control perspective: accelerations are controlled by the robot, while forces are chosen by the environment.¹ Pang and Trinkle (2000) have defined two notions of *contact stability* that abstract the intervention of the environment and focus on the choice of the robot. At the whole-body level, this notion translates as follows.

Definition 7 (Weak Contact Stability). *A contact is weakly stable when there exists a solution $(\ddot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{w}_1, \dots, \mathbf{w}_N)$ of the equations of motion satisfying the fixed contact mode for all contacting links. That is to say,*

- *there is no relative motion at any contact i : $\mathbf{J}_i^w \dot{\mathbf{q}} = 0$ and $\mathbf{J}_i^w \ddot{\mathbf{q}} = -\dot{\mathbf{J}}_i^w \dot{\mathbf{q}}$,*
- *actuated torques $\boldsymbol{\tau}$ are within torque limits: $|\boldsymbol{\tau}| \leq \boldsymbol{\tau}_{\max}$*
- *contact wrenches $\mathbf{w}_1^c, \dots, \mathbf{w}_N^c$ lie in their respective contact wrench cones.*

This formulation has been widely used in the literature. In approaches based on inverse dynamics, the first condition is first enforced kinematically, then torques and complementary forces are solved (Hauser, 2014; Righetti et al., 2013). The “weakness” in the definition above refers to the notions of strong and weak stability stated by Pang and Trinkle (2000). Strong stability happens when *all* solutions to the equations of motion satisfy the fixed contact mode. In what follows, we will always refer to contact stability in the weak sense.

Remember that the image of a polyhedral convex cone by a linear mapping is itself a polyhedral convex cone (c.f. Section 4.2). Thus, as all local contact wrenches $\{\mathbf{w}_i^c\}$

¹ From a game theoretical perspective, robot and environment are two players of a game. The strategy of the robot, i.e., selecting joint accelerations, reduces the choices that the environment can make in the current contact mode. For instance, an acceleration of the center of mass binds contact forces via the relation $\sum_i \mathbf{f}_i^c = m(\mathbf{g} - \ddot{\mathbf{p}}_G)$.

lie in wrench cones, the whole-body wrench cone w^c must lie in the (whole-body) contact wrench cone \mathcal{C} resulting from the linear mapping:

$$w_O^c = \sum_i w_{i,O}^c.$$

We will call \mathcal{C} the CWC for short. Since the gravito-inertial and contact wrenches are simply opposites (5.8), the gravito-inertial wrench w_O^{gi} also lies in a cone $-\mathcal{C}$, which is called the GIWC. In terms of whole-body dynamics, the notion of weak contact stability can be equivalently stated as:

Proposition 6 (Weak Contact Stability). *A motion of the robot is (weak-contact) stable if and only if the contact wrench (resp. gravito-inertial wrench) it generates belongs to the CWC (resp. GIWC).*

We will also say that a motion is *dynamically stable* when it satisfies weak contact stability.

5.2 Motivation for wrench reduction

Let us first consider the approach that consists in applying TOPP directly to humanoid trajectory optimization. It is *e.g.*, the approach followed by Suleiman et al. (2010), in which the authors took into account balance and sliding constraints, but not the actuation redundancy that arises in double support configurations. Meanwhile, Hauser (2014) dealt with both actuator limits and frictional contacts, using a polytope projection algorithm to deal with internal force redundancy. In both cases, redundancy was the object of a special treatment, namely avoided or projected.

As introduced in Chapter 3, TOPP computes a mapping $s(t)$ from an the index s of an original path $q(s)$ to the reparameterized time t . We will denote by q_s and q_{ss} (resp. \dot{q} and \ddot{q}) the derivatives with respect to the path variable s (resp. to the reparameterized time t). The two pairs are linked by the chain-rule:

$$\begin{aligned}\dot{q} &= \dot{s}q_s, \\ \ddot{q} &= \ddot{s}q_s + \dot{s}^2q_{ss}.\end{aligned}$$

As a consequence, constraints expressed as functions of $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ can be written equivalently as functions of the path derivatives (s, \dot{s}, \ddot{s}) .

Three types of constraints usually considered for humanoids: maintaining the ZMP inside its support polygon, as discussed in the previous Chapter, frictional contact (Definitions 4-5) and torque limits $|\tau_{\max}| \leq \tau_{\max}$. All three types of constraints boil down to a relation $f(s, \dot{s}, \ddot{s}) \leq 0$. We will show how each of them can be further factorized in:

$$\ddot{s}\mathbf{a}(s) + \dot{s}^2\mathbf{b}(s) + \mathbf{c}(s) \leq 0, \quad (5.9)$$

which is the input format for TOPP (c.f. Chapter 3). With the ability to compute vectors $(\mathbf{a}(s), \mathbf{b}(s), \mathbf{c}(s))$ at each index $s \in [0, T_{\text{ref}}]$ of the reference trajectory, it becomes possible to retime whole-body humanoid motions.

5.2.1 TOPP for ZMP constraints

Although it does not guarantee balance, the condition that the Zero-Moment Point (ZMP) lies inside the convex hull of ground contact points is physically consistent with the robot not tilting. In Pham and Nakamura (2012), the authors derived the TOPP form (5.9) of this constraint for rectangular support areas. We now provide a similar derivation for any convex polygonal area.

The coordinates of the ZMP defined on horizontal floor are given by:

$$\begin{bmatrix} x_Z \\ y_Z \end{bmatrix} = \frac{1}{\mathbf{e}_z \mathbf{f}^{gi}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tau_O^{gi} \quad (5.10)$$

The condition that (x_Z, y_Z) belongs to the convex hull of the ground contact points is given in H-representation by Figure 4.6. Let us write an arbitrary line of this system of inequalities as:

$$\alpha y_Z + \beta x_Z + \gamma \leq 0. \quad (5.11)$$

Introducing $\mathbf{u}_{\alpha\beta} := (-\alpha \ \beta \ 0)$ and $\mathbf{u}_\gamma := \mathbf{c}$, multiplying (5.11) by $-\mathbf{e}_z \mathbf{f}^{gi}$ yields

$$0 \geq \mathbf{u}_{\alpha\beta} \cdot \tau_O^{gi} + \mathbf{u}_\gamma \cdot \mathbf{f}^{gi}$$

Expanding \mathbf{f}^{gi} and τ_O^{gi} , the constraint becomes:

$$\begin{aligned} 0 &\geq \sum_i m_i (\mathbf{u}_{\alpha\beta} \cdot \mathbf{p}_{G_i} \times (\mathbf{g} - \ddot{\mathbf{r}}^i) + \mathbf{u}_\gamma \cdot (\mathbf{g} - \ddot{\mathbf{r}}^i)) \\ &\geq \sum_i m_i (\mathbf{g} - \ddot{\mathbf{r}}^i) \cdot (\mathbf{u}_{\alpha\beta} \times \mathbf{p}_{G_i} + \mathbf{u}_\gamma) \end{aligned}$$

Given the translation jacobian \mathbf{J}_{G_i} and hessian $\mathbf{H}_{G_i} := \frac{\partial^2 \mathbf{p}_{G_i}}{\partial \mathbf{q}^2}$ of each link's COM, one can write:

$$\begin{aligned} \dot{\mathbf{p}}_{G_i} &= \mathbf{J}_{G_i} \dot{\mathbf{q}} = \mathbf{J}_{G_i} \mathbf{q}_s \dot{s} \\ \ddot{\mathbf{p}}_{G_i} &= \mathbf{J}_{G_i} \ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{H}_{G_i} \dot{\mathbf{q}} = \mathbf{J}_{G_i} \mathbf{q}_s \ddot{s} + \dot{s}^2 (\mathbf{J}_{G_i} \mathbf{q}_s s + \mathbf{q}_s^\top \mathbf{H}_{G_i} \mathbf{q}_s) \end{aligned}$$

Using these expressions, (5.11) can finally be put in form (5.9) with:

$$\begin{aligned} \mathbf{a}_{\text{ZMP}}(s) &= - \sum_i m_i \mathbf{J}_{G_i} \mathbf{q}_s (\mathbf{u}_{\alpha\beta} \times \mathbf{p}_{G_i} + \mathbf{u}_\gamma), \\ \mathbf{b}_{\text{ZMP}}(s) &= - \sum_i m_i (\mathbf{J}_{G_i} \mathbf{q}_s s + \mathbf{q}_s^\top \mathbf{H}_{G_i} \mathbf{q}_s) \cdot (\mathbf{u}_{\alpha\beta} \times \mathbf{p}_{G_i} + \mathbf{u}_\gamma), \\ \mathbf{c}_{\text{ZMP}}(s) &= \mathbf{g} \cdot (\mathbf{u}_{\alpha\beta} \times \mathbf{p}_G + m \mathbf{u}_\gamma). \end{aligned}$$

This formulation allows the application of TOPP to maintain the ZMP inside its stability polygon.

5.2.2 TOPP for frictional contacts

Projecting on the last six coordinates (5.5) of the equations of motion gives:

$$\Pi(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{c}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}) = \Pi \mathbf{J}^\top \mathbf{f} \quad (5.12)$$

with Π the corresponding $6 \times n$ projector. Using the pseudo-inverse $(\Pi \mathbf{J}^\top)^\dagger$, we can express a least-square solution \mathbf{f}_0 to (5.12):

$$\mathbf{f}_0 = (\Pi \mathbf{J}^\top)^\dagger \Pi \cdot (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{c}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}) \quad (5.13)$$

$$\tau_O^{gi} = \mathbf{E}(\mathbf{q}) \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{E}(\mathbf{q}) \dot{\mathbf{q}}^\top \mathbf{c}(\mathbf{q}) \dot{\mathbf{q}} + \mathbf{E}(\mathbf{q}) \mathbf{g} \quad (5.14)$$

where $\mathbf{E}(\mathbf{q}) := \mathbf{S} \left(\mathbf{I}_n - (\Pi \mathbf{J}^\top)^\dagger \Pi \right)$ is an actuated torques projection matrix. With this last expression, we can express torque constraints $\boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}$ as

$$\begin{aligned} \mathbf{a}_{\text{lstsq}}(s) &= \mathbf{E}(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{q}_s, \\ \mathbf{b}_{\text{lstsq}}(s) &= \mathbf{E}(\mathbf{q}) \left[\mathbf{M}(\mathbf{q}) \mathbf{q}_{ss} + \mathbf{q}_s^\top \mathbf{c}(\mathbf{q}) \mathbf{q}_s \right], \\ \mathbf{c}_{\text{lstsq}}(s) &= \mathbf{E}(\mathbf{q}) \mathbf{g}(\mathbf{q}) - \boldsymbol{\tau}_{\max}. \end{aligned}$$

However, contact forces \mathbf{f}_i should also lie in their linearized friction cones (Definition 5), equivalently, $\mathbf{T} \cdot \mathbf{f} \leq 0$. One can use Quadratic Programming (QP) to take into account these inequalities when solving for \mathbf{f} at Equation (5.12), yet this approach yields a torque projector $\mathbf{E}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ with a non-linear dependency on $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, which makes it unsuitable for use with TOPP. This yields us to the observation that **redundancy in the unactuated force space is unsuitable for use with TOPP**.

5.2.3 Preliminary experiments and discussion

In an early attempt to solve this issue, we used solutions to the QP problem *with inequality constraints* only for specific values of the path tracking velocity \dot{s} , and use a linear model to describe how \mathbf{f} deviates from this solution when $\dot{s} < 1$ or $\dot{s} > 1$. All solutions to Equation (5.12) can be written:

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{f}_0(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \tilde{\mathbf{Q}}(\mathbf{q}) \mathbf{z} \quad (5.15)$$

where the expression of a particular solution $\mathbf{f}_0(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is given by Equation (5.13), and $\tilde{\mathbf{Q}}(\mathbf{q}) = (\mathbf{I} - (\Pi \mathbf{J}_C^\top)^\dagger \Pi \mathbf{J}_C^\top)$ is the projector on the nullspace of the solution space. We can use the vector $\mathbf{z} \in \mathbb{R}^{3k}$ to enforce the additional constraint $\mathbf{T} \mathbf{f} \leq 0$. Using the chain-rule $\dot{\mathbf{q}} = \dot{s} \mathbf{q}_s$ and $\ddot{\mathbf{q}} = \ddot{s} \mathbf{q}_s + \dot{s}^2 \mathbf{q}_{ss}$ in Equation (5.13), we can write the variation of \mathbf{f}_0 between the retimed state $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ and the initial trajectory state $(\mathbf{q}, \mathbf{q}_s, \mathbf{q}_{ss})$ as follows:

$$\begin{aligned} \mathbf{f}_0(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) &= \dot{s}^2 \mathbf{f}_0(\mathbf{q}, \mathbf{q}_s, \mathbf{q}_{ss}) + (1 - \dot{s}^2) \tilde{\mathbf{g}} + \ddot{s} \tilde{\mathbf{m}}, \\ \tilde{\mathbf{g}} &:= (\Pi \mathbf{J}_C^\top)^\dagger \Pi \mathbf{g}(\mathbf{q}), \\ \tilde{\mathbf{m}} &:= (\Pi \mathbf{J}_C^\top)^\dagger \Pi \mathbf{M}(\mathbf{q}) \mathbf{q}_s. \end{aligned}$$

We verify that $f_0(q, \dot{q}, \ddot{q}) = f_0(q, q_s, q_{ss})$ when $\dot{s} = 1$, which corresponds to the initial trajectory. Equation (5.15) becomes

$$f(q, \dot{q}, \ddot{q}) = \dot{s}^2 [f_0(q, q_s, q_{ss}) + \tilde{Q}(q)z_0] + (1 - \dot{s}^2) [\tilde{g} + \tilde{Q}(q)z_1] + \ddot{s} [\tilde{m} + \tilde{Q}(q)z_2]$$

Under this formulation, solutions to the general problem of finding (z_0, z_1, z_2) such that $Tf(q, \dot{q}, \ddot{q}) \leq 0$ still depend on both \dot{s} and \ddot{s} . Making the approximation of choosing z_i 's that only depend on q . First, taking $(\dot{s}^2, \ddot{s}) = (1, 0)$, we chose z_0 from:

$$\begin{aligned} \min_{z_0} \quad & \|z_0\| \\ \text{s.t.} \quad & T(f_0(q, q_s, q_{ss}) + \tilde{Q}(q)z_0) \leq 0 \end{aligned}$$

Note that this QP has a solution if and only if the input trajectory is feasible. Next, we derived z_1 using z_0 at, e.g., $(\dot{s}^2, \ddot{s}) = (2, 0)$:

$$\begin{aligned} \min_{z_1} \quad & \|z_1\| \\ \text{s.t.} \quad & T(2f_0(q, q_s, q_{ss}) - \tilde{g} + \tilde{Q}(q)z_0 - \tilde{Q}(q)z_1) \leq 0 \end{aligned}$$

Similarly for z_2 at, e.g., $(\dot{s}^2, \ddot{s}) = (1, 1)$:

$$\begin{aligned} \min_{z_2} \quad & \|z_2\| \\ \text{s.t.} \quad & T(f_0 + \tilde{m} + \tilde{Q}(q)z_0 + \tilde{Q}(q)z_2) \leq 0 \end{aligned}$$

Once z_0 , z_1 and z_2 are decided, the linearized Coulomb constraint becomes (5.9) with

$$\begin{aligned} a_{\text{friction}}(s) &= T(\tilde{m} + \tilde{Q}(q)z_2) \\ b_{\text{friction}}(s) &= T(f_0 - \tilde{g} + \tilde{Q}(q)(z_0 - z_1)) \\ c_{\text{friction}}(s) &= T(\tilde{g} + \tilde{Q}(q)z_1) \end{aligned}$$

Actuated torque limits are finally obtained from f using Equation (??), which gives

$$\begin{aligned} a_{\text{torque}}(s) &= S^T (M(q)q_s - \tilde{m} - \tilde{Q}(q)z_2) \\ b_{\text{torque}}(s) &= S^T (Mq_s s + q_s^T c(q)q_s) \\ &\quad + S^T (\tilde{g} - f_0 - \tilde{Q}(q)(z_0 - z_1)) \\ c_{\text{torque}}(s) &= S^T (g - \tilde{g} - \tilde{Q}(q)z_1) \end{aligned}$$

This method assumes that the motion is always feasible at points $(\dot{s}^2, \ddot{s}) = (1, 0), (2, 0)$ and $(1, 1)$.

We applied this method in an experiment with the HRP4-R humanoid robot (Kaneko et al., 2011), where we show how it can accelerate a slow quasi-static motion while still maintaining ZMP balance, frictional contact and actuated torque limits. We first designed a 53-second long reference motion satisfying static stability constraints by suitable positioning of the COM. Specifically, the motion starts by moving the COM to the left foot, then steps the right foot 15 cm forward, moves the COM to the right foot, steps the left foot 15 cm forward, and finally moves the COM to the center of the support area. This high-level description is converted to key postures using inverse kinematics. Trajectory chunks are then interpolated between those postures using B-spline curves with a fixed duration of 5.9 seconds for each segment. Foot contacts are maintained in the interpolated trajectories using the method of Yamane and Nakamura (2000) for closure of kinematic chains. Figure 5.2 shows a time-lapse of the input motion.

We used TOPP to retime this trajectory. We defined the support area for the ZMP constraints as a disc centered on the COM with radius 1 cm, resulting in conservative balance. The friction coefficient was set to 0.8 while the normal component of contact forces was enforced to a minimum 10 N (by changing the right-hand side of the linearized friction constraints). Actuated torque limits were set to 50% of the robot's limits. Figure 5.2 (bottom) shows a time-lapse of the retimed motion after application of TOPP. The duration of the retimed trajectory is 24 s, *i.e.*, $2\times$ faster than its input.

We ran simulations with more aggressive values of the parameters in OpenHRP using full actuated torque limits and a 5 cm radius around the COM for the ZMP. Figure 5.1 shows the retimed trajectory profile and MVCs thus obtained. The most constrained part of the problem occurs when the COM gets close to the limits of the support area, which saturates the contact constraints. The duration of the retimed trajectory in simulation is 12.3 s, more than $4\times$ faster than its input.

Note that, for now, we do not retime the complete trajectory as a whole because of the discontinuities in $(\mathbf{a}(s), \mathbf{b}(s), \mathbf{c}(s))$ that happen when switching between single and double support. Rather, we set small time intervals around these transitions

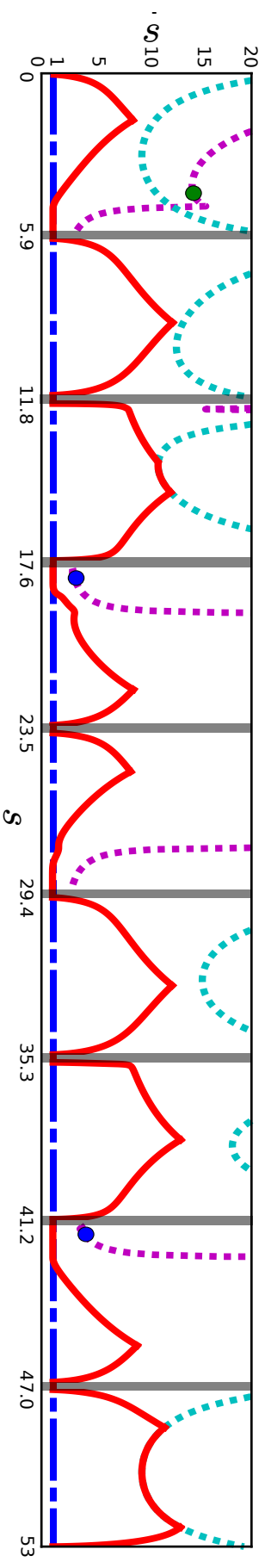
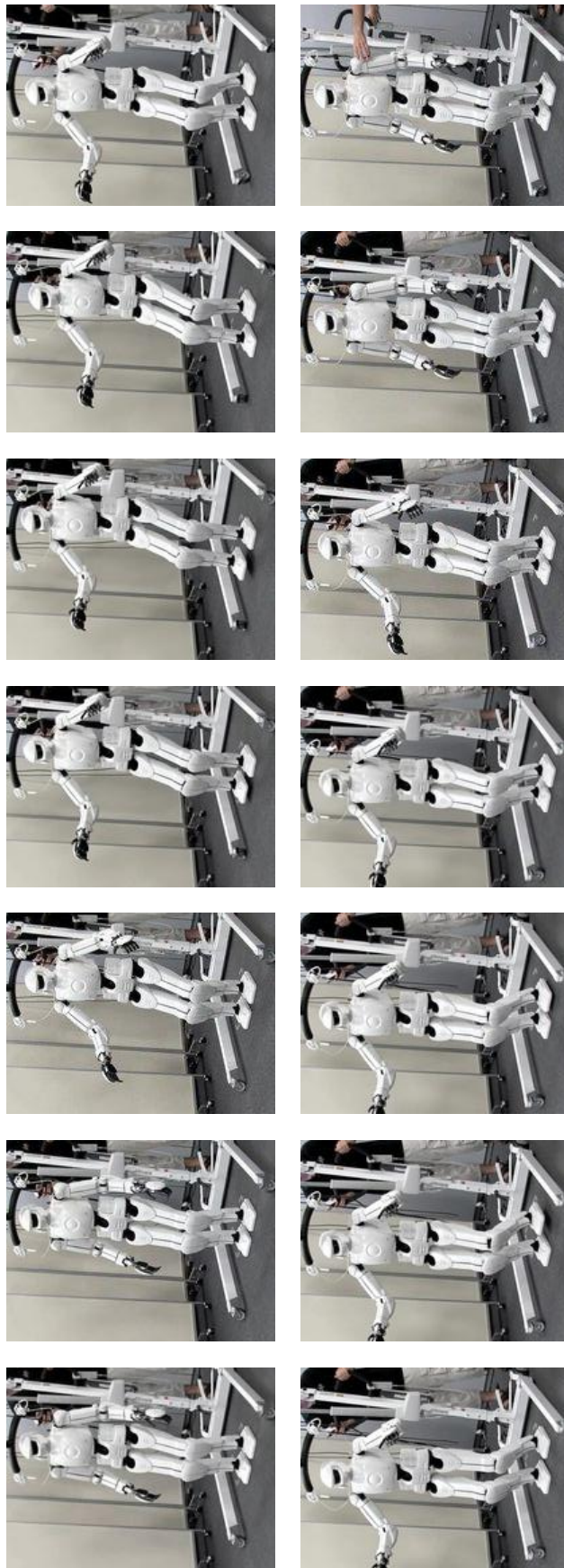


Fig. 5.1.: Retimed trajectory profiles in the (s, \dot{s}) space. Maximum Velocity Curves are depicted by dotted cyan and magenta lines. The dashed blue line represents the initial trajectory ($\dot{s} = 1$) while the red curve corresponds to the retimed trajectory. It may follow but never crosses the MVCs. Shaded gray areas show the intervals where we disabled retiming, forcing the retimed profile to go follow $\dot{s} = 1$. Blue and green dots indicate discontinuity of the MVC and singular points, respectively.

ORIGINAL MOTION



RETIMED MOTION

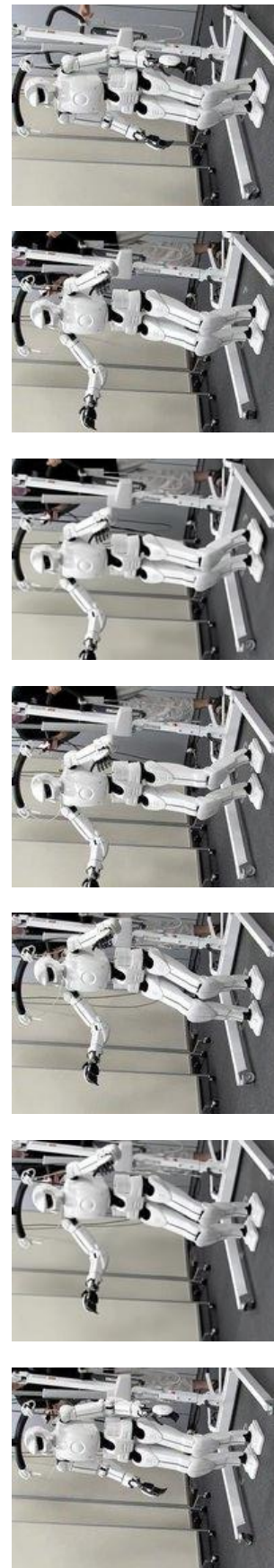


Fig. 5.2.: Time-lapse of the original and retimed trajectories. The interval between two frames is 53 s for the original and 24 s for the retimed trajectory.

during which retiming was disabled (gray areas in Figure 5.1). The duration of these intervals was set to 1.5 s on the real robot and 0.2 s in the OpenHRP experiment.

Discussion. The naive method proposed above can be applied to accelerate motions that are already stable but it does not solve the core issue, namely, that redundancy in the unactuated force space is unsuitable for time-optimal path tracking. Note that this limitation is not limited to TOPP itself, as other unrelated methods such as those based on QR decomposition Righetti et al. (2013) are equally unable to deal with linearly dependent (*i.e.*, redundant) contact forces. In our scope, such redundancy is a problem. We will now see how to solve it using contact wrenches at the whole-body level.

5.3 Gravito-inertial wrench cone

We now compute the gravito-inertial wrench cone following the approach that was proposed by Qiu et al. (2011). In matrix form, Equations (5.6)-(5.8) can be written as:

$$\mathbf{w}_O^{gi} = \sum_{\text{contact } i} \begin{bmatrix} -\mathbf{R}_i & \mathbf{0} \\ -[\overrightarrow{OC_i} \times] \mathbf{R}_i & -\mathbf{R}_i \end{bmatrix} \mathbf{w}_i^c = \mathbf{A} \mathbf{w}_{\text{all}},$$

where \mathbf{w}_{all} is the stacked vector of the contact wrenches \mathbf{w}_i^c , and $[\cdot \times]$ denotes the cross-product matrix of a vector. The key observation here is that \mathbf{A} only depends on the kinematic transform $(\overrightarrow{OC_i}, \mathbf{R}_i)$ between a the inertial frame at O and the contact frame, both which are *fixed* for each contact stance.

Property 5. *The gravito-inertial wrenches cone depends only on (1) the absolute positions and orientations of the contacting links, and (2) the local geometry and friction properties at the contact surfaces.*

The contact wrench \mathbf{w}_i^c is valid if and only if $\mathbf{w}_i^c \in \text{nonneg}(\mathbf{v}_1^i, \dots, \mathbf{v}_k^i)$ (V-representation), in matrix form $\mathbf{w}_i^c = \mathbf{V}^i \mathbf{z}_i$ with $\mathbf{z}_i \geq \mathbf{0}$. This implies that $\mathbf{w}_{\text{all}} = \mathbf{V}_{\text{all}} \mathbf{z}$ with $\mathbf{z} \geq \mathbf{0}$, where

$$\mathbf{V}_{\text{all}} := \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_N \end{bmatrix}.$$

It follows that $\mathbf{w}_O^{gi} = \mathbf{A}\mathbf{w}_{\text{all}} = \mathbf{A}\mathbf{V}_{\text{all}}\mathbf{z}$ for $\mathbf{z} \geq \mathbf{0}$, which is the V-representation of the GIWC. Using the double-description method (Section 4.2.2), one can now define $\mathbf{U}_{\text{stance}} := (\mathbf{A}\mathbf{V}_{\text{all}})^H$ to obtain the H-representation of this cone. The matrix $\mathbf{U}_{\text{stance}}$ thus constructed is such that

Property 6. *There exist valid wrenches at the contacts if and only if $\mathbf{U}_{\text{stance}}\mathbf{w}_O^{gi} \leq \mathbf{0}$.*

The matrix $\mathbf{U}_{\text{stance}}$ depends only on the absolute positions \mathbf{p}_i and orientations \mathbf{R}_i of the contacting links, as well as the friction coefficients at each contact surface.

5.3.1 Static multi-contact stability

Bretl and Lall (2008) showed that a general static stability criterion including friction could always be calculated as a COM stability polygon. They further proposed a recursive polygon expansion method to compute it. In this section, we provide an alternative method based on the previous derivation.

When the system is in static equilibrium, the two equations $\ddot{\mathbf{x}}_G = \ddot{\mathbf{y}}_G = 0$ add up to the four previously-mentioned $\ddot{\mathbf{z}}_G = 0$ and $\dot{\mathbf{L}}_G = \mathbf{0}$. In terms of the gravito-inertial wrench, these six equations become

$$\begin{bmatrix} \mathbf{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} = m \begin{bmatrix} \mathbf{g} \\ \overrightarrow{OG} \times \mathbf{g} \end{bmatrix}$$

By expanding the triple product $(\mathbf{n} \times \overrightarrow{OG} \times \mathbf{g})$, one can rewrite them equivalently as:

$$\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{1 \times 3} & \mathbf{n}^\top \end{bmatrix} \begin{bmatrix} \mathbf{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} \\ 0 \end{bmatrix}$$

$$\mathbf{p}_G = \mathbf{p}_O - (\mathbf{n}/mg) \times \boldsymbol{\tau}_O^{gi} + z_G \mathbf{n}$$

In concise form: $\mathbf{A}\mathbf{w}_O^{gi} = \mathbf{b}$ and $\mathbf{p}_G = \mathbf{C}\mathbf{w}_O^{gi} + \mathbf{d}$. Next, consider the stacked vector of contact forces $\mathbf{f}^{all} = [\mathbf{f}_1^c \cdots \mathbf{f}_n^c]^\top$. Linearized friction cones are given in half-space

representation by linear inequalities $\mathbf{F}_i \mathbf{f}_i^c \leq \mathbf{0}$. For instance, four-sided friction pyramids correspond to

$$\mathbf{F}_i = \begin{bmatrix} -1 & 0 & -\mu_i \\ +1 & 0 & -\mu_i \\ 0 & -1 & -\mu_i \\ 0 & +1 & -\mu_i \end{bmatrix} \mathbf{R}_i^\top.$$

Combining all \mathbf{F}_i 's in a block diagonal matrix \mathbf{F} yields an inequality $\mathbf{F} \mathbf{f}^{all} \leq \mathbf{0}$. Meanwhile, eq. (5.7)-(5.8) provide a linear mapping $\mathbf{w}_O^{gi} = \mathbf{W}_O^{gi} \mathbf{f}^{all}$ from contact forces to the gravito-inertial wrench. Summing up, the set of realizable contact forces in static stability is given in half-space representation by

$$\begin{aligned} \mathbf{F} \mathbf{f}^{all} &\leq \mathbf{0} \\ \mathbf{A} \mathbf{W}_O^{gi} \mathbf{f}^{all} &= \mathbf{b} \end{aligned}$$

Using the double description method, one can compute the vertex representation of this set as $\mathbf{f}^{all} = \sum_i \lambda_i \mathbf{g}_i$ ($\lambda_i > 0$).² The vertices of the COM stability polygon are finally given by $\mathbf{v}_i = \mathbf{C} \mathbf{g}_i + \mathbf{d}$.

Application to stair climbing

We demonstrate the applicability of the static stability condition by generating *quasi-statically stable* trajectories in a stair climbing experiment. A quasi-statically stable motion results from enforcing low accelerations in order to neglect the terms $\ddot{\mathbf{p}}_G$ and $\dot{\mathbf{L}}_G$ in the equations of motion.

Because we enforce slow velocities and keep the ZMP close to the COM, we do not need to check the stability condition at each time instant: each segment has a stationary stability polygon, which makes stability checking straightforward: when interpolating the linear COM trajectory $\mathbf{p}_G(s)$ ($s \in [0, 1]$) for a given segment, suffices to check whether its two extremities $\mathbf{p}_G(0)$ and $\mathbf{p}_G(1)$ lie in the support polygon.

Figure 5.3 shows the execution of the motion on the real robot. Compared to the simulation environment, we needed to perform an additional fitting of the COM

² Source code for the reduction of equality constraints in cdd is provided in the `examples/` folder of `pymanoid`. See the list of published software at the end of this manuscript.

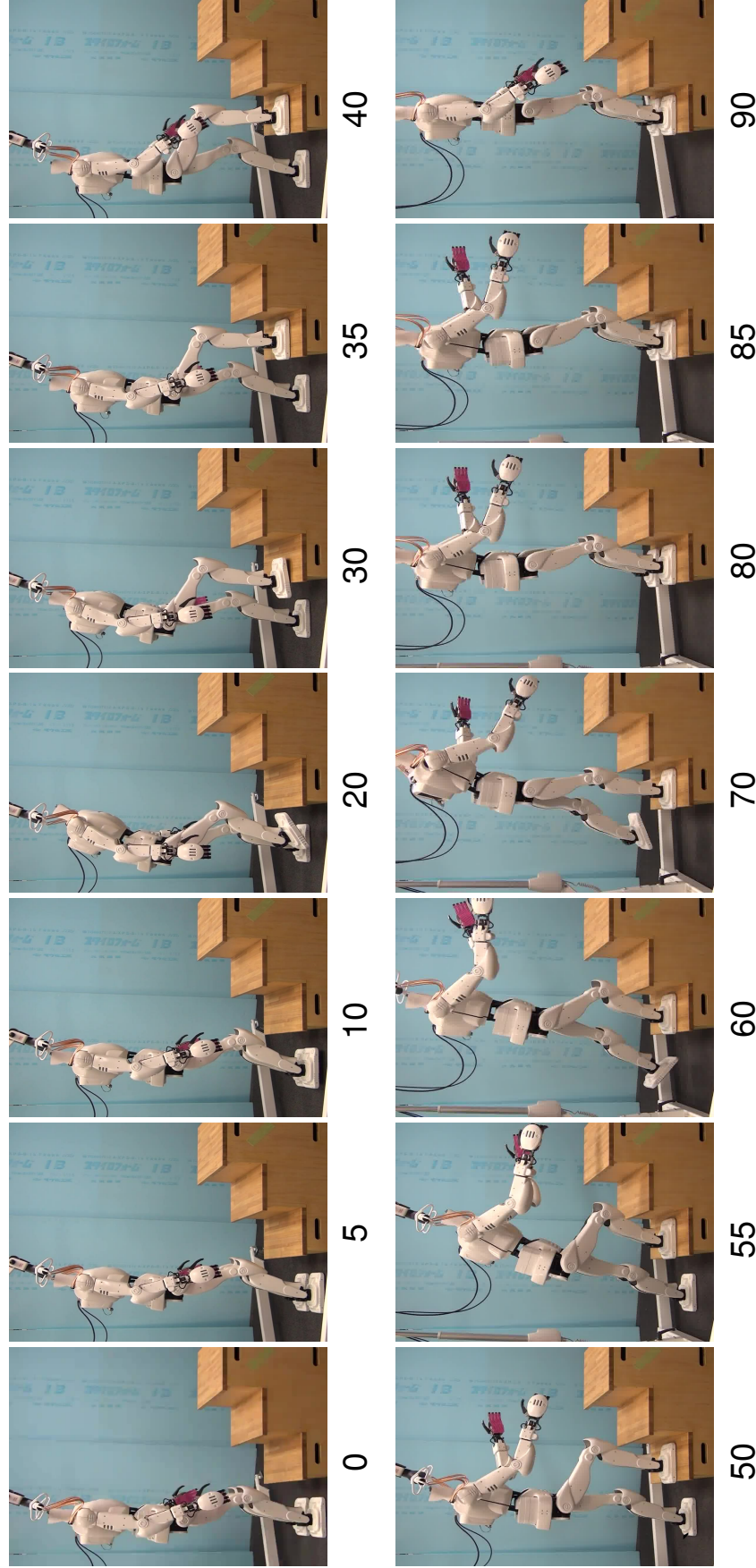


Fig. 5.3.: Execution of the planned motion by HRP-4 for climbing one step. The height of the step is 24 cm. The motion is generated at low velocity, which is the range of validity of our multi-contact stability criterion. The total execution duration is 1 min 30 s. Time stamps (in seconds) are indicated below each picture.

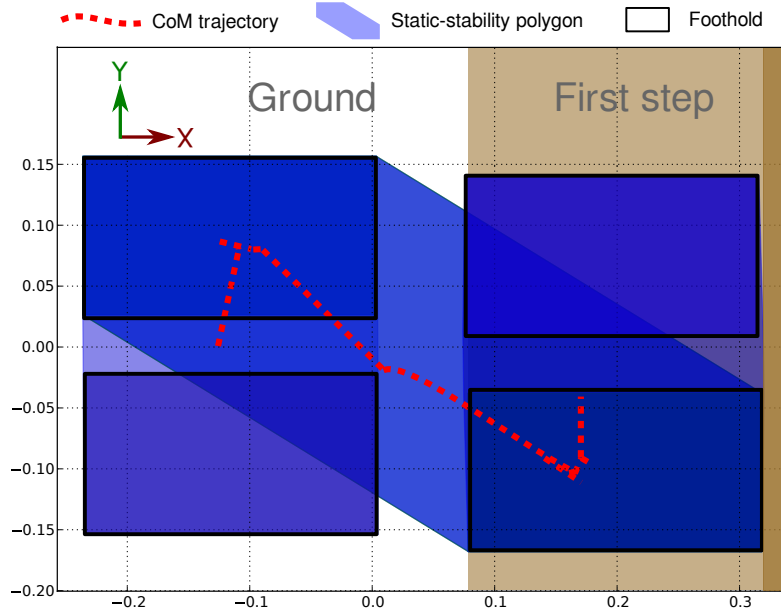


Fig. 5.4.: View of the generated CoM trajectory in the transverse plane. Consecutive support areas are represented by blue polygons while the staircase is in brown. Dotted red lines depict the CoM trajectory. Note that the sagittal vector is pointing rightward, so that the motion goes from left to right. The unit of both axes is the meter.

coordinates at the single-support postures. The reason for this tuning is that, as the motion deals with non-coplanar surfaces, we could not use HRP-4's stabilizer (which assumes all contacts are made with a horizontal floor). A general solution to avoid this would be to develop a stabilizer that takes into account the static stability polygon (rather than the traditional convex hull of ground contact points).

5.3.2 Robust multi-contact static stability

A static equilibrium configuration is said to be *robust* if not only the gravity wrench, but any wrench in some neighborhood around the gravity wrench can be generated by valid contact wrenches (Or and Rimon, 2006). In some limit situations, while simple static equilibrium may be satisfied (e.g., when the CoM is at the boundaries of the support polygon in flat terrain), robust static equilibrium adds an *informed* safety margin. Uncertainties such as model inaccuracies or unknown disturbances can be modeled within this framework.

Or and Rimon (2006) studied robust equilibrium for a robot in a 2D environment (one horizontal and one vertical directions) and for a polytopic neighborhood. They showed that the set of CoM positions that induce robust static equilibrium is no

longer an infinite vertical band, but a polygon. They also gave a “line sweep” algorithm to compute that polygon. Here, we show that the face representation of the GIWC provides a fast test for *robust static equilibrium in 3D*, which generalizes both (Bretl and Lall, 2008) (which discussed only simple equilibrium) and (Or and Rimon, 2006) (which discussed only the 2D case).

In the case of static equilibrium, we have $\ddot{\mathbf{p}}_G = \mathbf{0}$ and $\dot{\mathbf{L}}_G = \mathbf{0}$, so that $\mathbf{w}_O^{g_i} = (m\mathbf{g}, m\mathbf{p}_G \times \mathbf{g}) = \mathbf{A}\mathbf{w}_{\text{all}}$. We are now interested not only in the generation of the gravity wrench $(m\mathbf{g}, m\mathbf{p}_G \times \mathbf{g})$ but also in that of any wrench in a polytopic neighborhood $\mathcal{N}(\mathbf{p}_G)$ around it. Consider for simplicity the neighborhood defined as the convex hull of $\{(m\mathbf{g}_1, m\mathbf{p}_G \times \mathbf{g}_1), \dots, (m\mathbf{g}_K, m\mathbf{p}_G \times \mathbf{g}_K)\}$ where $\mathbf{g}_1, \dots, \mathbf{g}_K$ are vectors around \mathbf{g} .

Consider the sets \mathcal{M}_k defined as

$$\mathcal{M}_k := \left\{ \mathbf{p} : \mathbf{U}_{\text{stance}} \begin{pmatrix} \mathbf{g}_k \\ \mathbf{p} \times \mathbf{g}_k \end{pmatrix} \geq \mathbf{0} \right\} \quad (5.16)$$

and their intersection $\mathcal{M} := \bigcap_{k \in [1, K]} \mathcal{M}_k$.

Proposition 7. \mathcal{M} is the set of COM positions that ensure robust static equilibrium.

Proof. Assume that $\mathbf{p} \in \mathcal{M}$. Consider a wrench $\mathbf{w}^* \in \mathcal{N}(\mathbf{p})$. There exists $\lambda_1 \geq 0, \dots, \lambda_k \geq 0$ such that $\mathbf{w}^* = m(\lambda\mathbf{g}_1 + \dots \lambda\mathbf{g}_K, \mathbf{p} \times (\lambda\mathbf{g}_1 + \dots \lambda\mathbf{g}_K))$. Since $\mathbf{p} \in \mathcal{M}$, we have that, for all k ,

$$\mathbf{U}_{\text{stance}} \begin{pmatrix} \mathbf{g}_k \\ \mathbf{p} \times \mathbf{g}_k \end{pmatrix} \geq \mathbf{0},$$

thus there exist *valid* contact wrenches $\mathbf{w}_{\text{all}_1}, \dots, \mathbf{w}_{\text{all}_K}$ such that $(-m\mathbf{g}_k, -m\mathbf{p} \times \mathbf{g}_k) = \mathbf{A}\mathbf{w}_{\text{all}_k}$. By convexity, the contact wrench $\mathbf{w}_{\text{all}} = \sum_k \lambda_k \mathbf{w}_{\text{all}_k}$ is valid. On the other hand, by linearity, we have $\mathbf{w}^* = \mathbf{A}\mathbf{w}_{\text{all}}$. Together, these last two equations show that the wrench \mathbf{w}^* can be generated. Since \mathbf{w}^* is arbitrary in $\mathcal{N}(\mathbf{p})$, this implies that \mathbf{p} induces robust equilibrium.

Conversely, assume that \mathbf{p} induces robust equilibrium, *i.e.*, all wrenches $\mathbf{w} \in \mathcal{N}(\mathbf{p})$ can be generated. In particular there exists a valid contact wrench $\mathbf{w}_{\text{all}_1}$ that generates $(m\mathbf{g}_1, m\mathbf{p} \times \mathbf{g}_1)$. By construction of $\mathbf{U}_{\text{stance}}$, this implies that $\mathbf{p} \in \mathcal{M}_1$. By

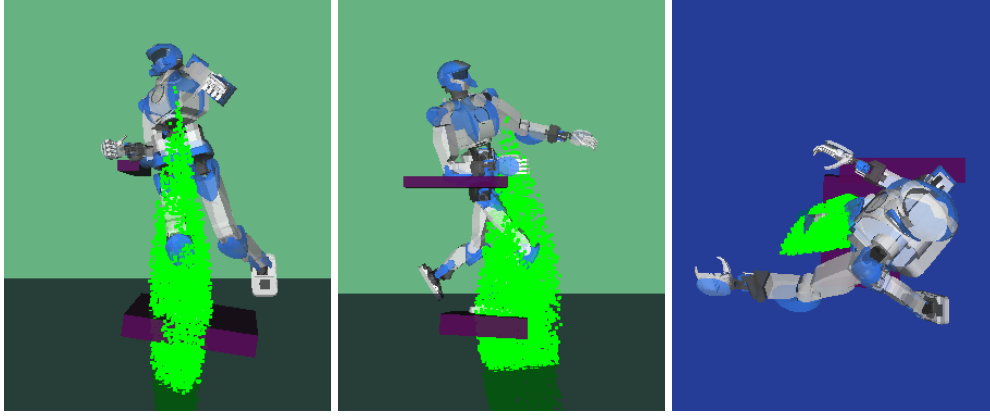


Fig. 5.5.: Testing robust static equilibrium. The robot right arm is supported by the ledge while its right foot is supported by an inclined box. The desired neighborhood around the gravity wrench is associated with $\{g_1, g_2, g_3, g_4\} = \{(0.15, 0, g), (-0.15, 0, g), (0, 0.15, g), (0, -0.15, g)\}$. We sampled one million random COM positions, those of which satisfy robust static equilibrium are depicted by a green dot. The aggregate of green dots outlines the shape of the robust static equilibrium wrench cone. The three images show views from the X, Y and Z directions respectively.

repeating this reasoning, one can show that $p \in \mathcal{M}_2, \dots, p \in \mathcal{M}_K$, which implies that $p \in \mathcal{M}$. \square

It can be noted that the \mathcal{M}_k are infinite right prisms with axis parallel to g_k . Thus, if the g_k are not collinear, then \mathcal{M} will not be a right prism, but a polyhedron. Nevertheless, from the development presented above, testing robust static equilibrium requires simply to pre-compute *once* U_{stance} , and subsequently, for each candidate COM position, to evaluate K matrix multiplications and comparisons as given in (5.16). Note that the algorithm in (Bretl and Lall, 2008) would require to perform K polytope projections in the pre-computation phase.

Figure 5.5 illustrates the proposed robust equilibrium test. For two surface contacts (right arm and right foot), the matrix U_{stance} had dimension 105×6 and could be computed in 3.5 ms on our 8-core 3.00 GHz microprocessor. Subsequently, each test took 0.1 ms.

5.3.3 Dynamic multi-contact stability

We can now resume the TOPP derivation from Section 5.2.2 on proper theoretical ground. Consider a path $\mathbf{p}(s)$ of the COM (note that we have dropped the subscript G for simplicity). Differentiating twice, we have

$$\ddot{\mathbf{p}} = \mathbf{p}_s \ddot{s} + \mathbf{p}_{ss} \dot{s}^2.$$

From its definition (5.4), the angular momentum \mathbf{L}_G can be expressed in Jacobian form³ $\mathbf{L}_G = \mathbf{J}_{\mathbf{L}_G}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{J}_{\mathbf{L}_G}(s)\mathbf{q}_s\dot{s}$. Therefore, one can always write

$$\dot{\mathbf{L}}_G = \mathbf{l}_1 \ddot{s} + \mathbf{l}_2 \dot{s}^2,$$

for some functions \mathbf{l}_1 and \mathbf{l}_2 . In general, there is no function \mathbf{l} such that $\mathbf{l}_1 = \mathbf{l}_s$ and $\mathbf{l}_2 = \mathbf{l}_{ss}$ (in other words, \mathbf{L}_G is non-holonomic). Exceptions include the cases when $\mathbf{L}_G = 0$ or when \mathbf{L}_G is the angular momentum of a single rigid body.

Substituting the expressions of $\ddot{\mathbf{p}}$ and $\dot{\mathbf{L}}_G$ into (5.6), we have:

$$\mathbf{w}_O^{gi} = \begin{bmatrix} m(\mathbf{g} - \mathbf{p}_s \ddot{s} - \mathbf{p}_{ss} \dot{s}^2) \\ m\mathbf{p} \times (\mathbf{g} - \mathbf{p}_s \ddot{s} - \mathbf{p}_{ss} \dot{s}^2) - \mathbf{l}_1 \ddot{s} - \mathbf{l}_2 \dot{s}^2 \end{bmatrix}.$$

Thus, the condition $\mathbf{U}_{\text{stance}} \mathbf{w}_O^{gi} \leq \mathbf{0}$ can be rewritten as

$$-\ddot{s} \mathbf{U}_{\text{stance}} \begin{bmatrix} m\mathbf{p}_s \\ m\mathbf{p} \times \mathbf{p}_s + \mathbf{l}_1 \end{bmatrix} - \dot{s}^2 \mathbf{U}_{\text{stance}} \begin{bmatrix} m\mathbf{p}_{ss} \\ m\mathbf{p} \times \mathbf{p}_{ss} + \mathbf{l}_2 \end{bmatrix} + \mathbf{U}_{\text{stance}} \begin{bmatrix} m\mathbf{g} \\ m\mathbf{p} \times \mathbf{g} \end{bmatrix} \leq \mathbf{0},$$

which is in the canonical form of (3.3) The centroidal trajectory, *i.e.*, the joint trajectory of the linear and angular momenta $(m\dot{\mathbf{p}}(t), \mathbf{L}_G(t))$, is thus the only piece of information required to formulate the contact stability constraint.

Implementation details. Our first attempts were consequently to interpolate a centroidal trajectory, re-time it with TOPP to satisfy contact constraints, and then interpolate a whole-body trajectory with the same linear and angular momenta by Inverse Kinematics (IK). This approach is however hampered by the difficulty in interpolating the angular momentum $\mathbf{L}_G(t)$. Because of its non-holonomy, it is impossible to integrate it into a position variable, as is the case with the linear

³ See the *pymanoid* source code for a sample implementation of angular-momentum Jacobians.

momentum. We experimented with the suggestion from (Popovic et al., 2005) to regulate $\mathbf{L}_G = 0$, but it resulted in large free limb movements with a tendency to get the limbs in inconvenient positions (e.g., hands behind the back), making the overall control task harder.

We subsequently opted for a different pipeline: interpolate the COM and end-effector trajectories first, compute a corresponding whole-body trajectory $\mathbf{q}(s)$ by inverse kinematics, and finally enforce contact stability along this trajectory by TOPP. The angular momentum $\mathbf{L}_G(\mathbf{q}, \mathbf{q}_s)$ will then result from the configurations computed by the IK solver.

Numerical TOPP solvers require relatively smooth velocity *and* acceleration profiles. Discontinuities in velocity or acceleration are allowed and properly dealt with by the TOPP library, but we found that the acceleration profiles returned by a velocity-based IK solver are too erratic for proper use with TOPP. To avoid this, we used an acceleration-based IK solver. The whole-body trajectory $\mathbf{q}(s)$ is computed as the double-integral of an acceleration trajectory $\ddot{\mathbf{q}}(s)$, where accelerations are computed as solutions to the following QP problem. Provided a reference trajectory $\mathbf{p}_G(s)$ and $\mathbf{p}_H(s)$ for the COM and a steered non-contacting link (e.g., a hand of the robot), minimize

$$\begin{aligned} & w_G \|\mathbf{J}_G \mathbf{q}_{ss} + \gamma(\dot{\mathbf{p}}_G^*(t) - \mathbf{J}_G \mathbf{q}_s) + \mathbf{q}_s^\top \mathbf{H}_G \mathbf{q}_s\|^2 \\ & + w_H \|\mathbf{J}_H \mathbf{q}_{ss} + \gamma(\dot{\mathbf{p}}_H^*(t) - \mathbf{J}_H \mathbf{q}_s) + \mathbf{q}_s^\top \mathbf{H}_H \mathbf{q}_s\|^2, \end{aligned}$$

such that

1. for each contact C_i , $\mathbf{J}_{C_i} \dot{\mathbf{q}}_{ss} = -\gamma \mathbf{J}_{C_i} \dot{\mathbf{q}}_s + \dot{\mathbf{q}}_s^\top \mathbf{H}_{C_i} \dot{\mathbf{q}}_s$, and
2. $K_{ss}(\dot{\mathbf{q}}_{\min} - \dot{\mathbf{q}}_s) \leq \dot{\mathbf{q}}_{ss} \leq K_{ss}(\dot{\mathbf{q}}_{\max} - \dot{\mathbf{q}}_s)$,

where $\dot{\mathbf{q}}_{\max} := -K_s(\mathbf{q} - \mathbf{q}_{\max})$ and $\dot{\mathbf{q}}_{\min} := -K_s(\mathbf{q} - \mathbf{q}_{\min})$. This problem can be readily addressed by many off-the-shelf QP solvers. We used CVXOPT⁴ which is free software and could deal efficiently with both equality and inequality constraints. The reader is referred to (Escande et al., 2014) for a more general solution to QP-based inverse kinematics.

⁴<http://cvxopt.org/>

In all experiments, we used $K_s = K_{ss} = \gamma = 10s^{-1}$, $w_G = 1$ and $w_{\text{link}} = 0.1$. COM and end-effector trajectories were interpolated as simple line segments.

Application to dynamic motion generation

We first illustrate this method on a common stair climbing motion. The staircase (red boxes, reconstructed from point cloud data) has a step height of 24 cm. The motion is quasi-statically stable (it can be executed at arbitrary slow velocities) and alternates single and double support segments where the projection of the COM is moved linearly from one support foot to the other. The retimed motion is shown in Figure 5.6. Computation times are reported in Table 5.1, where we detail the three consecutive computations: that of the gravito-inertial wrench matrix $\mathbf{U}_{\text{stance}}$ ($T_{\mathbf{U}_{\text{stance}}}$), of the constraint vectors $\mathbf{a}(s), \mathbf{b}(s), \mathbf{c}(s)$ (T_{abc}) and of the re-timing by our numerical TOPP solver (T_{TOPP}).

Tab. 5.1.: Computation times (ms) for the stair climbing motion, averaged over 10 runs on an 8-core 3.00 GHz processor. Segments are identified by their time stamps (see Figure 5.6).

Segment (s)	Size of $\mathbf{U}_{\text{stance}}$	$T_{\mathbf{U}_{\text{stance}}}$	T_{abc}	T_{TOPP}
0.0 – 0.4	114×6	4.2	2.0	130
0.4 – 1.0	16×6	1.3	2.0	350
1.0 – 1.2	162×6	4.5	2.2	560
1.2 – 2.0	16×6	1.3	2.0	360
2.0 – 2.8	162×6	4.3	2.2	150
Total	–	15.6	10.4	1,550

Next, let us apply our method to the setting depicted in Figure ???. The robot climbs a 10-cm box using two contacts: its right arm, set on a 90-cm high horizontal ledge, and its left foot, set on a 25° inclined stepping surface. The motion generated in this experiment may seem unnatural, as the robot could step on the box directly. Yet, the motivation for this setting is two-fold. First, stability throughout this motion cannot be checked by ZMP, as frictional contacts may be lost (the “sufficient friction” assumption does not apply) and the contact surfaces non-coplanar. Second, the solution to this constrained problem is not feasible at low velocities, which means COM-based methods such as (Bretl and Lall, 2008) cannot be applied.

Figure 5.7 shows the execution of the motion in our physics simulator. In this scenario, the friction coefficient of environment bodies (respectively the floor, box, arm support and inclined plane) was set to $\mu = 0.9$. Due to the high speed of the retimed segment (the combined duration of the right-foot and left-foot steps

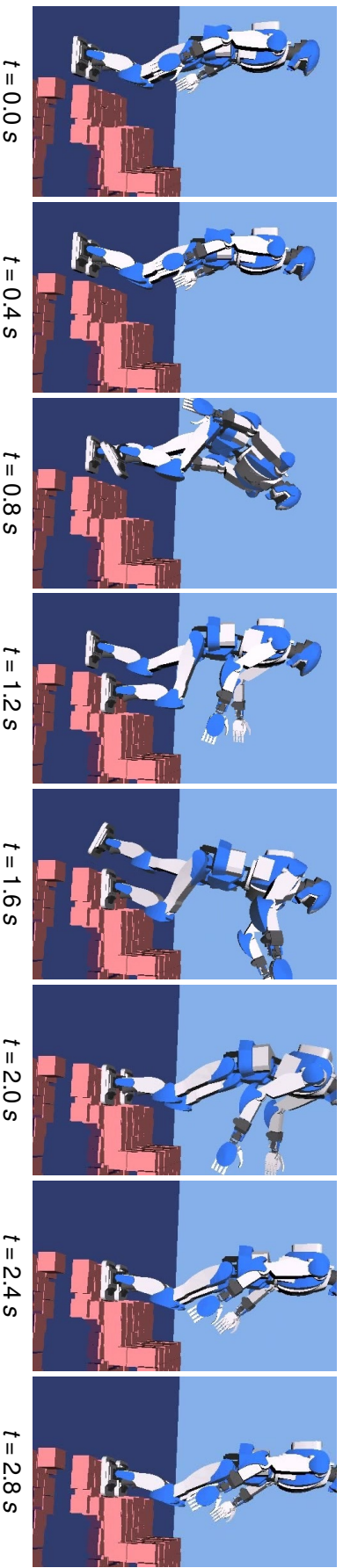


Fig. 5.6.: Stair climbing motion where contact stability is checked using our method. Time stamps are shown under each frame. The retimed motion is very fast as all underlying configurations are all statically stable.

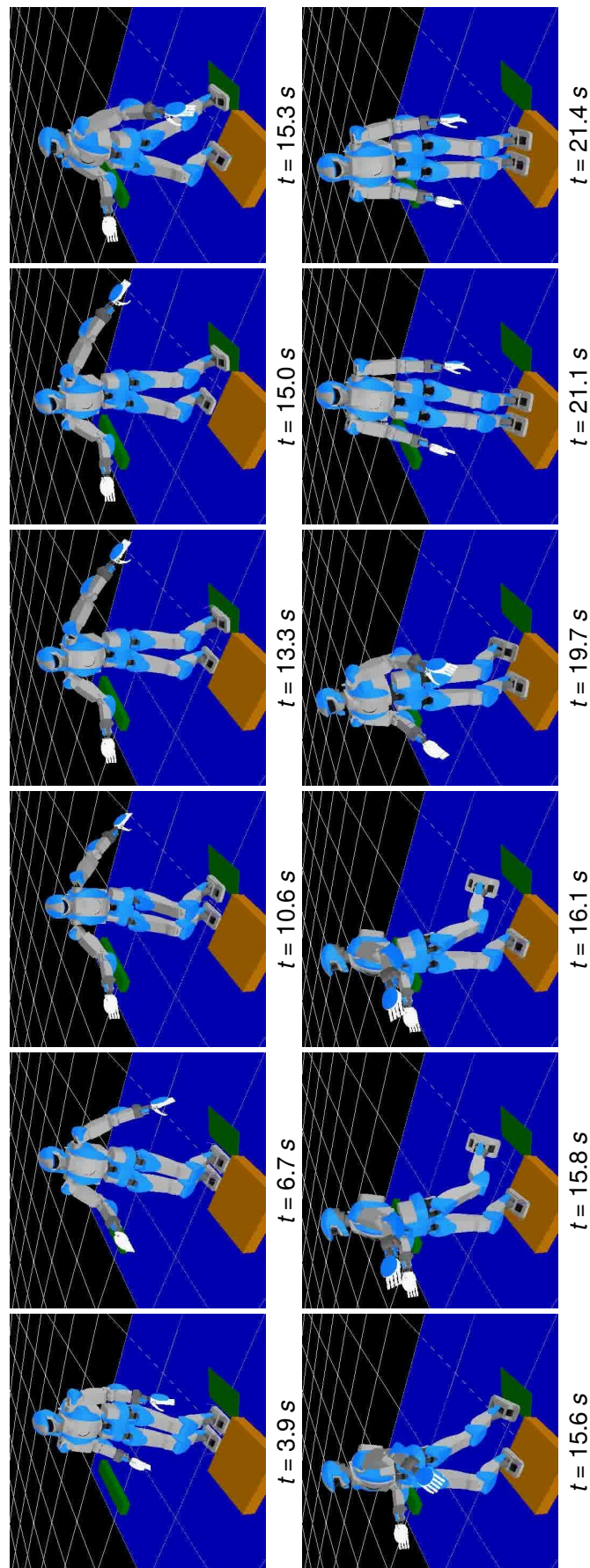


Fig. 5.7.: Execution of the multi-contact whole-body trajectory on a humanoid robot in a physics simulator. Time stamps are written under each frame. The dynamic part of the motion occurs between 15 s and 16 s, where movements need to be fast in order to maintain contact stability. This segment is dynamically but *not quasi-statically* stable: at least one contact is lost when the motion is executed at lower velocities.

is only two seconds), the position controller was not responsive enough to track the reference trajectory exactly. To palliate this, we made time-optimal trajectories slower by using a smaller friction coefficient $\mu = 0.7$ in the computations of the GIWC for TOPP (namely, $\mu = 0.7$ and contact surfaces scale by 0.7) in order to get a “safer” motion.

A further difficulty, compared to other experiments were humanoids walk on slightly inclined surfaces, was that we could not use the stabilizer module of the humanoid, which is designed for horizontal floors. Thus, the motion represented in Figure 5.7 was run with pure open-loop position control.

The transition between the first and second foot steps is the most challenging part of the motion. Its configurations are not statically stable, and it therefore needs a non-zero minimum velocity to be performed without falling. As COM trajectories are straight lines in our design, the problem of finding a feasible whole-body motion for this segment boiled down to finding a suitable COM velocity v . Manual trials being unsuccessful, we chose this velocity heuristically as $v^* = \arg \max_v \beta_v(0, 1)$, where β is an internal vector field from TOPP representing the maximum acceleration achievable along the path (see (Pham, 2014) for details). This heuristic provided feasible solutions in practice. Furthermore, from sampling neighboring vectors, we estimated the solid angle of valid COM velocity vectors around v^* to 0.1 steradian, *i.e.*, less than 0.8% of the orientation space.

Tab. 5.2.: Computation times (ms) for the box climbing motion, averaged over 10 runs on an 8-core 3.00 GHz processor. Segments are identified by their time stamps (see Figure 5.7).

Segment (s)	Size of $\mathbf{U}_{\text{stance}}$	$T_{\mathbf{U}_{\text{stance}}}$	T_{abc}	T_{TOPP}
0.0 – 6.7	113×6	4.4	2.4	420
6.7 – 15.0	84×6	3.7	2.5	310
15.0 – 15.3	55×6	3.0	2.5	60
15.3 – 21.4	118×6	4.0	2.5	320
Total	–	15.1	9.9	1,110

Performance discussion. Tables 5.2 and 5.1 show the performances of our method on the box and stair climbing motions. We compare these performances with those reported in previous work by Hauser (Hauser, 2014) where the recursive polygon expansion algorithm is used to retime humanoid trajectories under distributed contact forces and actuator limits. The two main differences between this work and our method are that (1) we calculate the GIWC rather than a recursive polygon

expansion, and (2) we use a numerical TOPP solver rather than a Sequential Linear Programming (SLP) solver. In order to make run times more comparable, we used the same path discretization resolution $N = 100$.

In accordance with previous work, we break down computation times as follows

- *Pre-computation of feasible sets* ($T_{\text{U}_{\text{stance}}} + T_{abc}$): the time reported in (Hauser, 2014) is 2.40 s, while our solution takes around 30 ms in both settings to perform this operation. When actuator torque is not limiting, it is therefore one to two orders of magnitude faster than previous work.
- *Computation of the velocity profile* (T_{TOPP}): the time reported in (Hauser, 2014) is 2.46 s, while our solution takes between 1 s and 1.5 s in the previous climbing motions. Two factors are concurring here to put execution times on the same magnitude: we use a numerical TOPP solver, an approach that is usually orders of magnitude faster than SLP; however, we do not prune redundant inequalities, which is a side benefit of recursive polygon expansion.

Overall, our approach cuts down to tens of milliseconds the pre-computations where previous work would spent half of the computing budget, while having comparable performances on the rest of the time-optimal re-timing. This comparison should of course be taken with care, as our robot, task, and computing environment are different from (Hauser, 2014). What we point out here is the difference in *orders of magnitude* on the computation times of feasible sets alone.

Conclusion

We have derived a general framework taking into account multi-contact stability for humanoid robots, a crucial element of kinodynamic motion planners for this category of robots. In particular, we showed how the general stability condition can be integrated into TOPP, which is the key to our planning framework (see Chapter 3). To the best of our knowledge, **this is the first time that a truly dynamic, non-quasi-static humanoid motion is discovered and reparameterized by TOPP** (previous works (Suleiman et al., 2010; Hauser, 2014) only reparameterized quasi-statically stable humanoid motions).

In the course of this study, we also identified that the question of *interpolating the angular momentum* was of prime importance for high-level planning of humanoid trajectories. In a primary attempt, we relied on an implicit interpolation resulting from Inverse Kinematics, but this approach does not give us control over the resulting momentum. We address this difficulty in the next Chapter.

Multi-contact ZMP

” 人の跡を求めず、
人の求めたところを求めよ。

— Variation on 松尾芭蕉
(俳人)

The ZMP is the dynamic quantity thanks to which roboticists solved the problem of walking on horizontal floors. One of its key properties is that *dynamic stability*, i.e., the balance of gravito-inertial forces by valid contact forces, implies that the ZMP lies in the convex hull of ground contact points, the so-called *support area* (Vukobratović and Stepanenko, 1972; Goswami, 1999). The support area thus provides a necessary (non-sufficient) condition for contact stability on horizontal floors.

For locomotion, a second key property of the ZMP lies in its coupling with the position of the center of mass (COM). By keeping a constant angular momentum and constraining the COM to lie on a plane, this relation simplifies into the Linear Inverted Pendulum (LIP) model (Kajita et al., 2001; Sugihara et al., 2002). In the LIP regime, the COM is “pushed away” from the ZMP under the linear dynamics of a point-mass at the tip of an inverted pendulum. The stabilization problem is then to control the position of the tip (COM) of the pendulum by moving its fulcrum (ZMP).

These main two merits (a geometric stability condition and linearized dynamics) are as well-known as the two main limitations of the ZMP: it does not account for friction, and it can only be applied on horizontal floors. The latter results from the definition of the ZMP as the point *on the ground* where the moment of gravito-inertial forces is parallel to the *vertical* (i.e., rolling and pitching moments are balanced) (Sardain and Bessonnet, 2004). In a general multi-contact scenario, each contact defines its own surface and there is no single “ground” plane. In a clas-

sis survey paper (Sardain and Bessonnet, 2004), Sardain and Bessonnet stated the problem to address as follows:

The generalization of the ZMP concept [to the case of multiple non-coplanar contacts] would be actually complete if we could define what is the pseudo-support-polygon, a certain projection of the three-dimensional (3-D) convex hull (built from the two real support areas) onto the virtual surface, inside which the pseudo-ZMP stays.

In this Chapter, we provide a complete construction of the area conjectured by Sardain and Bessonnet. Our first contribution is to characterize and calculate this area in arbitrary virtual planes. Our analysis provides a geometric construction along with a simple and fast calculation algorithm. Also, contrary to the assumption that “friction limits are not violated” usually made in the literature, the support areas we compute fully take friction into account.

From a practical point of view, locomoting systems usually regulate their linear and angular momenta. For example, a linear-pendulum model implies that the robot keeps a constant angular momentum around and a constant-height COM. These tasks restrain the set of realizable dynamic momenta, and thus contract the ZMP support area. Our second contribution here is a new algorithm to compute these contracted support areas, taking both frictional and dynamic-momentum constraints into account.

Combining these two advances, we design a whole-body controller for humanoids locomoting on arbitrary terrains. We take the ZMP plane *above* the COM and regulate the robot dynamics around that of a linear *non-inverted* pendulum. We showcase the applicability of the controller by locomoting a model of the HRP-4 humanoid robot in a challenging multi-contact scenario involving combinations of foot and hand contacts.

6.1 Previous work

Stability criteria. On horizontal floors, the support area for the ZMP is the CHCP. However, when the robot makes contact with different non-coplanar surfaces, the

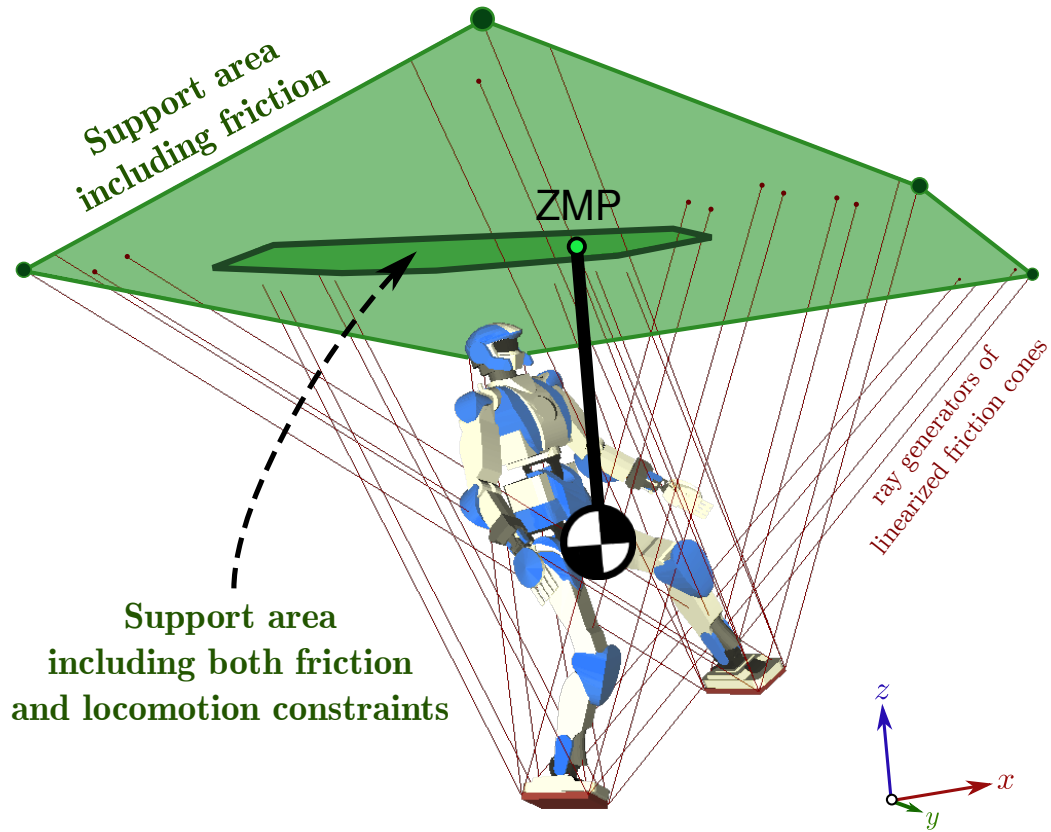


Fig. 6.1.: Overview of the construction we propose. The ZMP support area, including positive-pressure and frictional constraints, is computed in an arbitrary virtual plane (here, above the robot's head). For locomotion, linearized pendulum dynamics are obtained by regulation of the angular momentum. The shrinking of the support area incurred by this regulation is fully taken into account. A whole-body controller based on these developments finally enables multi-contact locomotion for arbitrary environments.

ZMP can no longer be defined as a point on the “ground” and the CHCP has no established connection with dynamic stability. Various attempts have been made in the literature to overcome this difficulty.

One line of research (Mitobe et al., 2004; Sardain and Bessonnet, 2004; Harada et al., 2006b; Inomata and Uchimura, 2010) conjectured that the convex hull of contact points (CHCP, a 3D volume in the general case) conveys the stability condition, and consequently sought to define a new point lying within this volume. Both (Mitobe et al., 2004) and (Inomata and Uchimura, 2010) assumed that the moments at centers of pressure and ZMP are all zeros, which is not the case in general¹ and thus results in a point that may not exist even in situations where

¹As pointed by Sardain and Bessonnet (Sardain and Bessonnet, 2004), the term “zero moment point” is misleading since the moment at these points has actually a non-zero component along the surface normal.

stability is possible. Harada et al. (Harada et al., 2006b) considered the CHCP as ZMP support volume when the robot makes two feet contact with a horizontal floor and hand contacts with the environment. They detailed how to project the support volume on the floor to obtain a ZMP support area. While their construction applies to general dynamic wrenches with non-zero angular momenta, like all approaches based on convex hull of contact points, it assumes infinite friction coefficients (see Appendix 6.4.2 for details). In our work, we will construct ZMP support areas that also take friction into account.

A parallel line of research kept the ZMP in a plane but relaxed the constraint that it coincides with the ground (Kagami et al., 2002; Sugihara et al., 2002; Shibuya et al., 2006; Sato et al., 2011). Kagami et al. (Kagami et al., 2002) had the insight that the ZMP could be taken relative to any plane normal, however they assumed that this plane should still pass through all contact points, which restricted their scope to a maximum of three arbitrary contacts points. Sugihara et al. (Sugihara et al., 2002) introduced the notion of “Virtual Horizontal Plane” (VHP) in which contact points are projected on a virtual plane via the line connecting them to the COM, and the convex hull of these points is then taken as ZMP support area. This construction is only valid in the linear pendulum model. Shibuya et al. (Shibuya et al., 2006) took the idea further by considering a virtual plane *above* the COM, resulting in marginally stable pendular dynamics rather than that of an unstable inverted pendulum. Sato et al. (Sato et al., 2011) applied the same idea to stair climbing. However, like CHCP, VHP support areas suppose infinite friction coefficients (see Appendix 6.4.2). The approach we propose now also relies on virtual plane and simple pendular dynamics, but we derive support areas also accounting for friction and establish that they are a necessary *and sufficient* condition for contact stability.

Breaking away from the notion of ZMP, another line of work has focused on building criteria that keep equivalence with full contact stability (Saida et al., 2003; Hirukawa et al., 2006; Qiu et al., 2011; Caron et al., 2015a). The seminal work by Saida et al. (Saida et al., 2003) impulsed a shift of paradigm from the ZMP to the gravito-inertial wrench. It also proposed to orient the virtual plane orthogonally to the resultant force, which reinstates the support area as convex hull of (projected) contact points – an idea that may have been overlooked by the literature so far. Next, Hirukawa et al. (Hirukawa et al., 2006) constructed the first

full stability criterion for the gravito-inertial wrench. However their construction was high-dimensional as it relied on the full vector of stacked contact forces. Later works (Qiu et al., 2011; Escande et al., 2013; Caron et al., 2015a) reduced these stacked variables to the gravito-inertial wrench itself using the double-description method. Compared to traditional ZMP solutions, this approach has the benefit of providing a full stability criterion, but at the cost of the non-linear dynamics of the gravito-inertial wrench. Furthermore, the nice geometric construction of the ZMP area is replaced by a considerably less intuitive six-dimensional cone. The method that we introduce reconciles full stability, linear-pendulum dynamics and a geometric support area.

Control. When it comes to control, the use of the ZMP is historically tied to the LIP model, which was introduced in (Kajita et al., 2001; Sugihara et al., 2002) and used in a wealth of subsequent research works (Kajita et al., 2003; Mitobe et al., 2004; Morisawa et al., 2005; Shibuya et al., 2006; Harada et al., 2006a; Harada et al., 2006b; Sato et al., 2011; Morisawa et al., 2014; Engelsberger et al., 2015; Tedrake et al., 2015). Kajita et al. (Kajita et al., 2003) brought in the technique of model predictive control as a way to generate COM trajectories from desired ZMP positions. Harada et al. (Harada et al., 2006a) proposed an analytical alternative with polynomial solutions for the coupled COM-ZMP trajectories. Recently, Tedrake et al. (Tedrake et al., 2015) exhibited a closed-form solution for the linear-quadratic regulator tracking a reference ZMP. However, these methods only apply to LIPs on horizontal floors.

Aiming for locomotion on rough terrains, Morisawa et al. (Morisawa et al., 2005) relaxed the planar constraint on the COM to arbitrary two-dimensional manifolds, while recent papers (Morisawa et al., 2014; Engelsberger et al., 2015) chose to control the Capture Point (CP) rather than the ZMP. As a control variable, the CP stabilizes the unstable dynamics of the LIP. In terms of support areas, though, the question is the same for the CP and the ZMP and was not addressed by these developments.² With the method proposed in this paper, we realize a control system with marginally stable dynamics, while at the same time deriving the support area corresponding to our control variable.

²On a related note, we observe that, to the best of our knowledge, none of the previous works using LIP found in the literature noticed that the ZMP area was shrunk by the LIP assumptions on the angular momentum and COM height.

As with stability criteria, solutions breaking away from control points were also explored in the whole-body control literature. The main alternative is to regulate contact forces directly, resulting in force distribution schemes where desired contact forces and torques are tracked by a whole-body controller (Hirukawa et al., 2006; Hyon et al., 2007; Lee and Goswami, 2010; Ott et al., 2011; Righetti et al., 2013). Force objectives can express whole-body tasks, such as tracking of desired COM or angular momentum, as well as local ones, such as minimizing friction forces (Ott et al., 2011) or end-effector torques (Lee and Goswami, 2010). Righetti et al. (Righetti et al., 2013) characterized force-distribution controllers for linear-quadratic objectives in the absence of inequality constraints. Overall, force distribution schemes yield fast computations and can cope with arbitrary contact conditions, but they lack the foresight of methods based on control points and support areas. Indeed, for locomotion, support areas provide both reachable COM locations and a stability margin (the point-to-boundary distance). Finding such indicators in the high-dimensional contact-force space is still elusive. In recent developments, (Nagasaka et al., 2012; Audren et al., 2014) added a level of foresight to their contact-force controllers via model-predictive control, while Zheng et al. (Zheng and Yamane, 2015) constructed a metric that can be used as wrench-space stability margin. We will now show that support areas can be derived in arbitrary multi-contact configurations as well, providing both COM reachability and stability margins suitable for locomotion.

6.2 ZMP support areas under frictional constraints

6.2.1 Linearized wrench cones

We pursue the framework outlined in Chapter 5. However, contrary to the previous derivation where we used a single wrench at each contact, we now suppose a set of contact points and their complementary contact forces. Let us denote by \mathbf{f}_{ij} the ray vectors of the linearized friction cones at each individual contact point. Friction constraints are then written:

$$\mathbf{f}_i^c = \sum_{\text{ray } j} \lambda_{ij} \mathbf{f}_{ij}, \quad \lambda_{ij} \geq 0. \quad (6.1)$$

The set of ray vectors $\{\mathbf{f}_{ij}\}$ can be computed directly from the contact frame and friction coefficient μ_i . For example, the expression of a four-sided pyramid is $\{\mathbf{n}_i \pm \frac{\mu_i}{\sqrt{2}} \mathbf{t}_i \pm \frac{\mu_i}{\sqrt{2}} \mathbf{b}_i\}$, with $(\mathbf{t}_i, \mathbf{b}_i, \mathbf{n}_i)$ the full orthonormal contact frame. Injecting the span combinations (6.1) into eq. (5.7) yields a span representation for the contact wrench cone:

$$\begin{bmatrix} \mathbf{f}^c \\ \boldsymbol{\tau}_O^c \end{bmatrix} = \sum_{i,j} \lambda_{ij} \begin{bmatrix} \mathbf{f}_{ij} \\ \overrightarrow{OC_i} \times \mathbf{f}_{ij} \end{bmatrix} \quad \lambda_{ij} \geq 0.$$

Let us define $\boldsymbol{\tau}_{O,ij} = \overrightarrow{OC_i} \times \mathbf{f}_{ij}$. After re-indexing the couples (i, j) into a single index i (counting the same contact point C_i multiple times accordingly), we get:

$$\begin{bmatrix} \mathbf{f}^c \\ \boldsymbol{\tau}_O^c \end{bmatrix} = \sum_i \lambda_i \begin{bmatrix} \mathbf{f}_i \\ \boldsymbol{\tau}_{O,i} \end{bmatrix}, \quad \lambda_i \geq 0. \quad (6.2)$$

We have thus obtained the span representation of the contact wrench cone. A motion is then dynamically stable if and only if its contact wrench can be written as (6.2) for a certain set of coefficients $\lambda_i \geq 0$.

6.2.2 Zero-tilting Moment Point

Let O denote a fixed reference point in the absolute frame, not necessarily on the floor. Let \mathbf{n} be a fixed unit space vector, not necessarily vertical. We are interested in computing the ZMP in the plane that contains O and that is orthogonal to \mathbf{n} , hereafter denoted by $\Pi(O, \mathbf{n})$. Note that, in the original horizontal-floor setting, O belongs to the ground plane and \mathbf{n} is vertical.

We follow the footsteps of Sardain and Bessonnet (Sardain and Bessonnet, 2004) in considering the *non-central axis* $\Delta_w(\mathbf{n})$ where the moment of a wrench \mathbf{w} is parallel to \mathbf{n} (note that (Sardain and Bessonnet, 2004) assumed that \mathbf{n} is vertical).

Definition 8 (Zero-tilting Moment Point of a wrench in a plane). *The zero-tilting moment point of a wrench $\mathbf{w} = (\mathbf{f}, \boldsymbol{\tau})$ in the plane $\Pi(O, \mathbf{n})$ is the point $Z \in \Pi(O, \mathbf{n})$ such that $\mathbf{n} \times \boldsymbol{\tau}_Z = \mathbf{0}$.*

We consider the ZMP Z of the contact (equivalently, gravito-inertial) wrench with respect to an arbitrary plane. Its defining equation $\mathbf{n} \times \boldsymbol{\tau}_Z^c = \mathbf{0}$ can be rewritten as

$$\mathbf{n} \times (\overrightarrow{ZO} \times \mathbf{f}^c) + \mathbf{n} \times \boldsymbol{\tau}_O^c = \mathbf{0}. \quad (6.3)$$

The first term of this equation expands to $(\mathbf{n} \cdot \mathbf{f}^c) \overrightarrow{ZO} - (\mathbf{n} \cdot \overrightarrow{ZO}) \mathbf{f}^c$, but since $\mathbf{n} \cdot \overrightarrow{ZO} = 0$ we have:

$$\mathbf{p}_Z = \frac{\mathbf{n} \times \boldsymbol{\tau}_O^c}{\mathbf{n} \cdot \mathbf{f}^c} + \mathbf{p}_O. \quad (6.4)$$

Note how, because of the ratio in the formula above, both the contact and gravito-inertial wrench define the same ZMP \mathbf{p}_Z . Also, when $\mathbf{n} \cdot \mathbf{f}^c = 0$, eq. (6.4) has no solution and cannot be used to define a point Z . This singularity is present in the horizontal-floor setting as well, where a horizontal resultant \mathbf{f}^{gi} yields a division by zero in the formula of the ZMP (Sardain and Bessonnet, 2004).

6.2.3 Construction of the support area

Equation (6.4) shows how the ZMP is a two-dimensional affine projection of the gravito-inertial wrench. Since contact stability is characterized by the GIWC, we define the support area as the set of ZMPs corresponding to realizable contact wrenches.

Definition 9. *The support area S of the ZMP in the plane $\Pi(O, \mathbf{n})$ is the image of the GIWC by the projection (6.4).*

The key idea to calculate this area is to use the span representation (6.2) of the CWC, which enables rewriting eq. (6.4) as

$$\mathbf{p}_Z = \frac{\sum_i \lambda_i (\mathbf{n} \times \boldsymbol{\tau}_{O,i})}{\sum_i \lambda_i (\mathbf{n} \cdot \mathbf{f}_i)} + \mathbf{p}_O, \quad \lambda_i \geq 0. \quad (6.5)$$

Next, define

$$\overrightarrow{OZ}_i := \frac{\mathbf{n} \times \boldsymbol{\tau}_{O,i}}{\mathbf{n} \cdot \mathbf{f}_i} = \frac{\mathbf{n} \times (\overrightarrow{OC}_i \times \mathbf{f}_i)}{\mathbf{n} \cdot \mathbf{f}_i}. \quad (6.6)$$

Denote by $p_i := (\mathbf{n} \cdot \mathbf{f}_i)$ the *virtual pressure* of the contact force generator \mathbf{f}_i through the virtual plane. Then,

$$\mathbf{p}_Z = \frac{\sum_i \lambda_i p_i \mathbf{p}_{Z_i}}{\sum_i \lambda_i p_i}, \quad \lambda_i \geq 0. \quad (6.7)$$

Note that, on a horizontal floor, all contact forces \mathbf{f}_i point upwards, \mathbf{n} is vertical and points upward, in such a way that $\lambda_i p_i > 0$ for all i . The point Z is then a convex combination of the Z_i 's. Furthermore, eq. (6.6) simplifies to $Z_i = C_i$, i.e., the vertices of the support area \mathcal{S} coincide with the contact points, themselves taken as the vertices of the contact polygon. We see here that our definition of the support area agrees with the horizontal-floor setting.

In general, however, virtual pressures p_i can be either positive or negative.³ Let us then partition the set of generator indices I into $I^+ := \{i \mid p_i > 0\}$ and $I^- := \{i \mid p_i < 0\}$. For any $S \subset I$, denote by $\sigma(S) := \sum_{i \in S} \lambda_i |p_i|$ and define

$$\begin{aligned} \alpha_i &:= \frac{+\lambda_i p_i}{\sigma(I^+)} \text{ for } i \in I^+, & \alpha &:= \frac{\sigma(I^+)}{\sigma(I)}. \\ \beta_i &:= \frac{-\lambda_i p_i}{\sigma(I^-)} \text{ for } i \in I^-, & \beta &:= \frac{\sigma(I^-)}{\sigma(I)}. \end{aligned}$$

Equation (6.5) becomes

$$\mathbf{p}_Z = \frac{1}{\alpha - \beta} \left[\alpha \sum_{i \in I^+} \alpha_i \mathbf{p}_{Z_i} - \beta \sum_{i \in I^-} \beta_i \mathbf{p}_{Z_i} \right].$$

Define the *positive-pressure polygon* as the convex hull of Z_i 's for $i \in I^+$: $\mathcal{P}^+ := \{\sum_{i \in I^+} \alpha_i \mathbf{p}_{Z_i}, \alpha_i \geq 0, \sum_i \alpha_i = 1\}$, and define the *negative-pressure polygon* \mathcal{P}^- *mutatis mutandis*. If one of these two polygons is empty, Z simply belongs to the other. Otherwise, the above expression can be rewritten as

$$\mathbf{p}_Z = \frac{\alpha \mathbf{p}_{Z^+} - \beta \mathbf{p}_{Z^-}}{\alpha - \beta}, \quad (6.8)$$

where $\alpha \geq 0, \beta \geq 0, Z^+ \in \mathcal{P}^+$ and $Z^- \in \mathcal{P}^-$.

We can now characterize the support area.

³We assume \mathbf{n} is chosen so that none of them is zero, which is easy to do since there is only a finite set of generators $\{\mathbf{f}_i\}$.

Proposition 8. *If one of the two polygons \mathcal{P}^+ or \mathcal{P}^- is empty, \mathcal{S} is equal to the other. Otherwise, Let $\mathcal{D} = \mathcal{P}^+ - \mathcal{P}^- = \text{conv}(\{\mathbf{r}_1, \dots, \mathbf{r}_k\})$ denote the vertices of the Minkowski difference of the two convex polygons. Then, the support area \mathcal{S} is the reunion of the two polygonal cones \mathcal{C}^+ and \mathcal{C}^- given by*

$$\begin{aligned}\mathcal{C}^+ &= \mathcal{P}^+ + \sum_i \mathbb{R}^+ \mathbf{r}_i, \\ \mathcal{C}^- &= \mathcal{P}^- + \sum_i \mathbb{R}^+ (-\mathbf{r}_i).\end{aligned}$$

In particular, when $\mathcal{P}^+ \cap \mathcal{P}^-$ has non-empty interior, \mathcal{S} covers the whole plane $\Pi(O, \mathbf{n})$.

Proof. eq. (6.8) can be reformulated as

$$\mathbf{p}_Z = \mathbf{p}_{Z^+} + \frac{\beta}{\alpha - \beta} \overrightarrow{Z^- Z^+} = \mathbf{p}_{Z^-} + \frac{\alpha}{\beta - \alpha} \overrightarrow{Z^+ Z^-}$$

Therefore, the set of points Z defined by this equation is

$$\begin{aligned}\mathcal{S} &:= \left\{ \mathbf{p}_{Z^+} + \frac{\beta}{\alpha - \beta} \overrightarrow{Z^- Z^+}, \alpha \geq \beta \geq 0, Z^\pm \in \mathcal{P}^\pm \right\} \\ &\cup \left\{ \mathbf{p}_{Z^-} + \frac{\alpha}{\beta - \alpha} \overrightarrow{Z^+ Z^-}, \beta \geq \alpha \geq 0, Z^\pm \in \mathcal{P}^\pm \right\}.\end{aligned}$$

Given the orderings of α and β , we can further simplify the ratios into a single positive scalar, so that $\mathcal{S} = \mathcal{C}^+ \cup \mathcal{C}^-$ with

$$\mathcal{C}^+ = \left\{ \mathbf{p}_{Z^+} + \lambda \overrightarrow{Z^- Z^+}, \lambda \geq 0, Z^\pm \in \mathcal{P}^\pm \right\}, \quad (6.9)$$

$$\mathcal{C}^- = \left\{ \mathbf{p}_{Z^-} + \lambda \overrightarrow{Z^+ Z^-}, \lambda \geq 0, Z^\pm \in \mathcal{P}^\pm \right\}. \quad (6.10)$$

The set $\mathcal{D} = \mathcal{P}^+ - \mathcal{P}^-$ is a convex polygon as Minkowski difference of two convex polygons. To conclude, we show that $\mathcal{C}^+ = \mathcal{P}^+ + \mathbb{R}^+ \mathcal{D}$. The inclusion \subset is straightforward from (6.9). Now, let $\mathbf{p}_C = \mathbf{p}_{Z_0^+} + \mu(\mathbf{p}_{Z_1^+} - \mathbf{p}_{Z_1^-})$ denote any point in $\mathcal{P}^+ + \mathbb{R}^+ \mathcal{D}$. Define

$$\mathbf{p}_{Z^+} := \frac{1}{1 + \mu} \mathbf{p}_{Z_0^+} + \frac{\mu}{1 + \mu} \mathbf{p}_{Z_1^+}.$$

One can check that $\mathbf{p}_C = \mathbf{p}_{Z^+} + \mu(\mathbf{p}_{Z^+} - \mathbf{p}_{Z^-})$, where Z^+ belongs to \mathcal{P}^+ as convex combination of two points from this convex polygon. Thus $C \in \mathcal{C}^+$, which estab-

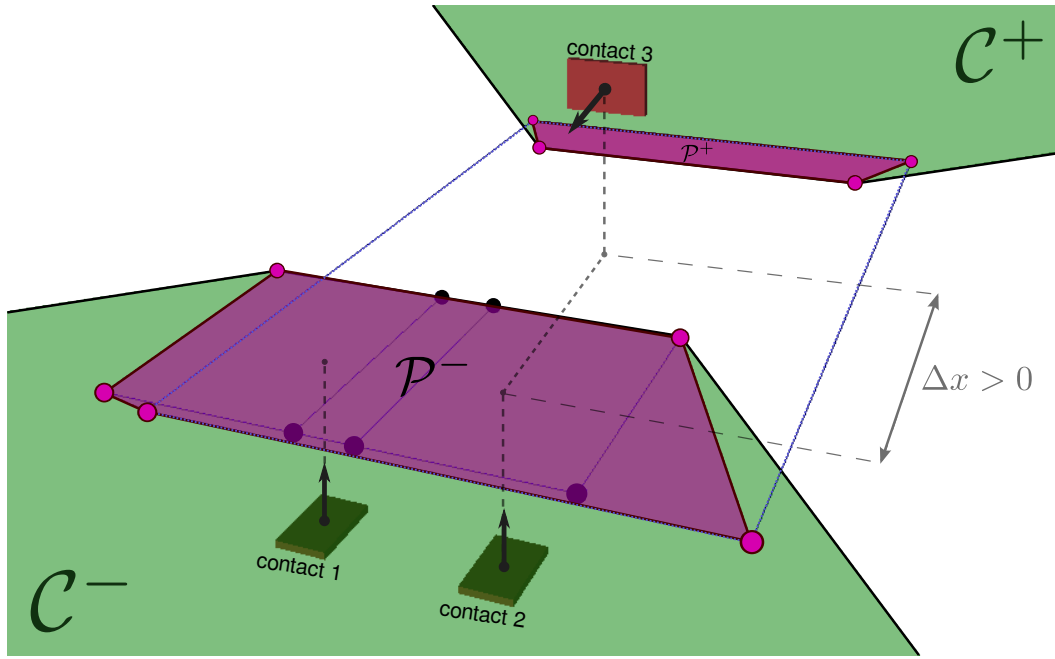


Fig. 6.2.: Example of computed support area in the dual-cone case. There are three contacts in total. Contacts 1 and 2 are made with the same horizontal surface and correspond to the feet of the robot. Contact 3 is located one meter above the others and made with a vertical surface in front of the robot. The virtual plane is taken horizontal and 50 cm above the feet's plane. Polygons \mathcal{P}^+ and \mathcal{P}^- used in the construction process are drawn in purple. Blue lines connect the vertices generated by contact 3. They illustrate that a single surface contact can yield vertices in both \mathcal{P}^+ and \mathcal{P}^- at the same time.

lishes the converse inclusion \supset . Finally, note how, when $\mathcal{P}^+ \cap \mathcal{P}^-$ has non-empty interior, \mathcal{D} contains a neighborhood of the origin and $\mathcal{C}^+ = \mathcal{P}^+ + \mathbb{R}^+ \mathcal{D}$ becomes the whole plane. \square

When both pressure polygons are non-empty, Proposition 8 provides the V-representation of the two polygonal cones. The dual (half-space or H-)representation of these cones provides the boundary segments and half-lines of the support area. General techniques such as the *double description method* (Fukuda and Prodon, 1996) can be employed here to convert from one representation to the other. However, since the problem at hand is two-dimensional, one can also use the following simple algorithm.

In 2D, the Minkowski difference has only two extreme rays $(\mathbf{r}_1, \mathbf{r}_2)$ which are easy to extract in linear time. Next, suppose w.l.o.g. that $\mathbf{r}_1 \times \mathbf{r}_2 > 0$, and denote by $\mathbf{v}_1 := \arg \min_i \mathbf{r}_1 \times \mathbf{v}_i$ and $\mathbf{v}_2 := \arg \max_i \mathbf{r}_2 \times \mathbf{v}_i$. Half-lines of the H-representation

are given by $\mathbf{v}_1 + \mathbb{R}^+ \mathbf{r}_1$, $\mathbf{v}_2 + \mathbb{R}^+$ and all line segments (in clockwise ordering of the convex hull) from \mathbf{v}_1 to \mathbf{v}_2 . See the implementation in *pymanoid*⁴ for details.

6.2.4 Geometric properties of the support area

We now provide a geometric formulation of the support area freed from the choice of the arbitrary reference point O used in the construction. A first thing to notice is that the only coordinate of O influencing \mathcal{S} is its position along the non-central axis $\Delta^c(\mathbf{n})$ of the wrench.

Proposition 9. *The support area \mathcal{S} does not depend on the coordinates of the reference point O in the plane $\Pi(O, \mathbf{n})$.*

Proof. By definition of the support area,

$$\mathcal{S} = \left\{ Z \in \Pi(O, \mathbf{n}) : \overrightarrow{OZ} = \frac{\mathbf{n} \times \boldsymbol{\tau}_O^c}{\mathbf{n} \cdot \mathbf{f}^c} \right\} \quad (6.11)$$

where $(\mathbf{f}^c, \boldsymbol{\tau}^c) \in \mathcal{C}^c$ ranges over the CWC. Now, choose a point $O' \in \Pi(O, \mathbf{n})$ and consider


$$\mathcal{S}' = \left\{ Z' \in \Pi(O, \mathbf{n}) = \Pi(O', \mathbf{n}) : \overrightarrow{O'Z'} = \frac{\mathbf{n} \times \boldsymbol{\tau}_{O'}^c}{\mathbf{n} \cdot \mathbf{f}^c} \right\}. \quad (6.12)$$

Consider one particular wrench $(\mathbf{f}^c, \boldsymbol{\tau}_O^c)$ and Z and Z' defined by the equalities inside (6.11) and (6.12). We have

$$\begin{aligned} \overrightarrow{O'Z'} &= \frac{\mathbf{n} \times \boldsymbol{\tau}_{O'}^c}{\mathbf{n} \cdot \mathbf{f}^c} = \frac{\mathbf{n} \times (\overrightarrow{O'O} \times \mathbf{f}^c) + \mathbf{n} \times \boldsymbol{\tau}_O^c}{\mathbf{n} \cdot \mathbf{f}^c} \\ &= \overrightarrow{O'O} - (\mathbf{n} \cdot \overrightarrow{O'O}) \frac{\mathbf{f}^c}{\mathbf{n} \cdot \mathbf{f}^c} + \overrightarrow{OZ} = \overrightarrow{O'Z}, \end{aligned}$$

Thus, $Z' = Z$, and since the wrench $(\mathbf{f}^c, \boldsymbol{\tau}^c) \in \mathcal{C}^c$ we considered is arbitrary, we have shown that $\mathcal{S} = \mathcal{S}'$. □

By contrast, the support area does change for displacements of O along the non-central axis $\Delta^c(\mathbf{n})$. Let us analyze the impact of this remaining coordinate by relax-

⁴ <https://github.com/stephane-caron/pymanoid>

ing the assumption $(\overrightarrow{OZ} \cdot \mathbf{n}) = 0$ into $(\overrightarrow{OZ} \cdot \mathbf{n}) = d_Z$, where d_Z is the axial coordinate of the virtual plane $\Pi(d_Z, \mathbf{n})$. The definition $\mathbf{n} \times \boldsymbol{\tau}_Z^c = \mathbf{0}$ expands to

$$\begin{aligned}\mathbf{n} \times \overrightarrow{OZ} \times \mathbf{f}^c &= \mathbf{n} \times \boldsymbol{\tau}_O^c \\ (\mathbf{n} \cdot \mathbf{f}^c) \overrightarrow{OZ} - d_Z \mathbf{f}^c &= \mathbf{n} \times \boldsymbol{\tau}_O^c,\end{aligned}$$

so that

$$\mathbf{p}_Z = \frac{\mathbf{n} \times \boldsymbol{\tau}_O^c}{\mathbf{n} \cdot \mathbf{f}^c} + \mathbf{p}_O + d_Z \frac{\mathbf{f}^c}{\mathbf{n} \cdot \mathbf{f}^c}. \quad (6.13)$$

Repeating the step from eq. (6.6),

$$\begin{aligned}\overrightarrow{OZ}_i &:= \frac{\mathbf{n} \times \boldsymbol{\tau}_{O,i}}{\mathbf{n} \cdot \mathbf{f}_i} + d_Z \frac{\mathbf{f}_i}{\mathbf{n} \cdot \mathbf{f}_i}, \\ \mathbf{p}_{Z_i} &= \mathbf{p}_{C_i} + (d_Z - d_i) \frac{\mathbf{f}_i}{\mathbf{n} \cdot \mathbf{f}_i}.\end{aligned} \quad (6.14)$$

We thus obtain the same equation (6.7), but this time the vertices Z_i of the support area have a different axial coordinate d_Z . It also appears from eq. (6.14) that Z_i is the intersection between the plane $\Pi(d_Z, \mathbf{n})$ and the line passing through C_i and directed by \mathbf{f}_i , the latter being a ray of the linearized friction cone. We have thus established that:

Property 7. *The vertices of the support area are located at the intersection between the virtual plane and the rays of the friction cones.*

This property gives a first geometric interpretation of the support area, yet it only provides its vertices. We saw in Prop. 8 how reconstructing the area from these vertices is not straightforward. Still, when a suitable plane orientation \mathbf{n} can be found so that \mathcal{S} is a polygon, we get a simple geometric characterization.

Corollary 2 (Geometric characterization of the support area). *When all virtual pressures $p_i := (\mathbf{n} \cdot \mathbf{f}_i)$ are positive, the support area is the convex hull of the intersection between linearized friction cones and the virtual plane.*

This result is coherent with the horizontal-floor setting where the virtual plane intersects friction cones at their apexes, *i.e.*, at contact points, and virtual pressures are all positive from contact unilaterality. A condition similar to the positivity of virtual pressures was also observed by Saida et al. (property 4.1 in (Saida et al., 2003)) for plane components of the gravito-inertial wrench.

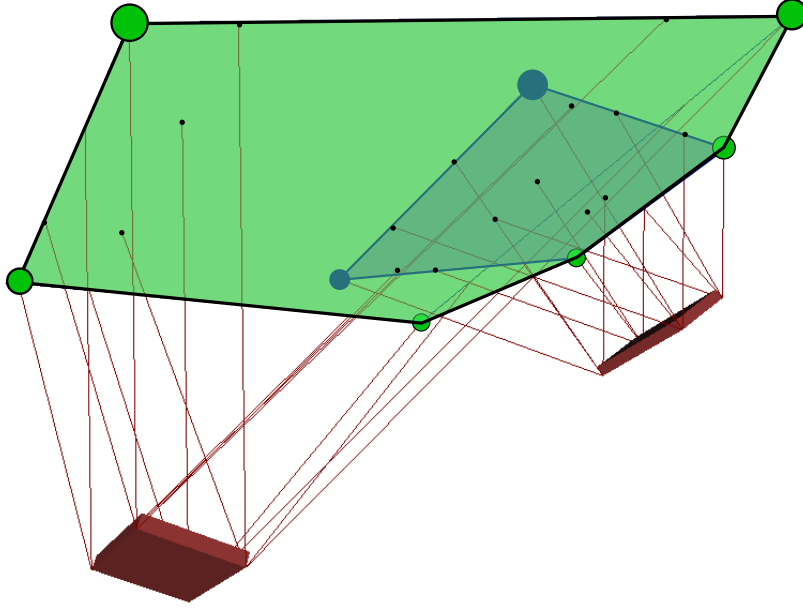


Fig. 6.3.: Geometric construction of the ZMP support area in the polygonal case. Ray generators of friction cones (red lines) are traced until they intersect the virtual plane, yielding points inside the support area (black dots). The support area (in green) is the convex hull of this set of points. The blue polygon corresponds to the individual support polygon of the right contact. These polygons can be used for an alternative construction of the ZMP support area suited to the problem of contact planning.

Figure 6.3 illustrates the geometric construction. The support area (in green) is the convex hull of the set of black points projected from friction rays (red lines). Alternatively, each individual contact surface projects its own support polygon (in blue for the right contact), and the ZMP support area is the convex hull of these individual polygons. This second construction is useful for contact planning as it provides a valuation criterion for new contacts, namely the expansion that they bring to the support area.

Finally, let us determine in which cases a polygonal ZMP support area can be found, which boils down to finding a suitable vector \mathbf{n} . Let \mathcal{C}_f denote the cone positively spanned by the \mathbf{f}_i 's. Then, the condition $\forall i, (\mathbf{n} \cdot \mathbf{f}_i) > 0$ is equivalent to $\mathbf{n} \in \mathcal{C}_f^*$, where \mathcal{C}_f^* is the *dual cone* of \mathcal{C}_f defined by

$$\mathcal{C}_f^* = \{\mathbf{y} : \forall \mathbf{x} \in \mathcal{C}_f, \mathbf{y} \cdot \mathbf{x} \geq 0\}.$$

The set of solutions \mathcal{C}_f^* can be computed from \mathcal{C}_f using *e.g.*, the double description method. In particular, $\mathcal{C}_f^* = \{\mathbf{0}\}$ if and only if $\mathcal{C}_f = \mathbb{R}^3$, *i.e.*, the \mathbf{f}_i 's positively span

the whole space. Thus, either contact forces can generate any arbitrary resultant force⁵ or one can find an orientation \mathbf{n} such that the support area is polygonal.

6.2.5 Generality of the concept

Although historically the ZMP has been mostly used in locomotion, the support areas constructed in this section are defined for arbitrary contact wrenches. As such, they also apply to related fields dealing with *mobility* under frictional constraints, such as grasping or workpiece fixturing.

As a mathematical object, support areas are 2D (non-linear) projections of the 6D contact wrench cone. From there, one could question the generality of this construction: is it the “best” we can do? Could there exist a 3D projection that would account for all three components of the resultant moment, rather than only two? For the interested reader, we provide some elements of answer to these questions in Appendix 6.4.3.

6.3 Multi-contact locomotion with pendulum control

6.3.1 Relationship between ZMP and COM

From a control point of view, the interest of the ZMP lies in its relationship with the acceleration of the COM.

Proposition 10. *The position of the ZMP relative to the COM is bound to the dynamic momentum by*

$$\ddot{\mathbf{p}}_G = \mathbf{g} + \frac{\gamma + \ddot{d}_G}{d_G - d_Z} \overrightarrow{ZG} + \frac{\mathbf{n} \times \dot{\mathbf{L}}_G}{m(d_G - d_Z)}, \quad (6.15)$$

where \mathbf{g} is the gravity vector. The constant $\gamma := -(\mathbf{n} \cdot \mathbf{g})$ coincides with the gravity constant g when \mathbf{n} is upward vertical.

⁵this condition is weaker than *force closure*, which usually assumes that contact forces can generate arbitrary resultant forces and torques

Proof. By definition of the gravito-inertial wrench,

$$\begin{aligned}\mathbf{f}^{gi} &= m(\mathbf{g} - \ddot{\mathbf{p}}_G), \\ (\mathbf{n} \cdot \mathbf{f}^{gi}) &= -m(\gamma + \ddot{d}_G).\end{aligned}$$

Expanding $\tau_O^{gi} = \overrightarrow{OG} \times \mathbf{f}^{gi} - \dot{\mathbf{L}}_G$ in eq. (6.13) yields

$$\begin{aligned}(\mathbf{n} \cdot \mathbf{f}^{gi})\overrightarrow{OZ} &= \mathbf{n} \times \overrightarrow{OG} \times \mathbf{f}^{gi} - \mathbf{n} \times \dot{\mathbf{L}}_G + d_Z \mathbf{f}^{gi} \\ (\mathbf{n} \cdot \mathbf{f}^{gi})\overrightarrow{GZ} &= (d_Z - d_G) \mathbf{f}^{gi} - \mathbf{n} \times \dot{\mathbf{L}}_G \\ m(\gamma + \ddot{d}_G)\overrightarrow{ZG} &= m(d_Z - d_G)(\mathbf{g} - \ddot{\mathbf{p}}_G) - \mathbf{n} \times \dot{\mathbf{L}}_G.\end{aligned}\quad (6.16)$$

Equation (6.15) is a rearrangement of the latter. \square

We saw in Section 6.2 how to construct support areas in virtual planes of arbitrary axial coordinate d_Z . We now establish that, from a control perspective, all planes on the same “side” of the COM ($d_Z > d_G$ or $d_Z < d_G$) are equivalent.

Proposition 11. *Let $Z \in \Pi(d_Z, \mathbf{n})$ denote the ZMP resulting from a given wrench by eq. (6.13). Then, eq. (6.15) yields the same COM acceleration regardless of the plane coordinate d_Z .*

Proof. Denote by $(\mathbf{f}^{gi}, \tau_O^{gi})$ the gravito-inertial wrench generating Z by (6.13). Let us differentiate (6.15) with respect to the parameter d_Z :

$$\frac{\partial \ddot{\mathbf{p}}_G}{\partial d_Z} = \frac{\gamma + \ddot{d}_G}{d_G - d_Z} \frac{\partial \overrightarrow{ZG}}{\partial d_Z} - \frac{[(\gamma + \ddot{d}_G)\overrightarrow{ZG} + \frac{\mathbf{n} \times \dot{\mathbf{L}}_G}{m}]}{(d_G - d_Z)^2}.$$

From (6.13), we have $\partial \overrightarrow{ZG} / \partial d_Z = -\mathbf{f}^{gi} / m(\gamma + \ddot{d}_G)$. Meanwhile, the factor between brackets can be replaced by (6.16), yielding:

$$\frac{\partial \ddot{\mathbf{p}}_G}{\partial d_Z} = \frac{-\mathbf{f}^{gi}}{m(d_G - d_Z)} - \frac{(d_Z - d_G)\mathbf{f}^{gi}}{m(d_G - d_Z)^2} = \mathbf{0}.\quad \square$$

Note that this proposition does *not* mean that two ZMPs with the same planar coordinates but in different planes yield the same COM acceleration. Equivalent ZMPs are aligned on the axis $\Delta^{gi}(\mathbf{n})$ directed by the resultant force \mathbf{f}^{gi} , and thus have different coordinates in different planes in general.

6.3.2 Linear pendulum models: LP and LIP

Taking \mathbf{n} as the upward vertical vector of the world frame $\mathbf{n} = \mathbf{e}_Z$, eq. (6.15) can be rewritten as

$$\ddot{x}_G = \frac{g + \ddot{z}_G}{z_G - z_Z}(x_G - x_Z) - \frac{\dot{L}_{Gy}}{m(z_G - z_Z)} \quad (6.17)$$

$$\ddot{y}_G = \frac{g + \ddot{z}_G}{z_G - z_Z}(y_G - y_Z) + \frac{\dot{L}_{Gx}}{m(z_G - z_Z)} \quad (6.18)$$

When the angular momentum is zero and $z_Z < z_G$, these expressions reduce to the well-known Linear Inverted Pendulum (LIP) model (Kajita et al., 2001; Sugihara et al., 2002):

$$\begin{cases} \ddot{x}_G = -\omega_{\text{IP}}^2(x_Z - x_G) \\ \ddot{y}_G = -\omega_{\text{IP}}^2(y_Z - y_G) \end{cases} \quad \omega_{\text{IP}} := \sqrt{\frac{g + \ddot{z}_G}{z_G - z_Z}} \Big|_{z_Z < z_G}$$

The assumption that $z_Z < z_G$ was taken for granted in previous works as the ZMP was supposed to lie on the ground. But our analysis now allows us to select the other side of the domain identified in Proposition 11, that is to say $z_Z > z_G$. In this new setting, equations (6.17)-(6.18) reduce to a Linear Pendulum (LP):

$$\begin{cases} \ddot{x}_G = \omega_{\text{P}}^2(x_Z - x_G) \\ \ddot{y}_G = \omega_{\text{P}}^2(y_Z - y_G) \end{cases} \quad \omega_{\text{P}} := \sqrt{\frac{g + \ddot{z}_G}{z_Z - z_G}} \Big|_{z_Z > z_G} \quad (6.19)$$

The key difference between the two approaches lies in the stability of the control law. The equilibrium position of an LIP is unstable in the sense that the COM is always diverging away from the ZMP. Consequently, to regulate the COM around a fixed position, the ZMP of an LIP needs to be constantly in motion. On the contrary, the equilibrium position of an LP is stable, with the COM always moving towards the ZMP. Taking the ZMP *above* the COM rather than below thus directly results in a stable⁶ control law, with no need for intermediate stabilization variables such as the Capture Point.

⁶ Since there is no damping term in eq. (6.19), the precise stability property is marginal stability, *i.e.*, the COM is either *at* or *orbiting* around the ZMP. Convergence can be added by *e.g.*, leveraging the angular momentum to generate a damping term.

6.3.3 ZMP support areas under LP regulation

In general, keeping the ZMP in the support area does not provide full contact stability as the ZMP only represents two out of the six wrench coordinates (the four remaining coordinates have constraints that ought to be checked as well). However, it becomes a full stability condition when the four other coordinates are regulated by $\ddot{z}_G = 0$ and $\dot{\mathbf{L}}_G = \mathbf{0}$, which yields LP dynamics. In this regime, Equations (6.17)-(6.18) become a one-to-one mapping between ZMP and dynamic-momentum coordinates. Adding these regulations consequently reduces the support area to a smaller polygon or cone, which we will call the *LP support area* for short.⁷ It turns out that the computation of this set by the double-description method is very similar to that of the COM static-stability polygon, which we gave in Section 5.3.1.

Suppose that p_G is known. The four dynamic-wrench equations for the LP regime become, for the gravito-inertial wrench,

$$\begin{bmatrix} \mathbf{n}^\top & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} = \begin{bmatrix} -mg \\ \overrightarrow{OG} \times \mathbf{f}^{gi} \end{bmatrix}.$$

Again, expanding the triple product $(\mathbf{n} \times \overrightarrow{OG} \times \mathbf{f}^{gi})$ in the expressions above, one can rewrite them equivalently as

$$\begin{bmatrix} z_G \mathbf{I}_3 & [\mathbf{n} \times] \\ (\mathbf{n} \times \overrightarrow{OG})^\top & -\mathbf{n}^\top \end{bmatrix} \begin{bmatrix} \mathbf{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} = \begin{bmatrix} -mg \overrightarrow{OG} \\ 0 \end{bmatrix}$$

In concise form: $\mathbf{A}' \mathbf{w}_O^{gi} = \mathbf{b}'$. The coordinates of the ZMP are given by eq. (6.13) as $p_Z = \mathbf{C}' \mathbf{w}_O^{gi} + \mathbf{d}'$. From there, the computation of the dynamic ZMP support area using the double description method is exactly the same as with static stability. We repeat it here to spare the reader some page-turning end-effector actuation.

Consider the stacked vector of contact forces $\mathbf{f}^{all} = [\mathbf{f}_1^c \dots \mathbf{f}_n^c]^\top$. Linearized friction cones are given in half-space representation by linear inequalities $\mathbf{F}_i \mathbf{f}_i^c \leq \mathbf{0}$. Combining all \mathbf{F}_i 's in a block diagonal matrix \mathbf{F} yields an inequality $\mathbf{F} \mathbf{f}^{all} \leq \mathbf{0}$. Meanwhile, eq. (5.7)-(5.8) provide a linear mapping $\mathbf{w}_O^{gi} = \mathbf{W}_O^{gi} \mathbf{f}^{all}$. Summing up,

⁷ Strictly speaking, our definition of an LP-regulated system includes $\dot{L}_{Gz} = 0$, which is not necessary to achieve linear pendulum dynamics.

the set of realizable contact forces in static stability is given in half-space representation by

$$\begin{aligned} \mathbf{F} \mathbf{f}^{all} &\leq \mathbf{0} \\ \mathbf{A}' \mathbf{W}_O^{g_i} \mathbf{f}^{all} &= \mathbf{b}' \end{aligned}$$

Using the double description method, one can compute the vertex representation of this set as $\mathbf{f}^{all} = \sum_i \lambda_i \mathbf{g}_i$ ($\lambda_i > 0$). The vertices of the ZMP stability polygon are finally given by $\mathbf{v}'_i = \mathbf{C}' \mathbf{g}_i + \mathbf{d}'$. Note that, contrary to static stability setting where the COM area is always a polygon, the LP support area for the ZMP may be either conical or polygonal. We observed both outcomes in simulations.

6.3.4 Trajectory generation for the LP model

We now design a trajectory generator for ZMP-COM trajectories based on the model-preview control formalism introduced in previous works (Kajita et al., 2003; Audren et al., 2014). We use the ZMP and COM as command and output variables respectively. First, we interpolate ZMP trajectories as line segments

$$\mathbf{p}_Z(t) = u(t) \mathbf{p}_0 + (1 - u(t)) \mathbf{p}_1.$$

The COM being constrained to a plane $\Pi(z_G, \mathbf{e}_Z)$ parallel to the one $\Pi(z_Z, \mathbf{e}_Z)$ of the ZMP, LP dynamics (6.19) yield:

$$\mathbf{p}_G(t) = v(t) \mathbf{p}_0 + (1 - v(t)) \mathbf{p}_1 + (z_G - z_Z) \mathbf{e}_Z,$$

where $\ddot{v} = \omega_P^2(u - v)$. Using line segments, the two-dimensional problem of controlling (x_G, y_G) from (x_Z, y_Z) is thus reduced to the one dimensional problem of controlling v from u .

We now define the state of our control problem by $\mathbf{x} = [v \ \dot{v}]^\top$ and its command by the linear position u of the ZMP. Discretizing the time interval into K steps of duration δt , the system's linear dynamics become

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \delta t \\ -\omega_P^2 \delta t & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0 \\ \omega_P^2 \delta t \end{bmatrix} u \quad (6.20)$$

Let $\mathbf{X} = [x_0^\top \cdots x_K^\top]^\top$ and $\mathbf{u} = [u_0 \cdots u_{K-1}]^\top$. Applying (6.20) repeatedly, we build the matrices Φ and Ψ such that $\mathbf{X} = \Phi \mathbf{x}_0 + \Psi \mathbf{u}$. We assume that the system starts with zero COM velocity, so that $\mathbf{x}_0 = \mathbf{0}$.

Finally, we formulate the trajectory generation problem as a Quadratic Program (QP) as follows:

$$\begin{aligned} \textbf{Objective:} \quad & \min \quad w_1 c_1(\mathbf{u}) + w_2 c_2(\mathbf{u}) \\ \textbf{Constraints:} \quad & \mathbf{0} \leq \mathbf{u} \leq \mathbf{1} \\ & \mathbf{x}_K = \Psi_{\text{last}} \mathbf{u} = [1 \ 0]^\top \\ & \mathbf{u}_{K-1} = \mathbf{1} \end{aligned}$$

The objective is the weighted sum of two terms:

$$\begin{aligned} c_1(\mathbf{u}) &= \frac{1}{K} \sum_k (v_k - u_k)^2 \\ c_2(\mathbf{u}) &= \sum_k (u_k - u_{k-1})^2 \end{aligned}$$

The first one minimizes COM accelerations while the second regularizes the ZMP trajectory. We chose the weights $w_1 = 1$ and $w_2 = 100$.

The constraints ensure respectively that:

1. the ZMP belongs to the line segment ($u(t) \in [0, 1]$)
2. the COM ends at the destination point ($v_K = 1$) with zero velocity ($\dot{v}_K = 0$),
3. the ZMP also ends at the destination point ($u_{K-1} = 1$).

With this method, maintaining dynamic stability is mostly enforced by including the segment $[p_0, p_1]$ inside the LP support area computed for the initial COM position $p_G(0)$. However, the COM will move as the robot performs the motion, which affects both the position and shape of the LP support area.

On a technical note, set aside the two regularization objectives, this optimization problems falls directly under the TOPP framework (c.f. Chapters 3 and 5). One could consequently trade smoothness of COM accelerations for Admissible Velocity

Propagation (AVP), allowing for a direct integration into a kinodynamic planner of COM trajectories.

6.3.5 Validation in simulations

We implemented the whole pipeline described so far to generate dynamically-stable multi-contact motions for a model of the HRP-4 humanoid robot. The scenario is depicted in Figure 6.4. The robot has to step on inclined platforms in order to reach its goal configuration on the right. Because there is no platform for its left foot in the middle of the course, the only way for it to complete the task is to use of the elevated “wall” platform with its left hand while keeping its right foot on the opposite tilted surface. Relying on these two simultaneous contacts, the humanoid can perform a long stride with its left leg, which would have been impossible to achieve in single-support.

As input given to solve this scenario, we assume that a contact planner provides a sequence of contact stances, where a stance provides both a reference position of the COM and a set of contact points. The first stage of our solution computes stance-to-stance COM trajectories. To move from stance i to stance $i + 1$, the controller considers the line segment $[\mathbf{p}_G^{(i)}, \mathbf{p}_G^{(i+1)}]$. The trajectory generator is called if this segment is included in the LP support area $\mathcal{S}(\mathbf{p}_G^{(i)}, z_Z)$ for the initial COM position. Otherwise, z_Z is increased until the segment is included in $\mathcal{S}(\mathbf{p}_G^{(i)}, z_Z)$. This condition was quite easy to fulfill in practice, as we observed that the region $\mathcal{S}(\mathbf{p}_G, z_Z)$ grows like the section by the plane $\Pi(z_Z, \mathbf{n})$ of a cone passing through G , *i.e.*, shrinking to a single point when $z_Z \rightarrow z_G$ and expanding proportionally to the distance $(z_Z - z_G)$. In practice, taking a ZMP one meter above the center of mass (HRP-4 is 1.5-meter tall) was enough to generate the complete motion depicted in Figure 6.4.⁸

Once the reference COM trajectory $\mathbf{p}_G(t)$ has been computed for the overall motion, whole-body joint-angles are generated by differential inverse kinematics (IK) under the following constraints (by decreasing task weight):

⁸ One may be tempted to take high plane coordinates z_Z , as it would only enlarge the corresponding support areas. However, we observed empirically that very high planes tend to generate numerical instabilities in both the double-description library *cddlib* (Fukuda and Prodon, 1996) and the trajectory generator described in 6.3.4.

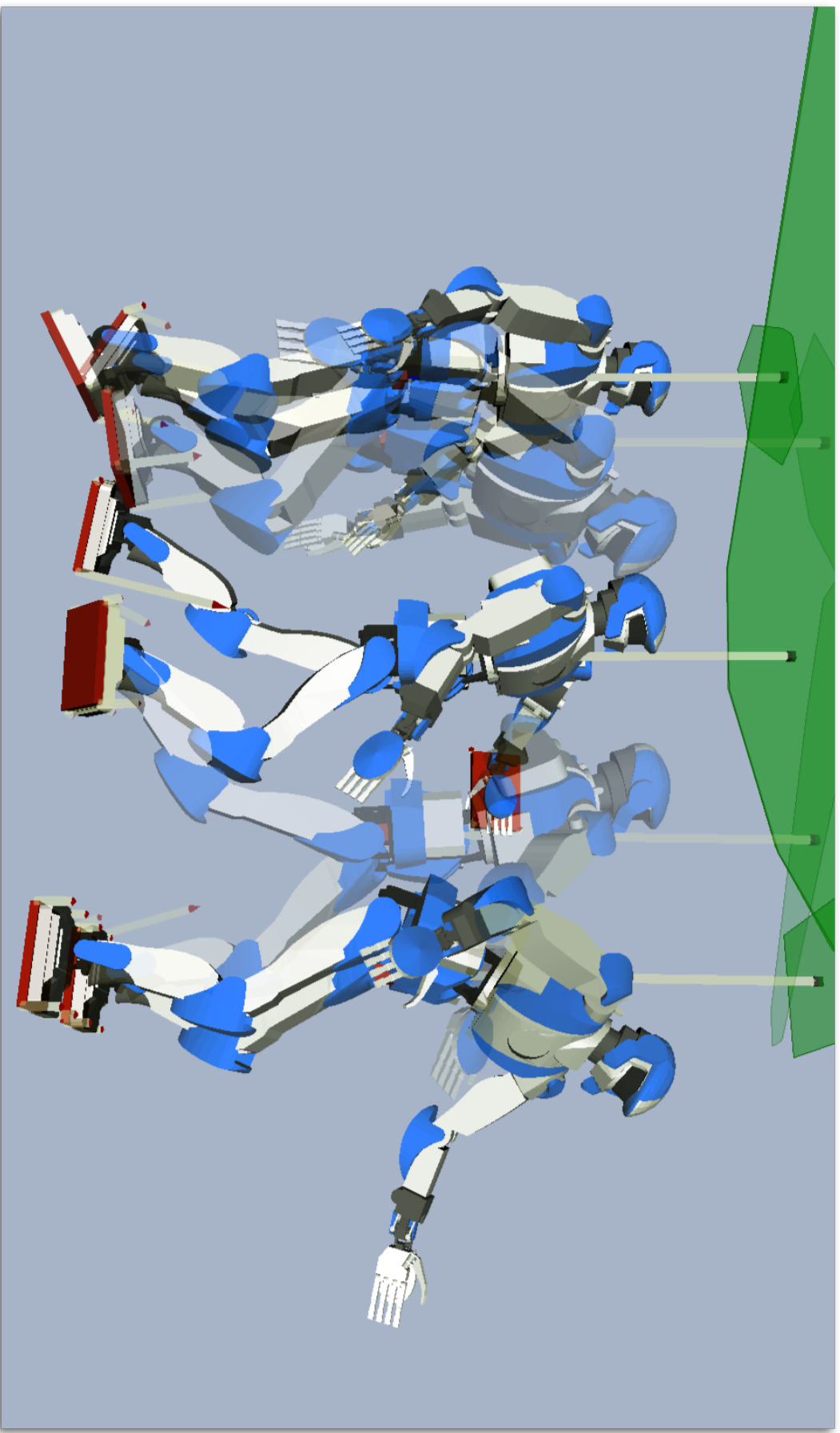


Fig. 6.4.: Snapshots of the motion generated by ZMP control of the COM *from above* under LP regulation. The scenario is designed so that the robot has to put its hand on the high platform (in the background) and right foot on the opposite tilted platform in order to perform an ample swing of the left leg that is otherwise impossible. In these simulations, the ZMP plane is taken one meter above the robot's COM. Green polygons in the plane above the robot's head are the respective support areas for each snapshot. The (virtual) linear pendula between ZMPs (points in the support polygons) and their attached COM are depicted by gray wires. The correctness of our dynamic-stability criterion was cross-validated by explicit computation of contact forces at each time instant (arrows at the corners of each contact surfaces).

TASK DESCRIPTION	GAIN [Hz]	WEIGHT
Contacting end-effector	1	100
Free end-effector	0.01	100
Center of mass tracking	1	1
Angular momentum variations	N/A	0.2
Velocity smoothness	N/A	1
Preferred joint-angles	0.05	0.1

Tab. 6.1.: Gains and weights used in the differential IK tracker (N/A: no gain for tasks regulating accelerations)


1. tracking of the contacting end-effector poses,
2. tracking of the COM trajectory,
3. minimum variations in angular momentum
4. preferred values for some joint-angles.

We used our own IK solver for the task. Similarly to (Lee and Goswami, 2010), this solver is formulated as a single-layer QP problem with linear inequality constraints. Gains and weights used in the simulations are reported in Table 6.1, while other simulation parameters are given in Table 6.2. Other implementation details are available in the source code *pymanoid*.⁹

Center-of-mass, feet and hands motions generated by locomotion are not compatible with a strict regulation of the angular-momentum. As such, a whole-body controller can only provide a “best effort” solution that tries to keep variations in angular-momentum to a minimum ($\dot{\mathbf{L}}_G \approx \mathbf{0}$), while LP support areas correspond to dynamic stability for $\dot{\mathbf{L}}_G = \mathbf{0}$. We therefore confirmed the validity of our approach by computing contact forces satisfying dynamic-equilibrium and frictional constraints at each time step of the generated motion. We confirmed that such forces always exist. Force vectors are depicted in Figure 6.4.

6.3.6 Performance

Overall, our controller prototype runs at around 30 Hz on an average laptop computer. Performance depends both on the number and relative positions of contacts.

⁹  <https://github.com/stephane-caron/pymanoid>

DESCRIPTION	SYMBOL	VALUE
Friction coefficient (all contacts)	μ	0.5
Number of traj. gen. timesteps	K	100
Duration of traj. gen. timesteps	δt	10 ms
Plane normal	\mathbf{n}	[0 0 1]
Step duration	T_S	2.5 [s]
Velocity limits	$\dot{\mathbf{q}}_{\max}$	0.5 [rad/s]

Tab. 6.2.: Simulation and trajectory generation parameters

In our experiments with *cdd* on the previous striding motion, single-support areas can be computed at around 100 Hz, while calculations slow down to around 10 Hz when the robot swings its left leg with hand on the wall. We note here that our code is not optimized: it suffers from two performance limitations imposed by the prototyping environment:

- The *global interpreter lock*, due to the Python interpreter, which prevents full use of all CPU cores on multi-threaded programs.
- The object-lock mechanism in OpenRAVE, which prevents parallelization of IK and support area computations due to their common use of an “environment” variable.

Finally, we found that some configurations in the previous striding motion (in particular when the free leg swings with a hand on the wall) throw *cdd* in an infinite loop. In this case, we timeout its execution and relaunch it as soon as possible. The overall stability of the whole motion can still be assessed by continuity: support areas “morph” from one COM position to the next, so that, even if the computation of the support area fails at time t , contact stability is still ensured if the ZMP is in the support area at $t - \delta t$ and $t + \delta t$.

6.4 Discussion

6.4.1 Plane coordinate selection

In our development, the coordinate of the virtual ZMP plane seems to be a free variable, which could be set to an arbitrary value by the user. In practice however, the plane coordinate has an impact on the ZMP-COM trajectories generated by our

MPC controller: a “low” ZMP-COM distance produces trajectories with back-and-forth oscillations, while a “high” distance yields bang-bang trajectories. We chose a suitable value empirically so as to hit the sweet spot where the COM trajectory is monotonic and varies smoothly. For instance, in the striding motion above, the altitude difference between COM and ZMP was set to $z_Z - z_G = 0.9$ m.

6.4.2 CHCP and infinite friction

Methods that take the Convex Hull of Contact Points (CHCP) as ZMP support areas rely by construction on the assumption that arbitrary contact forces can be exerted at each contact points, *i.e.*, that friction coefficients are infinite. This assumption was reasonable for *e.g.*, walking on horizontal floors, where the COM is high enough to lie inside contact friction cones. However, it becomes problematic in more general settings such as multi-contact locomotion or walking on low-friction floors (apart from sliding, foot yaw rotations due to insufficient friction have also been observed and studied (Cisneros et al., 2014; Caron et al., 2015b)).

The rationale for taking the CHCP as ZMP support area or volume goes as follows. First, the set \mathcal{S} of realizable ZMPs is convex. The ZMP can, in particular, be a center of pressure on a given contact surface, which in turn can be realized at any vertex of the corresponding contact polygon. Thus, the ZMP can be realized at any contact point, which makes \mathcal{S} a convex set containing all contact points. The smallest such set is the CHCP.

Friction comes into play with the resultant force. Suppose that the ZMP is located at a vertex C_k of a given contact polygon. From the analysis of Section 6.2, the resultant force \mathbf{f}^c must be realized by \mathbf{f}_k^c , while all other contact forces $\mathbf{f}_j^c = \mathbf{0}$ ($j \neq k$). (From eq. 6.7, $\mathbf{p}_Z = \mathbf{p}_{Z_k} = \mathbf{p}_{C_k}$ implies that only the λ_i 's corresponding to C_k can be strictly positive.) Therefore, the resultant \mathbf{f}^c of *all* contact forces must lie in the friction cone \mathcal{C}_k . This is impossible in situations such as the one depicted in Figure 6.5: assuming small linear and angular momentum, \mathbf{f}^c must be directed toward the vicinity of the center of mass (eq. (6.15)), and thus lies outside of \mathcal{C}_k . This is the reason why the actual ZMP support area in this case is smaller than the convex hull of contact points.

6.4.3 Can we do better than ZMP?

Proposition 9 implies that, while the moment τ_O^{gi} depends on the reference point O , its projection Z on Π does not depend on the plan coordinates of O . In this sense, the ZMP *decouples* the moment of a wrench from the position at which it is taken; yet at the cost of one dimension, as it only represents two out of the three moment coordinates. Could a “generalized” three-dimensional ZMP perform a similar decoupling for all three coordinates of the moment? Unfortunately the answer seems to be negative at first, at least in the following sense:

Property 8. *At least one coordinate of the ZMP depends on the reference point O . Formally, let $p_Z(O) = p_O + \mathbf{B}(\mathbf{f}) \tau_O$ denote any affine projection of the moment τ_O , where the matrix $\mathbf{B}(\mathbf{f})$ can depend non-linearly on \mathbf{f} . The set of displacements $\overrightarrow{OO'}$ that leave Z invariant is a vector space of dimension at most two.*

Proof. Let O and O' denote two points such that $p_Z(O') = p_Z(O)$. Then, $\overrightarrow{O'O} + \mathbf{B}(\mathbf{f}) \overrightarrow{OO'} \times \mathbf{f} = \mathbf{0}$, which one can write $\mathbf{C} \overrightarrow{OO'} = \overrightarrow{OO'}$ for $\mathbf{C} := \mathbf{B}(\mathbf{f})[-\mathbf{f} \times]$. The translation vector $\overrightarrow{OO'}$ thus belongs to the eigenspace \mathcal{E} of \mathbf{C} associated to the eigenvalue 1. To conclude, remark that $\dim(\mathcal{E}) \leq \text{rank}(\mathbf{C}) \leq \text{rank}([\mathbf{f} \times]) \leq 2$. \square

Let us then consider the remaining moment coordinate which is not represented by the ZMP. Our analysis following eq. (6.4) can be applied *mutatis mutandis* to this coordinate:

$$\frac{\mathbf{n} \cdot \tau_O^{gi}}{\mathbf{n} \cdot \mathbf{f}^{gi}} = \frac{\sum_i \lambda_i \mathbf{n} \cdot \overrightarrow{OC_i} \times \mathbf{f}_i}{\sum_i \lambda_i (\mathbf{n} \cdot \mathbf{f}_i)} = \frac{\sum_i \lambda_i (\mathbf{n} \cdot \mathbf{f}_i) \frac{\mathbf{n} \cdot \overrightarrow{OC_i} \times \mathbf{f}_i}{\mathbf{n} \cdot \mathbf{f}_i}}{\sum_i \lambda_i (\mathbf{n} \cdot \mathbf{f}_i)}.$$

A natural definition of the spatial point including all three coordinates (let us call it tentatively the \mathbf{n} -Moment Point or \mathbf{n} -MP) is then

$$\mathbf{p}_M = \mathbf{p}_O + \frac{\mathbf{n} \times \tau_O^{gi}}{\mathbf{n} \cdot \mathbf{f}} + \frac{\mathbf{n} \cdot \tau_O^{gi}}{\mathbf{n} \cdot \mathbf{f}} \mathbf{n}. \quad (6.21)$$

The vertices of its *support volume* \mathcal{V} can be computed in the same fashion as in Section 6.2 by

$$\mathbf{p}_{M_i} = \mathbf{p}_O + \frac{\mathbf{n} \times \overrightarrow{OC_i} \times \mathbf{f}_i}{\mathbf{n} \cdot \mathbf{f}_i} + \frac{\mathbf{n} \cdot \overrightarrow{OC_i} \times \mathbf{f}_i}{\mathbf{n} \cdot \mathbf{f}_i} \mathbf{n}.$$

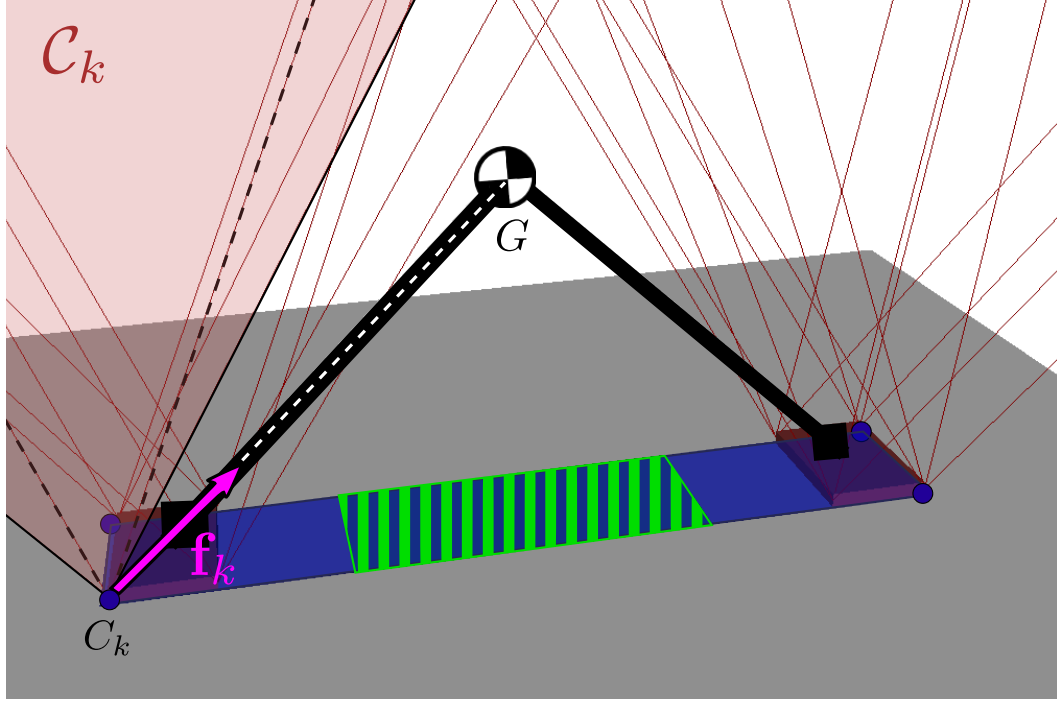


Fig. 6.5.: Situation where the ZMP support area (green stripes) is smaller than the convex hull of ground contact points (blue polygon). The robot has its two feet (transparent red boxes) set one meter apart on a horizontal floor (in gray). Its COM is 50 cm above ground, and the friction at contact is $\mu = 0.5$. The ZMP cannot be located at the corner C_k of the convex hull, as it would imply that the contact force exerted at this point (in magenta) lies outside the friction cone C_k (in red).

The geometric construction of support areas can also be applied *mutatis mutandis* to \mathcal{V} : when all virtual pressures $(\mathbf{n} \cdot \mathbf{f}_i)$ have the same sign, \mathcal{V} is the convex hull of the above vertices, while it is otherwise the union of two polyhedral convex cones built on the Minkowski difference of positive- and negative-pressure polyhedra. An implementation of this construction can be found in the *pymanoid* library.

The \mathbf{n} -MP is a three dimensional spatial point equivalent to the moment τ_O^{gi} , in the sense that one can be computed from the other by

$$\tau_O^{gi} = \overrightarrow{OM} \times (\mathbf{n} \cdot \mathbf{f}^{gi})\mathbf{n} + (\mathbf{n} \cdot \overrightarrow{OM})(\mathbf{n} \cdot \mathbf{f}^{gi})\mathbf{n}.$$

In other words, M represents the *screw coordinates* of the gravito-inertial wrench along the non-central axis $\Delta^{gi}(\mathbf{n})$, with magnitude $\mathbf{n} \cdot \mathbf{f}^{gi}$ and pitch $\mathbf{n} \cdot \overrightarrow{OM}$.

However, adding the third moment coordinates makes the shape of the support volume \mathcal{V} depend on the choice of the reference point O . Formally:

Property 9. *There is no non-empty subspace of displacements $\overrightarrow{OO'}$ of the reference point O , independent from the resultant \mathbf{f}^{gi} , that leaves the n -MP invariant.*

Proof. Consider a displacement $\overrightarrow{OO'}$ of O in the plane. From eq. (6.21), it results in a variation $\overrightarrow{OO'} \cdot \frac{\mathbf{n} \times \mathbf{f}^{gi}}{\mathbf{n} \cdot \mathbf{f}^{gi}}$ of the n -MP coordinate along \mathbf{n} . This term needs to be zero for any displacement leaving the n -MP invariant, thus $\overrightarrow{OO'}$ is parallel to either \mathbf{n} or \mathbf{f}^{gi} . The former would yield a variation of the plane coordinates of the n -MP (i.e., the ZMP). The latter is excluded, since we look for invariance independent from the gravito-inertial resultant. \square

Conclusion

We have derived the geometric construction of ZMP support areas in arbitrary planes from arbitrary contact wrench cones. Focusing on locomotion, we saw how the use of simplified control laws such as the linear pendulum model shrinks down the ZMP support area. Consequently, we proposed a novel algorithm to calculate this new area, and applied the concept to design a whole-body controller for locomotion across arbitrary contact situations, which we demonstrated on challenging simulation scenarios. It is, to the best of our knowledge, the **first time that the ZMP and Linear Pendulum models are applied for locomotion on arbitrary terrains**, with both implementation and solid theoretical foundations.

Conclusion

In this thesis, we have explored the questions of motion planning and control for humanoid robots with the aim to make humanoid motion planning practically fast so that it can run in a fast control loop. Our contributions towards this goal have developed three axes: kinodynamic decoupling, force-space curtailment, and dimensional reduction of the control space.

Kinodynamic decoupling

While we inspected the design of kinodynamic motion planners, we found out that motion planning was only successfully addressed for *geometric* problems, while the landscape of completeness and empirical results for *kinodynamic* planning was way more nuanced. In particular, proofs of correctness (probabilistic completeness) provided in the geometric case fail to generalize to kinodynamic systems such as humanoid robots, and other results published for such systems relied on assumptions far too complex to be checked on practical systems. We therefore derived our own proof of probabilistic completeness for a vast class of planners, namely those based on trajectory interpolation. This is **the first time that a completeness guarantee is established under assumptions that can be straightforwardly checked on actual robots**.

However, we found kinodynamic planning in the state space manifold (as is commonly done) intractable for humanoid motion planning. We consequently introduced in Chapter 3 an original method that can **decouple the kinematic and dynamic components of the kinodynamic problem**, allowing the motion planner to work only on the geometric configuration space of the robot, which has a much simpler structure than the state space manifold. To the best of our knowledge, **it is the first time that a kinodynamic planner in the configuration space is able to discover truly dynamic motions**, including the non-statically-stable ones that require constant non-zero accelerations.

Force-space curtailment

Narrowing down on humanoids, we reviewed in Chapter 4 the physics of contact and fundamental calculation techniques that allowed us to derive an original contribution, namely the **first-ever analytical formula of the contact wrench cone** of rectangular surfaces. This formula is of prime importance, as it allows the use of lower-dimensional coordinates to describe whole-body dynamics *without losing the complementary stability conditions*. Furthermore, the formula being derived analytically, it provides instantaneous computations at run time.

We then extended our developments to multi-contact scenarios in Chapter 5. We showed how the general stability condition of the gravito-inertial wrench cone can be integrated with TOPP, thus fitting in our kinodynamic planning framework, and asserted the feasibility of our solution in simulation experiments on challenging dynamical motions. Surprisingly, **this is the first time that a truly dynamic, non-quasi-static humanoid motion is discovered and reparameterized by TOPP**, as previous works all focused on retiming quasi-static humanoid motions.

Dimensional reduction of the control space

Wrench-based conditions provide a practical yet delicate framework for autonomous planning, as trajectory interpolation in the 6D wrench space is still an open question. We addressed this issue in Chapter 6 with a change of paradigm. We **generalized the notions of ZMP support area to multi-contact**, which is a novel and maybe the most significant contribution of this thesis. Based on this advance, we constructed and demonstrated a complete whole-body controller with which a humanoid can walk in challenging environments using a simple linear pendulum rooted at a ZMP above the robot's head. It is, to the best of our knowledge, the **first time that the ZMP and Linear Pendulum models are applied for locomotion on arbitrary terrains**.

These new conceptual tools integrate with our overall approach of reducing the state and control spaces of the robot for fast motion (re)planning. With the generalized ZMP and pendular control law, planning can be done in the three-dimensional *geometric* space of the COM, which is significantly smaller and structurally simpler than the space of generalized coordinates.

Planner in the loop

The outcome of our work is a kinodynamic planning framework, as depicted in Figure 1.1, in which the planning space is low-dimensional thanks to the reduced control space. Control trajectories are then converted to joint-angle trajectories in two decoupled steps, IK then TOPP. Meanwhile, despite curtailing contact force redundancy, our multi-stability conditions still fully capture the local contact dynamics; they ensure that local controllers always have a range of feasible solutions around which they can regulate local contact forces, centers of pressure and contact yaw moments. The overall pipeline finally illustrates how motion planning can be integrated in a fast control loop.

Elements of Contact planning

For space robots or manipulators with fixed bases, the notions of *motion* and *actuated joint trajectory* seem to coincide: given the time evolution of joint-angle values, one can apply the equation of motion to univocally compute the complete motion of the robot. This is because interactions between the robot and the environment are fully determined in such cases, with contact forces being nonexistent or fully determined (and unbounded) by fixed contacts, respectively. The story is of course different for limbed robots that locomote by making and breaking temporary contacts. In this case, contact forces are bounded and not fully determined: the environment can “choose” them in the nullspace of friction and momentum constraints.

A set of contact locations made at a given point in time is called a stance. By locations, we mean indifferently point (Escande et al., 2013) or surface contacts (the latter can be reduced to the former under proper assumptions, as shown in Chapter 4). In their early humanoid motion planner, Kuffner et al. (2002) applied RRT to plan new motions within a fixed stance. This solution could only move the free-flying coordinates of the humanoid inside the reachable space delimited by the fixed contact condition. Later developments (Bretl and Lall, 2008; Hauser et al., 2008) enabled a larger reachable space by using discrete stance changes: first, performing a graph search in a sample of the stance space (where nodes are stances and edges are steps), then planning a whole-body trajectory following the resulting step sequence.

A drawback in discretizing the stance space lies in the large impact of the sampling resolution on the algorithm’s performance: too sparse a sampling and no solution may be found, too large a sampling and the execution time may be prohibitive. To palliate this, (Bouyarmane and Kheddar, 2011; Escande et al., 2013) developed an alternative where the continuum of the stance space is explored, with guiding from heuristic cost and distance functions, rather than discretized.

In both approaches, planning takes place in the stance space and configurations are considered subsequently. Yet, the stance space is not an easy environment to plan in: its dimension grows linearly with the number of contact points or surfaces, and its free regions are conditioned by the geometric structure of the robot (*i.e.*, calls to an IK solver are required to compute obstacle-avoiding postures). In this appendix, we consider planning in the space of COM positions. That is, a COM trajectory is determined in the first place, and stances supporting it are computed subsequently.

A.1 Problem statement

The formulation is the same as in Chapter 5, with our n -DOF humanoid robot described by an $(n + 6)$ -dimensional vector of generalized coordinates \mathbf{q} , the last six Components of which describe the position and orientation of the free-flying link. We model the contact between the humanoid and its environment as a set of contact points¹ σ that is called a *stance* (Hauser et al., 2008). The equation of motion of the robot in the stance σ is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^\top \mathbf{C}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{S}^\top \boldsymbol{\tau} + \sum_{i \in \sigma} \mathbf{J}_i^\top(\mathbf{q}) \mathbf{f}_i, \quad (\text{A.1})$$

similarly to (5.1), with \mathbf{S} the projection on actuated coordinates. Due to the power limits, torques are bounded by

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}, \quad (\text{A.2})$$

while contact forces are constrained to lie in linearized friction cones

$$|f_{ix}| \leq \mu f_{iz}, \quad |f_{iy}| \leq \mu f_{iz}. \quad (\text{A.3})$$

We say that the configuration \mathbf{q} is *supported* by the stance σ , written $\mathbf{q} \in \mathcal{F}_\sigma$, when there exists a solution $\{\boldsymbol{\tau}, \mathbf{f}_1, \dots, \mathbf{f}_k\}$ to Equations (A.1), (A.2) and (A.3). Furthermore, two stances σ and σ' are *adjacent* when there exists a configuration $\mathbf{q} \in \mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'}$.

¹ We do not use the equivalent wrench formulation here.

A given stance σ allows for a given volume of free-link motions, but it is necessary to change stances in order to move the free-link in its full reachable space. We therefore consider motions with (a finite number of) switches in the supporting stance. Similarly to Hauser et al. (2008) and Escande et al. (2013), our goal is to compute a sequence of stances (and associated whole-body postures for the robot) connecting the initial stance to a goal area.

A.2 Contact planning in COM space

We will use the static stability criterion introduced and computed in Chapter 6 (*c.f.* Section 6.3.3). The support polygon thus computed only depends on the horizontal plane coordinates (x_G, y_G) of the COM. We choose to perform motion planning using a RRT in the two-dimensional space (x_G, y_G) .

The tree looks for a path between an initial and target COM positions. At each extension, arbitrary COM coordinates (x_G, y_G) are sampled from an estimate of the globally reachable region and the planner tries to extend one of its reached states in order to get closer to (x_G, y_G) . The overall process is reminded in Figure 6.

Algorithm 6 COM-RRT

Input: $p_{G,\text{start}}, p_{G,\text{goal}}$

Output: A feasible motion $(q(t), \sigma(t))$ such that $p_G(0) = p_{G,\text{start}}$ and $p_G(T) = p_{G,\text{goal}}$; or Failure

```

 $\sigma_{\text{start}} \leftarrow \text{GENERATE\_STANCE}(p_{G,\text{start}})$ 
 $\mathcal{T} \leftarrow \{(p_{G,\text{start}}, \sigma_{\text{start}})\}$ 
for  $i = 1$  to  $N$  do
   $p_G \leftarrow \text{SAMPLE}([X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}])$ 
   $\text{EXTEND}(\mathcal{T}, p_G)$ 
  if  $\text{EXTEND}(\mathcal{T}, p_{G,\text{goal}})$  then
    return  $\text{BACKTRACK\_TRAJECTORY}(\mathcal{T}, p_{G,\text{goal}})$ 
  end if
end for
return Failure

```

Differences from the generic RRT of (LaValle and Kuffner, 2001) occur in the EXTEND and GENERATE_STANCE functions. In the extension step, we use the k -nearest neighbors heuristic with $k = 10$ (Pham et al., 2013c) to raise the likelihood of a successful extension. The underlying metric $d(p_G, \sigma)$ is the distance from the target COM position p_G to the support polygon of the stance σ .

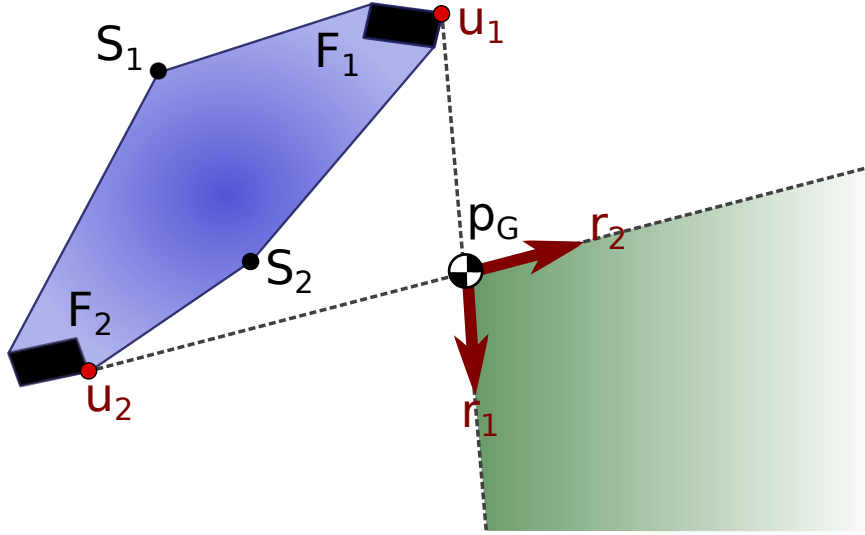


Fig. A.1.: Extension toward a candidate COM of a four-link stance with four non-coplanar contacts: two rectangular feet and two point sticks. Contact points and surfaces are drawn in black. The blue polygon represents the stance's support area. The green cone depicts the "eclipse" of the polygon by the COM. It corresponds to the positions at which the stick S_2 can be put so that the resulting stance stabilizes the candidate COM.

Algorithm 7 EXTEND() function

Input : tree \mathcal{T} , COM target p_G

Output : Success or Failure

```

1:  $\mathcal{N} \leftarrow \text{NEAREST\_STANCES}(\mathcal{T}, p_G)$ 
2:  $\mathcal{S} \leftarrow \{\text{EXTEND\_STANCE}(\sigma, p_G), \sigma \in \mathcal{N}\}$ 
3: if  $\mathcal{S} \neq \emptyset$  then
4:    $\mathcal{T} \leftarrow \mathcal{T} \cup \left\{ \arg \min_{\sigma \in \mathcal{S}} d(p_G, \sigma) \right\}$ 
5:   return Success
6: end if
7: return Failure

```

The core routine here is EXTEND_STANCE, which we will now explain. Let us denote by v_1, \dots, v_m the vertices of the support area, in trigonometric order. We define the *extension set* of the stance as

$$\text{ext}(\sigma) = \{v \mid p_G \in \text{conv}(\{v_1, \dots, v_m, v\})\}$$

This extension set turns out to be a cone that one can compute as follows. First, take the sequence of signed distances

$$s_i := \frac{v_{i+1} - v_i}{\|v_{i+1} - v_i\|} \times (p_G - v_i),$$

with the convention that the $m + 1$ index loops to 1. When p_G is outside of the support polygon, this sequence has positive and negative entries. Furthermore, all positive (resp. negative) elements are consecutive. Then, define the two points where the signed distance changes sign as

$$\begin{aligned} \mathbf{u}_1 &:= \mathbf{v}_{j_1} \text{ s.t. } s_{j_1} > 0 \wedge s_{j_1-1} < 0 \\ \mathbf{u}_2 &:= \mathbf{v}_{j_2} \text{ s.t. } s_{j_2} < 0 \wedge s_{j_2-1} > 0 \end{aligned}$$

Finally, $\text{ext}(\sigma)$ is given by the cone defined by the apex p_G and the two rays $\mathbf{r}_1 = (p_G - \mathbf{u}_1)$ and $\mathbf{r}_2 = (p_G - \mathbf{u}_2)$. Figure 8 illustrates this procedure.

We extend the stance by either setting a free link $\ell \in \text{free}(\sigma)$ or moving a contacting link to intersect the extension cone. In the latter case, we consider only the links that can be moved without affecting the side of the support polygon opposite to the target COM. This set is computed by mapping each vertex \mathbf{v}_i to its closest link $\ell(\mathbf{v}_i)$ (for the point-to-volume 3D euclidean metric) and using the signed distances to determine the side of each vertex. *In fine*,

$$\text{candidates}(\sigma) := \text{free}(\sigma) \cup \{\ell(\mathbf{v}_j) \mid s_j < 0\} - \{\ell(\mathbf{u}_1), \ell(\mathbf{u}_2)\}$$

Our overall stance-extension algorithm is summarized in Figure 8.

Algorithm 8 EXTEND_STANCE() function

Input : initial stance σ , COM target p_G

Output : stance σ' stabilizing p_G , or Failure

```

1:  $(j_1, j_2), (\mathbf{r}_1, \mathbf{r}_2) \leftarrow \text{ext}(\sigma)$ 
2: for each link  $\ell \in \text{candidates}(\sigma)$  do
3:    $(x', y') \leftarrow \text{SAMPLE\_CONE}(p_G, \mathbf{r}_1, \mathbf{r}_2)$ 
4:    $\mathbf{t} \leftarrow \text{GROUND\_POSE\_AT}(x', y')$ 
5:    $\sigma' \leftarrow \text{GENERATE\_POSTURE}(\sigma \cup \{(\ell, \mathbf{t})\})$ 
6:   if  $\sigma'$  was found then
7:     return  $\sigma'$ 
8:   end if
9: end for
10: return Failure

```

As illustrated by the call to GENERATE_POSTURE in the above pseudo-code, all the pipeline we described rests on an inverse-geometry solver. Posture Generation is the geometric problem of finding a vector of generalized coordinates \mathbf{q} satisfying a

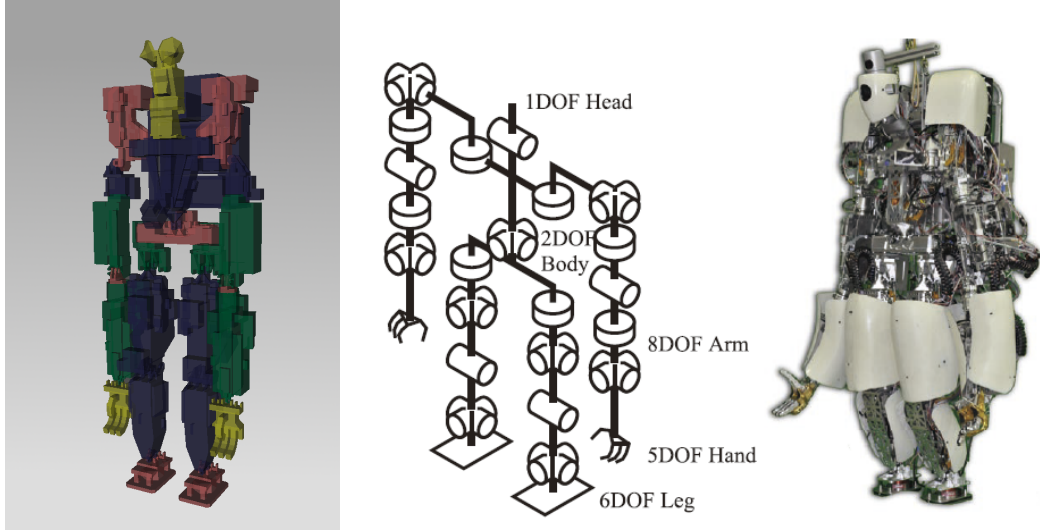


Fig. A.2.: HYDRA humanoid robot. **Left:** geometric model in false colors. **Middle:** kinematic structure of the robot. **Right:** picture of the real robot.

set of constraints such as DOF limits or, in a given stance σ , that all contacts $i \in \sigma$ are made. We solve the posture generation problem by inverse kinematics using the prioritized kinematic control framework from (Kanoun, 2012). The reader is referred to (Brossette et al., 2014) for a more general approach to multi-contact posture generation.

A.3 Experiment

We conduct our experiments in OpenRAVE (Diankov, 2010) with a model of the HYDRA humanoid robot developed in our laboratory, which is depicted in Figure A.2. In its current version, the robot has 41 degrees of freedom and is actuated by Electro-Hydrostatic Actuation (EHA). See Kaminaga and Nakamura (2015), Sato et al. (2015), and Kang et al. (2015) for details.

The setting of the experiment is a rubble field, as depicted in Figure A.3. Following the idea from Khatib and Chung (2014), we replace the robot's hands by walking sticks. The sticks give the robot a wider geometric range, in turn allowing for larger support areas. We ran COM-RRT with $p_{G,start}$ set to the middle of the rubble field. For now, our implementation of the GENERATE_STANCE function samples foot and stick poses around the COM until a solution is found.

Figure A.3 shows a stance sequence computed by our solution, with the underlying RRT depicted in Figure A.4. The humanoid starts in a configuration where its legs stand on two non-coplanar rubble blocks and are almost crossed. It first moves its right, then left stick before performing a left step. Two stick moves later, it performs a second left step, this time reaching a second block left of the first one, immediately followed by a right step putting both feet on the same block. Finally, after repositioning the two sticks, it performs an additional right step on the next block forward.

Conclusion

We have sketched how to implement a contact planner supporting the whole-body planning machinery developed in the core of this thesis, in the particular case of the static stability criterion. Contact planning and high-level planning interact as depicted in Figure 1.1. In the present experiment, we showed how a COM static-stability polygons can be used to guide contact planning, with new contacts supporting further motions of the COM, and so forth. A contact planner designed to follow the dynamic stability areas from Chapter 6 can be developed in a similar fashion.

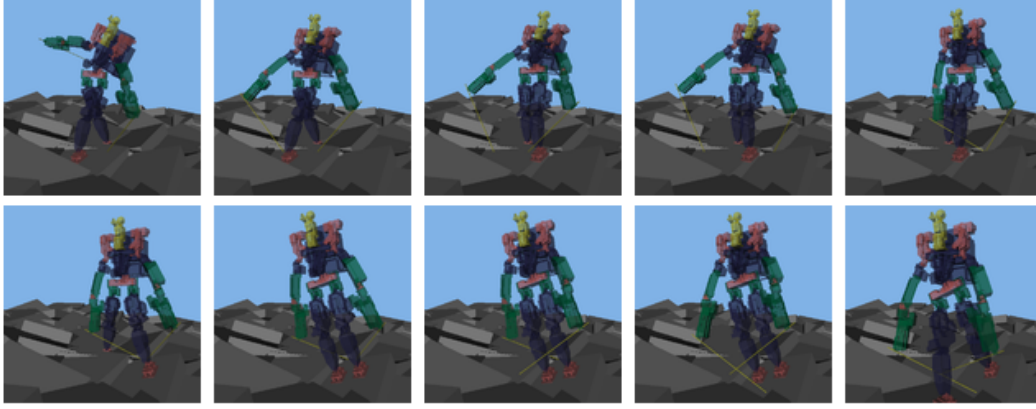


Fig. A.3.: Stance sequence generated by our planner. A stick-carrying variant of the HYDRA humanoid evolves on a randomly-generated rubble-field. Our planner's state space is the two-dimensional plane of horizontal COM position. In the sequence above, each stance results from an extension toward a COM sub-goal.

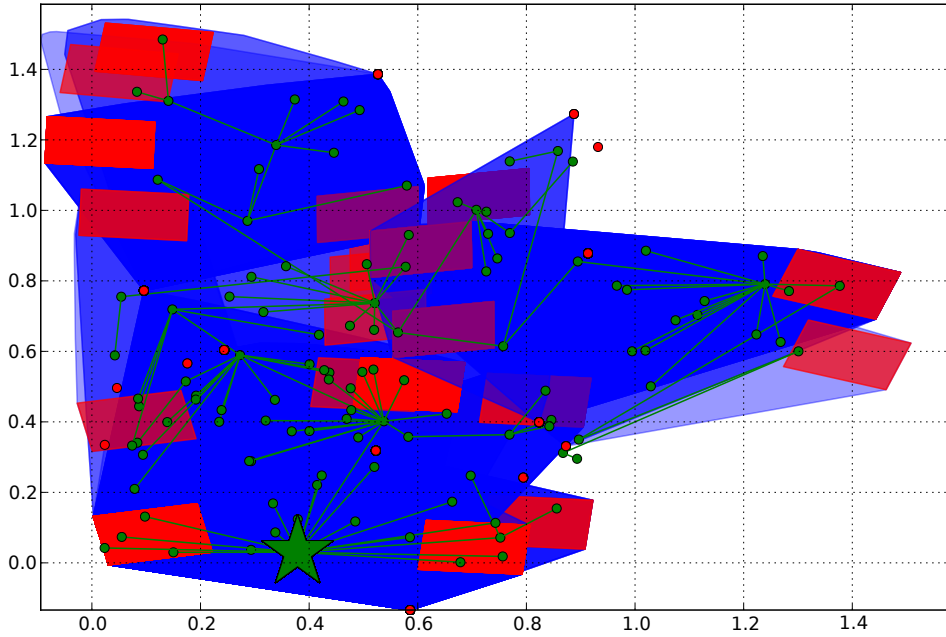


Fig. A.4.: Exploration of the RRT in the horizontal plane for the stance sequence depicted in Figure A.3. COM positions and trajectories are represented by green dots and lines, respectively. The green star corresponds to the starting COM position. Contact locations, either rectangular surfaces or stick points, are drawn in red while the blue areas correspond to superposed support polygons (plotted with transparency for readability).

Components developed for the DARPA Robotics Challenge

During the DARPA Robotics Challenge¹, robots were expected to solve a number of tasks under teleoperation by a human operator. Execution time and teleoperation bandwidth were restricted, so that teams needed to implement some level of autonomy on their robots. Yet, a human operator could still provide meaningful input to the robot on a regular basis via the team's teleoperation software, which was called Operator Control System (OCS) during the challenge. In this Appendix, we describe the design of the OCS we developed in Team Hydra in preparation for the challenge.

Like many teams, we chose ROS², on top of which we developed our own components. Our interface is based on *RViz*, the ROS visualization software, which comes with various extension commodities such as panels and interactive markers. Panels are areas of the interface where one can add arbitrary widgets connected to arbitrary commands. Interactive markers will be detailed later on.

To realize the valve-turning task, we have developed a number of components, split between two categories corresponding to perception and execution. Perception components include the construction of an environment model, identification of the valve position, and estimation of contact locations. Execution components include walking pattern generation, inverse kinematics and trajectory generation.

¹ <http://www.theroboticschallenge.org/>

² The “Robot Operating System <http://www.ros.org>, Operating System”, which is actually not an operating system but stands as a middleware running on Linux that also provides a software distribution and compilation tools.

B.1 Perception

Environment

We build a 3D point-cloud model of the environment using a 2D scanning-type laser range sensor UTM-X002S (Hokuyo Automatic Co. Ltd) tilted along the pitch axis by a MX-64 motor (Dynamixel). For this purpose, we used the ROS packages `hokuyo_tilter` and `pointcloud_tools` implemented by Kiyoshi Irie (Chiba Institute of Technology). Calibration between the environment model with respect to the humanoid’s kinematic chain was done by manual identification of the kinematic transform between the sensor base frame and the robot’s body. This approach has limited precision, as validation is done by a human operator, and is sensible to joint-calibration errors. Consequently, we prepared for execution-time updates: when limbs of the humanoid appear in its field-of-vision, the operator can tune the transform so as to match them with the kinematic model.

Here, the human operator implements a feedback loop between the point cloud and the robot’s kinematic model. This feedback loop could be automated, *e.g.*, by sampling points on the robot’s 3D mesh and using point registration methods to match them in the point-cloud. The underlying problem is to calibrate the sensor’s base transforms with respect to the robot’s kinematic chain. Birbach et al. (2015) demonstrated the feasibility of an autonomous solution to this problem using visual markers for simultaneous calibration of several sensory inputs.

Mesh identification

The position, orientation and size of the valve are input by the human operator with feedback from the point cloud. To enable this input, we developed a model-fitting interface based using a custom RViz panel and the `interactive_markers` library³. In this library, a “marker” consists in:

- a 3D mesh or a primitive shape (*e.g.*, a box),
- a set of “controls”,
- a right-click menu.

³http://wiki.ros.org/interactive_markers

The marker controls are graphical handles by which the operator can translate or rotate the mesh along any of its degrees of freedom (see Figure B.1).

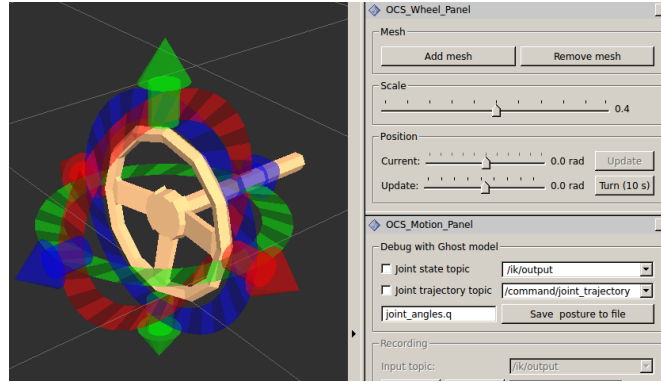


Fig. B.1.: Manipulation marker for a wheel mesh model. Control handles for the six degrees of freedom (translation and rotation) are depicted in red, green and blue for the x -, y - and z -axis respectively.

Algorithm 9 Model-fitting Procedure

1. Add the valve marker
(button from the valve panel)
 2. Enable marker controls
(right-click menu of the valve marker)
 3. Translate and rotate the valve mesh to match the position and orientation given by the point cloud
(marker controls)
 4. Adjust the scale of the mesh
(slider from the valve panel)
 5. Iterate 3 and 4 until fitting is complete
 6. Disable marker controls
(right-click menu of the valve marker)
-

The operator then follows the model-fitting procedure described in Algorithm 9. Overall, there are seven parameters to fit: the mesh's scale, position and orientation. Although the complete execution of the procedure takes less than a minute to a trained operation, this task ought to be solved by the software as well, *e.g.*, using point-cloud registration methods (Rusu et al., 2009).

Contact locations

The next step is to estimate the contact locations on the valve. To enable natural input of contact coordinates with respect to a 3D mesh, we developed a wrapper to

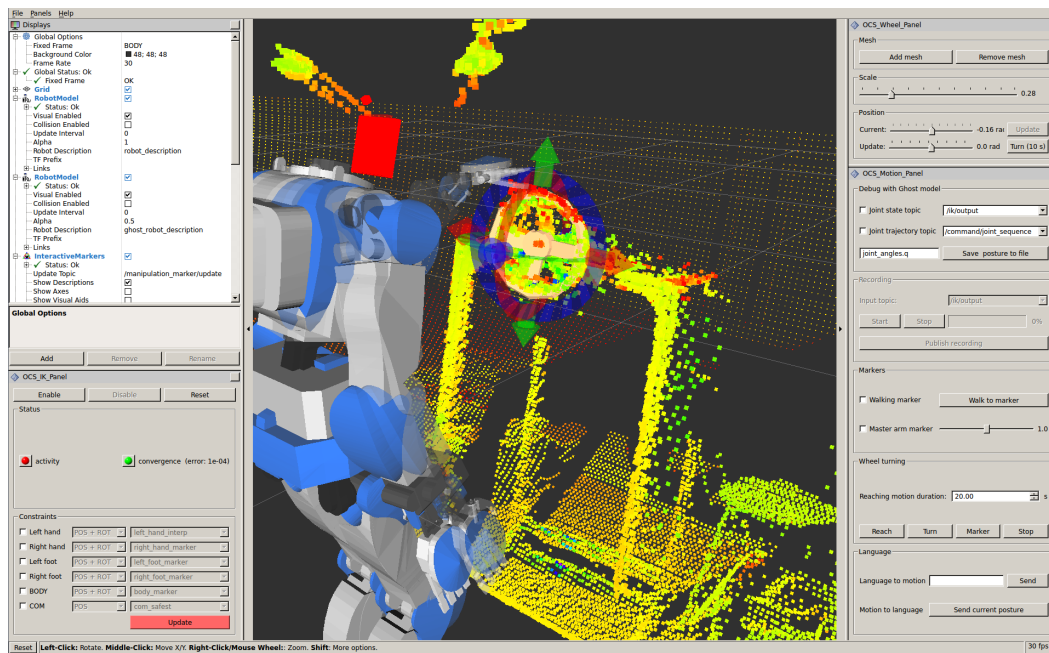



Fig. B.2.: Interface of the Operator Control System with a model of the HRP4-R humanoid robot. Its head has been replaced with an adjustable plate (in red) for the laser-range sensor frame. Panels on the left: default “Displays” from RViz and a custom panel for IK control. Panels on the right: Ghost model, Walking, Language and Wheel. The ghost model is used to preview IK results before execution on the real robot. Walking markers can be used to control the Walking Pattern Generator (Santacruz and Nakamura, 2012). The language panel is used to query a large human-behavior database for postures (Takano and Nakamura, 2008). Finally, the wheel panel allows to add/remove the valve marker, update its scale, send the reaching and turning commands. It displays two sliders: one for the estimate of the current wheel rotation, and the second for the desired turn angle. (Points above the robot head are artifacts from the sensor.)

interactive_markers, the manipulation_markers library, which is publicly available at:

 https://github.com/stephane-caron/manipulation_markers

Manipulation markers consist of

- a 3D-meshed “parent” marker, and
- a set of primitive-meshed “contact” markers.

Contact markers are used to calculate the end-effector poses (*i.e.*, position and orientation) by which the robot makes contact with their parent mesh. They can be added on-the-fly by the operator via the right-click menu. Because their kinematic transform are defined in the parent marker’s reference frame, moving the parent marker (in our case, turning the valve) automatically updates the position and orientation of each of its contact markers.

B.2 Motion generation and control

Our turning strategy relies on a single contact between the palm of the robot’s left hand and the outer rim of the valve. To avoid the need to move the feet while turning, we further decided to apply only quarter-turns at a time, with the robot periodically reaching for the top of the valve to start the motion again.

Walking Pattern Generator

First of all, the robot has to walk to the valve. To achieve this purpose, we used the Walking Pattern Generator (WPG) from (Santacruz and Nakamura, 2012), which allowed us to abstract the task to the 2D ground position of the Center-of-Mass (COM). Using an interactive marker on the foot horizontal place, the operator evaluates the point-cloud data before inputting the goal location of the COM to the WPG. Then, after a 10-second refresh interval, he or she can reiterate the process until the robot is standing in front of the valve. Once a satisfactory location is reached, we fix the foot locations on the ground and turn the valve with no additional call to the WPG.

Inverse Kinematics

The kinematic transform of a contact marker with respect to the robot's body frame can be used to establish contact via Inverse Kinematics (IK) and position control. For the former, we used RoKi (RoKi July 2015), a software library from the Motor Intelligence Laboratory (Osaka University). Given a set of end-effector poses (*i.e.*, position and orientation) and a COM position (maintained above the foot support area for stability), it could perform the whole-body inverse kinematics of our 34-DOF robot model within milliseconds.

Interpolation of reaching trajectories

Using the reduction from joint-space to work-space coordinates provided by the IK, we only control the left hand pose for the rest of the task. When the operator sets the contact marker on the valve, we compute two via-points from the current hand pose to the contact pose. The first one lies between the hand and the valve. The second one is closer to contact and has a *z*-coordinate *above* that of the contact point in order for the left hand to make contact with a downward-pointing incidence vector. Finally, a 20-second trajectory is interpolated between the four left-hand poses (initial, via-point 1, via-point 2, contact) by using linear interpolation for the end-effector position and spherical linear interpolation (Slerp) for its orientation.

Control of the Valve Turning Motion

Once contact is made, we generate the valve turning motion by turning the valve manipulation marker in RViz and forwarding the resulting IK output (where the left hand is bound to the contact marker) to the position controller. However, to cope with the limited bandwidth constraint, we further cut the operator input to a single turning angle. As depicted in Figure B.2, to execute of a quarter-turn, the operator sets the desired angle ("Update" slider) and presses the command button. The relative angle is then the only data sent to the robot field PC over the network, at which point the latter will update its internal model of the valve and perform the IK locally.

B.3 Valve turning task

The framework we described has been implemented as part of the *openth* software stack. We tested the valve-task software with an HRP4-R humanoid robot and a steel valve mounted on a metal frame. (Another application of *openth* has been reported by Ishiguro et al. (2015) for the car-driving task.) In the laboratory environment, it took the operator 10 minutes to execute the task.

Position estimation errors accumulate during execution of the task. In the present experiment, after reaching a third time for the top of the valve, the humanoid did not properly establish contact. The operator could notice that the valve did not turn with the hand (e.g., by looking at its rims in the point cloud view) and adapt to the situation by translating or downscaling the wheel mesh in the OCS.

Contact state estimation ought to be done at the robot level. Torque-controlled humanoids, or position-controlled ones with force-torque sensors in their wrist, can estimate the contact force directly. The estimation is still possible with previous generation humanoids like HRP4 where force-torque sensors are typically located in the ankles. Yet, it is more involved as forces are only observed after propagation along the kinematic chain.

Acknowledgments

The work presented in this Appendix was supported by the New Energy and Industrial Technology Development Organization (NEDO), The International R&D and Demonstration Project on Robotic Field / Research and Development of Disaster-Response Robot Open Platform (FY2014–FY2015).

The complete source code for the valve-turning OCS includes ROS packages developed by Kiyoshi Irie (Chiba Institute of Technology) for the Hokuyo sensor and Dynamixel motor, and by Yasuhiro Ishiguro for the integration between ROS and OpenRTM (Ishiguro et al., 2015).

Bibliography

- Audren, Hervé, Joris Vaillant, Abderrahmane Kheddar, et al. (2014). “Model preview control in multi-contact motion-application to a humanoid robot”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, pp. 4030–4035 (cit. on pp. 124, 137).
- Balkcom, Devin J and Jeffrey C. Trinkle (2002). “Computing wrench cones for planar rigid body contact tasks”. In: *The International Journal of Robotics Research* 21.12, pp. 1053–1066 (cit. on pp. 66, 67).
- Bialkowski, Joshua, Michael Otte, and Emilio Frazzoli (2013). “Free-configuration biased sampling for motion planning”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, pp. 1272–1279 (cit. on p. 25).
- Birbach, Oliver, Udo Frese, and Berthold Bäuml (2015). “Rapid calibration of a multi-sensorial humanoid’s upper body: An automatic and self-contained approach”. In: *The International Journal of Robotics Research* 34.4-5, pp. 420–436 (cit. on p. 160).
- Bobrow, J.E., S. Dubowsky, and JS Gibson (1985). “Time-optimal control of robotic manipulators along specified paths”. In: *The International Journal of Robotics Research* 4.3, pp. 3–17 (cit. on pp. 23, 41).
- Bouyarmane, Karim and Abderrahmane Kheddar (2011). “Multi-contact stances planning for multiple agents”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, pp. 5246–5253 (cit. on p. 151).
- Bretl, Timothy and Sanjay Lall (2008). “Testing static equilibrium for legged robots”. In: *Robotics, IEEE Transactions on* 24.4, pp. 794–807 (cit. on pp. 16, 105, 109, 110, 113, 151).
- Brossette, Stanislas, Adrien Escande, Joris Vaillant, et al. (2014). “Integration of non-inclusive contacts in posture generation”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, pp. 933–938 (cit. on p. 156).
- Bullo, Francesco and Kevin M Lynch (2001). “Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems”. In: *Robotics and Automation, IEEE Transactions on* 17.4, pp. 402–412 (cit. on p. 23).

- Caron, Stéphane and Yoshihiko Nakamura (2015a). “Planning with the Center-of-Mass rather than Stances for Humanoids Walking on Uneven Terrains”. In: *14th World Congress in Mechanism and Machine Science* (cit. on p. 177).
- (2015b). “Teleoperation System Design of Valve Turning Motions in Degraded Communication Conditions”. In: *Proceedings of the 33rd Annual Conference of the Robotics Society of Japan* (cit. on pp. 178, 179).
- Caron, Stéphane, Yoshihiko Nakamura, and Quang-Cuong Pham (2013). “Kinodynamic Motion Planners based on Velocity Interval Propagation”. In: *Proceedings of the 31st Annual Conference of the Robotics Society of Japan* (cit. on p. 178).
- Caron, Stéphane, Quang-Cuong Pham, and Yoshihiko Nakamura (2014a). “Completeness of randomized kinodynamic planners with state-based steering”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, pp. 5818–5823 (cit. on p. 178).
- Caron, Stéphane, Yoshihiko Nakamura, and Quang-Cuong Pham (2014b). “Kinodynamic Motion Retiming for Humanoid Robots”. In: *Proceedings of the 32nd Annual Conference of the Robotics Society of Japan* (cit. on p. 178).
- Caron, Stéphane, Quang-Cuong Pham, and Yoshihiko Nakamura (2015a). “Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics”. In: *Robotics: Science and System* (cit. on pp. 23, 122, 123, 177).
- (2015b). “Stability of Surface Contacts for Humanoid Robots: Closed-Form Formulae of the Contact Wrench Cone for Rectangular Support Areas”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE (cit. on pp. 143, 178).
- Cisneros, Rafael, Kazuhito Yokoi, and Eiichi Yoshida (2014). “Yaw moment compensation by using full body motion”. In: *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*. IEEE, pp. 119–125 (cit. on p. 143).
- Del Prete, Andrea, Steve Tonneau, and Nicolas Mansard (2015). “Fast Algorithms to Test Robust Static Equilibrium for Legged Robots”. URL: <https://hal.archives-ouvertes.fr/hal-01201060> (cit. on p. 68).
- Diankov, R. (2010). “Automated Construction of Robotic Manipulation Programs”. PhD thesis. Carnegie Mellon University, Robotics Institute. URL: http://www.programmingvision.com/rosen_diankov_thesis.pdf (cit. on pp. 55, 156).
- Donald, Bruce, Patrick Xavier, John Canny, and John Reif (1993). “Kinodynamic motion planning”. In: *Journal of the ACM (JACM)* 40.5, pp. 1048–1066 (cit. on pp. 15, 22, 27).
- Englsberger, Johannes, Christian Ott, and Alin Albu-Schaffer (2015). “Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion”. In: *Robotics, IEEE Transactions on* 31.2, pp. 355–368 (cit. on p. 123).

- Escande, Adrien, Abderrahmane Kheddar, and Sylvain Miossec (2013). “Planning contact points for humanoid robots”. In: *Robotics and Autonomous Systems* 61.5, pp. 428–442 (cit. on pp. 16, 123, 151, 153).
- Escande, Adrien, Nicolas Mansard, and Pierre-Brice Wieber (2014). “Hierarchical quadratic programming: Fast online humanoid-robot motion generation”. In: *The International Journal of Robotics Research*, p. 0278364914521306 (cit. on p. 112).
- Fourier, Joseph (1827). “Histoire de l’Académie, partie mathématique (1824)”. In: *Mémoires de l’Académie des sciences de l’Institut de France* 7 (cit. on p. 71).
- Fukuda, Komei (2000). “Polyhedral computation FAQ”. In: (cit. on p. 69).
- Fukuda, Komei and Alain Prodon (1996). “Double description method revisited”. In: *Combinatorics and computer science*. Springer, pp. 91–111 (cit. on pp. 70, 71, 129, 139).
- Goswami, Ambarish (1999). “Postural stability of biped robots and the foot-rotation indicator (FRI) point”. In: *The International Journal of Robotics Research* 18.6, pp. 523–533 (cit. on p. 119).
- Harada, Kensuke, Shuuji Kajita, Kenji Kaneko, and Hirohisa Hirukawa (2006a). “An analytical method for real-time gait planning for humanoid robots”. In: *International Journal of Humanoid Robotics* 3.01, pp. 1–19 (cit. on p. 123).
- (2006b). “Dynamics and balance of a humanoid robot during manipulation tasks”. In: *Robotics, IEEE Transactions on* 22.3, pp. 568–575 (cit. on pp. 121–123).
- Hauser, Kris (2013). “Fast Interpolation and Time-Optimization on Implicit Contact Submanifolds.” In: *Robotics: Science and Systems*. Citeseer (cit. on pp. 36, 41).
- (2014). “Fast interpolation and time-optimization with contact”. In: *The International Journal of Robotics Research* 33.9, pp. 1231–1250 (cit. on pp. 68, 95, 96, 116, 117).
- Hauser, Kris and Victor Ng-Thow-Hing (2010). “Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, pp. 2493–2498 (cit. on p. 38).
- Hauser, Kris, Timothy Bretl, Jean-Claude Latombe, Kensuke Harada, and Brian Wilcox (2008). “Motion planning for legged robots on varied terrain”. In: *The International Journal of Robotics Research* 27.11-12, pp. 1325–1349 (cit. on pp. 151–153).
- Hirukawa, Hirohisa, Shizuko Hattori, Kensuke Harada, et al. (2006). “A universal stability criterion of the foot contact of legged robots-adios zmp”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, pp. 1976–1983 (cit. on pp. 15, 122, 124).

- Hsu, David, J-C Latombe, and Rajeev Motwani (1997). "Path planning in expansive configuration spaces". In: *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*. Vol. 3. IEEE, pp. 2719–2726 (cit. on pp. 15, 26, 27).
- Hsu, David, Robert Kindel, Jean-Claude Latombe, and Stephen Rock (2002). "Randomized kinodynamic motion planning with moving obstacles". In: *The International Journal of Robotics Research* 21.3, pp. 233–255 (cit. on pp. 15, 26, 27).
- Hyon, Sang-Ho, Joshua G Hale, and Gordon Cheng (2007). "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces". In: *Robotics, IEEE Transactions on* 23.5, pp. 884–898 (cit. on p. 124).
- Inomata, Kentaro and Yutaka Uchimura (2010). "3DZMP-based control of a humanoid robot with reaction forces at 3-dimensional contact points". In: *Advanced Motion Control, 2010 11th IEEE International Workshop on*. IEEE, pp. 402–407 (cit. on p. 121).
- Ishiguro, Yasuhiro, Hiroshi Kaminaga, and Yoshihiko Nakamura (2015). "Design of a system for remote car driving in a degraded communication environment". In: *The 33rd Annual Conference of the Robotics Society of Japan*. Language: Japanese. Original title: 劣悪通信環境下における遠隔自動車運転操作のシステム設計 (cit. on p. 165).
- Johnson, J. and K. Hauser (2012). "Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path". In: *IEEE International Conference on Robotics and Automation*, pp. 2035–2041 (cit. on p. 46).
- Jolly, K. G., R. Sreerama Kumar, and R. Vijayakumar (2009). "A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits". In: *Robotics and Autonomous Systems* 57.1, pp. 23–33 (cit. on p. 36).
- Kagami, Satoshi, Tomonobu Kitagawa, Koichi Nishiwaki, et al. (2002). "A fast dynamically equilibrated walking trajectory generation method of humanoid robot". In: *Autonomous Robots* 12.1, pp. 71–82 (cit. on p. 122).
- Kajita, Shuuji, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa (2001). "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation". In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 239–246 (cit. on pp. 119, 123, 135).
- Kajita, Shuuji, Fumio Kanehiro, Kenji Kaneko, et al. (2003). "Biped walking pattern generation by using preview control of zero-moment point". In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. Vol. 2. IEEE, pp. 1620–1626 (cit. on pp. 123, 137).
- Kaminaga, Hiroshi and Yoshihiko Nakamura (2015). "Electronics of the hydraulically-actuated humanoid robot Hydra". In: *The 33rd Annual Conference of the Robotics Society of Japan*. Language: Japanese. Original title: 油圧駆動ヒューマノイドロボット Hydra のエレクトロニクス (cit. on p. 156).

- Kanehiro, Fumio, Hirohisa Hirukawa, and Shuuji Kajita (2004). “OpenHRP: Open architecture humanoid robotics platform”. In: *The International Journal of Robotics Research* 23.2, pp. 155–165 (cit. on p. 87).
- Kaneko, Kenji, Fumio Kanehiro, Mitsuharu Morisawa, et al. (2011). “Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, pp. 4400–4407 (cit. on pp. 87, 101).
- Kang, Tianyi, Hiroshi Kaminaga, and Yoshihiko Nakamura (2015). “Development of the 5-DOF hydraulically-actuated hand of Hydra”. In: *The 33rd Annual Conference of the Robotics Society of Japan*. Language: Japanese. Original title: 5自由度油圧駆動Hydraハンドの開発 (cit. on p. 156).
- Kanoun, Oussama (2012). “Real-time prioritized kinematic control under inequality constraints for redundant manipulators”. In: *Robotics: Science and Systems*. Vol. 7, p. 145 (cit. on p. 156).
- Karaman, S. and E. Frazzoli (2013). “Sampling-based Optimal Motion Planning for Non-holonomic Dynamical Systems”. In: *IEEE Conference on Robotics and Automation (ICRA)* (cit. on p. 28).
- Karaman, Sertac and Emilio Frazzoli (2010). “Optimal kinodynamic motion planning using incremental sampling-based methods”. In: *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, pp. 7681–7687 (cit. on p. 27).
- (2011). “Sampling-based algorithms for optimal motion planning”. In: *The International Journal of Robotics Research* 30.7, pp. 846–894 (cit. on pp. 26–28).
- Kavraki, Lydia E, Petr Svestka, J-C Latombe, and Mark H Overmars (1996). “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. In: *Robotics and Automation, IEEE Transactions on* 12.4, pp. 566–580 (cit. on pp. 15, 22, 23).
- Khatib, Oussama and Shu-Yun Chung (2014). “SupraPeds: humanoid contact-supported locomotion for 3D unstructured environments”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, pp. 2319–2325 (cit. on p. 156).
- Kuffner, James J, Koichi Nishiwaki, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue (2001). “Footstep planning among obstacles for biped robots”. In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 500–505 (cit. on p. 15).
- Kuffner, James J, Satoshi Kagami, Koichi Nishiwaki, Masayuki Inaba, and Hirochika Inoue (2002). “Dynamically-stable motion planning for humanoid robots”. In: *Autonomous Robots* 12.1, pp. 105–118 (cit. on pp. 41, 151).
- Kunz, Thomas and Mike Stilman (2014). “Probabilistically complete kinodynamic planning for robot manipulators with acceleration limits”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, pp. 3713–3719 (cit. on p. 26).

- Kunz, Tobias and Mike Stilman (2015). “Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete”. In: *Algorithmic Foundations of Robotics XI*. Springer, pp. 233–244 (cit. on p. 27).
- Latombe, Jean-Claude (1991). *Robot motion planning*. Vol. 124. Springer US (cit. on p. 22).
- Laumond, J.-P. (1998). *Robot Motion Planning and Control*. New York: Springer-Verlag (cit. on pp. 23, 25).
- LaValle, S.M. (2006). *Planning algorithms*. Cambridge Univ Press (cit. on p. 22).
- LaValle, Steven M and James J Kuffner (2000). “Rapidly-exploring random trees: Progress and prospects”. In: (cit. on pp. 15, 22, 38, 51).
- (2001). “Randomized kinodynamic planning”. In: *The International Journal of Robotics Research* 20.5, pp. 378–400 (cit. on pp. 15, 23, 25–27, 35, 153).
- Lee, Sung-Hee and Ambarish Goswami (2010). “Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground”. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, pp. 3157–3162 (cit. on pp. 124, 141).
- Lertkultanon, Puttichai and Quang-Cuong Pham (2014). “Dynamic non-prehensile object transportation”. In: *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*. IEEE, pp. 1392–1397 (cit. on p. 51).
- Lozano-Perez, Tomas (1983). “Spatial planning: A configuration space approach”. In: *Computers, IEEE Transactions on* 100.2, pp. 108–120 (cit. on p. 15).
- Mitobe, Kazuhisa, Shinichiro Kaneko, Tomohiro Oka, Yasuo Nasu, and Genci Capi (2004). “Control of legged robots during the multi support phase based on the locally defined ZMP”. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 3. IEEE, pp. 2253–2258 (cit. on pp. 121, 123).
- Morisawa, Mitsuharu, Shuuji Kajita, Kenji Kaneko, et al. (2005). “Pattern generation of biped walking constrained on parametric surface”. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, pp. 2405–2410 (cit. on p. 123).
- Morisawa, Mitsuharu, Nobuyuki Kita, Shin’ichiro Nakaoka, et al. (2014). “Biped locomotion control for uneven terrain with narrow support region”. In: *System Integration (SII), 2014 IEEE/SICE International Symposium on*. IEEE, pp. 34–39 (cit. on p. 123).
- Motzkin, Theodor (1952). *The theory of linear inequalities*. Tech. rep. DTIC Document. URL: <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=AD0604770> (cit. on p. 71).

- Nagasaka, K., T. Fukushima, and H. Shimomura (2012). “Whole-body control of a humanoid robot based on generalized inverse dynamics and multi-contact stabilizer that can take account of contact constraints”. In: *Robotics Symposium (in Japanese)*. Vol. 17 (cit. on p. 124).
- Or, Yizhar and Elon Rimon (2006). “Computation and graphical characterization of robust multiple-contact postures in two-dimensional gravitational environments”. In: *The International Journal of Robotics Research* 25.11, pp. 1071–1086 (cit. on pp. 108, 109).
- Ott, Christian, Maximo Roa, Gerd Hirzinger, et al. (2011). “Posture and balance control for biped robots based on contact force optimization”. In: *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, pp. 26–33 (cit. on p. 124).
- Pang, J-S and J Trinkle (2000). “Stability characterizations of rigid body contact problems with coulomb friction”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 80.10, pp. 643–663 (cit. on p. 95).
- Papadopoulos, Georgios, Hanna Kurniawati, and Nicholas M Patrikalakis (2014). “Analysis of Asymptotically Optimal Sampling-based Motion Planning Algorithms for Lipschitz Continuous Dynamical Systems”. In: *arXiv preprint arXiv:1405.2872* (cit. on pp. 25–27).
- Perez, Alejandro, Robert Platt, George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez (2012). “LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, pp. 2537–2542 (cit. on pp. 26, 38).
- Pham, Q.-C. (2014). “A general, fast, and robust implementation of the time-optimal path parameterization algorithm”. In: *IEEE Transactions on Robotics* 6, pp. 1533–1540. DOI: 10.1109/TR0.2014.2351113 (cit. on pp. 41, 44, 45, 51, 116, 179).
- Pham, Quang-Cuong and Yoshihiko Nakamura (2012). “Time-optimal path parameterization for critically dynamic motions of humanoid robots”. In: *IEEE-RAS International Conference on Humanoid Robots* (cit. on p. 97).
- Pham, Quang-Cuong, Stéphane Caron, and Yoshihiko Nakamura (2013a). “Kinodynamic Planning in the Configuration Space via Velocity Interval Propagation”. In: *Robotics: Science and System* (cit. on p. 38).
- (2013b). “Kinodynamic Planning in the Configuration Space via Velocity Interval Propagation”. In: *Robotics: Science and System* (cit. on pp. 51, 56, 178).
- (2013c). “Kinodynamic Planning in the Configuration Space via Velocity Interval Propagation”. In: *Robotics: Science and System* (cit. on p. 153).

- Pham, Quang-Cuong, Stéphane Caron, Puttichai Lertkultanon, and Yoshihiko Nakamura (2014). “Planning Truly Dynamic Motions: Path-Velocity Decomposition Revisited”. In: *CoRR* abs/1411.4045. URL: <http://arxiv.org/abs/1411.4045> (cit. on p. 56).
- Popovic, Marko B, Ambarish Goswami, and Hugh Herr (2005). “Ground reference points in legged locomotion: Definitions, biological trajectories and control implications”. In: *The International Journal of Robotics Research* 24.12, pp. 1013–1032 (cit. on p. 112).
- Qiu, Zhapeng, Adrien Escande, Alain Micaelli, and Thomas Robert (2011). “Human motions analysis and simulation based on a general criterion of stability”. In: *International Symposium on Digital Human Modeling* (cit. on pp. 15, 68, 104, 122, 123).
- Righetti, Ludovic, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan, and Stefan Schaal (2013). “Optimal distribution of contact forces with inverse-dynamics control”. In: *The International Journal of Robotics Research* 32.3, pp. 280–298 (cit. on pp. 95, 104, 124).
- RoKi (July 2015). RoKi - Robot Kinetics library, Motor Intelligence Lab, Osaka University. Available online at <http://www.mi.ams.eng.osaka-u.ac.jp/software/roki.html> (cit. on p. 164).
- Rusu, Radu Bogdan, Nico Blodow, and Michael Beetz (2009). “Fast point feature histograms (FPFH) for 3D registration”. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 3212–3217 (cit. on p. 161).
- Saida, Takao, Yasuyoshi Yokokohji, and Tsuneo Yoshikawa (2003). “FSW (feasible solution of wrench) for multi-legged robots”. In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. Vol. 3. IEEE, pp. 3815–3820 (cit. on pp. 15, 122, 131).
- Santacruz, Carlos and Yoshihiko Nakamura (2012). “Analytical real-time pattern generation for trajectory modification and footstep replanning of humanoid robots”. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, pp. 2095–2100 (cit. on pp. 162, 163).
- Sardain, P. and G. Bessonnet (2004). “Forces acting on a biped robot. center of pressure-zero moment point”. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 34.5, pp. 630–637 (cit. on pp. 83, 84, 119–121, 125, 126).
- Sato, Shunsuke, Hiroshi Kaminaga, and Yoshihiko Nakamura (2015). “Design of the joint electro-hydrostatic actuation mechanism for a life-size humanoid robot”. In: *The 33rd Annual Conference of the Robotics Society of Japan*. Language: Japanese. Original title: 等身大ヒューマノイドロボットのEHAによる関節駆動機構の設計 (cit. on p. 156).

- Sato, Tomoya, Sho Sakaino, Eijiro Ohashi, and Kouhei Ohnishi (2011). “Walking trajectory planning on stairs using virtual slope for biped robots”. In: *Industrial Electronics, IEEE Transactions on* 58.4, pp. 1385–1396 (cit. on pp. 122, 123).
- Shibuya, Maki, Tomoyuki Suzuki, and Kouhei Ohnishi (2006). “Trajectory planning of biped robot using linear pendulum mode for double support phase”. In: *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*. IEEE, pp. 4094–4099 (cit. on pp. 122, 123).
- Shiller, Z. and S. Dubowsky (1985). “On the optimal control of robotic manipulators with actuator and end-effector constraints”. In: *IEEE International Conference on Robotics and Automation*, pp. 614–620 (cit. on p. 43).
- Shiller, Z. and Y.R. Gwo (1991). “Dynamic motion planning of autonomous vehicles”. In: *IEEE Transactions on Robotics and Automation* 7.2, pp. 241–249 (cit. on p. 42).
- Shiller, Z. and H.H. Lu (1992). “Computation of path constrained time optimal motions with dynamic singularities”. In: *Journal of dynamic systems, measurement, and control* 114, p. 34 (cit. on p. 43).
- Shin, Kang and N McKay (1986). “Selection of near-minimum time geometric paths for robotic manipulators”. In: *Automatic Control, IEEE Transactions on* 31.6, pp. 501–511 (cit. on p. 41).
- Shin, Kang G and Neil D McKay (1985). “Minimum-time control of robotic manipulators with geometric path constraints”. In: *Automatic Control, IEEE Transactions on* 30.6, pp. 531–541 (cit. on p. 41).
- Shkolnik, Alexander, Matthew Walter, and Russ Tedrake (2009). “Reachability-guided sampling for planning under differential constraints”. In: *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, pp. 2859–2865 (cit. on pp. 25, 36).
- Škrjanc, Igor and Gregor Klančar (2010). “Optimal cooperative collision avoidance between multiple robots based on Bernstein–Bézier curves”. In: *Robotics and Autonomous systems* 58.1, pp. 1–9 (cit. on p. 36).
- Slotine, J.J.E. and H.S. Yang (1989). “Improving the efficiency of time-optimal path-following algorithms”. In: *IEEE Transactions on Robotics and Automation* 5.1, pp. 118–124 (cit. on pp. 41, 44).
- Sugihara, Tomomichi (2011). “Solvability-unconcerned inverse kinematics by the Levenberg–Marquardt method”. In: *Robotics, IEEE Transactions on* 27.5, pp. 984–991 (cit. on p. 179).
- Sugihara, Tomomichi, Yoshihiko Nakamura, and Hirochika Inoue (2002). “Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control”. In: *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*. Vol. 2. IEEE, pp. 1404–1409 (cit. on pp. 83, 119, 122, 123, 135).

- Suleiman, Wael, Fumio Kanehiro, Eiichi Yoshida, Jean-Paul Laumond, and André Monin (2010). “Time parameterization of humanoid-robot paths”. In: *Robotics, IEEE Transactions on* 26.3, pp. 458–468 (cit. on pp. 96, 117).
- Takano, Wataru and Yoshihiko Nakamura (2008). “Integrating whole body motion primitives and natural language for humanoid robots”. In: *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*. IEEE, pp. 708–713 (cit. on p. 162).
- Tedrake, Russ (2009). “LQR-Trees: Feedback motion planning on sparse randomized trees”. In: (cit. on p. 26).
- Tedrake, Russ, Scott Kuindersma, Robin Deits, and Kanako Miura (2015). “A closed-form solution for real-time ZMP gait generation and feedback stabilization”. In: *Proceedings of the International Conference on Humanoid Robotics* (cit. on p. 123).
- Trinkle, Jeffrey C, J-S Pang, Sandra Sudarsky, and Grace Lo (1997). “On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 77.4, pp. 267–279 (cit. on p. 69).
- Verscheure, D., B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl (2009). “Time-optimal path tracking for robots: A convex optimization approach”. In: *IEEE Transactions on Automatic Control* 54.10, pp. 2318–2327 (cit. on p. 41).
- Vukobratović, Miomir and J Stepanenko (1972). “On the stability of anthropomorphic systems”. In: *Mathematical biosciences* 15.1, pp. 1–37 (cit. on pp. 16, 119).
- Walker, Michael W and David E Orin (1982). “Efficient dynamic computer simulation of robotic mechanisms”. In: *Journal of Dynamic Systems, Measurement, and Control* 104.3, pp. 205–211 (cit. on pp. 29, 56).
- Wieber, P-B (2006). “Holonomy and nonholonomy in the dynamics of articulated motion”. In: *Fast motions in biomechanics and robotics*. Springer, pp. 411–425 (cit. on pp. 15, 23, 87, 91, 94).
- Wieber, Pierre-Brice (2002). “On the stability of walking systems”. In: *Proceedings of the international workshop on humanoid and human friendly robotics* (cit. on p. 23).
- Yamane, Katsu and Yoshihiko Nakamura (2000). “Dynamics computation of structure-varying kinematic chains and its application to human figures”. In: *Robotics and Automation, IEEE Transactions on* 16.2, pp. 124–134 (cit. on p. 101).
- Yershova, Anna, Léonard Jaillet, Thierry Siméon, and Steven M LaValle (2005). “Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain”. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, pp. 3856–3861 (cit. on p. 22).

Zheng, Yu and Katsu Yamane (2015). “Generalized Distance Between Compact Convex Sets: Algorithms and Applications”. In: *Robotics, IEEE Transactions on* 31.4, pp. 988–1003 (cit. on p. 124).

Zlajpah, L. (1996). “On time optimal path control of manipulators with bounded joint velocities and torques”. In: *IEEE International Conference on Robotics and Automation*. Vol. 2. IEEE, pp. 1572–1577 (cit. on p. 51).

Publications and software from this thesis

Submitted to journals

- Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “ZMP support areas for multi-contact mobility under frictional constraints.” Submitted to *IEEE Transactions on Robotics*.
- Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “Completeness of Randomized Kinodynamic Planners with State-based Steering.” Submitted to *Robotics and Autonomous Systems*.
- Quang-Cuong Pham, Stéphane Caron, Puttichai Lertkultanon, Yoshihiko Nakamura. “Admissible Velocity Propagation: Beyond Quasi-Static Path Planning for High-Dimensional Robots.” Submitted to *International Journal of Robotics Research*.
- Zhang Tianwei, Stéphane Caron, Yoshihiko Nakamura. “Humanoid Stair Climbing based on dedicated Plane Segment Estimation and Multi-contact Motion Generation.” Submitted to *International Journal of Humanoid Robotics*.

International conferences


IFTOMM 2015: Caron and Nakamura (2015a)

Stéphane Caron, Yoshihiko Nakamura. “Planning with the Center-of-Mass rather than Stances for Humanoids Walking on Uneven Terrains.” *14th World Congress in Mechanism and Machine Science*, Taipei, Taiwan, 25-30 October, 2015

RSS 2015: Caron et al. (2015a)


Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics.” *Robotics: Science and Systems (RSS)*, Roma,

Italy, July 2015.

 <https://github.com/stephane-caron/rss-2015>

ICRA 2015: Caron et al. (2015b)

Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “Stability of Surface Contacts for Humanoid Robots Closed-Form Formulae of the Contact Wrench Cone for Rectangular Support Areas.” *The 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, USA, May 2015.

 <https://github.com/stephane-caron/icra-2015>

ICRA 2014: Caron et al. (2014a)

Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “Completeness of Randomized Kinodynamic Planners with State-based Steering.” *The 2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong-Kong, China, June 2014.

RSS 2013: Pham et al. (2013b)

Quang-Cuong Pham, Stéphane Caron, Yoshihiko Nakamura. “Kinodynamic planning in the configuration space via Admissible Velocity Propagation.” *Robotics: Science and Systems (RSS)*, Berlin, Germany, June 2013.

National Conferences

RSJ 2015: Caron and Nakamura (2015b)

Stéphane Caron, Yoshihiko Nakamura. “Teleoperation System Design of Valve Turning Motions in Degraded Communication Conditions.” *The 33rd Annual Conference of the Robotics Society of Japan (RSJ)*, Tokyo, Japan, September 2015.

RSJ 2014: Caron et al. (2014b)

Stéphane Caron, Quang-Cuong Pham, Yoshihiko Nakamura. “Kinodynamic Motion Retiming for Humanoid Robots.” *The 32nd Annual Conference of the Robotics Society of Japan (RSJ)*, Fukuoka, Japan, September 2014.


RSJ 2013: Caron et al. (2013)

Stéphane Caron, Yoshihiko Nakamura, Quang-Cuong Pham. “Kinodynamic Motion Planners based on Velocity Interval Propagation.” *The 31st Annual Conference of the Robotics Society of Japan (RSJ)*, Tokyo, Japan, September 2013.

Software


pymanoid:

Backbone of the software stack described in Chapters 4-6. Features a custom numerical Inverse Kinematics (IK) solver, Jacobians and Hessians for the center-of-mass, ZMP and angular momentum, as well as the computation of the gravito-inertial wrench cone using the Double-description method.

 <https://github.com/stephane-caron/pymanoid>
(08e9544d24c3b3d02b7f5ab0e853ab7555dd59f3)⁴

manipulation_markers:

ROS package for manipulation tasks that we developed and used in Team Hydra while preparing for the DARPA Robotics Challenge (see Appendix B. This code is part of our controller for the valve turning task, which we demonstrated in Caron and Nakamura (2015b).


 https://github.com/stephane-caron/manipulation_markers
(3bcf1f20471525368bc6b53a130589b273d46f18)⁴

openth

Software developed by Team Hydra while preparing for the DARPA Robotics Challenge. My work in the team focused on the integration of the Inverse Kinematics solver (*Roki*⁵ from Sugihara (2011)) with the visualization and manipulation components. In particular, I developed the controller for the valve turning task, which we demonstrated in Caron and Nakamura (2015b).


TOPP:

The *Time-Optimal Path Parameterization* library by Quang-Cuong Pham. Given an input trajectory, TOPP finds its time-optimal retiming under a set of dynamic constraints, e.g., velocity limits, torque limits, and more generally any constraint that can be put in quadratic form. See (Pham, 2014) for details.

 <https://github.com/quangounet/TOPP>
(ef1688db4fc49b4dcba98e361696c9caadbe5631)⁴

AVP-RRT:

Implementation of the motion planners used in Chapter 2. The package includes a benchmarking framework (multi-processing, data logging and analysis tools) that we developed to compare our new planner to existing ones.

 <https://github.com/stephane-caron/rss-2013>
(d3d9b50bb582c23a4ee83408b26bc4d84469e)⁴

⁴ Revision number at the time of writing this manuscript.

⁵ <http://www.mi.ams.eng.osaka-u.ac.jp/open-e.html>

List of Figures

1.1	Computational framework proposed in this thesis. For fast-rate outputs, high-level motion planning is done on low-dimensional models of the full robot dynamics, which we obtain by reduction of the force space (Chapters 4 and 5) and a proper generalization of the ZMP to multi-contact scenarios (Chapter 6). The output of the planner is then conveyed as objectives to a whole-body path tracker. We limit dimensional expansion by a decoupling of kinematic and dynamic constraints presented in Chapter 3. Theoretical guarantees are provided for the high-level planner (Chapter 2) and its integration with time-optimal control (Chapter 3). The tracker computes a reference joint-angle trajectory, which is independently tracked at best by each joint controller. Up-to-date joint states are fed back to the motion planner for plan updates. Contact planning (in gray) is another constitutive part of motion planning. Appendix A provides technical elements on this component, however the main focus of this thesis is on the high-level planner. . . .	18
2.1	Illustration of the extension routine of randomized planners. To grow the roadmap toward the sample x' , the planner selects a number of parents $\text{PARENTS}(x') = \{P_1, P_2, P_3\}$ from which it applies the $\text{STEER}(P_i, x')$ method.	25
2.2	Single (A) and double (B) pendulums. Under torque bounds, these systems must swing back and forth several times before they can reach for the upright position, as depicted in (B) (lighter images represent earlier times).	30
2.3	Phase-space portrait of the roadmap constructed by RRT using the second-order continuous (SOC1) interpolation. The planner found a successful trajectory (red line) after 26,300 extensions. This planner is probabilistically complete (Theorem 1) thanks to the fact that SOC1 curves satisfy Assumption 5.	39

2.4	Roadmap constructed by RRT after 100,000 extensions using the Bezier interpolation. Reachable states are distributed in two major areas: a central, diamond shape corresponding to the states that the planner can connect at any rate, and two cones directed towards the goal ($\theta = \pi$ or $\theta = -\pi$). Even after several days of computations, this planner could not find a successful motion plan. Our completeness theorem does not apply to this planner because Bezier curves do not satisfy Assumption 5.	39
3.1	A: Illustration for Maximum Velocity Curve (MVC) and Concatenated Limiting Curve (CLC). The optimal velocity profile follows the green β -profile, then a portion of the CLC, and finally the yellow α -profile. B: Illustration for the Switch Point Lemma.	44
3.2	Illustration for step A (computation of the LC's). Left: illustration for case A1. A profile that crosses an α -CLC violates the α bound. Right: illustration for case A3.	47
3.3	Illustration for step B: one can determine the maximum final velocity by integrating forward from $(0, \dot{s}_{\text{beg}}^*)$	48
3.4	Illustration for the predicate IS_VALID: one can assess whether a final velocity \dot{s}_{test} is valid by integrating backward from $(s_{\text{end}}, \dot{s}_{\text{test}})$	50
3.5	Illustration for AVP-RRT. The horizontal plane represents the configuration space while the vertical axis represents the path velocity space. Black areas represent configuration space obstacles. A vertex in the tree is composed of a configuration (blue disks), the incoming path from the parent (blue curve), and the interval of admissible velocities (bold magenta segments). At each tree extension step, one interpolates a smooth, collision-free path in the configuration space and propagates the interval of admissible velocities along that path using AVP. The fine magenta line shows one possible valid velocity profile (which is guaranteed to exist by AVP) “above” the path connecting q_{start} and q_{new}	53

- 3.6 Performances of AVP-RRT compared to that of a state-space RRT. Left: percentage of trials that have reached the goal area at given time instants for $\tau_{\max} = (11, 7)$. Right: individual plots for each trial. Each curve shows the distance to the goal as a function of time for a given instance (red: AVP-RRT, blue: RRT-40). Dots indicate the time instants when a trial successfully terminated. Stars show the mean values of termination times. 57
- 3.7 Performances of AVP-RRT compared to that of a state-space RRT. Left: percentage of trials that have reached the goal area at given time instants for $\tau_{\max} = (11, 5)$. Right: individual plots for each trial. Each curve shows the distance to the goal as a function of time for a given instance (red: AVP-RRT, blue: RRT-40). Dots indicate the time instants when a trial successfully terminated. Stars show the mean values of termination times. 57
- 3.8 RRT, k NN-RRT and VIP-RRT ran over 40 instances of a double inverted pendulum with torque limits (8 Nm on the first joint and 4 Nm on the second one). The horizontal axis shows computation time, while the vertical axis represents the number of successful planners (left) and the distance to the goal area (right) averaged over all runs. 58
- 3.9 Swinging up a fully-actuated double pendulum. A typical solution for the case $(\tau_1^{\max}, \tau_2^{\max}) = (11, 5)$ N·m, with trajectory duration 1.88 s (see also the attached video). **A:** The tree in the (θ_1, θ_2) space. The final path is highlighted in magenta. **B:** snapshots of the trajectory, taken every 0.1 s. Snapshots taken near the beginning of the trajectory are lighter. **C:** Velocity profiles in the (s, \dot{s}) space. The MVC is in cyan. The various velocity profiles (CLC, Φ , Ψ , cf. Section 3.2.2) are in black. The final, optimal, velocity profile is in dashed blue. The vertical dashed red lines correspond to vertices where 0 is a valid velocity, which allowed a discontinuity of the path tangent at that vertex. **D:** Torques profiles. The torques for joint 1 and 2 are respectively in red and in blue. The torque limits are in dotted line. Note that, in agreement with time-optimal control theory, at each time instant, at least one torque limit was saturated (the small overshoots were caused by discretization errors). 59

3.10	Illustration for the existence of an admissible velocity profile above an approximated path in the proof of completeness for AVP-RRT.	61
3.11	Approximation of a given smooth path in the proof of completeness for AVP-RRT.	61
4.1	Two different contact modes. Left: a sphere-to-plane contact yields a point positional constraint: one point on the link coincides with one point on the surface. This constraint binds the three linear DOF between the link frame and the environment frame, resulting in a three-dimensional contact force. The three angular DOF are not constraint, thus there is no contact torque. Right: a surface-to-surface contact yields a full positional constraint: the link and environment frame coincide. All six relative DOF are constrained, resulting in a six-dimensional contact wrench (force and torque).	66
4.2	Simple case of a block on table pushed by an external force f_a	67
4.3	Friction cone at the contact point C_i with local contact frame (t_i, b_i, n_i) , represented respectively alone (left), with the outer (middle) and inner (right) linear approximations. (This figure is inspired from Fig. 1 in Trinkle et al. (1997).)	69
4.4	Projection of a 2D polygon by elimination of the x coordinate (red interval \mathcal{P}_y) or of the y coordinate (blue interval \mathcal{P}_x).	71
4.5	Contact in the surface plane. The physics of contact is a continuous distribution of stress and pressure fields (illustrated on the left). We show that, under Coulomb friction, this distribution is equivalent (in terms of the resulting wrench) to contact forces lying in frictions cones at the corners of the contact polygon (right).	74
4.6	Complete expression of the ZMP support polygon, accounting for the complete contact stability condition (<i>i.e.</i> , taking into account not only roll and pitch, as is usually done, but also slippage and yaw rotations). Most of these inequalities are redundant and can be eliminated once the position of the COM is determined. However, in the general case any of them can be saturated.	86

4.7	ZMP support area (in green) for the complete contact stability condition, <i>i.e.</i> , taking into account translational and rotational friction. The foot dimensions are that of HRP-4. The area was computed for a static friction coefficient $\mu = 0.1$ (icy surface) and a COM height of 0.8 m. The polygon has eight edges; the other thirty-two inequalities of the complete system (Figure 4.6) could be eliminated once the position of the COM (red disc) was determined.	87
4.8	Snapshots of the retimed motion. Total duration is 7.3 s. Time stamps are shown below each frame. The motion stresses all components of the wrench cone. The first segment stresses the pitch by moving the COP forward. The second stresses the roll through arm motions. Meanwhile, the yaw component is stressed by chest-pitch motions. Finally, the waist performs an elliptic motion (back and forth, up and down) throughout the whole motion, thus stressing the translation of the contact foot.	89
4.9	Zoom on the contact forces computed by OpenHRP at the left foot. Each thin purple segment corresponds to one force. Time stamps are shown below each column. First row: retimed motion with CWC enforced. Second row: 10% acceleration of the motion, CWC is not enforced. Around 4.2 s, the contact surface degenerates to a line contact on the left edge of the foot. Third row: 15% acceleration of the motion, CWC is not enforced. The contact surface degenerates to a line before contact is lost and the humanoid falls.	90
5.1	Retimed trajectory profiles in the (s, \dot{s}) space. Maximum Velocity Curves are depicted by dotted cyan and magenta lines. The dashed blue line represents the initial trajectory ($\dot{s} = 1$) while the red curve corresponds to the retimed trajectory. It may follow but never crosses the MVCs. Shaded gray areas show the intervals where we disabled retiming, forcing the retimed profile to go follow $\dot{s} = 1$. Blue and green dots indicate discontinuity of the MVC and singular points, respectively.	102
5.2	Time-lapse of the original and retimed trajectories. The interval between two frames is 3.5 s. Total durations are 53 s for the original and 24 s for the retimed trajectory.	103

5.3	Execution of the planned motion by HRP-4 for climbing one step. The height of the step is 24 cm. The motion is generated at low velocity, which is the range of validity of our multi-contact stability criterion. The total execution duration is 1 min 30 s. Time stamps (in seconds) are indicated below each picture.	107
5.4	View of the generated COM trajectory in the transverse plane. Consecutive support areas are represented by blue polygons while the staircase is in brown. Dotted red lines depict the COM trajectory. Note that the sagittal vector is pointing rightward, so that the motion goes from left to right. The unit of both axes is the meter.	108
5.5	Testing robust static equilibrium. The robot right arm is supported by the ledge while its right foot is supported by an inclined box. The desired neighborhood around the gravity wrench is associated with $\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4\} = \{(0.15, 0, g), (-0.15, 0, g), (0, 0.15, g), (0, -0.15, g)\}$. We sampled one million random COM positions, those of which satisfy robust static equilibrium are depicted by a green dot. The aggregate of green dots outlines the shape of the robust static equilibrium wrench cone. The three images show views from the X, Y and Z directions respectively.	110
5.6	Stair climbing motion where contact stability is checked using our method. Time stamps are shown under each frame. The retimed motion is very fast as all underlying configurations are all statically stable.	114
5.7	Execution of the multi-contact whole-body trajectory on a humanoid robot in a physics simulator. Time stamps are written under each frame. The dynamic part of the motion occurs between 15 s and 16 s, where movements need to be fast in order to maintain contact stability. This segment is dynamically but <i>not quasi-statically</i> stable: at least one contact is lost when the motion is executed at lower velocities.	115

6.1	Overview of the construction we propose. The ZMP support area, including positive-pressure and frictional constraints, is computed in an arbitrary virtual plane (here, above the robot's head). For locomotion, linearized pendulum dynamics are obtained by regulation of the angular momentum. The shrinking of the support area incurred by this regulation is fully taken into account. A whole-body controller based on these developments finally enables multi-contact locomotion for arbitrary environments.	121
6.2	Example of computed support area in the dual-cone case. There are three contacts in total. Contacts 1 and 2 are made with the same horizontal surface and correspond to the feet of the robot. Contact 3 is located one meter above the others and made with a vertical surface in front of the robot. The virtual plane is taken horizontal and 50 cm above the feet's plane. Polygons \mathcal{P}^+ and \mathcal{P}^- used in the construction process are drawn in purple. Blue lines connect the vertices generated by contact 3. They illustrate that a single surface contact can yield vertices in both \mathcal{P}^+ and \mathcal{P}^- at the same time.	129
6.3	Geometric construction of the ZMP support area in the polygonal case. Ray generators of friction cones (red lines) are traced until they intersect the virtual plane, yielding points inside the support area (black dots). The support area (in green) is the convex hull of this set of points. The blue polygon corresponds to the individual support polygon of the right contact. These polygons can be used for an alternative construction of the ZMP support area suited to the problem of contact planning.	132

- 6.4 Snapshots of the motion generated by ZMP control of the COM *from above* under LP regulation. The scenario is designed so that the robot has to put its hand on the high platform (in the background) and right foot on the opposite tilted platform in order to perform an ample swing of the left leg that is otherwise impossible. In these simulations, the ZMP plane is taken one meter above the robot's COM. Green polygons in the plane above the robot's head are the respective support areas for each snapshot. The (virtual) linear pendula between ZMPs (points in the support polygons) and their attached COM are depicted by gray wires. The correctness of our dynamic-stability criterion was cross-validated by explicit computation of contact forces at each time instant (arrows at the corners of each contact surfaces). 140
- 6.5 Situation where the ZMP support area (green stripes) is smaller than the convex hull of ground contact points (blue polygon). The robot has its two feet (transparent red boxes) set one meter apart on a horizontal floor (in gray). Its COM is 50 cm above ground, and the friction at contact is $\mu = 0.5$. The ZMP cannot be located at the corner C_k of the convex hull, as it would imply that the contact force exerted at this point (in magenta) lies outside the friction cone C_k (in red). 145
- A.1 Extension toward a candidate COM of a four-link stance with four non-coplanar contacts: two rectangular feet and two point sticks. Contact points and surfaces are drawn in black. The blue polygon represents the stance's support area. The green cone depicts the "eclipse" of the polygon by the COM. It corresponds to the positions at which the stick S_2 can be put so that the resulting stance stabilizes the candidate COM. 154
- A.2 HYDRA humanoid robot. **Left:** geometric model in false colors. **Middle:** kinematic structure of the robot. **Right:** picture of the real robot. 156
- A.3 Stance sequence generated by our planner. A stick-carrying variant of the HYDRA humanoid evolves on a randomly-generated rubble-field. Our planner's state space is the two-dimensional plane of horizontal COM position. In the sequence above, each stance results from an extension toward a COM sub-goal. 158

A.4	Exploration of the RRT in the horizontal plane for the stance sequence depicted in Figure A.3. COM positions and trajectories are represented by green dots and lines, respectively. The green star corresponds to the starting COM position. Contact locations, either rectangular surfaces or stick points, are drawn in red while the blue areas correspond to superposed support polygons (plotted with transparency for readability).	158
B.1	Manipulation marker for a wheel mesh model. Control handles for the six degrees of freedom (translation and rotation) are depicted in red, green and blue for the x -, y - and z -axis respectively.	161
B.2	Interface of the Operator Control System with a model of the HRP4-R humanoid robot. Its head has been replaced with an adjustable plate (in red) for the laser-range sensor frame. Panels on the left: default “Displays” from RViz and a custom panel for IK control. Panels on the right: Ghost model, Walking, Language and Wheel. The ghost model is used to preview IK results before execution on the real robot. Walking markers can be used to control the Walking Pattern Generator (Santacruz and Nakamura, 2012). The language panel is used to query a large human-behavior database for postures (Takano and Nakamura, 2008). Finally, the wheel panel allows to add/remove the valve marker, update its scale, send the reaching and turning commands. It displays two sliders: one for the estimate of the current wheel rotation, and the second for the desired turn angle. (Points above the robot head are artifacts from the sensor.)	162

List of Tables

3.1	Comparing the success rate and search time of the configuration-space AVP-RRT planner versus its state-space counterpart. On instances with severe torque constraints, AVP-RRT always found solutions, and did so much faster than the state-space RRT in all our experiments.	56
5.1	Computation times (ms) for the stair climbing motion, averaged over 10 runs on an 8-core 3.00 GHz processor. Segments are identified by their time stamps (see Figure 5.6).	113
5.2	Computation times (ms) for the box climbing motion, averaged over 10 runs on an 8-core 3.00 GHz processor. Segments are identified by their time stamps (see Figure 5.7).	116
6.1	Gains and weights used in the differential IK tracker (N/A: no gain for tasks regulating accelerations)	141
6.2	Simulation and trajectory generation parameters	142

List of Abbreviations

A

AVP Admissible Velocity Propagation 45

C

CHCP Convex Hull of Contact Points 85, 120

CLC Concatenated Limiting Curve 44

COM Center Of Mass 82, 93, 101, 152

COP Center Of Pressure 80, 88

CP Capture Point 123

CWC Contact Wrench Cone 75, 96

E

EHA Electro-Hydrostatic Actuation 156

G

GIW Gravito-Inertial Wrench 94

GIWC Gravito-Inertial Wrench Cone 15, 96

I

IK Inverse Kinematics 111

L

LC Limiting Curve..... 44

LIP Linear Inverted Pendulum..... 135

LP Linear Pendulum 135

M

MVC Maximum Velocity Curve 43

O

OCS Operator Control System 159, 165

P

PRM Probabilistic Roadmap 15, 23

R

RRT Rapidly-exploring Random Trees 23, 153

T

TOPP Time-Optimal Path Parameterization 15, 41, 97

Z

ZMP Zero-tilting Moment Point 16, 82, 85, 88, 119

Nomenclature

C

\mathcal{C} Configuration space of the robot	21
conv Convex hull	69
$C(q)$ Joint Coriolis tensor	29

F

f^c Contact force	67
f^{gi} Resultant force of the gravito-inertial wrench	94

G

G Center of mass (point)	82, 93
P Linear momentum	93
g Gravity constant	83
$g(q)$ Joint gravity vector	29

J

J_{C_j} Translation jacobian computed at contact point C_j	91
J_i^{rot} Rotation jacobian of link i	92

L

L_G Angular momentum, taken at the center of mass 93

M

m Total mass of the robot 83

$M(q)$ Joint inertia matrix 29

N

n Degree of freedom of the robot 29

nonneg Non-negative span, that is, the set of positive linear combinations 69

O

ω_i Angular velocity of link i 92

Q

q Vector of joint angles 21

\dot{q} Vector of joint-angle velocities 21

\ddot{q} Vector of joint-angle accelerations 29

R

R_i Rotation matrix representing the orientation of link i 93

S

s Path index variable 42

T

t Time	21
τ Vector of actuated joint torques.....	29
U	
u Vector of control inputs (typically, actuated joint torques)	22
W	
w^c Contact wrench (resulting from all contact forces)	94
w^{gi} Gravito-inertial wrench.....	94
X	
\mathcal{X} State space (position and velocity) of the robot.....	21
x First-order state variable (that is, position and velocity) of the robot.....	21
Z	
Z Zero-tilting moment point.....	82
$[\cdot \times]$ Cross-product matrix of a vector	104

Colophon

This thesis was typeset with \LaTeX 2 ϵ . It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at <http://cleanthesis.der-ric.de/>.

Declaration

This work was completed solely and only with the help of the references provided in the bibliography. *Nil igitur fieri de nihilo posse fatendum est.*

Tokyo, February 1st, 2016, and March 10th, 2017

Stéphane Caron

