Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Automatique-Productique**

Arrêté ministériel : 7 août 2006

Présentée par

Alexander Sherikov

Thèse dirigée par **Bernard Brogliato**
et co-encadrée par **Pierre-Brice Wieber**

préparée au sein **de l'INRIA**
dans **l'École Doctorale EEATS**

Balance preservation and task prioritization in whole body motion control of humanoid robots

Thèse soutenue publiquement le **23 Mai 2016**,
devant le jury composé de :

Olivier Stasse
Directeur de Recherche CNRS au LAAS, Rapporteur
Pascal Morin
Chargé de Recherche INRIA en détachement à l'UPMC, Rapporteur
Abderrahmane Kheddar
Directeur de Recherche CNRS au LIRMM, Président
Bernard Brogliato
Directeur de Recherche INRIA, Directeur de thèse
Pierre-Brice Wieber
Chargé de Recherche INRIA, Co-encadrant de thèse

Abstract

One of the greatest challenges in robot control is closing the gap between the motion capabilities of humans and humanoid robots. The difficulty lies in the complexity of the dynamical systems representing the said robots: their nonlinearity, underactuation, discrete behavior due to collisions and friction, high number of degrees of freedom. Moreover, humanoid robots are supposed to operate in non-deterministic environments, which require advanced real time control. The currently prevailing approach to coping with these difficulties is to impose various limitations on the motions and employ approximate models of the robots. In this thesis, we follow the same line of research and propose a new approach to the design of balance preserving whole body motion controllers. The key idea is to leverage the advantages of whole body and approximate models by mixing them within a single predictive control problem with strictly prioritized objectives.

Balance preservation is one of the primary concerns in the control of humanoid robots. Previous research has already established that anticipation of motions is crucial for this purpose. We advocate that anticipation is helpful in this sense as a way to maintain capturability of the motion, *i.e.*, the ability to stop. We stress that capturability of anticipated motions can be enforced with appropriate constraints. In practice, it is common to anticipate motions using approximate models in order to reduce computational effort, hence, a separate whole body motion controller is needed for tracking. Instead, we propose to introduce anticipation with an approximate model into the whole body motion controller. As a result, the generated whole body motions respect the capturability constraints and the anticipated motions of an approximate model take into account whole body constraints and tasks. We pose our whole body motion controllers as optimization problems with strictly prioritized objectives. Though such prioritization is common in the literature, we believe that it is often not properly exploited. We, therefore, propose several examples of controllers, where prioritization is useful and necessary to achieve desired behaviors. We evaluate our controllers in two simulated scenarios, where a whole body task influences walking motions of the robot and the robot optionally exploits a hand contact to maintain balance while standing.

Résumé

Un des plus grands défis dans la commande des robots est de combler l'écart entre la capacité de mouvement de l'humain et des robots humanoïdes. La difficulté réside dans la complexité des systèmes dynamiques représentant les robots humanoïdes: la nonlinéarité, le sous-actionnement, le comportement non-lisse en raison de collisions et de frottement, le nombre élevé de degrés de liberté. De plus, les robots humanoïdes sont censés opérer dans des environnements non-déterministes, qui exigent une commande temps réel avancée. L'approche qui prévaut actuellement pour faire face à ces difficultés est d'imposer diverses restrictions sur les mouvements et d'employer des modèles approximatifs des robots. Dans cette thèse, nous suivons la même ligne de recherche et proposons une nouvelle approche pour la conception de contrôleurs corps entier qui preservent l'équilibre. L'idée principale est de tirer parti des avantages des modèles approximatifs et de corps entier en les mélangeant dans un seul problème de contrôle prédictif avec des objectifs strictement hiérarchisés.

La préservation de l'équilibre est l'une des principales préoccupations dans la commande des robots humanoïdes. Des recherches antérieures ont déjà établi que l'anticipation des mouvements est essentiel à cet effet. Nous préconisons que l'anticipation est utile dans ce sens comme un moyen de maintenir la capturabilité du mouvement, *i.e.*, la capacité de s'arrêter. Nous soulignons que capturabilité des mouvements prévus peut être imposée avec des contraintes appropriées. Dans la pratique, il est fréquent d'anticiper les mouvements du robot à l'aide de modèles approximatifs afin de réduire l'effort de calcul, par conséquent, un contrôleur séparé de mouvement du corps entier est nécessaire pour le suivi. Au lieu de cela, nous proposons d'introduire l'anticipation avec un modèle approximatif directement dans le contrôleur corps entier. En conséquence, les mouvements du corps entier générés respectent les contraintes de capturabilité et les mouvements anticipes du modèle approximatif prennent en compte les contraintes et les tâches désirées pour le corps entier. Nous posons nos contrôleurs du mouvement du corps entier comme des problèmes d'optimisation avec des objectifs strictement hiérarchisés. Bien que cet ordre de priorité soit commun dans la littérature, nous croyons qu'il est souvent mal exploité. Par conséquent, nous proposons plusieurs exemples de contrôleurs, où la hiérarchisation est utile et nécessaire pour atteindre les comportements souhaités. Nous évaluons nos contrôleurs dans deux scénarios simulés, où la tâche du corps entier du robot influence la marche et le robot exploite éventuellement un contact avec la main pour maintenir son équilibre en étant debout.

Acknowledgements

First and foremost, I would like to express my highest gratitude to Pierre-Brice Wieber and Dimitar Dimitrov, who guided me through the past years of my study. Their insights and advices were invaluable for my work. I would also like to thank Bernard Brogliato and jury members Abderrahmane Kheddar, Olivier Stasse, and Pascal Morin for their time and interest in this work.

Some of my research was performed in collaboration with Camille Brasseur and Nestor Bohorquez-Dorante from BIPOP team, and Joven Agravante from LIRMM. I must thank them for their effort and feedback.

I would like to personally thank Alejandro Blumentals and Jan Michalczyk, as well as other present and past members of BIPOP team, for their general assistance and support. I am also grateful to Diane Courtiol and Myriam Etienne from INRIA for their administrative assistance during my stay in France.

As an active user of free software during most of my professional life, I cannot forget the people who invest their time and effort in development of software tools and packages, which I employed during my work on this thesis: `Asymptote`, `FreeBSD`, `git`, `Maxima`, `Octave`, `qpOASES`, `TeX Live`, `Vim`, and many others. Thank you all!

Contents

List of Figures

List of Acronyms

CAS	Computer Algebra System
CoM	Center of Mass
CoP	Center of Pressure
CPU	Central Processing Unit
FSM	Finite State Machine
KKT	Karush-Kuhn-Tucker
LIPM	Linear Inverted Pendulum Model
MMPC	Mixed Model Predictive Control
MPC	Model Predictive Control
PD	Proportional-Derivative
PLLS	Prioritized Linear Least-Squares
QP	Quadratic Program
SCARA	Selective Compliance Assembly Robot Arm
ZMP	Zero Moment Point

Chapter 1

Introduction

1.1 Context and motivation

Humanoid robotics is undoubtedly one of the most fascinating areas in robotics research. The reason for this is that the creation of robots, which are comparable to humans at least in their motion capabilities, is expected to have a tremendous social and industrial impact [Kemp 2008]. At the present moment, we are, however, quite far from this ultimate goal, even though some remarkable progress has been made in the recent years [DRC 2015]. While we hope that, eventually, humanoid robots will perform dangerous and tedious tasks in our place, some valuable gains from studying these robots can be obtained in short term. First of all, humanoid robots are very versatile and a challenging research and educational platform, since they have to perform a large variety of activities, such as sensing, manipulation, locomotion, interaction with the environment, and others. Furthermore, these robots can be employed in entertainment or play a role as an interactive interface for humans [Kemp 2008]. Control algorithms developed for humanoid robots may find applications in prosthetic devices and exoskeletons, or in the generation of natural and functional motions of animated characters [Van De Panne 1997].

In the present thesis, we limit our focus to the motion control of humanoid robots. This is a difficult problem due to the intrinsic complexity of the dynamical systems representing the said robots: their nonlinearity, underactuation, discrete behavior due to collisions and friction, high number of degrees of freedom. Even the modeling and simulation of these systems is a challenging problem on its own. Moreover, humanoid robots are supposed to operate in non-deterministic environments, which require advanced real time control. This real time control must often be realized using relatively weak on-board computers, since a network connection to a more powerful computer is not available or unreliable. Hence, there is a need for the development of controllers, which adequately trade off computational complexity with quality and generality of realizable motions. This is usually achieved by imposing various limitations on the motions and employing approximate models of the robots. Approximate models lack expressiveness of whole body models, but their utilization results in smaller demands of computational resources.

The complexity of the kinematic structure of humanoid robots makes them redundant with respect to typical motion objectives, *e.g.*, the robot can perform different tasks with its hands simultaneously. Consequently, it is common to employ prioritization of these objectives in the controllers in order to exploit this redundancy [Kanoun 2009, Saab 2013, Sentis 2007]. Sometimes, however, prioritization is not well justified and is not meaningful.

The first goal of the present work is to demonstrate that the advantages of whole body and an approximate model can be leveraged by mixing them within a single predictive control problem. The second goal is to design controllers, in which prioritization of objectives is

useful and necessary to achieve new behaviors.

1.2 Contribution

- One of the primary contributions of the present work is the introduction of a novel approach to the design of whole body motion controllers for humanoid robots. The key idea of the proposed approach is to employ the whole body model and an approximate model of the robot simultaneously within a single controller [Sherikov 2014, Sherikov 2015]. The role of the whole body model is to allow instantaneous whole body motion control, while the approximate model is used for anticipation to ensure long term balance. Such a mix of models enables interplay between them, in particular, instantaneous whole body tasks can influence motions anticipated with an approximate model.

  The resulting controller can be perceived as a whole body *Model Predictive Control* (MPC) problem, where the whole body model is replaced by an approximate model everywhere except the current time instant. Hence, in comparison with the whole body MPC, our controllers sacrifice quality of the prediction for computational performance.
- The second contribution consists in several practical applications of strict prioritization of the objectives in the proposed whole body motion controllers and general MPC:
  - We propose to employ prioritization of state constraints in an MPC to implement a time optimal controller for an industrial manipulator [Homsi 2016b].
  - We introduce prioritization in contact force distribution in order to avoid the application of an optional hand contact force unless it is necessary for the execution of a whole body task or the preservation of balance [Sherikov 2015].
- Finally, there is a number of technical contributions:
  - We implement coupling between the whole body model and two approximate models within a single controller and consider possible difficulties in this coupling [Sherikov 2014, Sherikov 2015].
  - We discuss the construction of capturability constraints, which ensure that the robot can be stop without a fall, for several approximate models.
  - Our whole body motion controllers with prioritized objectives serve as a testbed for a general purpose solver of *Prioritized Linear Least-Squares* (PLLS) problems described in [Dimitrov 2015].
  - We formulate the whole body motion controllers so that they are solved by the said solver in the most computationally efficient way.
  - We evaluate the proposed controllers in simulations and develop the necessary software for performing these simulations.
  - We contribute to the development of MPC schemes for carrying objects in collaboration with humans [Agravante 2016b] and for walking with varying *Center of Mass* (CoM) height [Brasseur 2015b].

1.2.1 List of publications

The work on this thesis resulted in the following publications in peer-reviewed conferences

- A. Sherikov, D. Dimitrov and P.-B. Wieber. *Whole body motion controller with long-term balance constraints.* In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 444–450, Nov 2014

- A. Sherikov, D. Dimitrov and P.-B. Wieber. *Balancing a humanoid robot with a prioritized contact force distribution.* In Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on, Nov 2015
- C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov and P.-B. Wieber. *A robust linear MPC approach to online generation of 3D biped walking motion.* In Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on, Nov 2015
- D. J. Agravante, A. Sherikov, P.-B. Wieber, A. Cherubini and A. Kheddar. *Walking pattern generators designed for physical collaboration.* In Robotics and Automation (ICRA), 2016 IEEE International Conference on, May 2016
- S. A. Homsi, A. Sherikov, D. Dimitrov and P.-B. Wieber. *A hierarchical approach to minimum-time control of industrial robots.* In Robotics and Automation (ICRA), 2016 IEEE International Conference on, May 2016

The author also contributed to two journal papers, which are currently under review

- D. Dimitrov, A. Sherikov and P.-B. Wieber. Efficient resolution of potentially conflicting linear constraints in robotics. Preprint, August 2015
- D. J. Agravante, A. Cherubini, A. Sherikov, P.-B. Wieber and A. Kheddar. Human-Humanoid Collaborative Carrying. working paper or preprint, 2016

1.3 Outline

This dissertation is composed of 5 main Chapters 2–6, a conclusive Chapter 7 and appendices. We start with discussion of our general point of view on balance preservation of humanoid robots in Chapter 2. We declare anticipation and capturability, *i.e.*, the ability to stop, to be the basic concepts, which we employ for the construction of balance preserving controllers. The next, rather technical, Chapter 3 is devoted to the presentation of whole body and approximate models employed in this work. In Chapter 4, we move on to anticipation using approximate models. We also consider capturability enforcing constraints for various approximate models, and introduce the idea of mixing whole body and approximate models. Chapter 5 is focused on the *Prioritized Linear Least-Squares* (PLLS) optimization framework, which is used to define our controllers. We outline the idea of prioritization of objectives and give several examples of its applications. Finally, we discuss techniques, which allow for efficient solution of PLLS problems. In Chapter 6, we consider two whole body motion controllers presented in [Sherikov 2014, Sherikov 2015] in detail and discuss their performance in simulations. In addition to that, we overview results of several works performed in collaboration.

1.4 Notation

Software names

Names of programs and software libraries, names of constants, variables and functions that are used in programs are typed in a monospaced font: `Eigen`, `LexLS`.

General scalars, vectors, matrices

- Vectors and matrices are denoted by letters in a bold font: $\boldsymbol{v}$, $\boldsymbol{M}$, $\boldsymbol{\mathcal{A}}$.
- Scalars are denoted using the standard italic of calligraphic font: $N, n, \mathcal{K}$.
- $(\cdot)^\top$ – transpose of a matrix or a vector.

- $(\cdot)^\times$ – a skew-symmetric matrix used for representation of a cross product of two three dimensional vectors as a product of a matrix and a vector:

$$\boldsymbol{v} = \begin{bmatrix} v^x \\ v^y \\ v^z \end{bmatrix}, \quad \boldsymbol{v}^\times = \begin{bmatrix} 0 & -v^z & v^y \\ v^z & 0 & -v^x \\ -v^y & v^x & 0 \end{bmatrix}. \tag{1.1}$$

- Block diagonal matrices:

$$\begin{gathered} \underset{2}{\operatorname{diag}}(\boldsymbol{M}) = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}, \quad \underset{k=1\ldots2}{\operatorname{diag}}\left(\boldsymbol{M}_k\right) = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_2 \end{bmatrix}, \\ \operatorname{diag}(\boldsymbol{M}, \boldsymbol{R}) = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix}. \end{gathered} \tag{1.2}$$

- Stacked vectors and matrices:

$$\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_n \end{bmatrix}, \quad \boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_n) = \begin{bmatrix} \boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{M}_n \end{bmatrix}. \tag{1.3}$$

- Inequalities between vectors $\boldsymbol{v} \geq \boldsymbol{r}$ are interpreted component-wise.

Special matrices and vectors

- $\boldsymbol{I}$ – an identity matrix. $\boldsymbol{I}_n$ – $n \times n$ identity matrix.
- $\boldsymbol{I}_{(\cdot)}$ – a selection matrix, examples:
  - $\boldsymbol{I}_{\boldsymbol{\tau}}$ – a torque selection matrix;
  - $\boldsymbol{I}_\alpha$, where α is a combination of $\{x, y, z\}$, selects components of a 3d vector:

$$\boldsymbol{I}_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I}_{xz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{I}_{zy} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{1.4}$$

- $\boldsymbol{0}$ – a matrix of zeros. $\boldsymbol{0}_{n,m}$ – $n \times m$ matrix of zeros.
- $\boldsymbol{J} = (\boldsymbol{J}_\uparrow, \boldsymbol{J}_\circlearrowleft)$ – a Jacobian matrix with translational $\boldsymbol{J}_\uparrow$ and rotational $\boldsymbol{J}_\circlearrowleft$ parts.

Reference frames

- Frames are denoted using a sans-serif font: A. All considered frames are orthonormal.
- ${}^{\mathsf{A}}\boldsymbol{v}$ – vector expressed in frame A.
- ${}^{\mathsf{A}}\boldsymbol{R}_{\mathsf{B}}$ – rotation matrix from from frame B to frame A.
- The global frame is implicit and is not denoted by any letter, *e.g.*, $\boldsymbol{R}_{\mathsf{B}}$ rotates from frame B to the global frame.

Sets

- The sets are denoted using a blackboard bold font: $\mathbb{A}$.
- $\mathbb{R}$ is the set of real numbers.
- $\mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$ are the sets of non-negative and positive real numbers.
- $\mathbb{R}^n$ is the set of real-valued vectors.
- $\mathbb{R}^{n \times m}$ is the set of real-valued matrices.

Other

- Function names in mathematical expressions are written in the regular font: $\operatorname{func}(\boldsymbol{x}, \boldsymbol{y})$.
- $\|\cdot\|_2$ denotes the Euclidean norm.

Chapter 2

Balance of a humanoid robot

One of the primary topics of the present thesis is preservation of balance by humanoid robots. Though this problem has been studied extensively, the definition of balance itself is often too vague or is limited to a particular application, such as walking. Therefore, we start by defining the balance preservation problem in a general and informal way using the concepts of *viability theory* [Aubin 2011], which has already been employed in [Wieber 2002, Koolen 2012, Zaytsev 2015a]. Thereafter we discuss other related concepts in robotics and control and interpret the existing approaches to balance preservation using the viability concepts.

2.1 Balance preservation as a viability problem

Consider a general dynamical system

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t), & \text{(2.1a)} \\ \boldsymbol{x} \in \mathbb{X}(\boldsymbol{x}, t) \subset \mathbb{R}^n, & \text{(2.1b)} \\ \boldsymbol{u} \in \mathbb{U}(\boldsymbol{x}, t) \subset \mathbb{R}^m, & \text{(2.1c)} \end{cases}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are the state and control vectors subject to varying constraints. Violation of the constraints indicates a failure, *i.e.*, an undesirable or infeasible state-control pair. An infinitely long evolution of the $(\boldsymbol{x}, \boldsymbol{u})$ pair is called *viable*, if it always complies with the constraints, and, hence, never results in a failure. Accordingly, we call a state $\boldsymbol{x}$ *viable*, if at least one viable evolution starts from it. All viable states define the *viability kernel* $\mathbb{V}$. In practice, we are interested in reaching some *target set* $\mathbb{T} \subset \mathbb{V}$. The set of states from which the target can be reached is referred to as the *capture basin* $\mathbb{C}$ ($\mathbb{T} \subset \mathbb{C} \subset \mathbb{V}$). For convenience, we also introduce the set $\mathbb{C}_H \subset \mathbb{C}$, such that the target can be reached from it within a time interval H. Whenever $\boldsymbol{x} \in \mathbb{C}$ we call $\boldsymbol{x}$ *capturable*.

Intuitively, preservation of balance means avoiding *falls* at all future moments. This definition immediately suggests interpretation using the viability concepts presented above. Let us assume that we have a set of constraints, violation of which indicates a fall. Then the balance is ensured if the state of the robot stays within the viability kernel $\mathbb{V}$ constructed based on these constraints. In general, it is difficult to check if a state is inside of the viability kernel, since the computation of $\mathbb{V}$ appears to be intractable in the case of such complex systems as humanoid robots [Wieber 2002]. However, it is usually possible to isolate a target subset $\mathbb{T} \subset \mathbb{V}$ consisting of states, which can be easily demonstrated to be balanced. For example, $\mathbb{T}$ may contain states, in which the robot is stopped, or states comprising a cycle in $\mathbb{V}$. Later in this chapter we will see that all common approaches to balance preservation explicitly or implicitly make use of $\mathbb{T}$ or the respective capture basin $\mathbb{C}$.

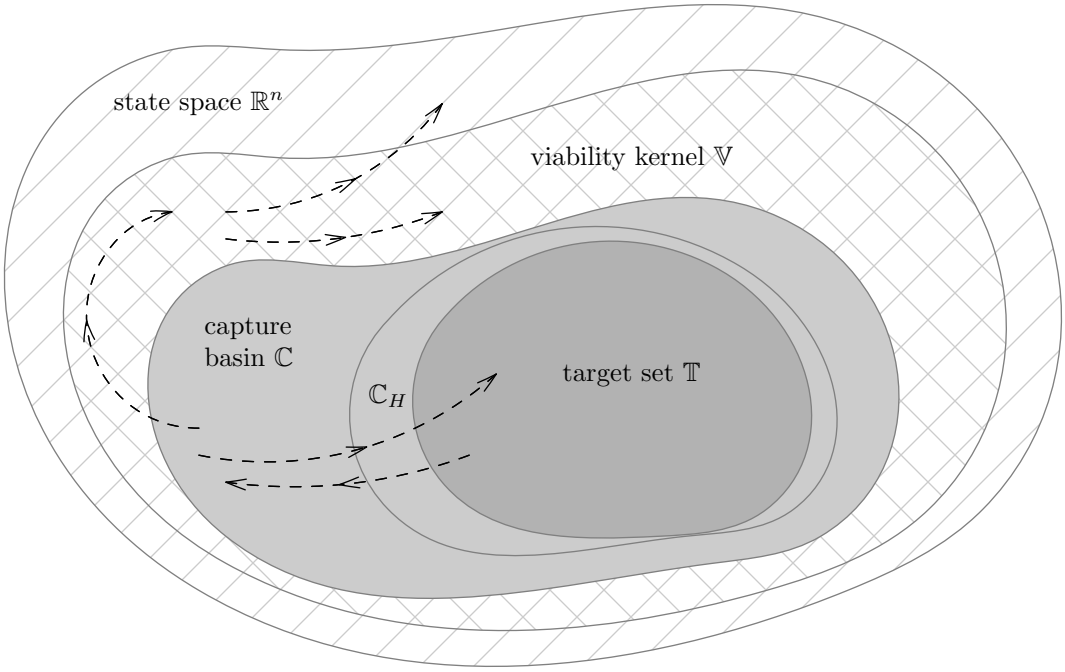


Figure 2.1: Viability kernel, capture basin, and target set. Possible evolutions are shown with dashed curves.

Generally speaking, falling is not the only type of a failure, which may happen to a robot. Depending on the context, failures may be defined as violations of hardware limits, collisions with the environment, and even undesirable behaviors, such as bending of the torso while walking. If these failures can be indicated by some constraints, they can also be described in terms of viability theory. Therefore, while controlling the robot we should aim at staying within the viability kernel $\mathbb{V}$ defined with respect to constraints reflecting all possible failures. However, in practice it is usually necessary to consider constraints of different types separately in sequential schemes for control or planning to tackle the concomitant computational complexity. In the rest of the thesis the sets $\mathbb{V}$, $\mathbb{C}$, and $\mathbb{T}$ always take into account the constraints due to balance preservation and, optionally, other constraints, which are added depending on the context. Thus, by saying that a particular state is viable or capturable we imply that a fall can be avoided starting from it.

Though the main control goal is to avoid failures and, in particular, preserve balance, we argue that this is not sufficient in practice, since we almost always want the robot to eventually stop, or at least be able to stop. In other words, the ultimate control goal is not to stay within $\mathbb{V}$, but rather to stay within the capture basin $\mathbb{C}$ corresponding to the target set $\mathbb{T}$ composed of the states in which the robot is stopped. This choice of control goal may appear to be of a philosophical importance due to negligible difference between the capture basin $\mathbb{C}$ and viability kernel $\mathbb{V}$ constructed for some simplified models of humanoid robots [Zaytsev 2015b]. However, our point of view has an advantage when it is necessary to design a controller without explicit knowledge of $\mathbb{C}$ and $\mathbb{V}$: ensuring $\boldsymbol{x} \in \mathbb{C}_H$ is easier than $\boldsymbol{x} \in \mathbb{V}$.

2.1.1 Related terms and concepts

Legged locomotion is one of the most common tasks of humanoid robots. At the same time, stepping allows to compensate for perturbations, for example, pushes. For these reasons, the general problem of balance preservation is often perceived as a problem of making proper steps. The number of steps required to stop the robot, and positions of these steps are of

particular interest, which was recognized as early as 70's [Yamashita 1974]. Consequently, suitable terminology was introduced in the recent works [Pratt 2006, Koolen 2012], where a certain state is called *K-step capturable*, if the robot can stop by making no more than K steps. Though this term does not originate from viability theory, it does not contradict with the definition of capturability given above. In particular, the relation between the capture basin $\mathbb{C}$ when the target is to stop the robot and the set of K-step capturable states $\mathbb{K}_K$ is described by

$$\mathbb{C} = \bigcup_{K \geq 0} \mathbb{K}_K. \tag{2.2}$$

Some researchers focused on the case, when it is necessary to maintain a desired walking or running cycle [Carver 2009, Zaytsev 2015a]. Carver studied *K-step deadbeat* control laws, which allow to compensate for disturbances in K steps while running [Carver 2009]. Zaytsev introduced *K-step controllability* as generalization of K-step capturability to the cases, when the goal can also be to maintain a given walking speed [Zaytsev 2015b]. This case can also be described using the notion of capturability as defined here: the relation between the set of K-step controllable states and the capture basin is equivalent to (2.2).

Viability theory can be applied to many problems in robotics. Hence, it is not surprising, that some of the application-specific terminology introduced in the field is equivalent to the notions of viability or capturability. For example, *inevitable collision states* defined in the context of collision avoidance comprise the complement of a viability kernel [Fraichard 2003]. Another example is the *alternative safe behaviors* constructed in [Rubrecht 2012] to avoid violation of the control constraints of a manipulator. These behaviors are to be triggered to safely stop the robot, when there is no other way to avoid violation of the constraints. Essentially, this allows to maintain capturability at all times. Schouwenaars introduced *terminal feasible invariant sets*, "*in which a vehicle can remain for an indefinite period of time without colliding with obstacles or other vehicles*", *i.e.*, without violating constraints of the system [Schouwenaars 2006, Chapter 4]. These sets are conceptually equivalent to the target sets $\mathbb{T}$ considered here. Consequently, existence of a trajectory, which ends in a terminal feasible invariant set, establishes that the current state of a vehicle is capturable with respect to the system constraints.

Related concepts are also used in general control theory. For example, *recursive feasibility* property of *Model Predictive Control* (MPC) implies feasibility with respect to constraints at each future control iteration [Rossiter 2003, Chapter 8]. Hence, recursive feasibility implies viability. Some researchers have already pointed out the connection between viability theory and the notions of *controllability* and *reachability* [Zaytsev 2015a, Wieber 2015] or *control invariant sets* [Wieber 2002]. Typically, these notions are employed for simple cases of linear systems or systems with non-varying constraints [Sontag 1998, Blanchini 2008, Borrelli 2015, Kerrigan 2000]. As a result, there exists a vast variety of extensions and specifications meant to adapt controllability, reachability, or control invariant sets to nonlinear and hybrid systems [Sontag 1998, Tomlin 2003, Bemporad 2000, Lewis 1999], to obstacle avoidance constraints [LaValle 2006, Chapter 15] or general varying constraints [Rawlings 2009, Chapter 2]. We believe that such variety may lead to a confusion, which can only be resolved by rigorous definitions. For this reason, here we resort to the concepts of viability theory.

2.2 Balance preservation in different settings

2.2.1 Standing

The states, in which the robot can stay infinitely long without falling, are called *statically balanced*. A particular state is statically balanced, if it complies with the hardware limitations

and the vertical projection of the *Center of Mass* (CoM) of the robot stays within the *support area*, which depends on the configuration of the contacts with the environment [Wieber 2002, Bretl 2008, de Santos 2005]. Though the statically balanced states *per se* are of little interest for motion control, they can be used to define the target set $\mathbb{T}$ and the corresponding capture basin $\mathbb{C}$. Such a definition of $\mathbb{T}$ and $\mathbb{C}$ is very natural, since we usually want the robot to stop at some point in the future.

It is common to employ the conditions on the CoM for static balance when the robot is not motionless. The corresponding motions are said to be *quasi-statically balanced*. This approach is widely used to reject disturbances while standing or avoid falls while performing simple tasks. One way to achieve this is to maintain a statically balanced reference configuration of the body or simply a position of the CoM. A less restrictive alternative is to allow motion of the CoM projection within the support area. Numerous examples of this approach can be found in the literature using various control methods: standard inverse kinematics [Escande 2014] or dynamics [Collette 2007], passivity-based [Hyon 2007, Ott 2011], MPC [Henze 2014].

2.2.2 Regular walk

One can observe that walking is an intrinsically cyclic process (see Figure 2.2). Consequently, any balanced regular gait has a corresponding cycle of states in $\mathbb{V}$. In order to produce such a gait, the system has to stay on this cycle comprising the target set $\mathbb{T}$ and return to it after perturbations.

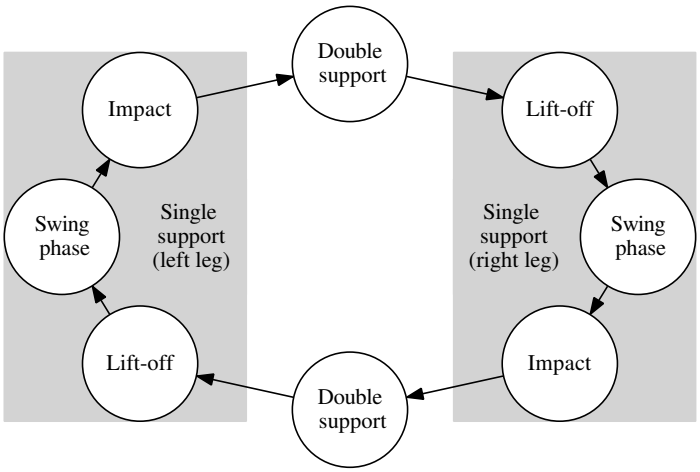


Figure 2.2: Walking cycle of a biped including two *single support* phases, when only one foot is on the ground, and two *double support* phases, when both feet are on the ground. One single support and one adjacent double support comprise a *step*.

This goal can be achieved without any actuation by specially designed mechanical systems called *passive walkers* [McGeer 1990]. They start walking when positioned on sloped surfaces, where the gravitational force allows to restore energy lost at foot impacts. Walking on a flat ground can be enabled by exploiting different sources of energy, for example, heating of the feet [Nemoto 2015] or limited actuation [Collins 2005]. During the walk, the state of passive walkers evolves along a semi-stable limit cycle $\mathbb{T}$, whose basin of attraction is the capture basin $\mathbb{C}$. Therefore, they can tolerate small perturbations.

The cyclic nature of walking is also exploited in the control of actuated robots. In particular, the control of a robot as a hybrid system is significantly simplified due to the sequence of discrete collision events known in advance. The design and stability analysis of hybrid system controllers is often further simplified with the help of *virtual constraints* which shape the

walking motion. This approach is referred in the literature as *hybrid zero dynamics* or *hybrid restriction dynamics* [Westervelt 2003, Chevallereau 2003, Grizzle 2006, Westervelt 2007]. Typically, it deals with only planar walking, however, some attempts have been made to apply it to three dimensional walking as well [Chevallereau 2009]. Other researches have also extended this approach to compliant robots [Sreenath 2010].

A completely different approach to generation of periodic gaits was inspired by the neurobiological research. This research indicated a major role of groups of neurons which generate periodic signals in locomotion of animals. Emulation of such groups called *central pattern generators* has been successfully employed in robot locomotion control, even though the balance of the resulting motion is not usually rigorously proven [Fukuoka 2003, Nakanishi 2004, Ijspeert 2008].

The common disadvantage of all these methods is the lack of flexibility: they are limited to walking and cannot produce acyclic gaits. Sometimes it is also unclear how to start and terminate the walk or change the walking gait without falling.

2.2.3 General motion and locomotion

Potential practical applications of humanoid robots, such as disaster response [DRC 2015] or airplane assembly [COM 2016], lie far beyond the scope of standing or regular walking. Hence, the problem of balance preservation while performing much more general motions must be addressed. This is usually achieved as a by-product of motion planning [LaValle 2006], which ensures execution of the motion goals without falling. We can perceive the motion planning as a procedure of finding a viable evolution connecting an initial state $\boldsymbol{x}_s$ and a final state $\boldsymbol{x}_f \in \mathbb{T}$. Whenever $\boldsymbol{x}_s \notin \mathbb{C}$ the procedure must necessarily fail. Note that the sets $\mathbb{T}$ and $\mathbb{C}$ are assumed to be defined based on the constraints taking into account all possible failures.

Determination of a single viable evolution appears to be easier than determination of the whole $\mathbb{V}$ and $\mathbb{C}$ sets. Nevertheless, it is a very challenging problem, which is usually solved with the help of various approximations and heuristics. Motion planning in complex situations is often decoupled into several stages [LaValle 2006, Chapter 14]. This strategy is ubiquitous in planning for legged robots. One of the most commonly used stages is planning of *stances* – configurations of the contacts of the robot with the environment [Bouyarmane 2012, Bretl 2006, Zucker 2011]. Note that, as we have already indicated in Section 2.1, some of the constraints are neglected on different stages to reduce computational costs.

An important property of a planning algorithm is *completeness*, *i.e.*, the ability to find a solution if one exists. In the case of legged robots, it is common to resort to weaker *probabilistic completeness* [Kuffner Jr. 2002, Hauser 2008, Dalibard 2013, Shkolnik 2011] or to *resolution completeness* [Zucker 2011]. The former guarantees that the probability of finding a solution approaches 1 as the computation time increases, the latter is valid for certain resolution of the state space. Sometimes the completeness is sacrificed by the approaches solely based on optimization, which may get stuck in a local minimum and, therefore, fail to find a solution [Dai 2014, Kanoun 2010].

In spite of an abundance of techniques making motion planning computationally cheap, it still cannot be performed in real-time. Real-time replanning is, however, crucial for adaptation to unexpected changes in the environment and compensation for strong external perturbations [Nishiwaki 2009]. One way to address this issue is to adjust the precomputed plan if the reality deviates from it. This may still be intractable in real-time, and a more common alternative is generation of a short-term plan, which we call motion *anticipation* or motion *preview*. The performance gain in this case is achieved due to limited preview horizon, lack of completeness, and, in most cases, approximated representations of the robot [Kajita 2003, Nagasaka 2012, Audren 2014, Nishiwaki 2009]. Low computational cost allows regeneration of a short-term plan at high frequency, which

enables highly reactive motions. Anticipation can be used to return to the precomputed plan or can even render it unnecessary for simple motions such as locomotion [Azevedo 2002, Wieber 2006b, Herdt 2010, Tassa 2014, Sherikov 2014].

A planned motion is balanced if it complies with all the constraints at all times and terminates in a final state. The final state, however, usually cannot be reached, when anticipation over a limited horizon is performed. Consequently, there is a possibility of reaching a non-viable state. We are not aware of possibilities to ensure viability without knowledge of the viability kernel $\mathbb{V}$. Nevertheless, it is possible to ensure more restrictive capturability by imposing that the robot must be able to stop or reach certain walking cycle in the end of preview horizon, *i.e.*, by imposing that the anticipated motion reaches $\mathbb{T}$ [Sherikov 2014, Sherikov 2015, Schouwenaars 2006]. Existence of a motion complying with such constraint implies that the current state $\boldsymbol{x}_0 \in \mathbb{C}_H$, where H is the length of the preview horizon. The requirement to stop does not undermine the execution of the motion goals provided that two conditions are met:

1. the preview horizon is regularly shifted in the future and the anticipated motion is recomputed,
2. the preview horizon is long enough.

The first condition implies that the moment, when $\mathbb{T}$ should be reached, is continuously postponed. Hence, it is not imposed that the robot stops, but rather that it maintains the capacity to stop, *i.e.*, capturability. The second condition is also straightforward: if the robot is required to reach $\mathbb{T}$ the very next moment, it hardly has enough freedom to execute the motion goals. This naturally leads to the question about sufficient length of the preview horizon. Currently, we do not have an answer for the general case: the choice may well be application dependent. However, the existing research on walking of both humans and robots strongly suggests that the length of a preview horizon should cover 2 or 3 steps [Zaytsev 2015a, Carver 2009, Kajita 2003, Koolen 2012, Sparrow 2005, Vukobratović 1970].

The idea of maintaining capturability by imposing that $\mathbb{T}$ is reached at the end of a preview horizon is recurrent in the literature. Implementation, however, is dependent on the particular method chosen for anticipation: the motion may be a result of solution of a boundary value problem, or an MPC problem with terminal constraints [Mansour 2011, Henze 2014, Sherikov 2014, Morisawa 2007, Takenaka 2009]. The target set $\mathbb{T}$ is usually defined so that the robot's ability to stop [Mansour 2011, Henze 2014] is ensured or cyclicity of motions is enforced [Takenaka 2009].

2.3 Conclusion

We have defined the balance preservation problem using notions of viability theory and briefly reviewed existing approaches to this problem based on our definition. Henceforth, we are going to focus on online control and balancing of humanoid robots performing general motions. We assume that a global motion plan is not provided or only partial information, such as planned contact positions, is given. Therefore, we completely rely on motion anticipation with limited preview horizon. In the preceding section it is indicated, that capturability can be enforced in such a situation by imposing appropriate conditions on the final previewed state. Commonly, these conditions ensure the capacity of the robot to stop or maintain a cyclic motion. We believe the first option to be more appropriate, since it allows for generation of both cyclic and acyclic motions. Hence, in the following chapters we narrow the notion of capturability by assuming that it implies the capacity of the robot to reach a statically balanced state, *i.e.*, to stop.

Chapter 3

Modeling of a humanoid robot

A very abstract model of a humanoid robot (2.1) was sufficient for the discussion in Chapter 2, but it is of no use for practical applications. Therefore, in the present chapter we specify and study the structure of humanoid robot models. Though we employ only the whole body model for control (Section 3.1) and approximate linear models for anticipation (Section 3.3), we also consider approximate nonlinear models in Section 3.2 to demonstrate the ancestral relationships between these models. These relationships can be illustrated with the following tree

- Whole body (WB) model (Section 3.1)
  - Nonlinear momenta-based (NMB) model (Subsection 3.2.1)
    - Nonlinear point-mass (NPM) model (Subsection 3.2.2)
      - Linear point-mass models (CPPMJ and CPPMdZ) with planar *Center of Mass* (CoM) motion (Subsection 3.3.2)
      - Linear point-mass model (CNPM) with nonplanar CoM motion (Subsection 3.3.3)
    - Linear momenta-based (CMB) model (Subsection 3.3.1)

We present detailed derivations of the models and try to be explicit about all assumptions made in the process. Most of the assumptions are introduced to linearize models in order to fit them in the *Prioritized Linear Least-Squares* (PLLS) framework (Chapter 5), which we employ for both control and anticipation.

3.1 Whole body dynamical model

We start by considering the whole body model of a humanoid robot. A humanoid robot is a chain of $n+1$ links interconnected by n joints. In general, joints can be rotary or prismatic, actuated or not actuated. While all these joint types can be incorporated in the model without much effort, we limit our discussion to actuated rotary joints only, since the other types are rare in humanoid robots. The links composing the robot can be in contact with the environment, but are not fixed to any support. Therefore, the robot must be modeled taking contacts and contact friction into account. We use *Coulomb's law* as a friction model [Popov 2010, Chapter 10] and make the following additional assumptions:

AS-3.1 coefficients of static and kinetic (dynamic) friction are equal and constant [Popov 2010, Chapter 10];

AS-3.2 there are no rolling contacts;

AS-3.3 the environment and links of the robot are rigid;

AS-3.4 the robot contacts only the objects, whose positions are fixed in the environment;

AS-3.5 there is no friction in the joints;

AS-3.6 there are no noises in measurements of the state and no inaccuracies in parameters of the robot.

Assumption 3.1 is ubiquitous in the literature. Assumptions 3.2, 3.3 and 3.4 are not valid in some settings, but we do not consider such settings and thus avoid the burden of more accurate modeling. The last two assumptions can never be fulfilled in practice. However, friction in the joints can be modeled, if the corresponding parameters of the robot are provided, or it can be concealed by joint-level position controllers of a robot such as `HRP-2` [Kaneko 2004]. The problem of uncertainty, on the other hand, is not addressed by modeling, but rather by robust control and techniques for estimation of parameters and state [Siciliano 2009, Chapter 8], [Christensen 2008]. We leave such topics outside of the scope of this thesis.

Based on the listed assumptions, we present the whole body model in the following Subsection 3.1.1. Subsections 3.1.2 and 3.1.3 focus on constraints included in the model and their linearization. Subsection 3.1.4 presents a whole body model with linear constraints, while concluding Subsection 3.1.5 overviews the control of the robot using this model.

3.1.1 Humanoid robot as a complementarity system

Under Assumptions 3.1 to 3.6 the humanoid robot is described by a *complementarity system* [Brogliato 2003, Hurmuzlu 2004], whose form at a given time instant with M contacts is [Trinkle 1997]

$$\begin{cases} \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{I}_{\boldsymbol{\tau}}\boldsymbol{\tau} + m\boldsymbol{J}_{com}^{\top}(\boldsymbol{q})\boldsymbol{g} + \boldsymbol{J}_{\uparrow,p}^{\top}(\boldsymbol{q})\boldsymbol{f}_p, & \text{(3.1a)} \\ \underline{\boldsymbol{b}} \le \mathrm{A}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{\tau}) \le \bar{\boldsymbol{b}}, & \text{(3.1b)} \\ \ddot{p}_i^n \ge 0, & \text{(3.1c)} \\ f_i^n \ge 0, & \text{(3.1d)} \\ \ddot{p}_i^n f_i^n = 0, & \text{(3.1e)} \\ \kappa_i^2 (f_i^n)^2 - (\boldsymbol{f}_i^t)^{\top}\boldsymbol{f}_i^t \ge 0, & \text{(3.1f)} \\ \kappa_i f_i^n \dot{\boldsymbol{p}}_i^t - \boldsymbol{f}_i^t \|\dot{\boldsymbol{p}}_i^t\|_2 = 0, & \text{(3.1g)} \end{cases}$$

where $i \in \{1, ..., M\}$,

$$(\ddot{\boldsymbol{p}}_1, \ldots, \ddot{\boldsymbol{p}}_M) = \boldsymbol{J}_{\uparrow,p}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,p}(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad \text{(3.2a)}$$
$$(\boldsymbol{f}_1, ..., \boldsymbol{f}_M) = \boldsymbol{f}_p, \quad \text{(3.2b)}$$

and variables have the following meaning

$\boldsymbol{q} = (\boldsymbol{q}', \boldsymbol{r}, \boldsymbol{\mathcal{E}})$	$\in \mathbb{R}^{n+3+3}$	vector of generalized coordinates including n joint angles $\boldsymbol{q}'$, position of the base $\boldsymbol{r}$, and orientation of the base represented with Euler angles $\boldsymbol{\mathcal{E}}$
$\boldsymbol{\tau}$	$\in \mathbb{R}^n$	vector of joint torques
$\boldsymbol{f}_i$	$\in \mathbb{R}^3$	i-th contact force
$\boldsymbol{p}_i$	$\in \mathbb{R}^3$	position of the i-th contact
$\boldsymbol{H}(\boldsymbol{q})$	$\in \mathbb{R}^{(n+6)\times(n+6)}$	inertia matrix
$\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}})$	$\in \mathbb{R}^{n+6}$	vector of Coriolis and centrifugal terms
$\boldsymbol{J}_{com}$	$\in \mathbb{R}^{3\times(n+6)}$	Jacobian of the CoM (see Appendix A.4)
$\boldsymbol{J}_{\uparrow,p}$	$\in \mathbb{R}^{3M\times(n+6)}$	translational Jacobian of the contact points

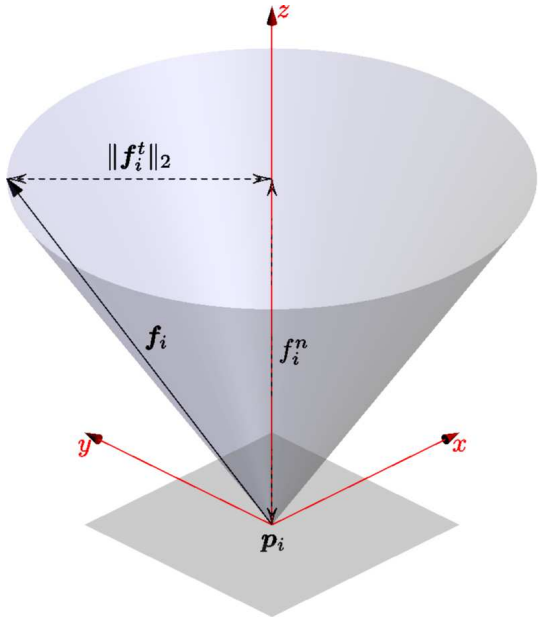


Figure 3.1: Friction cone and the local frame i of the i-th contact. Square patch indicates the contact surface.

$\boldsymbol{I_\tau} = (\boldsymbol{I}_n, \boldsymbol{0}_{6,n})$	$\in \mathbb{R}^{(n+6)\times n}$	torque selection matrix
$\boldsymbol{g}$	$\in \mathbb{R}^3$	vector of gravitational acceleration
κ_i	$\in \mathbb{R}_{\geq 0}$	friction coefficient of the i-th contact
m	$\in \mathbb{R}_{>0}$	total mass of the system
$\mathrm{A}, \underline{\boldsymbol{b}}, \bar{\boldsymbol{b}}$		function, which expresses application dependent constraints on the state and controls, and its bounds

Superscripts $(\cdot)^t$ and $(\cdot)^n$ denote tangential and normal components of vectors with respect to the contact surfaces as shown in Figure 3.1. These components are obtained by expressing vectors in local frames i using rotation matrices ${}^{\mathsf{i}}\boldsymbol{R}$:

$$\begin{bmatrix} \ddot{\boldsymbol{p}}_i^t \\ \ddot{p}_i^n \end{bmatrix} = {}^{\mathsf{i}}\boldsymbol{R}\ddot{\boldsymbol{p}}_i, \quad \begin{bmatrix} \dot{\boldsymbol{p}}_i^t \\ \dot{p}_i^n \end{bmatrix} = {}^{\mathsf{i}}\boldsymbol{R}\dot{\boldsymbol{p}}_i, \quad \begin{bmatrix} \boldsymbol{f}_i^t \\ f_i^n \end{bmatrix} = {}^{\mathsf{i}}\boldsymbol{R}\boldsymbol{f}_i. \tag{3.3}$$

Each frame i is fixed to the contact point and its z axis is normal to the contact surface.

Individual components of system (3.1) have the following interpretation

(3.1a) is the standard equation of dynamics of a multibody system (see Appendix A). The equation establishes relationship between motion of the joints and forces acting on the robot.

(3.1b) represents constraints of a particular robot and setting, for example, mechanical constraints. These constraints are discussed later in Subsection 3.1.2.

(3.1c) prevents interpenetration of the contact surface and the contacting body.

(3.1d) indicates that the contacts are unilateral, *i.e.*, the robot can push on the contact surface, but cannot pull on it.

(3.1e) is the complementarity condition, which states that a contact force cannot be applied at a contact surface if the contact point detaches from this surface.

(3.1f) bounds tangential components of the contact force depending on the norm of its normal component in accordance with Coulomb's friction law.

(3.1g) indicates that the tangential component of the contact force is opposite to the sliding velocity of the contact point.

In order to complete the model we have to consider one more important aspect, which manifests itself in the discrete nature of changes of the set of contacts. These changes, or *switches* of the mode of the system, indicate that the system at hand is *hybrid*. The state of such a system is often discontinuous at the instant of a switch, and a *state reinitialization rule* must be defined [Brogliato 2003]. In the case of the considered system with contacts we employ an *impact law* described in Appendix B as a reinitialization rule.

3.1.2 Mechanical constraints

The general inequality (3.1b) comprises mechanical limits of the joints and motors

$$\underline{\boldsymbol{q}}' \leq \boldsymbol{q}' \leq \bar{\boldsymbol{q}}', \quad \underline{\dot{\boldsymbol{q}}}' \leq \dot{\boldsymbol{q}}' \leq \bar{\dot{\boldsymbol{q}}}', \quad \underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}', \tag{3.4a}$$

$$\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}}. \tag{3.4b}$$

In general, constraints (3.1b) are not limited to (3.4), but such cases are not considered in this work.

Enforcement of constraints (3.4a) is not straightforward, when the model is used exclusively for instantaneous control. In this case, the current state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is given and cannot be constrained, while the next state is not explicitly computed. For this reason, constraints on the joint angles and velocities are imposed through the joint accelerations. Hence, (3.4a) boils down to [Rubrecht 2012]

$$\underline{\ddot{\boldsymbol{q}}}'(\boldsymbol{q}', \dot{\boldsymbol{q}}') \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'(\boldsymbol{q}', \dot{\boldsymbol{q}}'), \tag{3.5}$$

where bounds on accelerations incorporate bounds on joint angles and velocities as well. We discuss computation of $\underline{\ddot{\boldsymbol{q}}}'(\boldsymbol{q}', \dot{\boldsymbol{q}}')$ and $\bar{\ddot{\boldsymbol{q}}}'(\boldsymbol{q}', \dot{\boldsymbol{q}}')$ in Appendix C.

3.1.3 Contact constraints and assumptions

One of the primary sources of nonlinearity of system (3.1) are the contact constraints. In order to linearize them we make a number of approximations and assumptions. We start by assuming that

AS-3.7 There is no sliding, *i.e.*, $\left\|\dot{\boldsymbol{p}}_i^t\right\|_2 = 0$. In order to enforce this, constraint (3.1g), which is not needed now, is replaced with

$$\ddot{\boldsymbol{p}}_i^t = \boldsymbol{0}. \tag{3.6}$$

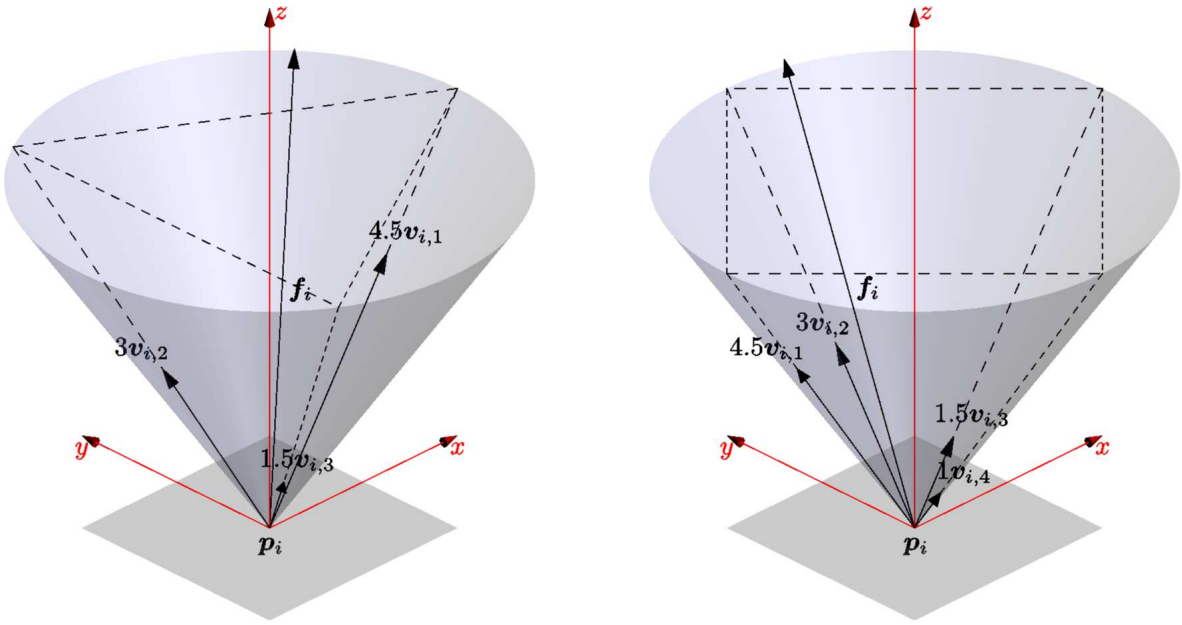
While being useful in manipulation [Howe 1996] sliding is less common in locomotion [Miura 2013], and is usually prevented to simplify control.

AS-3.8 Contacting bodies do not detach, and, consequently, constraint

$$\ddot{p}_i^n = 0 \tag{3.7}$$

must be imposed instead of nonpenetration constraint (3.1c) and complementarity condition (3.1e). This is also a typical assumption. Whenever it is necessary to break one of the contacts, constraint (3.7) is replaced by a motion task for the contacting body.

The next step is to linearize the quadratic constraint (3.1f), which bounds the i-th tangential contact force with respect to the normal contact force in accordance with Coulomb's friction law. In order to linearize it, we replace the second order cones with pyramids as



(a) Friction cone. The force vector is represented by a weighted sum of vectors: $\boldsymbol{f}_i = 4.5\boldsymbol{v}_{i,1} + 3\boldsymbol{v}_{i,2} + 1.5\boldsymbol{v}_{i,3}$.

(b) Friction cone. The force vector is represented by a weighted sum of vectors: $\boldsymbol{f}_i = 4.5\boldsymbol{v}_{i,1} + 3\boldsymbol{v}_{i,2} + 1.5\boldsymbol{v}_{i,3} + \boldsymbol{v}_{i,4}$.

Figure 3.2: Linear approximation of a friction cone.

shown in Figure 3.2. Then for a pyramid with three faces (Figure 3.2a) the constraint is expressed as

$$\begin{bmatrix} (\boldsymbol{v}_{i,1} \times \boldsymbol{v}_{i,2})^\top \\ (\boldsymbol{v}_{i,2} \times \boldsymbol{v}_{i,3})^\top \\ (\boldsymbol{v}_{i,3} \times \boldsymbol{v}_{i,1})^\top \end{bmatrix} \boldsymbol{f}_i \geq \boldsymbol{0}, \tag{3.8}$$

where $\boldsymbol{v}_{i,\{1,2,3\}}$ are unity vectors coinciding with the edges of the pyramid, and cross product is used to obtain normals of the faces. Note that (3.8) implies unilaterality constraint (3.1d) $f_i^n \geq 0$, which becomes redundant and should be removed from the system. There exist several approaches to representation of the linearized constraints [Kuindersma 2014], and the one used in (3.8) is not optimal when it comes to solution of the system of constraints. We perform a change of variables to express these constraints as simple bounds, which are beneficial for solvers as indicated in Subsection 5.4.1.2:

$$\boldsymbol{f}_i = \begin{bmatrix} \boldsymbol{v}_{i,1} & \boldsymbol{v}_{i,2} & \boldsymbol{v}_{i,3} \end{bmatrix} \boldsymbol{\lambda}_i = \boldsymbol{V}_i \boldsymbol{\lambda}_i, \quad \boldsymbol{\lambda}_i \geq \boldsymbol{0}. \tag{3.9}$$

Accuracy of the linear approximation increases with the number of faces of pyramids, which is typically chosen to be 3 or 4 (Figures 3.2a and 3.2b respectively). We use pyramids with three faces, because it results in the least number of inequality constraints.

The number of inequality constraints is further reduced by representing a foot contact force by a single wrench $\hat{\boldsymbol{f}}_i = (\boldsymbol{f}_i, \boldsymbol{\mu}_i) = (\boldsymbol{V}_i \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$ applied at $\boldsymbol{p}_i$ instead of multiple 3-dimensional forces applied at several points on the foot. The force component of the wrench is subject to Coulomb's friction constraint as above, while the moment is constrained by

$$f_i^n \underline{\boldsymbol{\mu}}_i \leq {}^{\mathrm{i}}\boldsymbol{\mu}_i \leq f_i^n \bar{\boldsymbol{\mu}}_i, \quad \text{or} \quad f_i^n \underline{\boldsymbol{\mu}}_i \leq {}^{\mathrm{i}}\boldsymbol{R}\boldsymbol{\mu}_i \leq f_i^n \bar{\boldsymbol{\mu}}_i \tag{3.10}$$

where $\underline{\boldsymbol{\mu}}_i$ and $\bar{\boldsymbol{\mu}}_i$ are constant vectors derived in Appendix D.1, ${}^{\mathrm{i}}\boldsymbol{R}$ transforms the moment to the local frame of i-th contact. This constraint is linear with respect to $\boldsymbol{\lambda}_i$ and $\boldsymbol{\mu}_i$ and in

the following is stated as

$$\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i}\begin{bmatrix}\boldsymbol{\lambda}_i\\ \boldsymbol{\mu}_i\end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}. \tag{3.11}$$

When a foot contact is represented by three or more contact points, the orientation of the foot is fixed. In order to prevent rotation of the foot, when the foot contact is represented by a single point $\boldsymbol{p}_i$, it is necessary to fix angular acceleration with the constraint

$$\dot{\boldsymbol{\omega}}_i = \boldsymbol{J}_{\circlearrowleft,i}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\circlearrowleft,i}\dot{\boldsymbol{q}} = \boldsymbol{0}. \tag{3.12}$$

3.1.4 Whole body model with linear constraints

Substitution of the constraints obtained in Subsections 3.1.2 and 3.1.3 into the complementarity system (3.1) results in the whole body model with linear constraints

Model WB (Whole Body)

$$\begin{cases}
\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{h} = \boldsymbol{I}_{\boldsymbol{\tau}}\boldsymbol{\tau} + m\boldsymbol{J}_{com}^{\top}\boldsymbol{g} + \sum_{i=1}^{M}\boldsymbol{J}_i^{\top}\hat{\boldsymbol{f}}_i, & \text{(3.13a)}\\
\begin{bmatrix}\ddot{\boldsymbol{p}}_i\\ \dot{\boldsymbol{\omega}}_i\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,i}\\ \boldsymbol{J}_{\circlearrowleft,i}\end{bmatrix}\ddot{\boldsymbol{q}} + \begin{bmatrix}\dot{\boldsymbol{J}}_{\uparrow,i}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{J}}_{\circlearrowleft,i}\dot{\boldsymbol{q}}\end{bmatrix} = \boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} = \boldsymbol{0}, & \text{(3.13b)}\\
\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}}, & \text{(3.13c)}\\
\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}', & \text{(3.13d)}\\
\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i}\begin{bmatrix}\boldsymbol{\lambda}_i\\ \boldsymbol{\mu}_i\end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}, & \text{(3.13e)}\\
\boldsymbol{\lambda}_i \geq \boldsymbol{0}, & \text{(3.13f)}
\end{cases}$$

where $\hat{\boldsymbol{f}}_i = (\boldsymbol{f}_i, \boldsymbol{\mu}_i) = (\boldsymbol{V}_i\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$, $\boldsymbol{J}_i$ is the contact point Jacobian including both translational and rotational parts. Dependence of $\boldsymbol{H}$, $\boldsymbol{h}$, $\underline{\ddot{\boldsymbol{q}}}'$, $\bar{\ddot{\boldsymbol{q}}}'$, and Jacobians on $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ is omitted for brevity. Equation (3.13b) incorporates constraints (3.6), (3.7), and (3.12).

3.1.5 Controlling the robot

The primary interest of our modeling effort is to allow control of a humanoid robot using its model. It turns out, that we have already introduced elements of control in WB model by adding constraint

$$\ddot{\boldsymbol{p}}_i = \boldsymbol{J}_{\uparrow,i}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,i}\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{3.14}$$

in order to support Assumptions 3.7 and 3.8. This constraint is clearly not physical, but rather dictated by our control goals. It falls within the framework of *operational space control* [Khatib 1987] or more general *task function control approach* [Samson 1991, Chapter 3], both of which focus on control of the end-effectors of the robot rather than its joints. Hence, we call (3.14) a *task*, whose purpose is to maintain constant position of the end-effector corresponding to the contact point. In a similar way we define tasks for other parts of the robot's body

$$\boldsymbol{J}_{ee}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{ee}\dot{\boldsymbol{q}} = \ddot{\boldsymbol{y}}_{ee}^{\heartsuit}, \tag{3.15}$$

where $\boldsymbol{J}_{ee}$ and $\ddot{\boldsymbol{y}}_{ee}^{\heartsuit}$ are the Jacobian and the desired acceleration of a certain end-effector. $\ddot{\boldsymbol{y}}_{ee}^{\heartsuit}$ can be obtained with a simple PD-controller such as

$$\ddot{\boldsymbol{y}}_{ee}^{\heartsuit} = \mathcal{K}_{p,ee}(\boldsymbol{y}_{ee}^{\heartsuit} - \boldsymbol{y}_{ee}) - \mathcal{K}_{d,ee}\dot{\boldsymbol{y}}_{ee}, \tag{3.16}$$

which drives the end-effector to the desired position $\boldsymbol{y}_{ee}^{\heartsuit}$ given current position $\boldsymbol{y}$ and velocity $\dot{\boldsymbol{y}}$, and positive scalar gains $\mathcal{K}_{p,ee}$, $\mathcal{K}_{d,ee}$. Examples of various end-effector tasks for humanoid robots can be found in [Kanoun 2009, Chapter 4].

Tasks, however, are not limited to control of the end-effectors. For example, the joint-level task

$$\ddot{\boldsymbol{q}}' = \mathcal{K}_{p,\boldsymbol{q}'}(\boldsymbol{q}'^{\heartsuit} - \boldsymbol{q}') - \mathcal{K}_{d,\boldsymbol{q}'}\dot{\boldsymbol{q}}' \tag{3.17}$$

employs a PD-controller to maintain the desired joint configuration $\boldsymbol{q}'^{\heartsuit}$. Furthermore, tasks can also be posed as inequalities [Kanoun 2009, Chapter 3]

$$\underline{\ddot{\boldsymbol{y}}}_{ee} \leq \boldsymbol{J}_{ee}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{ee}\dot{\boldsymbol{q}} \leq \bar{\ddot{\boldsymbol{y}}}_{ee}. \tag{3.18}$$

Thus, we conclude that the difference between tasks and constraints in the model is only a matter of interpretation: as we are going to see in Chapter 5, the only important thing for numerical computations is prioritization of the constraints or tasks with respect to each other. Hence, in the following we use terms "task" and "constraint" interchangeably.

All whole body motion controllers employed in this work include tasks for control of motion of the CoM and orientation of the body, which are related to linear and angular momenta respectively. Both are important for balance preservation: in order to stop the robot it is necessary to nullify its momenta. Furthermore, control over vertical motion of the CoM helps to prevent falls, which cannot be prevented by the constraints in WB model. The desired values for momenta are obtained by anticipating the motion of the robot with approximate linear models described later in this chapter.

While controlling the robot using the task function approach, we may encounter several types of problems. The first problem arises when imposed tasks do not require all degrees of freedom of the system. This implies that the robot is *redundant* with respect to the tasks and there is an infinite number of control inputs that achieve our goals [Samson 1991, Chapter 4]. We can easily alleviate this issue by adding tasks, which, however, increase the chance of triggering the second problem: a *conflict* of the tasks with each other and the physical constraints. Both redundancy and conflicts are important subjects of this thesis and are discussed in Chapter 5. Another important question is dealing with online addition, removal, or transformation of the tasks, which is not considered here [Lee 2012].

3.2 Nonlinear approximate models

The whole body (WB) model reviewed in Section 3.1 allows for the control of the robot, but it varies nonlinearly with the state of the system and is rather complicated. We have already indicated in Subsection 2.2.3, that such complexity is not always needed when online motion anticipation is performed. In this case it is common to employ approximate models, which are derived from the complete model under certain assumptions. In the present section we consider two approximate nonlinear models, which we do not employ *per se*, but use them later in Section 3.3 for the construction of linear models.

3.2.1 Nonlinear momenta-based model

First, let us consider the friction constraints (3.13e), (3.13f) in WB model. It is interesting to observe that they primarily limit the aggregate motion of the robot, *i.e.*, the rate of its spatial momentum. That is, as long as the rate of momentum complies with these constraints, the joint motion can be arbitrary, provided that the motors can produce sufficient torques and joint limits are satisfied. We demonstrate this by rewriting the equation of dynamics using derivations from Appendix A as

$$\underbrace{\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\\ \boldsymbol{H}_3\end{bmatrix}}_{\boldsymbol{H}}\underbrace{\begin{bmatrix}\ddot{\boldsymbol{q}}'\\ \ddot{\boldsymbol{r}}\\ \dot{\boldsymbol{\mathcal{E}}}\end{bmatrix}}_{\ddot{\boldsymbol{q}}} + \underbrace{\begin{bmatrix}\boldsymbol{h}_1\\ \boldsymbol{h}_2\\ \boldsymbol{h}_3\end{bmatrix}}_{\boldsymbol{h}} = \begin{bmatrix}\boldsymbol{\tau}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} + m\underbrace{\begin{bmatrix}\boldsymbol{J}_{com,1}^{\top}\\ \boldsymbol{J}_{com,2}^{\top}\\ \boldsymbol{J}_{com,3}^{\top}\end{bmatrix}}_{\boldsymbol{J}_{com}^{\top}}\boldsymbol{g} + \sum_{i=1}^{M}\underbrace{\begin{bmatrix}\boldsymbol{J}_{i,1}^{\top}\\ \boldsymbol{J}_{i,2}^{\top}\\ \boldsymbol{J}_{i,3}^{\top}\end{bmatrix}}_{\boldsymbol{J}_i^{\top}}\hat{\boldsymbol{f}}_i, \tag{3.19}$$

and use the first line of this equation to eliminate $\boldsymbol{\tau}$ from WB model to obtain

$$\begin{cases} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_{\mathcal{E}}^{\top} \end{bmatrix} \begin{bmatrix} {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{h}_2 \\ \boldsymbol{h}_3 \end{bmatrix} = m \begin{bmatrix} \boldsymbol{J}_{com,2}^{\top} \\ \boldsymbol{J}_{com,3}^{\top} \end{bmatrix} \boldsymbol{g} + \sum_{i=1}^{M} \begin{bmatrix} \boldsymbol{J}_{i,2}^{\top} \\ \boldsymbol{J}_{i,3}^{\top} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix}, , & \text{(3.20a)} \\ \begin{bmatrix} \ddot{\boldsymbol{p}}_i \\ \dot{\boldsymbol{\omega}}_i \end{bmatrix} = \boldsymbol{J}_i \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i \dot{\boldsymbol{q}} = \boldsymbol{0}, & \text{(3.20b)} \\ \boldsymbol{f}_i = \boldsymbol{V}_i \boldsymbol{\lambda}_i, & \text{(3.20c)} \\ \underline{\boldsymbol{\tau}} \leq \boldsymbol{H}_1 \ddot{\boldsymbol{q}} + \boldsymbol{h}_1 - m \boldsymbol{J}_{com,1}^{\top} \boldsymbol{g} - \sum_{i=1}^{M} \boldsymbol{J}_{i,1}^{\top} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \leq \bar{\boldsymbol{\tau}}, & \text{(3.20d)} \\ \underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}', & \text{(3.20e)} \\ \boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}, & \text{(3.20f)} \\ \boldsymbol{\lambda}_i \geq \boldsymbol{0}, & \text{(3.20g)} \end{cases}$$

where ${}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}}$ and ${}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}}$ are rates of linear and angular momenta respectively, and $\boldsymbol{T}_{\mathcal{E}}$ transforms derivatives of the Euler angles to angular velocities (see Appendix A.1.2). Assuming that

AS-3.9 constraints of the joints and motors (3.20d), (3.20e), and contact preservation task (3.20b) are always satisfied,

we neglect the structure of the robot and focus on its momenta and contact forces. Both the contact forces and momenta are crucial for ensuring balance in motion anticipation: control over the former allows to avoid tipping and slipping, over the latter – to stop motion of the robot as indicated in Subsection 3.1.5. Under Assumption 3.9 the model boils down to constrained Newton-Euler equations for a rigid body (see also Appendix A)

Model NMB (Nonlinear Momenta-Based)

$$\begin{cases} \begin{bmatrix} {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}} \end{bmatrix} = m \begin{bmatrix} \boldsymbol{I} \\ (\boldsymbol{c} - \boldsymbol{r})^{\times} \end{bmatrix} \boldsymbol{g} + \sum_{i=1}^{M} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ (\boldsymbol{p}_i - \boldsymbol{r})^{\times} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix}, & \text{(3.21a)} \\ \boldsymbol{f}_i = \boldsymbol{V}_i \boldsymbol{\lambda}_i, & \text{(3.21b)} \\ \boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}, & \text{(3.21c)} \\ \boldsymbol{\lambda}_i \geq \boldsymbol{0}, & \text{(3.21d)} \\ \text{proxy constraints.} & \text{(3.21e)} \end{cases}$$

Here, the term *proxy constraints* is used as in [Zaytsev 2015b, Chapter 3] and denotes constraints, which allow to take into account (3.20d), (3.20e), and (3.20b) indirectly through the variables still present in NMB model. For example, one may choose to reflect kinematic limits of the robot by constraining distance between CoM $\boldsymbol{c}$ and contact points $\boldsymbol{p}_i$. Proxy constraints are typically rough approximations, but are often necessary to support Assumption 3.9.

Equation (3.21a) has several properties, which are important to mention:

- It is more convenient to work with Newton-Euler equation when the base position coincides with the CoM $\boldsymbol{c} = \boldsymbol{r}$ resulting in

$$\begin{bmatrix} {}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} + \sum_{i=1}^{M} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ (\boldsymbol{p}_i - \boldsymbol{c})^{\times} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix}. \tag{3.22}$$

- Another important property of (3.21a) is nonholonomy of its lower part corresponding to the rate of angular momentum [Wieber 2006a]. This means that (3.21a) constrains

rotational motion of the system, but not its orientation. The best known illustration of this fact is a falling cat (see Figure 3.3), which changes its orientation in mid-air without using any contacts with the environment to finally land on its paws. Since there is no clear connection between orientation of the robot and its angular momentum, it is also unclear what are the desired values for angular momentum. It is presumed that the angular momentum should be kept small, but not equal to zero [Wieber 2006a]. In the present work, we constrain the angular momentum by including it in a capturability constraint (see Subsection 4.3.1). Though not necessary in our simulations, it is also reasonable to bound it.

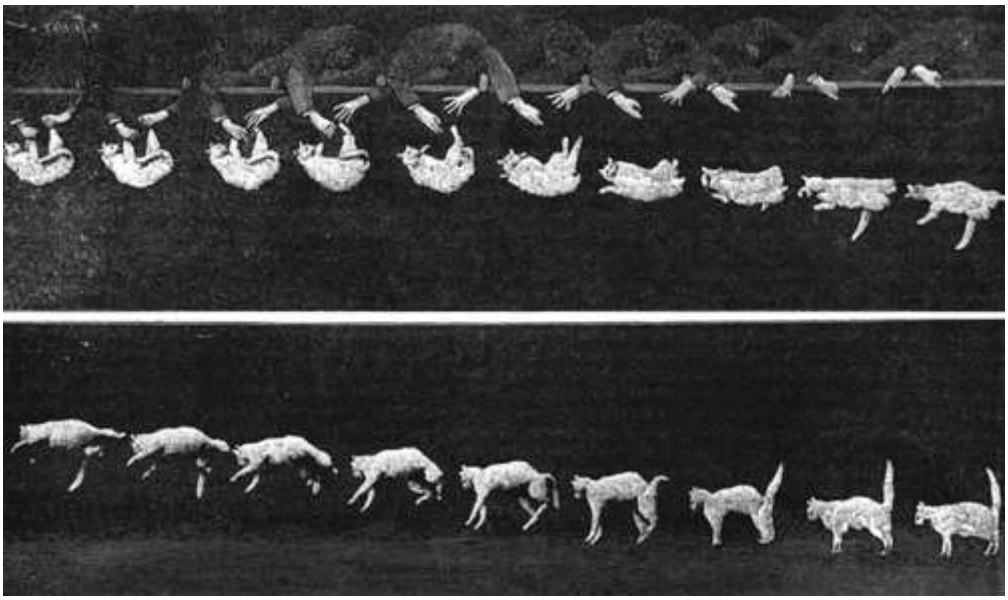


Figure 3.3: Cat changes its orientation while falling even though its angular momentum is initially zero and conserved during a fall. Snapshots of a short film recorded by Étienne-Jules Marey in 1894 [Wikipedia 2015a].

3.2.2 Nonlinear point-mass model

In order to construct this model we introduce additional assumptions. Also, we facilitate derivations by manipulating the position and orientation of the global frame. In practice, however, it may be more convenient to utilize models in convenient working frames and transform data to the global frame when necessary. In many cases it is beneficial to choose the orientation of the global (working) frame so that

AS-3.10 its z axis is parallel to the gravity vector $\boldsymbol{g}$, *i.e.*, $\boldsymbol{g}^{xy} = \mathbf{0}$.

Also, we simplify derivations by taking the orientation of c to be the same as the orientation of the global (working) frame, *i.e.*, $\boldsymbol{\mathcal{P}} = {}^{\mathsf{c}}\boldsymbol{\mathcal{P}}$ and $\dot{\boldsymbol{\mathcal{P}}} = {}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{P}}}$.

The characteristic feature of the point-mass model is the absence of angular momentum ${}^{\mathsf{c}}\boldsymbol{\mathcal{L}} = \mathbf{0}$, since all the mass of the robot is concentrated in a point. Note that in spite of the popularity of such point-mass approximations, it is not strictly necessary to neglect angular momentum to construct a linear model.

We start with NMB model, but this time we express it with respect to the global (working)

frame

$$\begin{cases} \begin{bmatrix} \dot{\boldsymbol{\mathcal{P}}} \\ \dot{\boldsymbol{\mathcal{L}}} \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}} + \boldsymbol{c} \times {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{P}}} \end{bmatrix} = \begin{bmatrix} m\ddot{\boldsymbol{c}} \\ {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}} + m\boldsymbol{c} \times \ddot{\boldsymbol{c}} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{g} \\ m\boldsymbol{c} \times \boldsymbol{g} \end{bmatrix} + \sum_{i=1}^{M} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{p}_i^{\times} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix}, & \text{(3.23a)} \\ \boldsymbol{f}_i = \boldsymbol{V}_i \boldsymbol{\lambda}_i, & \text{(3.23b)} \\ \boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}, & \text{(3.23c)} \\ \boldsymbol{\lambda}_i \geq \boldsymbol{0}, & \text{(3.23d)} \\ \text{proxy constraints.} & \text{(3.23e)} \end{cases}$$

Then we assume that

AS-3.11 the support (ground) surface is flat, is spanned by x and y axes of the global (working) frame, and intersects z axis at p^z,

and divide the set of all contacts into two groups:

1. one contains all $\{1, ..., M_s\}$ support contacts, such that $p_i^z = p^z$;
2. another contains all remaining contacts.

For simplicity, all forces corresponding to the contacts of the second group are reduced to a single wrench $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ acting on the CoM. This wrench is used, for example, to model interaction with a human-collaborator [Agravante 2016b], but usually it is taken to be zero. Once this is done, the equation for the rates of momenta is transformed to

$$\begin{cases} m\ddot{\boldsymbol{c}} - m\boldsymbol{g} - \boldsymbol{f}_{ext} = \sum_{i=1}^{M_s} \boldsymbol{f}_i = \boldsymbol{f}_s & \text{(3.24a)} \\ {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}} + \boldsymbol{c} \times \boldsymbol{f}_s - \boldsymbol{\mu}_{ext} = \sum_{i=1}^{M_s} (\boldsymbol{p}_i \times \boldsymbol{f}_i + \boldsymbol{\mu}_i), & \text{(3.24b)} \end{cases}$$

where $\boldsymbol{f}_s$ is the total support surface reaction force. Each contact position can be represented by a sum of two vectors $\boldsymbol{p}_i = (\boldsymbol{p}_i^{xy}, 0) + (\boldsymbol{0}, p^z)$, hence

$$\sum_{i=1}^{M_s} \boldsymbol{p}_i \times \boldsymbol{f}_i = \begin{bmatrix} \boldsymbol{0} \\ p^z \end{bmatrix} \times \underbrace{\left(\sum_{i=1}^{M_s} \boldsymbol{f}_i \right)}_{\boldsymbol{f}_s} + \sum_{i=1}^{M_s} \begin{bmatrix} \boldsymbol{p}_i^{xy} \\ 0 \end{bmatrix} \times \boldsymbol{f}_i \tag{3.25}$$

Substitution of (3.25) into the equation for the rate of angular momentum yields

$${}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}} + \begin{bmatrix} 0 & p^z - c^z & c^y \\ c^z - p^z & 0 & -c^x \\ -c^y & c^x & 0 \end{bmatrix} \boldsymbol{f}_s - \boldsymbol{\mu}_{ext} = \sum_{i=1}^{M_s} \left(\begin{bmatrix} 0 & 0 & p_i^y \\ 0 & 0 & -p_i^x \\ -p_i^y & p_i^x & 0 \end{bmatrix} \boldsymbol{f}_i + \boldsymbol{\mu}_i \right). \tag{3.26}$$

We avoid nonlinearity of this equation by assuming that

AS-3.12 moments about the z axis μ_i^z can be arbitrary and the z component of constraint (3.23c) can be omitted.

Then ${}^{\mathrm{c}}\dot{\mathcal{L}}^z$ can be always set to any desired value, for example zero, and the respective equation can be neglected. Provided that

AS-3.13 $f_s^z \neq 0$, which implies $m\ddot{c}^z \neq mg^z + f_{ext}^z$, *i.e.*, the robot interacts with the support surface,

we divide the remaining equations for the rates of angular momentum by the total vertical force f_s^z

$$\frac{1}{f_s^z}\left({}^{\mathsf{c}}\dot{\mathcal{L}}^{xy} + \begin{bmatrix} 0 & p^z - c^z & c^y \\ c^z - p^z & 0 & -c^x \end{bmatrix} \boldsymbol{f}_s - \boldsymbol{\mu}_{ext}^{xy}\right) = \frac{1}{f_s^z}\sum_{i=1}^{M_s}\left(f_i^z \begin{bmatrix} p_i^y \\ -p_i^x \end{bmatrix} + \boldsymbol{\mu}_i^{xy}\right) = \frac{\boldsymbol{\mu}_s^{xy}}{f_s^z} \tag{3.27}$$

where $\boldsymbol{\mu}_s^{xy}$ is the total moment created by the surface contacts. Then the rightmost expression is equal to $(z^y, -z^x)$ as shown in Appendix D.1, and (z^x, z^y) is the position of the *Center of Pressure* (CoP) of all support contacts. In this context the CoP is also known as *Zero Moment Point* (ZMP), since the moments about the x and y axes are zero in this point [Vukobratović 2004]. The CoP must stay within the *support area* – the convex hull of all support contacts illustrated in Figure 3.4: $\boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy})$. This constraint on the position of the CoP is equivalent to the remaining constraints on x and y components of contact moments in (3.23c) as follows from Appendices D.1 and D.2.

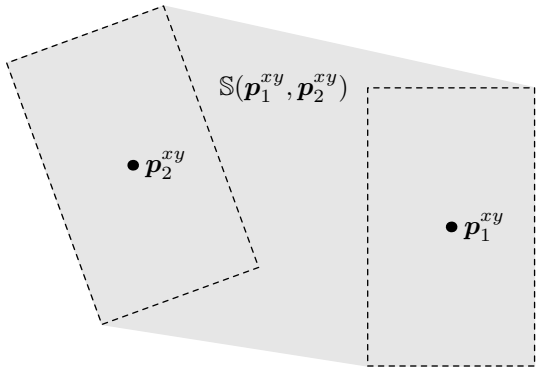


Figure 3.4: Grey area represents support area $\mathbb{S}(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$ of two rectangular foot contacts.

Now we drop individual support contact forces and moments $(\boldsymbol{f}_i, \boldsymbol{\mu}_i)$ with $i \in \{1, ..., M_s\}$ to obtain

$$\begin{cases} \boldsymbol{f}_s = m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) - \boldsymbol{f}_{ext} = \sum_{i=1}^{M_s} \boldsymbol{V}_i \boldsymbol{\lambda}_i, & (3.28a) \\ \frac{1}{f_s^z}\left({}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}}^{xy} - \boldsymbol{\mu}_{ext}^{xy} - (c^z - p^z)\begin{bmatrix} f_s^y \\ -f_s^x \end{bmatrix}\right) + \begin{bmatrix} c^y \\ -c^x \end{bmatrix} = \begin{bmatrix} z^y \\ -z^x \end{bmatrix}, & (3.28b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.28c) \\ \boldsymbol{\lambda}_i \geq \boldsymbol{0}, & (3.28d) \\ f_s^z \neq 0, & (3.28e) \\ \text{proxy constraints.} & (3.28f) \end{cases}$$

The system is further simplified if

AS-3.14 The rate of angular momentum is always zero ${}^{\mathsf{c}}\boldsymbol{\mathcal{L}}^{xy} = \boldsymbol{0}$, which implies that the system models a point-mass. This is an ubiquitous assumption in the literature, even though it is not required to construct a linear model [Wieber 2015].

AS-3.15 The friction coefficients are the same for all contacts, so that $\boldsymbol{V}_i = \boldsymbol{V}$ and $\boldsymbol{f}_s = \boldsymbol{V}\sum_{i=1}^{M_s}\boldsymbol{\lambda}_i$ or $\boldsymbol{f}_s = \boldsymbol{V}\boldsymbol{\lambda}$ with $\boldsymbol{\lambda} \geq \boldsymbol{0}$, which makes variables $\boldsymbol{\lambda}_i$ unnecessary.

Assumption 3.15 implies that the total surface contact force is constrained to the same friction cone as individual surface contact forces.

Thus, the system takes the following form

Model NPM (Nonlinear Point-Mass)

$$\begin{cases} \boldsymbol{z} = \boldsymbol{c}^{xy} - \zeta \boldsymbol{f}_s^{xy}/m + \dfrac{1}{f_s^z}\begin{bmatrix} -\mu_{ext}^y \\ \mu_{ext}^x \end{bmatrix}, & (3.29a) \\ \zeta = \dfrac{m(c^z - p^z)}{f_s^z}, & (3.29b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.29c) \\ f_s^z \neq 0, & (3.29d) \\ \boldsymbol{f}_s = m(\ddot{\boldsymbol{c}} - \boldsymbol{g}) - \boldsymbol{f}_{ext} = \boldsymbol{V}\boldsymbol{\lambda}, \quad \boldsymbol{\lambda} \geq \boldsymbol{0}, & (3.29e) \\ \text{proxy constraints}, & (3.29f) \end{cases}$$

where the first equation is nonlinear with respect to motion of the CoM and external force $\boldsymbol{f}_{ext}$, and constraint (3.29c) is nonlinear with respect to positions and orientations of the contacts as explained in Appendix D.2.

3.3 Linear approximate models

In the present work we employ two types of approximate models: the first one (Subsection 3.3.1) is tailored for 3-dimensional multi-contact settings, the second (Subsections 3.3.2 and 3.3.3) – for walking. All of these models are based on the models described in Section 3.2 and are linear. Though nonlinear models find more and more applications in practice, they are generally more demanding for computational resources [Koenemann 2015].

3.3.1 Momenta-based model with noncoplanar contacts

The linear model for preview of linear and angular momenta derived in this section was originally proposed in [Nagasaka 2012] in Japanese and later restated with some minor extensions in [Audren 2014, Sherikov 2015] in English.

The component-wise equation of rate of angular momentum in NMB model has the following form

$$\begin{bmatrix} {}^{\mathrm{c}}\dot{\mathcal{L}}^x \\ {}^{\mathrm{c}}\dot{\mathcal{L}}^y \\ {}^{\mathrm{c}}\dot{\mathcal{L}}^z \end{bmatrix} = \sum_{i=1}^{M} \left(\begin{bmatrix} 0 & -(p_i^z - c^z) & p_i^y - c^y \\ p_i^z - c^z & 0 & -(p_i^x - c^x) \\ -(p_i^y - c^y) & p_i^x - c^x & 0 \end{bmatrix} \begin{bmatrix} f_i^x \\ f_i^y \\ f_i^z \end{bmatrix} + \begin{bmatrix} \mu_i^x \\ \mu_i^y \\ \mu_i^z \end{bmatrix} \right). \tag{3.30}$$

In order to avoid nonlinearity in this equation we assume that

AS-3.16 Contact points $\boldsymbol{p}_i$ are given constants.

AS-3.17 c^z is a given constant, which implies that the CoM acceleration along the z axis is zero and

$$\dot{\mathcal{P}}^z = m\ddot{c}^z = mg^z + \sum_{i=1}^{M} f_i^z = 0. \tag{3.31}$$

AS-3.18 Rate of angular momentum about the z axis is arbitrary. In other words, the computed rotational motion about the z axis is always feasible. Validity of this assumption is not completely clear, but no problems in simulations were reported so far. This assumption is opposite to Assumption 3.12, which allows for arbitrary contact moments about the z axis in the point-mass (NPM) model.

Since the vertical motion of the CoM is prohibited by Assumption 3.17 and ${}^{\mathrm{c}}\dot{\mathcal{L}}^z$ is neglected due to Assumption 3.18, we focus on the linear and angular momenta about the x

and y axes:

$$\dot{\boldsymbol{\mathcal{P}}}^{xy} = m\ddot{\boldsymbol{c}}^{xy} = \underbrace{m\boldsymbol{g}^{xy}}_{\tilde{\boldsymbol{b}}} + \sum_{i=1}^{M} \boldsymbol{f}_i^{xy} \tag{3.32}$$

$${}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}}^{xy} = \sum_{i=1}^{M} \left(\underbrace{\begin{bmatrix} 0 & -(p_i^z - c^z) & p_i^y \\ p_i^z - c^z & 0 & -p_i^x \end{bmatrix}}_{\tilde{\boldsymbol{B}}_i} \boldsymbol{f}_i + \boldsymbol{I}_{xy}\boldsymbol{\mu}_i \right) + \underbrace{\begin{bmatrix} 0 & g^z \\ -g^z & 0 \end{bmatrix}}_{\tilde{\boldsymbol{A}}} \underbrace{(m\boldsymbol{c}^{xy})}_{\hat{\boldsymbol{\mathcal{P}}}^{xy}} \tag{3.33}$$

Then we construct a linear continuous-time model with M control inputs $\hat{\boldsymbol{f}}_i = (\boldsymbol{f}_i, \boldsymbol{\mu}_i)$ and state vector $\boldsymbol{x} = (m\boldsymbol{c}^{xy}, \boldsymbol{\mathcal{P}}^{xy}, {}^{\mathrm{c}}\boldsymbol{\mathcal{L}}^{xy}) = (\hat{\boldsymbol{\mathcal{P}}}^{xy}, \boldsymbol{\mathcal{P}}^{xy}, {}^{\mathrm{c}}\boldsymbol{\mathcal{L}}^{xy})$:

Model CMB (Continuous Momenta-Based)

$$\begin{cases} \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{P}}^{xy} \\ \dot{\boldsymbol{\mathcal{P}}}^{xy} \\ {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}}^{xy} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \tilde{\boldsymbol{A}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \hat{\boldsymbol{\mathcal{P}}}^{xy} \\ \boldsymbol{\mathcal{P}}^{xy} \\ {}^{\mathrm{c}}\boldsymbol{\mathcal{L}}^{xy} \end{bmatrix}}_{\boldsymbol{x}} + \sum_{i=1}^{M} \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_{xy} & \boldsymbol{0} \\ \tilde{\boldsymbol{B}}_i & \boldsymbol{I}_{xy} \end{bmatrix}}_{\boldsymbol{B}_i} \underbrace{\begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\mu}_i \end{bmatrix}}_{\hat{\boldsymbol{f}}_i} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{b}} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{b}} & \text{(3.34a)} \\ \boldsymbol{f}_i = \boldsymbol{V}_i \boldsymbol{\lambda}_i, & \text{(3.34b)} \\ \sum_{i=1}^{M} f_i^z = -mg^z, & \text{(3.34c)} \\ \boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}, & \text{(3.34d)} \\ \boldsymbol{\lambda}_i \geq \boldsymbol{0}, & \text{(3.34e)} \\ \text{proxy constraints.} & \text{(3.34f)} \end{cases}$$

Note that if Assumption 3.10 holds, *i.e.*, $\boldsymbol{g}^{xy} = \boldsymbol{0}$, the model is simplified due to $\boldsymbol{b} = \boldsymbol{0}$. Assumption 3.10 may be too restrictive in some applications [Audren 2014], but in the simulations discussed in Section 6.2 it always holds true.

3.3.2 Linear point-mass model with planar CoM motion

The subject of the present section is the *linear point-mass model*, which was originally proposed by Kajita [Kajita 2001, Kajita 2003], later it was employed and extended in [Diedam 2008, Herdt 2010, Agravante 2016b] and many other works. The best known variant of this model is interpreted as an *inverted pendulum* with a weightless leg and a mass constrained to a plane as shown in Figure 3.5. Hence, it is often referred to as *Linear Inverted Pendulum Model* (LIPM) in the literature.

NPM model is usually linearized with respect to the CoM motion by taking c^z to be constant and, consequently, $\ddot{c}^z = 0$ (Assumption 3.17) [Kajita 2003]. In addition to this the following assumptions are commonly made

AS-3.19 Wrench $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ is known, or $f_{ext}^z = 0$. This is necessary to linearize the model with respect to the external wrench [Agravante 2016b]. In the following we assume that $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ is a known constant for simplicity.

AS-3.20 Constraint (3.29c) on the CoP positions is expressed with a set of linear equality and inequality constraints. The conditions under which this is true are discussed in Appendix D.2.

AS-3.21 Proxy constraints (3.29f) and constraints on the surface contact forces 3.29d and 3.29e are always satisfied. They can, however, be imposed explicitly if necessary.

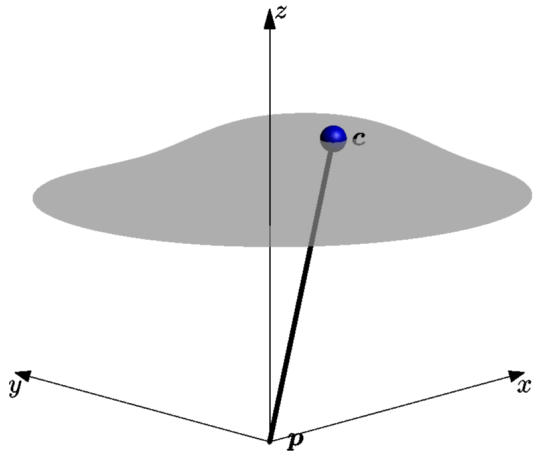


Figure 3.5: Inverted pendulum with a mass $\boldsymbol{c}$ constrained to a plane.

Then the model is reduced to

$$\begin{cases} \boldsymbol{z} = \boldsymbol{c}^{xy} - \zeta \ddot{\boldsymbol{c}}^{xy} + \mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}), & \text{(3.35a)} \\ \zeta = \dfrac{m(c^z - p^z)}{-mg^z - f^z_{ext}}, & \text{(3.35b)} \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & \text{(3.35c)} \end{cases}$$

where the function $\mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ represents the contribution of the external wrench and gravity to the CoP position:

$$\mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}) = \zeta\left(\boldsymbol{g}^{xy} + \frac{\boldsymbol{f}^{xy}_{ext}}{m}\right) + \frac{1}{-mg^z - f^z_{ext}} \begin{bmatrix} -\mu^y_{ext} \\ \mu^x_{ext} \end{bmatrix}, \tag{3.36}$$

The model can be further combined with a double or triple integrator to obtain a second or third order linear model. There is a number of ways to realize such a combination. For example, the CoP position can be

- an output of second or third order model [Kajita 2003, Herdt 2010, Agravante 2016b];
- a part of the state of a third order model [Kajita 2010];
- the control input of a second order or third order model [Sherikov 2014].

Furthermore, the differential equation (3.35a) can be transformed to expose its stable and unstable parts [Englsberger 2011, Krause 2012, Takenaka 2009], see also Subsection 4.3.2. Another common approach taken in the literature consist in collapsing the surface contacts to point contacts, which greatly simplifies constraint (3.35c) [Englsberger 2011]: if $M_s = 1$ then $\boldsymbol{z} = \boldsymbol{p}_1^{xy}$, if $M_s = 2$ then $\boldsymbol{z}$ lies on a segment $[\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy}]$, *etc.* The differences between the various versions of the model are often subtle, but there are certain general considerations to keep in mind while choosing a particular version:

- Force sensors in the feet of modern robots allow for the determination of the position of the CoP [Englsberger 2014, Kaneko 2004, Kaneko 2009], which suggests its inclusion into the state of the model.
- Third order models imply a smooth variation of CoM acceleration contrary to second order models. A discontinuous change of CoM acceleration results in a discontinuous change of the CoP position, which are difficult to realize for humanoid robots [Kajita 2010].

- Discretization of models controlled with piece-wise constant CoM acceleration or *jerk* (derivative of acceleration) leads to some minor difficulties in the satisfaction of constraint (3.35c). Moreover, the number of times this constraint is imposed can be larger than in other cases and its satisfaction is not guaranteed at all times. These issues are discussed later in Subsection 4.2.2.1, where the models are discretized to be used in *Model Predictive Control* (MPC) schemes.

In the following we are going to consider two third order models, whose state includes positions, velocities, and accelerations of the CoM in the x-y plane

$$\boldsymbol{x} = (c^x, \dot{c}^x, \ddot{c}^x, c^y, \dot{c}^y, \ddot{c}^y). \tag{3.37}$$

The models differ in their control inputs:

1. The first one is controlled by the third derivative (*jerk*) of the CoM position [Kajita 2003, Herdt 2010, Agravante 2016b]

Model CPPMJ (Continuous Planar Point-Mass controlled with CoM Jerk)

$$\begin{cases} \dot{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{A}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \tilde{\boldsymbol{B}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{B}} \end{bmatrix} \dddot{\boldsymbol{c}}^{xy}, & (3.38a) \\ \boldsymbol{z} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x} + \mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}), & (3.38b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.38c) \end{cases}$$

where

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \tilde{\boldsymbol{D}} = \begin{bmatrix} 1 & 0 & -\zeta \end{bmatrix}. \tag{3.39}$$

We do not use this model in our controllers and present it here due to its prevalence in the literature.

2. The second one is controlled by the CoP velocity, which is obtained by differentiation of (3.35a) under assumption that $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ is constant [Sherikov 2014]

Model CPPMdZ (Continuous Planar Point-Mass controlled with $\dot{\boldsymbol{z}}$)

$$\begin{cases} \dot{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{A}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \tilde{\boldsymbol{B}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{B}} \end{bmatrix} \dot{\boldsymbol{z}}, & (3.40a) \\ \boldsymbol{z} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x} + \mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}), & (3.40b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.40c) \end{cases}$$

where

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \frac{1}{\zeta} & 0 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\zeta} \end{bmatrix}, \quad \tilde{\boldsymbol{D}} = \begin{bmatrix} 1 & 0 & -\zeta \end{bmatrix}. \tag{3.41}$$

Here we assumed that

AS-3.22 $\zeta \neq 0$ or $c^z - p^z \neq 0$, *i.e.*, the CoM position is not in the same plane as contacts.

If Assumption 3.22 does not hold, CPPMJ and CPPMdZ models become equivalent.

3.3.3 Linear point-mass model with nonplanar CoM motion

Walking with planar motion of the CoM (Assumption 3.17) is considered to be unnatural and inefficient in terms of energy. In order to address these issues we proposed to construct a model, which allows for vertical motion of the CoM [Brasseur 2015b]. This is achieved by making the model robust to the motion of the CoM along the z axis.

Let us consider NPM model

$$\begin{cases} \boldsymbol{z} = \boldsymbol{c}^{xy} - \zeta\left(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy} - \dfrac{\boldsymbol{f}_{ext}^{xy}}{m}\right) + \underbrace{\dfrac{1}{m(\ddot{c}^z - g^z) - f_{ext}^z}}_{\zeta/(mc^z)} \begin{bmatrix} -\mu_{ext}^y \\ \mu_{ext}^x \end{bmatrix}, & (3.42a) \\ \zeta = \dfrac{mc^z}{m(\ddot{c}^z - g^z) - f_{ext}^z}, & (3.42b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.42c) \end{cases}$$

where some of the constraints are omitted for simplicity and the support area is assumed to be linear with respect to contact points (Assumptions 3.20 and 3.21). One can observe that if

AS-3.23 the external wrench is equal to zero $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}) = \boldsymbol{0}$,

the position of the CoP is linear with respect to parameter ζ:

$$\begin{cases} \boldsymbol{z}^{xy} = \boldsymbol{c}^{xy} - \zeta(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}), & (3.43a) \\ \zeta = \dfrac{c^z - p^z}{\ddot{c}^z - g^z}, & (3.43b) \\ \boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}). & (3.43c) \end{cases}$$

Thus, if ζ is bounded $\underline{\zeta} \leq \zeta \leq \bar{\zeta}$, the CoP position $\boldsymbol{z}$ lies on a line segment between two points $\underline{\boldsymbol{z}}$ and $\bar{\boldsymbol{z}}$ as shown in Figure 3.6, where

$$\underline{\boldsymbol{z}} = \boldsymbol{c}^{xy} - \underline{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}), \quad \bar{\boldsymbol{z}} = \boldsymbol{c}^{xy} - \bar{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}). \tag{3.44}$$

At the same time, $\underline{\boldsymbol{z}} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy})$ and $\bar{\boldsymbol{z}} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy})$ imply $\boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy})$ due

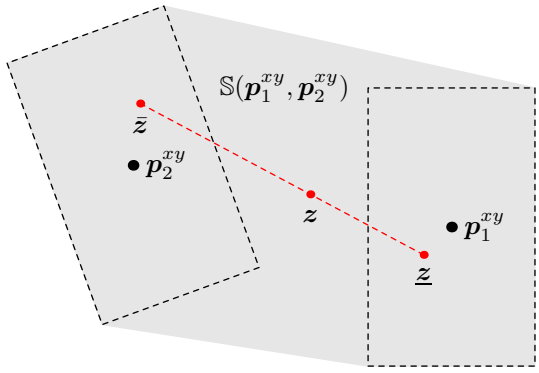


Figure 3.6: Grey area represents support area $\mathbb{S}(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$ of two rectangular foot contacts. The CoP position lying on a line segment between two points is shown in red.

to convexity of the support area (see Figure 3.6). Hence, we construct a system of linear

constraints on the 3-dimensional motion of the CoM as follows

$$\begin{cases} \underline{\boldsymbol{z}} = \boldsymbol{c}^{xy} - \underline{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}), & (3.45a) \\ \bar{\boldsymbol{z}} = \boldsymbol{c}^{xy} - \bar{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}), & (3.45b) \\ \underline{\zeta}(\ddot{c}^{z} - g^{z}) \leq c^{z} - p^{z} \leq \bar{\zeta}(\ddot{c}^{z} - g^{z}), & (3.45c) \\ \underline{\boldsymbol{z}} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.45d) \\ \bar{\boldsymbol{z}} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.45e) \end{cases}$$

where $\underline{\zeta}$ and $\bar{\zeta}$ are predefined constants. This system is then combined with the triple integrator to model motion of the CoM

Model CNPM (Continuous Nonplanar Point-Mass)

$$\begin{cases} \dot{\boldsymbol{x}} = \underset{3}{\operatorname{diag}}\left(\tilde{\boldsymbol{A}}\right)\boldsymbol{x} + \underset{3}{\operatorname{diag}}\left(\tilde{\boldsymbol{B}}\right)\dddot{\boldsymbol{c}}, & (3.46a) \\ \underline{\zeta}(\ddot{c}^{z} - g^{z}) \leq c^{z} - p^{z} \leq \bar{\zeta}(\ddot{c}^{z} - g^{z}), & (3.46b) \\ \boldsymbol{c}^{xy} - \underline{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}) \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.46c) \\ \boldsymbol{c}^{xy} - \bar{\zeta}(\ddot{\boldsymbol{c}}^{xy} - \boldsymbol{g}^{xy}) \in \mathbb{S}(\boldsymbol{p}_1^{xy}, ..., \boldsymbol{p}_{M_s}^{xy}), & (3.46d) \\ \boldsymbol{x} = (c^{x}, \dot{c}^{x}, \ddot{c}^{x},\ \ c^{y}, \dot{c}^{y}, \ddot{c}^{y},\ \ c^{z}, \dot{c}^{z}, \ddot{c}^{z}), & (3.46e) \end{cases}$$

where

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.47}$$

3.4 Limitations of the approximate models

We conclude our survey of approximate models with a reflection upon their limitations and the means of relaxing these limitations. In order to do this, we investigate the assumptions made during the derivation of these models:

- We abstract from the complex structure of the robot's body (Assumption 3.9). Hence, we lose the ability to express some of the whole body tasks and constraints. For example, the point-mass model does not allow to specify a reaching task for a hand. Therefore, when it is necessary to bias motion previewed with this model by a hand task, it is common to resort to various *ad hoc* approaches [Yoshida 2006, Nishiwaki 2003, Fukumoto 2004]. Whole body constraints are often approximated by proxy constraints (Section 3.2). For example, the position of the CoM is limited with respect to foot positions in [Brasseur 2015b, Dellin 2012] to avoid kinematic infeasibility. Another general approach to take into account whole body tasks and constraints is to employ *Mixed Model Predictive Control* (MMPC), which was proposed in [Sherikov 2014] and is described in Section 4.5.
- We presume that the rate of angular momentum is zero in the point-mass model (Assumption 3.14) or takes arbitrary values in the momenta based model (Subsection 3.3.1). Neither of this is true in reality: execution of limb motions implies certain values of rate of angular momentum. These values can be estimated using multi-mass models, some of which are linear [Herdt 2012, Chapter 3], [Lafaye 2014, Shimmyo 2013, Takenaka 2009].
- Bilinearity of the rate of angular momentum with respect to the CoM motion and contact forces as follows from Equation (3.48) is addressed with two assumptions:

  - The angular momentum about the z axis is typically disregarded (Assumptions 3.12 and 3.18). Significance of implications of this is currently unclear to us.
  - Motion of the CoM is fixed to a plane (Assumption 3.17). There exists a number of approaches to cope with this [Nishiwaki 2011, Feng 2013], in particular, we proposed to introduce robustness with respect to nonplanar CoM motion [Brasseur 2015b], Subsection 3.3.3.

$$\begin{bmatrix} {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}} \end{bmatrix} = \begin{bmatrix} m\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} + \sum_{i=1}^{M} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ (\boldsymbol{p}_i - \boldsymbol{c})^{\times} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{R}_{\mathrm{i}}{}^{\mathrm{i}}\boldsymbol{\mu}_i \end{bmatrix}. \tag{3.48}$$

- Bilinearity of the rate of angular momentum with respect to the contact positions and contact forces as follows from Equation (3.48) is dealt with by predetermining either contact positions or magnitudes of the contact forces. For example, we fix contact positions in the momenta based model (Assumption 3.16). On the other hand, in the case of a single contact, when the force is determined by the CoM motion, the contact position may vary (see [Herdt 2010] and Appendix D.2). When it is undesirable to predetermine contact positions and forces, it is common to employ nonlinear models [Dai 2014, Tassa 2014].

It is important to note that some of these limitations can be lifted in discrete-time models, since such models allow for different parameters on different discretization intervals. For example, in discrete-time the number and positions of contacts in the momenta-based model may vary, as well as the height of the contact surface p^z in the point-mass model with nonplanar CoM motion. This topic is discussed in more detail in Subsection 4.2.4.

3.5 Conclusion

This chapter serves as a survey of the whole body model of a humanoid robot and approximate models derived from it. The models are presented in a unified view, which is not very common in the literature, where the focus is typically made on particular models and applications. The author's contribution to the subject consists in a participation to collaborative works which employ approximate models [Agravante 2016b, Brasseur 2015b].

Chapter 4

Anticipation using approximate models

We have indicated in Subsection 2.2.3, that awareness of the future is crucial for balance preservation in a general setting. It is, however, often sufficient to look into the future for a limited time horizon. This can be achieved with *Model Predictive Control* (MPC) [Rawlings 2009, Maciejowski 2002], which is the subject of the present chapter. The discussion begins with a brief overview of MPC in Section 4.1, which is followed by Sections 4.2 and 4.3, where we discretize the continuous-time approximate models constructed in Section 3.3 and derive capturability constraints for them. In the last two Sections 4.5 and 4.6 we introduce *Mixed Model Predictive Control* (MMPC) and discuss the choice of duration of sampling intervals.

4.1 Overview of Model Predictive Control

The name of the *Model Predictive Control* (MPC) paradigm stresses two of its important components: a model of the system, and a prediction of its evolution. In this thesis, we employ linear discrete-time models of the form

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k \in \{0, ..., N-1\} & \text{(4.1a)} \\ \boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1}, & \text{(4.1b)} \\ \boldsymbol{u}_k \in \mathbb{U}_k. & \text{(4.1c)} \end{cases}$$

The prediction is used to choose a sequence of N control inputs $(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1})$, such that the future states $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ comply with the constraints of the model. In some cases, the constraints uniquely determine the future evolution of the model and controls can be found analytically. In general, however, a selection criterion for controls is needed, which is typically expressed as a least-squares objective, for example,

$$\underset{\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1}}{\text{minimize}} \quad \sum_{k=0}^{N-1} \left(\|\boldsymbol{\Gamma}_{\boldsymbol{u}} \boldsymbol{u}_k\|_2^2 + \|\boldsymbol{\Gamma}_{\boldsymbol{x}} \boldsymbol{x}_{k+1}\|_2^2 \right), \tag{4.2}$$

where $\boldsymbol{\Gamma}_u$ and $\boldsymbol{\Gamma}_x$ are weighting matrices. Given the constraints of the model and a selection criterion for controls we express an MPC problem as a *Quadratic Program* (QP) [Nocedal 2006, Boyd 2004], which can be solved with off-the-shelf software, for example, `qpOASES` [Ferreau 2014].

The key feature of MPC is that the control problem is resolved periodically in order to realize state feedback. Hence, not all N control inputs are applied, instead, the MPC problem

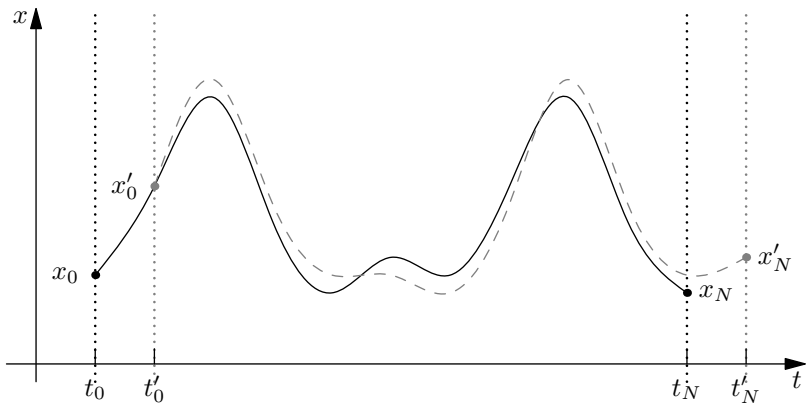


Figure 4.1: Shift of the preview horizon in MPC. Trajectory of state x previewed starting from time t_0 is recomputed at time t'_0. Length of the preview horizon is $H = t_N - t_0 = \sum_{k=0}^{N-1} T_k$, where N is the number of sampling intervals and T_k is the duration of k-th interval.

is updated and resolved after a short time, usually one sampling interval, as illustrated in Figure 4.1.

4.2 Discretization of approximate models

Standard approaches to MPC rely on discrete-time models. For this reason, we discretize the linear continuous-time models constructed in Section 3.3 with the help of `Maxima` *Computer Algebra System* (CAS) [MAX 2016]. Discretization is performed in the standard way with *zero-order hold* for controls, *i.e.*, the discrete-time models have constant controls during a sampling interval [Bao-Cang 2010, Chapter 1].

In all considered discrete-time models, k denotes the index of a sampling interval and T_k is the duration of this interval.

4.2.1 Momenta-based model

Discretization of CMB model yields

Model MB

$$\begin{cases} \boldsymbol{x}_{k+1} = \underbrace{\begin{bmatrix} \boldsymbol{I} & T_k\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ T_k\tilde{\boldsymbol{A}} & \frac{T_k^2}{2}\tilde{\boldsymbol{A}} & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{A}_k} \boldsymbol{x}_k + \sum_{i=1}^{M_k} \underbrace{\begin{bmatrix} \frac{T_k^2}{2}\boldsymbol{I}_{xy} & \boldsymbol{0} \\ T_k\boldsymbol{I}_{xy} & \boldsymbol{0} \\ T_k\tilde{\boldsymbol{B}}_{k,i} + \frac{T_k^3}{6}\tilde{\boldsymbol{A}}\boldsymbol{I}_{xy} & T_k\boldsymbol{I}_{xy} \end{bmatrix}}_{\boldsymbol{B}_{k,i}} \hat{\boldsymbol{f}}_{k,i} + \underbrace{\begin{bmatrix} \frac{T_k^2}{2}\tilde{\boldsymbol{b}} \\ T_k\tilde{\boldsymbol{b}} \\ \frac{T_k^3}{6}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{b}} \end{bmatrix}}_{\boldsymbol{b}}, & \text{(4.3a)} \\ \boldsymbol{f}_{k,i} = \boldsymbol{V}_{k,i}\boldsymbol{\lambda}_{k,i}, & \text{(4.3b)} \\ \sum_{i=1}^{M_k} f_{k,i}^z = -mg^z, & \text{(4.3c)} \\ \boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},k,i} \begin{bmatrix} \boldsymbol{\lambda}_{k,i} \\ \boldsymbol{\mu}_{k,i} \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},k,i}, & \text{(4.3d)} \\ \boldsymbol{\lambda}_{k,i} \geq \boldsymbol{0}, & \text{(4.3e)} \\ \text{proxy constraints}, & \text{(4.3f)} \end{cases}$$

where the state vector $\boldsymbol{x}$ and matrices $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{b}}$ are defined as in Subsection 3.3.1, contact wrench $\hat{\boldsymbol{f}}_{k,i} = (\boldsymbol{f}_{k,i}, \boldsymbol{\mu}_{k,i})$ is constant during T_k, M_k is the number of contacts during the k-th interval,

and $\tilde{\boldsymbol{B}}_{k,i}$ in contrast with $\tilde{\boldsymbol{B}}_i$ used in continuous-time CMB model allows for different positions of contacts $\boldsymbol{p}_{k,i}$ during different intervals

$$\tilde{\boldsymbol{B}}_{k,i} = \begin{bmatrix} 0 & -(p^z_{k,i} - c^z) & p^y_{k,i} \\ p^z_{k,i} - c^z & 0 & -p^x_{k,i} \end{bmatrix}. \tag{4.4}$$

4.2.2 Point-mass models with planar CoM motion

In the following subsections we discretize CPPMJ and CPPMdZ models presented in Subsection 3.3.2. For simplicity we assume that

AS-4.1 the external wrench $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ and orientation of the gravity $\boldsymbol{g}$ do not change within the preview horizon;

AS-4.2 the vertical position of contact points p^z is constant.

It is possible to generalize the derivations to situations, when these assumptions do not hold. This, however, would require a larger number of constraints on the *Center of Pressure* (CoP) positions.

4.2.2.1 Model controlled with the CoM jerk

Discretization of CPPMJ model yields

Model PPMJ (Planar Point-Mass controlled with CoM Jerk)

$$\begin{cases} \boldsymbol{x}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{A}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \tilde{\boldsymbol{B}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{B}}_k \end{bmatrix} \dddot{\boldsymbol{c}}^{xy}_k, & (4.5a) \\ \boldsymbol{z}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x}_{k+1} + \mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}), & (4.5b) \\ \boldsymbol{z}_{k+1} \in \mathbb{S}(\boldsymbol{p}^{xy}_{k+1,1}, ..., \boldsymbol{p}^{xy}_{k+1,M_s}), & (4.5c) \end{cases}$$

where $k \in \{0, ..., N-1\}$, $\boldsymbol{x}_k = (c^x_k, \dot{c}^x_k, \ddot{c}^x_k, c^y_k, \dot{c}^y_k, \ddot{c}^y_k)$,

$$\tilde{\boldsymbol{A}}_k = \begin{bmatrix} 1 & T_k & T_k^2/2 \\ 0 & 1 & T_k \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_k = \begin{bmatrix} T_k^3/6 \\ T_k^2/2 \\ T_k \end{bmatrix}, \quad \tilde{\boldsymbol{D}} = \begin{bmatrix} 1 & 0 & -\zeta \end{bmatrix}, \quad \zeta = \frac{m(c^z - p^z)}{-mg^z - f^z_{ext}}. \tag{4.6}$$

$$\mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}) = \zeta \left(\boldsymbol{g}^{xy} + \frac{\boldsymbol{f}^{xy}_{ext}}{m} \right) + \frac{1}{-mg^z - f^z_{ext}} \begin{bmatrix} -\mu^y_{ext} \\ \mu^x_{ext} \end{bmatrix} \tag{4.7}$$

Note that, if one of the parameters in (4.5b) changes at the boundary between the k-th and $k+1$ sampling intervals, there is a discontinuity in the CoP position at this instant. Hence, if, for example, the external force $\boldsymbol{f}_{ext}$ changes at this instant, the number of constraints on the CoP position must be doubled. For the same reason it is necessary to impose the CoP constraints twice for each sampling interval in the case of the second order model based on a double integrator and controlled with the CoM acceleration. In order to avoid this complication we introduced Assumptions 4.1 and 4.2.

Satisfaction of the constraints on $\boldsymbol{z}_k$ and $\boldsymbol{z}_{k+1}$ in the models controlled by the CoM acceleration or jerk does not guarantee their satisfaction during the k-th sampling interval. In order to illustrate this we consider the k-th sampling interval of the system controlled by the CoM jerk. Let $\boldsymbol{x}_k$ be an initial state, $\dddot{\boldsymbol{c}}^{xy}_k$ – the constant jerk applied during T_k, $\boldsymbol{x}_t$ – the state of the system at some $t \in [0, T_k]$. Position of the CoP during the sampling interval can be found as

$$z^\alpha_t = \tilde{\boldsymbol{D}}_k \begin{bmatrix} c^\alpha_t \\ \dot{c}^\alpha_t \\ \ddot{c}^\alpha_t \end{bmatrix} = \tilde{\boldsymbol{D}}_k \left(\tilde{\boldsymbol{A}}_t \begin{bmatrix} c^\alpha_k \\ \dot{c}^\alpha_k \\ \ddot{c}^\alpha_k \end{bmatrix} + \tilde{\boldsymbol{B}}_t \dddot{c}^\alpha_k \right), \tag{4.8}$$

where $\alpha \in \{x, y\}$, $(\boldsymbol{f}_{k,ext}, \boldsymbol{\mu}_{k,ext}) = \mathbf{0}$ for simplicity, and

$$\tilde{\boldsymbol{A}}_t = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_t = \begin{bmatrix} t^3/6 \\ t^2/2 \\ t \end{bmatrix}. \tag{4.9}$$

Hence, the CoP position at time t depends cubically on time t:

$$z_t^\alpha = \frac{\dddot{c}_k^\alpha}{6} t^3 + \frac{\ddot{c}_k^\alpha}{2} t^2 + (\dot{c}_k^\alpha - \dddot{c}_k^\alpha \zeta_k) t - \ddot{c}_k^\alpha \zeta_k + c_k^\alpha. \tag{4.10}$$

Similarly, this dependence is quadratic in the case of a second order model controlled by the CoM acceleration. Therefore, satisfaction of the CoP constraints at time 0 and T_k, as is usually enforced by MPC schemes, does not guarantee their satisfaction at $t \in (0, T_k)$. The systems controlled by the CoP position or its velocity are not subject to this problem. This problem, however, is typically not critical, since the support areas are intentionally shrunk due to the addition of safety margins [Wieber 2015]. The size of these margins can be estimated by computing maxima of the polynomial (4.10).

4.2.2.2 System controlled with the CoP velocity

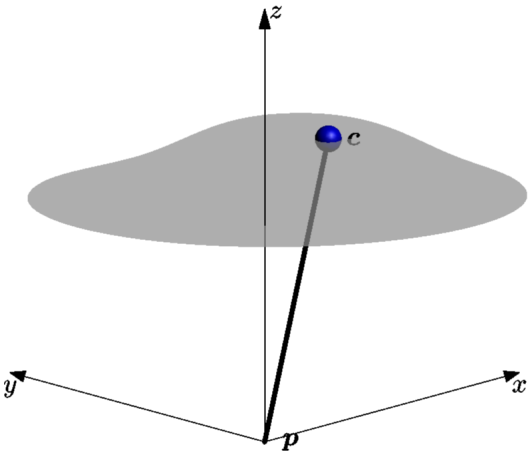


Figure 4.2: Inverted pendulum with a mass $\boldsymbol{c}$ constrained to a plane.

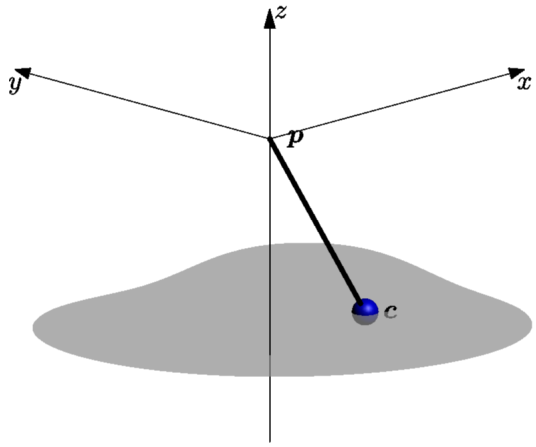


Figure 4.3: Pendulum with a mass $\boldsymbol{c}$ constrained to a plane.

Discretized version of CPPMdZ is

Model PPMdZ (Planar Point-Mass controlled with $\dot{\boldsymbol{z}}$)

$$\begin{cases} \boldsymbol{x}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{A}}_k & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{A}}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \tilde{\boldsymbol{B}}_k & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{B}}_k \end{bmatrix} \dot{\boldsymbol{z}}_k, & \text{(4.11a)} \\ \boldsymbol{z}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x}_{k+1} + \mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}), & \text{(4.11b)} \\ \boldsymbol{z}_{k+1} \in \mathbb{S}(\boldsymbol{p}_{k+1,1}^{xy}, ..., \boldsymbol{p}_{k+1,M_s}^{xy}), & \text{(4.11c)} \end{cases}$$

where all terms are defined as in PPMJ model, except matrices $\tilde{\boldsymbol{A}}_k$ and $\tilde{\boldsymbol{B}}_k$, which differ depending on the sign of ζ

$$\zeta = \frac{m(c^z - p^z)}{-mg^z - f_{ext}^z}. \tag{4.12}$$

Note that we already have $\zeta \neq 0$ due to Assumption 3.22.

- The variable ζ is negative, when $c^z - p^z > 0$ and $-mg^z < f^z_{ext}$, or $c^z - p^z < 0$ and $-mg^z > f^z_{ext}$. This is possible in two cases: (**i**) the CoM is below the support surface and f^z_{ext} does not cancel the gravity, (**ii**) the CoM is above the support surface and f^z_{ext} cancels the gravity. In any case, the robot is hanging and its dynamics resembles dynamics of a standard pendulum shown in Figure 4.3. We are not aware of publications, where this version of the model was employed, but it may be useful in some settings, for example, to realize locomotion on monkey bars as in [Dai 2014], which relies on a nonlinear model. So, whenever $\zeta < 0$, the matrices are defined as

$$\tilde{\boldsymbol{A}}_k = \begin{bmatrix} 1 & \sin\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right)\sqrt{\|\zeta\|} & \zeta\cos\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) - \zeta \\ 0 & \cos\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) & \sin\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right)\sqrt{\|\zeta\|} \\ 0 & -\frac{1}{\sqrt{\|\zeta\|}}\sin\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) & \cos\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_k = \begin{bmatrix} T_k - \sin\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right)\sqrt{\|\zeta\|} \\ 1 - \cos\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) \\ \frac{1}{\sqrt{\|\zeta\|}}\sin\left(\frac{T_k}{\sqrt{\|\zeta\|}}\right) \end{bmatrix}. \tag{4.13}$$

- In the case of positive ζ, which corresponds to *Linear Inverted Pendulum Model* (LIPM) (Figure 4.2), the matrices have the following form

$$\tilde{\boldsymbol{A}}_k = \begin{bmatrix} 1 & \sinh\left(\frac{T_k}{\sqrt{\zeta}}\right)\sqrt{\zeta} & \zeta\cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) - \zeta \\ 0 & \cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) & \sinh\left(\frac{T_k}{\sqrt{\zeta}}\right)\sqrt{\zeta} \\ 0 & \frac{1}{\sqrt{\zeta}}\sinh\left(\frac{T_k}{\sqrt{\zeta}}\right) & \cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_k = \begin{bmatrix} T_k - \sinh\left(\frac{T_k}{\sqrt{\zeta}}\right)\sqrt{\zeta} \\ 1 - \cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) \\ -\frac{1}{\sqrt{\zeta}}\sinh\left(\frac{T_k}{\sqrt{\zeta}}\right) \end{bmatrix}. \tag{4.14}$$

  Note that substitution of negative ζ in the obtained matrices yields (4.13), hence, they are expressed in (4.14) in a more general form.

4.2.3 Point-mass model with nonplanar CoM motion

Discretization of CNPM model presented in Subsection 3.3.3 is trivial as in the case of CPPMJ model with planar CoM motion (see Subsection 4.2.2.1).

4.2.4 Variation of discrete-time models with time

One can observe that state transition matrices, control matrices, and constraints in discrete-time models can be chosen differently depending on the sampling interval k. Hence, some parameters of the models, which were assumed to be constant in continuous-time case, can vary in discrete-time case if their changes are known in advance and synchronized with the boundaries of sampling intervals. For example,

- The number and positions of contacts can be changed in momenta-based model as hinted in Subsection 4.2.1. Therefore, this model can be used to realize locomotion [Nagasaka 2012, Audren 2014].
- Variation of the discrete-time models with planar CoM motion is used to change foot contacts and orientations of the feet without compromising linearity [Herdt 2010], Subsection 4.4.1.
- The point-mass model with nonplanar CoM motion can be utilized for walking on an uneven terrain, if height of the contact surface p^z_k as well as its orientation are changed appropriately [Brasseur 2015b].

Variation of the discrete-time models is typically realized with the help of external routines with respect to MPC. For example, a preliminary MPC problem is solved in [Herdt 2012,

Chapter 2] to determine orientations of the foot contacts. In the same work, a *Finite State Machine* (FSM) is used to set durations and alternation of left and right foot contacts during a regular walk. Though durations of the contacts are usually decided externally when approximate models are employed, in some settings this can be partially avoided with the help of motion anticipation and task prioritization as described in Section 6.2 and [Sherikov 2015].

4.3 Capturability constraints

We claimed in Subsection 2.2.3 that balance is maintained as long as it is possible to stop the robot, *i.e.*, capture it, without violation of the system constraints. A capturability constraint on the final state-control pair $(\boldsymbol{x}_N, \boldsymbol{u}_{N+1})$ formalizes conditions under which the robot can be stopped. In this regard the capturability constraint is similar to the so called *terminal constraint*, which is employed in *dual-mode* MPC [Mayne 2000], [Rossiter 2003, Chapter 6]. In this version of MPC a terminal constraint is imposed at the end of the preview horizon to make sure that the system reaches some *terminal set*, where a simple local controller can stabilize the system. Here we take a similar approach, but we want a local controller to drive the system to *any* of its statically balanced states instead of *a given desired* state. Our approach is conceptually similar to constraining the final state-control pair to a *terminal feasible invariant set* as suggested in [Schouwenaars 2006].

There are several conceptual questions regarding capturability constraints, which we would like to address before going into details of construction of these constraints for particular models:

- Shouldn't we aim for stability rather than capturability? So far we cannot give a decisive answer. Capturability is in general less restrictive as indicated above. On the other hand, closed loop behavior of an MPC with capturability constraints may be worse, for example, due to different statically balanced states being chosen on different control iterations.
- Why many applications of MPC for balance preservation are successful without capturability constraints [Herdt 2010, Kajita 2003] and what kind of improvements such constraints can bring? The answer to the first part of the question lies in the fact that a finite preview horizon *approximates* an infinite preview horizon, which, in turn, guarantees viability, *i.e.*, that the robot will not fall [Wieber 2008]. A capturability constraint, on the other hand, *guarantees* capturability, *i.e.*, the ability to stop, even with a finite preview horizon. Hence, an MPC problem with capturability constraint is more reliable in balance preservation and does not need the preview horizon to be as long as in the case without such constraint.
- What is the impact of a capturability constraint on feasibility of the underlying optimization problem? A capturability constraint, as well as any other hard constraint, may cause infeasibility [Rossiter 2003, Chapter 8], which means that the chosen model of the robot cannot be stopped within the chosen preview horizon. This infeasibility is equivalent to a conflict between tasks in whole body motion control reviewed in Subsection 3.1.5 and, thus, can be addressed with the same numerical tools, which are discussed in Chapter 5.

In the following subsections we construct capturability constraints for the approximate models introduced in Section 3.3 and discretized in Section 4.2. In all cases we derive constraints assuming that

AS-4.3 The model is time-invariant, starting from the end of the preview horizon, which implies, in particular, that the number and positions of the contacts do not change.

Note that our capturability constraints ensure recursive feasibility of time-invariant systems in the same way as the usual terminal constraints [Mayne 2000].

It is also presumed that the system constraints are imposed on the final state-control pair in addition to the derived capturability constraints.

4.3.1 Momenta-based model

In its simplest form a capturability constraint imposes that the robot is in a statically balanced state. A statically balanced state must necessarily be a part of a fixed point, *i.e.*, such state-control pair $(\boldsymbol{x}_k, \boldsymbol{u}_k)$ that [Scheinerman 1996, Chapter 8]

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k. \tag{4.15}$$

In the case of momenta-based (MB) model, this condition results in the following constraints, where indices of state and control variables are omitted for simplicity:

- $\boldsymbol{\mathcal{P}}^{xy} = \boldsymbol{0}$ – no linear momentum in the x-y plane, *i.e.*, the CoM velocity is zero $\dot{\boldsymbol{c}}^{xy} = \boldsymbol{0}$;
- $m\boldsymbol{g}^{xy} + \sum_{i=1}^{M} \boldsymbol{f}_i^{xy} = \boldsymbol{0}$ – no forces in the x-y plane, *i.e.*, the CoM acceleration is zero $\ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}$;
- the rate of change of the angular momentum about the x and y axes is also zero:

$$\begin{aligned} {}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}}^{xy} &= \begin{bmatrix} \hat{\mathcal{P}}^y g^z + \sum_{i=1}^{M} (f_i^z p_i^y + \boldsymbol{I}_x \boldsymbol{\mu}_i - f_i^y (p_i^z - c^z)) \\ -\hat{\mathcal{P}}^x g^z + \sum_{i=1}^{M} (-f_i^z p_i^x + \boldsymbol{I}_y \boldsymbol{\mu}_i + f_i^x (p_i^z - c^z)) \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathcal{P}}^y g^z - mc^z g^y + \sum_{i=1}^{M} (f_i^z p_i^y + \boldsymbol{I}_x \boldsymbol{\mu}_i - f_i^y p_i^z) \\ -\hat{\mathcal{P}}^x g^z + mc^z g^x + \sum_{i=1}^{M} (-f_i^z p_i^x + \boldsymbol{I}_y \boldsymbol{\mu}_i + f_i^x p_i^z) \end{bmatrix} = \boldsymbol{0}. \end{aligned} \tag{4.16}$$

Note that the fixed points of the approximate system include states with non-zero angular momentum ${}^{\mathsf{c}}\boldsymbol{\mathcal{L}}^{xy} \neq \boldsymbol{0}$, even though the real system cannot store angular momentum [Stephens 2010]. Consequently, a fixed point of this approximate model does not always correspond to a fixed point of the whole body model and does not imply that the latter is captured. We alleviate this issue by imposing that ${}^{\mathsf{c}}\boldsymbol{\mathcal{L}}^{xy}$ is zero as well. Furthermore, provided that $m\boldsymbol{g}^{xy} + \sum_{i=1}^{M} \boldsymbol{f}_i^{xy} = \boldsymbol{0}$ holds, we can also determine ${}^{\mathsf{c}}\dot{\mathcal{L}}^z$ using the equation (3.30) and constrain it:

$$\begin{aligned} {}^{\mathsf{c}}\dot{\mathcal{L}}^z &= \sum_{i=1}^{M} \left(\begin{bmatrix} -(p_i^y - c^y) & p_i^x - c^x & 0 \end{bmatrix} \boldsymbol{f}_i + \boldsymbol{I}_z \boldsymbol{\mu}_i \right) \\ &= \sum_{i=1}^{M} (p_i^x f_i^y - p_i^y f_i^x + \boldsymbol{I}_z \boldsymbol{\mu}_i) + \hat{\mathcal{P}}^x g^y - \hat{\mathcal{P}}^y g^x = 0, \end{aligned} \tag{4.17}$$

even though ${}^{\mathsf{c}}\dot{\mathcal{L}}^z$ depends nonlinearly on the state and control variables in general.

Thus, the complete capturability constraint is defined as

$$\begin{cases} \boldsymbol{\mathcal{P}}^{xy} = \boldsymbol{0}, & \text{(4.18a)} \\ {}^{\mathsf{c}}\boldsymbol{\mathcal{L}}^{xy} = \boldsymbol{0}, & \text{(4.18b)} \\ \dot{\boldsymbol{\mathcal{P}}}^{xy} = m\boldsymbol{g}^{xy} + \sum_{i=1}^{M} \boldsymbol{f}_i^{xy} = \boldsymbol{0}, & \text{(4.18c)} \\ {}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}} = -\boldsymbol{g} \times \begin{bmatrix} \hat{\boldsymbol{\mathcal{P}}}^{xy} \\ mc^z \end{bmatrix} + \sum_{i=1}^{M} (\boldsymbol{p}_i \times \boldsymbol{f}_i + \boldsymbol{\mu}_i) = \boldsymbol{0}. & \text{(4.18d)} \end{cases}$$

Note that the capturability constraint imposed in [Sherikov 2015] for the same model is not entirely correct.

4.3.2 Point-mass models with planar CoM motion

In the present section we construct capturability constraints for the two third order point-mass models introduced in Subsection 3.3.2 and discretized in Subsection 4.2.2. We approach the derivation of these constraints using the idea of dual-mode MPC instead of identification of fixed points as in the previous case of the momenta-based model. To be more precise, we use capturability constraints to specify a set of states, starting from which the system can converge to a statically balanced state with the help of a simple linear feedback controller. Strictly speaking, *convergence* to a statically balanced state contradicts with our definition of capturability given in Chapter 2, since this definition requires that the system reaches a statically balanced state in *finite* time. However, we consider this contradiction to be insignificant.

We construct linear feedback controllers in such a way, that they maintain a constant position of the CoP. This implies that constraints on this position will not be violated in the future. Though the controllers are supposed to be discrete-time, we consider their continuous-time counterparts as well to emphasize relations and differences between variants of the model with different control variables.

In order to simplify presentation, we exploit the fact, that the state transition and control matrices in systems are block diagonal and each block corresponds to motion along the x or y axis. This allows us to limit our analysis to blocks $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{A}}_k$, $\tilde{\boldsymbol{B}}_k$ defined in Subsections 3.3.2 and 4.2.2 respectively.

4.3.2.1 Continuous-time model controlled by the CoM jerk

In this section we work with CPPMJ model. Our goal is to derive a controller, which maintains constant position of the CoP or, in other words, maintains zero velocity of the CoP. Velocity of the CoP can be found by differentiation of the equation (3.38b) (assuming constant external wrench $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$):

$$\dot{\boldsymbol{z}} = \dot{\boldsymbol{c}}^{xy} - \zeta \dddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}. \tag{4.19}$$

If $\zeta = 0$, the CoP velocity is zero when the CoM velocity is zero. Hence, the capturability constraint is

$$\dot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \quad \ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \quad \dddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}. \tag{4.20}$$

If $\zeta \neq 0$, we substitute $\dddot{\boldsymbol{c}}^{xy} = \dot{\boldsymbol{c}}^{xy}/\zeta$ into the model. The obtained model is identical to CPPMdZ model with control input equal to zero $\dot{\boldsymbol{z}} = \boldsymbol{0}$. This case is analyzed in the following subsection, while the discrete-time case of the model controlled by the CoM jerk is considered afterwards.

4.3.2.2 Continuous and discrete-time models controlled by the CoP velocity

Position of the CoP in CPPMdZ model can always be maintained with a trivial linear controller $\dot{\boldsymbol{z}} = \boldsymbol{0}$. Hence, we can analyze stability of the system with the help of eigen decomposition of the state transition matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{A}}_k$

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \frac{1}{\zeta} & 0 \end{bmatrix}, \quad \tilde{\boldsymbol{A}}_k = \begin{bmatrix} 1 & \sinh\left(\frac{T_k}{\sqrt{\zeta}}\right)\sqrt{\zeta} & \zeta\cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) - \zeta \\ 0 & \cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) & \sinh\left(\frac{T_k}{\sqrt{\zeta}}\right)\sqrt{\zeta} \\ 0 & \frac{1}{\sqrt{\zeta}}\sinh\left(\frac{T_k}{\sqrt{\zeta}}\right) & \cosh\left(\frac{T_k}{\sqrt{\zeta}}\right) \end{bmatrix}. \tag{4.21}$$

Eigen decomposition allows to identify unstable eigenvalues and nullify unstable modes of the system [Scheinerman 1996, Muske 1993]. Eigenvalues of matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{A}}_k$ are

$$\left(-\frac{1}{\sqrt{\zeta}}, \frac{1}{\sqrt{\zeta}}, 0\right) \text{ and } \left(e^{-\frac{T_k}{\sqrt{\zeta}}}, e^{\frac{T_k}{\sqrt{\zeta}}}, 1\right) \tag{4.22}$$

respectively, where e is Euler's number. Thus, stability of the systems is determined by the sign of ζ

- When $\zeta < 0$ matrix $\tilde{\boldsymbol{A}}$ has two purely imaginary and one zero eigenvalues. In order to suppress oscillatory behavior in this case it is necessary to set the CoM velocities and accelerations to zero:
$$\dot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \quad \ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}. \tag{4.23}$$
Analysis of the discrete-time matrix $\tilde{\boldsymbol{A}}_k$ results in the same conclusion.
- If, on the opposite, $\zeta > 0$, matrix $\tilde{\boldsymbol{A}}$ has one unstable positive, one negative and one zero eigenvalues. The unstable mode is nullified when
$$\dot{\boldsymbol{c}}^{xy} + \sqrt{\zeta}\ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}. \tag{4.24}$$
If this equation holds, the system with zero input converges to a fixed point with $\boldsymbol{c}^{xy} = \boldsymbol{z}$, $\dot{\boldsymbol{c}}^{xy} = \boldsymbol{0}$, and $\ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}$. The same is true for the discrete-time system.

Note that due to (4.24)
$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{c}}^{xy} + \sqrt{\zeta}\dddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \tag{4.25}$$
where $\boldsymbol{\xi}$ is the *capture point*, which is defined as the point on the ground where the robot should step to stop [Koolen 2012, Takenaka 2009, Englsberger 2011, Hof 2005]. In our case $\boldsymbol{\xi}$ is the position of the CoP, which lies within the support area and to which $\boldsymbol{c}^{xy}$ converges. The capturability constraint (4.24) ensures that this point does not move, which is implied in the other works where the second order model is considered [Koolen 2012, Englsberger 2011].

4.3.2.3 Discrete-time model controlled by the CoM jerk

In the case of the discrete-time version of CPPMJ model our goal is to construct controller such that $\boldsymbol{z}_k = \boldsymbol{z}_{k+1}$ and
$$\boldsymbol{z}_k - \boldsymbol{z}_{k+1} = \tilde{\boldsymbol{D}}\begin{bmatrix} c_k^\alpha \\ \dot{c}_k^\alpha \\ \ddot{c}_k^\alpha \end{bmatrix} - \tilde{\boldsymbol{D}}\left(\tilde{\boldsymbol{A}}_k\begin{bmatrix} c_k^\alpha \\ \dot{c}_k^\alpha \\ \ddot{c}_k^\alpha \end{bmatrix} + \tilde{\boldsymbol{B}}_k\dddot{c}_k^\alpha\right) = 0, \tag{4.26}$$
where $\alpha \in \{x, y\}$. When $\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}_k = 0$ this equation is satisfied if $\dot{c}_k^\alpha = 0$ and $\ddot{c}_k^\alpha = 0$. Otherwise, we can find $\dddot{c}_k^\alpha$ from (4.26) and substitute it into the equation of dynamics of the system to obtain a new state transition matrix
$$\left(\tilde{\boldsymbol{A}}_k + \tilde{\boldsymbol{B}}_k\left(\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}_k\right)^{-1}\tilde{\boldsymbol{D}}\left(\boldsymbol{I}_{3,3} - \tilde{\boldsymbol{A}}_k\right)\right), \tag{4.27}$$
whose eigenvalues are
$$\left(\frac{\sqrt{3}T_k\sqrt{12\zeta + T_k^2} + 6\zeta + 2T_k^2}{6\zeta - T_k^2}, \frac{-\sqrt{3}T_k\sqrt{12\zeta + T_k^2} + 6\zeta + 2T_k^2}{6\zeta - T_k^2}, 1\right). \tag{4.28}$$
We do not go further in the analysis due to subtle dependence of the eigenvalues on parameters ζ and T_k. When the parameters are known, appropriate constraints can be easily found. These constraints, however, are not the same as in the case of continuous-time version of this system. Alternatively, it is possible to explicitly impose a complete stop with
$$\dot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \quad \ddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \quad \dddot{\boldsymbol{c}}^{xy} = \boldsymbol{0}, \tag{4.29}$$

4.3.3 Point-mass model with nonplanar CoM motion

A simple linear controller capable of stopping the system without violation of the constraints does not seem to exist. Therefore, we do not use the same approach for derivation of the capturability constraint as in Subsection 4.3.2, instead, we suggest to explicitly stop the CoM motion with

$$\dot{\boldsymbol{c}} = \mathbf{0}, \quad \ddot{\boldsymbol{c}} = \mathbf{0}, \quad \dddot{\boldsymbol{c}} = \mathbf{0}. \tag{4.30}$$

4.3.4 Infeasibility due to variation of a model with time

We have already briefly discussed infeasibility of the capturability constraints in the introduction of this section. In the conclusion we would like to point out a particular type of infeasibility, which is caused by variation of models with time (see Subsection 4.2.4).

Consider a situation, when the model changes in the very end of the preview horizon due to a change in the contact stance. It is usually impossible to change the contact stance and completely stop within the same final sampling interval, since these goals overconstrain the final control input. This issue, however, is not observed in the case of the point-mass model with planar CoM motion when convergence to a statically balanced state is exploited (see Subsection 4.3.2.2). In other cases, various heuristics can be employed to avoid infeasibility of the capturability constraint.

4.4 Point-mass model with foot motion

We do not employ any of the point-mass models presented in Section 4.2 for walking *per se*, but rather an augmented version of PPMdZ model. The augmented version includes a linear system, which reflects changes of positions of the feet on the ground in the preview horizon as in [Herdt 2010]. For the sake of computational performance all inequality constraints in the augmented version are expressed as simple bounds (see Subsection 5.4.1.2).

4.4.1 Modeling of changes of the foot positions

We model changes of the foot positions on the ground in order to facilitate their automatic adjustment, which allows for disturbance compensation and tracking of desired walking speed [Herdt 2010]. Future foot positions can be found using the following discrete-time linear system

$$\begin{cases} \hat{\boldsymbol{p}}_{j+1} = \hat{\boldsymbol{p}}_j + \hat{\boldsymbol{R}}_\mathrm{j} \Delta\hat{\boldsymbol{p}}_j, \quad j \in \{0, ..., K-1\}, & \text{(4.31a)} \\ \Delta\hat{\boldsymbol{p}}_j \in \mathbb{F}_j, & \text{(4.31b)} \end{cases}$$

where $\Delta\hat{\boldsymbol{p}}_j$ is the distance between the j-th and $j+1$ foot positions expressed in frame j fixed to j-th foot, $\hat{\boldsymbol{R}}_\mathrm{j}$ transforms $\Delta\hat{\boldsymbol{p}}_j$ to the global frame, and $\mathbb{F}_j$ defines a polygonal area of allowed positions of $\hat{\boldsymbol{p}}_{j+1}$ with respect to $\hat{\boldsymbol{p}}_j$ [Stasse 2009, Herdt 2010]. K is the number of adjustable foot positions in the preview horizon. $\hat{\boldsymbol{p}}_0$ denotes the current support foot position and is, therefore, fixed.

Combination of (4.31) with PPMdZ model gives

$$\begin{cases} \boldsymbol{x}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{A}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \tilde{\boldsymbol{B}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{B}}_k \end{bmatrix} \dot{\boldsymbol{z}}_k, & k \in \{0, ..., N-1\}, & (4.32a) \\ \hat{\boldsymbol{p}}_{j+1} = \hat{\boldsymbol{p}}_j + \hat{\boldsymbol{R}}_j \Delta\hat{\boldsymbol{p}}_j, & j \in \{0, ..., K-1\}, & (4.32b) \\ \boldsymbol{z}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x}_{k+1}, & & (4.32c) \\ (\boldsymbol{p}^{xy}_{k+1,1}, ..., \boldsymbol{p}^{xy}_{k+1,M_s}) = \mathrm{F}_{k+1}(\hat{\boldsymbol{p}}_0, ..., \hat{\boldsymbol{p}}_K), & & (4.32d) \\ \Delta\hat{\boldsymbol{p}}_j \in \mathbb{F}_j, & & (4.32e) \\ \boldsymbol{z}_{k+1} \in \mathbb{S}(\boldsymbol{p}^{xy}_{k+1,1}, ..., \boldsymbol{p}^{xy}_{k+1,M_s}), & & (4.32f) \end{cases}$$

where the function F_{k+1} selects foot positions at the instant $k+1$. Here we assumed that the gravity vector is aligned with the z axis and there is no external wrench acting on the model, *i.e.*, $\mathrm{Z}(\zeta, \boldsymbol{g}, \boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext}) = \boldsymbol{0}$. We indicated in Appendix D.2 that the constraints on the position of CoP during double supports depend nonlinearly on foot positions (see Appendix D.2). In order to avoid this nonlinearity, we impose the CoP constraints only when the system is in single support [Herdt 2010], or approximate double support constraints with single support constraints as explained in Appendix D.2. In either case the number of contacts $M_s = 1$, and the model is simplified to

$$\begin{cases} \boldsymbol{x}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{A}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_k \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \tilde{\boldsymbol{B}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{B}}_k \end{bmatrix} \dot{\boldsymbol{z}}_k, & k \in \{0, ..., N-1\}, & (4.33a) \\ \hat{\boldsymbol{p}}_{j+1} = \hat{\boldsymbol{p}}_j + \hat{\boldsymbol{R}}_j \Delta\hat{\boldsymbol{p}}_j, & j \in \{0, ..., K-1\}, & (4.33b) \\ \boldsymbol{z}_{k+1} = \begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x}_{k+1}, & & (4.33c) \\ \boldsymbol{p}^{xy}_{k+1} = \mathrm{F}_{k+1}(\hat{\boldsymbol{p}}_0, ..., \hat{\boldsymbol{p}}_K), & & (4.33d) \\ \Delta\hat{\boldsymbol{p}}_j \in \mathbb{F}_j, & & (4.33e) \\ \boldsymbol{z}_{k+1} \in \mathbb{S}(\boldsymbol{p}^{xy}_{k+1}), & & (4.33f) \end{cases}$$

4.4.2 Conversion of the constraints to simple bounds

Constraints on the CoP positions are expressed as simple bounds in frames fixed to the feet provided that the feet are rectangular (see Appendix D.2). Hence, we redefine the output of the system (4.33) as follows:

$$\hat{\boldsymbol{z}}_{k+1} = {}^{\mathsf{k+1}}\boldsymbol{R} \left(\begin{bmatrix} \tilde{\boldsymbol{D}} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{D}} \end{bmatrix} \boldsymbol{x}_{k+1} - \boldsymbol{p}^{xy}_{k+1} \right), \tag{4.34}$$

where ${}^{\mathsf{k+1}}\boldsymbol{R}$ is the rotation matrix representing orientation of the foot at instant $k+1$. Then constraints on the CoP position are changed to

$$\underbrace{-\begin{bmatrix} w \\ \ell \end{bmatrix}}_{\underline{\boldsymbol{z}}} \leq \hat{\boldsymbol{z}}_{k+1} \leq \underbrace{\begin{bmatrix} w \\ \ell \end{bmatrix}}_{\bar{\boldsymbol{z}}}, \tag{4.35}$$

where $2w$ and 2ℓ are the width and the length of the foot sole (Appendix D.1).

The structure of the constraints on the CoP positions is lost after condensing of the MPC problem as described in Appendix E. We avoid this by changing the control inputs of the model to $\hat{\boldsymbol{z}}_{k+1}$. For this purpose we reformulate the output equation again

$$\begin{aligned}\hat{\boldsymbol{z}}_{k+1} &= {}^{\mathsf{k+1}}\boldsymbol{R}\left(\begin{bmatrix}\tilde{\boldsymbol{D}} & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{D}}\end{bmatrix}\boldsymbol{x}_{k+1} - \boldsymbol{p}^{xy}_{k+1}\right)\\ &= {}^{\mathsf{k+1}}\boldsymbol{R}\left(\begin{bmatrix}\tilde{\boldsymbol{D}} & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{D}}\end{bmatrix}\boldsymbol{x}_{k} + T_k\dot{\boldsymbol{z}}_k - \boldsymbol{p}^{xy}_{k+1}\right),\end{aligned} \tag{4.36}$$

then find $\dot{\boldsymbol{z}}_k$ using it

$$\dot{\boldsymbol{z}}_k = \underbrace{-\frac{1}{T_k}\begin{bmatrix}\tilde{\boldsymbol{D}} & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{D}}\end{bmatrix}}_{\boldsymbol{D}_k}\boldsymbol{x}_k + \underbrace{\frac{1}{T_k}\begin{bmatrix}\boldsymbol{R}_{\mathsf{k+1}} & \boldsymbol{I}_2\end{bmatrix}}_{\boldsymbol{E}_k}\begin{bmatrix}\hat{\boldsymbol{z}}_{k+1}\\ \boldsymbol{p}^{xy}_{k+1}\end{bmatrix}, \tag{4.37}$$

and substitute it in the equation of dynamics of the system

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \begin{bmatrix}\tilde{\boldsymbol{A}}_k & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_k\end{bmatrix}\boldsymbol{x}_k + \frac{1}{T_k}\begin{bmatrix}\tilde{\boldsymbol{B}}_k & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{B}}_k\end{bmatrix}\left(-\begin{bmatrix}\tilde{\boldsymbol{D}} & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{D}}\end{bmatrix}\boldsymbol{x}_k + \boldsymbol{R}_{\mathsf{k+1}}\hat{\boldsymbol{z}}_{k+1} + \boldsymbol{p}^{xy}_{k+1}\right)\\ &= \underbrace{\begin{bmatrix}\tilde{\boldsymbol{A}}_k - \frac{1}{T_k}\tilde{\boldsymbol{B}}_k\tilde{\boldsymbol{D}} & \boldsymbol{0}\\ \boldsymbol{0} & \tilde{\boldsymbol{A}}_k - \frac{1}{T_k}\tilde{\boldsymbol{B}}_k\tilde{\boldsymbol{D}}\end{bmatrix}}_{\boldsymbol{A}_k}\boldsymbol{x}_k + \underbrace{\begin{bmatrix}\frac{1}{T_k}\tilde{\boldsymbol{B}}_k & \boldsymbol{0}\\ \boldsymbol{0} & \frac{1}{T_k}\tilde{\boldsymbol{B}}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{\mathsf{k+1}} & \boldsymbol{I}_2\end{bmatrix}}_{\boldsymbol{B}_k}\begin{bmatrix}\hat{\boldsymbol{z}}_{k+1}\\ \boldsymbol{p}^{xy}_{k+1}\end{bmatrix}.\end{aligned} \tag{4.38}$$

Constraints on the foot positions are also expressed as simple bounds on $\Delta\hat{\boldsymbol{p}}_j$ provided that $\mathbb{F}_j$ defines a rectangular area. In this case the model of the system takes the following form

Model PPMZ (Planar Point-Mass controlled with CoP position)

$$\begin{cases}\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\begin{bmatrix}\hat{\boldsymbol{z}}_{k+1}\\ \boldsymbol{p}^{xy}_{k+1}\end{bmatrix}, & k \in \{0,...,N-1\}, & (4.39a)\\ \hat{\boldsymbol{p}}_{j+1} = \hat{\boldsymbol{p}}_j + \hat{\boldsymbol{B}}_j\Delta\hat{\boldsymbol{p}}_j, & j \in \{0,...,K-1\}, & (4.39b)\\ \dot{\boldsymbol{z}}_k = \boldsymbol{D}_k\boldsymbol{x}_k + \boldsymbol{E}_k\begin{bmatrix}\hat{\boldsymbol{z}}_{k+1}\\ \boldsymbol{p}^{xy}_{k+1}\end{bmatrix}, & & (4.39c)\\ \boldsymbol{p}^{xy}_{k+1} = \mathrm{F}_{k+1}(\hat{\boldsymbol{p}}_0,...,\hat{\boldsymbol{p}}_K), & & (4.39d)\\ \underline{\boldsymbol{b}}_{\boldsymbol{p},j} \le \Delta\hat{\boldsymbol{p}}^{xy}_j \le \bar{\boldsymbol{b}}_{\boldsymbol{p},j}, & & (4.39e)\\ \underline{\boldsymbol{z}} \le \hat{\boldsymbol{z}}_{k+1} \le \bar{\boldsymbol{z}}, & & (4.39f)\end{cases}$$

Note that the structure of simple bounds on $\Delta\hat{\boldsymbol{p}}_{1,...,K}$ is also lost after condensing. In order to avoid this it is necessary to express $\boldsymbol{p}^{xy}_{1,...,N}$ using $\Delta\hat{\boldsymbol{p}}_{1,...,K}$ after condensing.

4.4.3 Motion of the feet in the air

PPMZ model accounts for foot positions on the ground, but neglects motion of the feet in the air. We use 3-rd order polynomials to represent this motion [Sherikov 2014, Nishiwaki 2009]. Hence, acceleration of the foot in the air at a given control instant is found using a linear function of $\Delta\hat{\boldsymbol{p}}_0$:

$$\ddot{\boldsymbol{s}}_0 = \boldsymbol{\mathcal{A}}_{sa}\Delta\hat{\boldsymbol{p}}_0 + \boldsymbol{b}_{sa}, \tag{4.40}$$

where $\boldsymbol{\mathcal{A}}_{sa}$ and $\boldsymbol{b}_{sa}$ are constructed as described in Appendix F. In a similar way we find x and y components of the foot jerk

$$\dddot{\boldsymbol{s}}^{xy} = \boldsymbol{\mathcal{A}}_{sj}\Delta\hat{\boldsymbol{p}}_0 + \boldsymbol{b}_{sj}, \tag{4.41}$$

which is constant for the whole foot trajectory. We will use $\ddot{\boldsymbol{s}}_0$ and $\dddot{\boldsymbol{s}}^{xy}$ in whole body motion controllers considered in Chapter 6.

4.5 Mixed Model Predictive Control

The traditional approach to control of humanoid robots using anticipation with approximate models is divided into two main sequential stages [Kajita 2003, Herdt 2010, Morisawa 2007, Nishiwaki 2009]:

1. anticipation of trajectories of the CoM, CoP, momenta, or contact forces using MPC or analytic formulas;
2. tracking of the generated trajectories with a whole body motion controller taking into account whole body tasks and constraints.

In the following this approach is referred to as *two-stage control*. It has several disadvantages:

- As we have learned in Section 3.2, whole body constraints are neglected in simplified models (Assumption 3.9) and must be approximated with proxy constraints. Hence, the generated trajectories can be infeasible for the whole body model. Though an infeasible trajectory may still be executed approximately [Kanehiro 2009], it is always preferable to avoid such situations.
- The whole body tasks cannot in general be expressed with an approximate model, and, thus, cannot be taken into account during anticipation without some *ad hoc* task-specific techniques [Yoshida 2006, Nishiwaki 2003, Fukumoto 2004].

In order to address these disadvantages we propose to merge the two sequential stages into one [Sherikov 2014, Sherikov 2015], Subsection 5.3.4.

Here we call our approach *Mixed Model Predictive Control* (MMPC). It consists in using both the whole body model and an approximate model within the same preview horizon. This approach should not be confused with *Multi-Model Predictive Control*, where the model may differ from one MPC iteration to another, but each preview horizon is formed using a single model [Rossiter 2003, Chapter 15]. MMPC is also different from the *Distributed Model Predictive Control*, where a complex system is broken down into several simpler interacting systems [Rawlings 2009, Chapter 6]. We are not aware of previous works using MMPC, though conceptually similar ideas were proposed for motion planning in [Dai 2014, Kanoun 2010].

We summarize advantages and limitations of MMPC in Subsection 4.5.2 after discussion of some technical details of mixing models in the preview horizon in Subsection 4.5.1.

4.5.1 Composition of the preview horizon

In order to avoid nonlinearity we introduce the whole body model of a robot only at the current instant of the preview horizon, while the rest of it is formed using an approximate model. Such combination of the models can be interpreted as a whole body MPC, where the whole body model is replaced by a simplified model everywhere except the current time instant [Sherikov 2015]. Alternatively, we can say that we construct a whole body motion controller with built-in anticipation for ensuring long-term balance [Sherikov 2014].

The initial state-control pair $(\boldsymbol{x}_0, \boldsymbol{u}_0)$ is shared by both coupled models. Therefore, when the whole body model is coupled with the momenta-based (MB) model, they share initial momenta and the contact wrenches applied on the first interval of the preview horizon $\hat{\boldsymbol{f}}_{0,\{0,\dots,M\}}$ [Sherikov 2015]. Coupling with third order point-mass models PPMdZ and PPMJ is not as trivial [Sherikov 2014]:

- The first reason for this is mismatch of the orders of models: accelerations are part of the state in the approximate models and are control variables in the whole body model. In order to alleviate this issue we include the initial CoM acceleration $\ddot{\boldsymbol{c}}_0^{xy}$ into the control vector $\boldsymbol{u}_0$ rather than state $\boldsymbol{x}_0$.
- The second problem is due to Assumption 3.14, which requires the rate of angular momentum to be zero in the point-mass models. Consequently, the same change in the CoP position results in different motions of CoM in approximate and whole body models, since the latter generates nonzero rate of angular momentum most of the time. Enforcing both the CoM and CoP motion of a point-mass model on the whole body model is infeasible or leads to unnatural and excessive upper body motions. We address this problem by coupling these models only through the CoM motion using the task

$$\boldsymbol{I}_{xy}\left(\boldsymbol{J}_{com}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{com}\dot{\boldsymbol{q}}\right) = \ddot{\boldsymbol{c}}_0^{xy}. \tag{4.42}$$

More details on the coupling is given in Chapter 6.

4.5.2 Advantages and limitations

The best quality of motion anticipation can be achieved using the complete whole body model. This quality, however, comes at a price of high computational requirements. Anticipation based on approximate models, on the other hand, is computationally cheap, but cannot directly take into account whole body tasks and constraints. In this regard, simultaneous exploitation of different models allows MMPC to compromise quality of the prediction with computational complexity.

Formulations of MMPC, which are considered in this thesis, are posed in linear least-squares optimization framework (Chapter 5). This implies that they are always computationally cheaper than anticipation with the whole body model [Koenemann 2015]. Furthermore, MMPC does not lose much in this aspect to two-stage control, since in the case of the MMPC a separate whole body motion controller is not required.

Since the whole body model is employed only at the current instant in the considered MMPC formulations, whole body tasks and constraints are taken into account only instantaneously. However, contrary to the two-stage control, these tasks and constraints directly influence the anticipated motion.

One of the side effects of the integration of the whole body motion controller in MMPC is that the whole MMPC problem must be resolved at the same frequency as the traditional whole body motion control is performed. This might be considered as a drawback in comparison to the two-stage control, where the rate of update of anticipated motion can be chosen independently. We do not see this as a major issue, since frequent regeneration of anticipated motion was shown to be advantageous [Nishiwaki 2009], and we believe that the modern hardware is capable of performing MMPC online. There are, however, some technical difficulties related to the discretization of approximate models, which become more apparent in MMPC. We discuss these difficulties in the following Section 4.6.

4.6 Duration and sampling of the preview horizon

Performance of an MPC controller is to a large extent determined by duration of the preview horizon and its sampling. Values of these parameters are tuned for the best performance and depend on each other and properties of the controlled system [Rossiter 2003, Chapter 5].

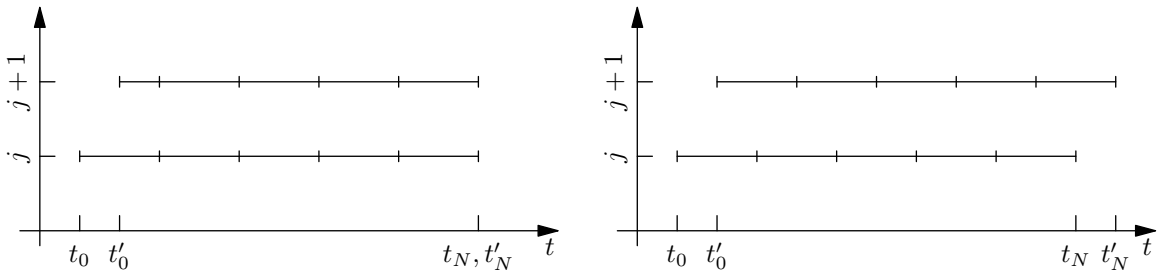
Duration of the preview horizon $H = t_N - t_0$ of an MPC problem based on the approximate models considered in this thesis is usually in the order of seconds [Kajita 2003, Herdt 2010,

Audren 2014]. In the case of walking it corresponds to 2-3 steps of the robot. The preview horizon is typically sampled uniformly, *i.e.*, $T_k = T$ for all $k \in \{0, ..., N-1\}$ and $H = NT$.

In general, the duration of a sampling interval T is chosen by taking into account several factors:

- Frequency at which sensor readings of the state are updated.
- Computational complexity of the underlying optimization problem. The complexity grows with N, *i.e.*, with the number of variables and constraints, but not with H.
- Frequency at which the constraints must be enforced.

The last two factors dominate the choice of T in the MPC applications considered here. In particular, in [Herdt 2010] T is chosen in such a way that imposing constraints on the CoP position during double supports is avoided. This is achieved in the following way: (**i**) duration of a transitional double support between two adjacent single supports is taken to be equal to $T = 100$ $[ms]$, (**ii**) the single support CoP constraints are imposed at the beginning and end of the double support omitting the less restrictive double support CoP constraint. Explicit double support constraints are avoided, since they are nonlinear with respect to the contact positions, when these positions are not predetermined (see [Herdt 2010] and Appendix D.2). Even when double support constraints are imposed explicitly, reducing T below 100 $[ms]$ is undesirable due to increase of computational complexity.



(a) Reduction of the first sampling interval. **(b)** Shift of the whole preview horizon.

Figure 4.4: Sampling of the preview horizon on iterations j and $j+1$ with $T_c = T/2$ and $t'_0 = t_0 + T_c$.

The desired frequency of whole body motion control is higher than 200 $[Hz]$, which corresponds to duration of a control interval T_c in the order of milliseconds [Kuindersma 2014, Herzog 2015, Saab 2013]. Therefore, if an MPC problem is resolved at such frequency, in the form of MMPC or as a part of two-stage control, it is necessary to address discrepancy between T and T_c. In this work we tried to employ two heuristics for this purpose:

- Reducing the first sampling interval T_0 by T_c on each control iteration (Figure 4.4a). This can be interpreted as gradual removal of implicit *move blocking* constraints, which impose that the control input is constant during T_0 (see also [Cagienard 2007]).
- Shifting the whole preview horizon by T_c on each control iteration (Figure 4.4b). The time instants at which constraints are imposed are shifted too. Thus, there is a problem of synchronization of changes in the model with the boundaries of sampling intervals as suggested in Subsection 4.2.4. For example, consider a situation when a switch from a double to a single support is scheduled at time $t_0 + T$. If the constraint on the CoP position is shifted from $t_0 + T$ with the preview horizon, there is a risk that the anticipated motion does not comply with this constraint at the instant of the switch. If we preserve constraints at such instants, the underlying optimization problem often becomes infeasible since the considered heuristic impairs recursive feasibility even in the case of time-invariant systems.

In either case, the structure of the MPC problem changes significantly from one control iteration to another due to changes in constraints. As a result, the solutions obtained on subsequent iterations in general are not equal on interval $[t'_0, t_N]$, even in an ideal situation. In our experience, this results in variations with period T in solutions. Similar behavior observed in [Henze 2014] is presumably caused by the same reason. The second heuristic, however, yields much better results, which are demonstrated later in Chapter 6.

In conclusion of this discussion, it is necessary to emphasize that further investigation of the sampling issue is needed, since neither of the proposed heuristics is completely satisfactory.

4.7 Conclusion

This chapter develops on the ideas and results of Chapters 2 and 3. We introduce *Model Predictive Control* (MPC) as the tool for motion anticipation and discretize approximate models constructed in Chapter 3, so that they can be used in MPC. Then we construct capturability constraints for these models to enforce capturability of anticipated motion as suggested in Chapter 2. In addition to this, we present our *Mixed Model Predictive Control* (MMPC) approach to whole body motion control and anticipation [Sherikov 2014, Sherikov 2015], and discuss some technical details of MPC implementation.

Chapter 5

Prioritized Linear Least-Squares Optimization

Whole-body motion control and motion anticipation problems considered in Chapters 3 and 4 fall within the framework of *Prioritized Linear Least-Squares* (PLLS) optimization, which is the topic of the present chapter. We briefly outline the general concepts behind PLLS optimization in Sections 5.1 and 5.2 without going into details of the algorithms for solution of PLLS problems. In the following Section 5.3 we present a number of examples of PLLS problems used in our works. Section 5.4 lists a few general techniques, which allow for increasing the computational performance of the PLLS solvers.

5.1 Introduction to Linear Least Squares Optimization

The basic building block of the PLLS framework is a system of linear inequalities

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}, \tag{5.1}$$

where $\boldsymbol{\chi}$ is the vector of decision variables, $\boldsymbol{\mathcal{A}}$ is some matrix, $\underline{\boldsymbol{b}}$ and $\bar{\boldsymbol{b}}$ are vectors of lower and upper bounds. This system is satisfied in the least-squares sense, and, hence, can be posed as a *Quadratic Program* (QP)

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}}{\text{minimize}} \quad & \|\boldsymbol{v}\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} - \boldsymbol{v} \leq \bar{\boldsymbol{b}}, \end{aligned} \tag{5.2}$$

where vector $\boldsymbol{v}$ contains *violations* of the respective inequalities [Bramley 1994, Escande 2014]. A solution $(\boldsymbol{\chi}^\star, \boldsymbol{v}^\star)$ of (5.2) always exists, but may be nonunique.

In the rest of the thesis we call a system of inequalities in the form (5.1) an *objective*, individual inequality in an objective is called a *constraint*. For convenience, constraints in an objective are subdivided into groups called *tasks*, since they often correspond to tasks of the whole body motion control introduced in Subsection 3.1.5. We presume that all objectives are satisfied in the least-squares sense, therefore, vectors of violations $\boldsymbol{v}$ are usually omitted for simplicity. We also presume, that systems of constraints are solved using an *active set* strategy [Nocedal 2006, Chapter 16]. Many presentational choices of this chapter are dictated by the intrinsic properties of active set strategies, whose key idea is iterative *activation* and *deactivation* of inequality constraints in the search for a solution. A constraint is called *active* if it holds as an equality, *i.e.*, equality constraints with $\underline{\boldsymbol{b}} = \bar{\boldsymbol{b}}$ are always active, inequality constraints are active when their bounds are reached or violated.

5.2 Prioritization of objectives

Let us consider an equality objective composed of two tasks

$$\begin{bmatrix}\gamma_1\boldsymbol{\mathcal{A}}_1\\ \gamma_2\boldsymbol{\mathcal{A}}_2\end{bmatrix}\boldsymbol{\chi} = \begin{bmatrix}\gamma_1\boldsymbol{b}_1\\ \gamma_2\boldsymbol{b}_2\end{bmatrix}, \tag{5.3}$$

where γ_1 and γ_2 are positive scalars. This objective corresponds to the multicriterion optimization problem [Boyd 2004, Chapter 4]

$$\underset{\boldsymbol{\chi}}{\text{minimize}} \quad \gamma_1^2\|\boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} - \boldsymbol{b}_1\|_2^2 + \gamma_2^2\|\boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} - \boldsymbol{b}_2\|_2^2. \tag{5.4}$$

Tasks are said to be compatible, if they are violated to the same extent when combined into an objective and when satisfied independently from each other. Weights γ_1 and γ_2 determine a trade-off between incompatible tasks, and have no effect otherwise.

In many practical situations trade-offs between tasks are unacceptable, since some of the tasks have *strictly* (or *infinitely*) higher priority than others. The classic example of such strict prioritization is *The Three Laws of Robotics* introduced by Isaac Asimov in his science-fiction works [Wikipedia 2015b]:

Hierarchy (5.1)

1: *A robot may not injure a human being or, through inaction, allow a human being to come to harm.*

2: *A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.*

3: *A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.*

A set of tasks organized in accordance with their priorities is called a *hierarchy* or *stack of tasks* [Mansard 2009].

Let us assign strictly higher priority to the first task in (5.3) to obtain the hierarchy composed of two objectives

Hierarchy (5.2)

1: $\boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} = \boldsymbol{b}_1$

2: $\boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} = \boldsymbol{b}_2$

where weights γ_1 and γ_2 are meaningless and therefore omitted. Hierarchies of such form were originally introduced in the field of robotics to exploit *redundancy* of manipulators with respect to the primary objective [Liégeois 1977]. The secondary objective is then used to express preferences in the way of execution of the primary objective, which can be realized nonuniquely. Prioritization prevents degradation of performance of the primary objective due to secondary objectives.

5.2.1 Solving a hierarchy

The classic way to obtain the solution of Hierarchy 5.2 is to perform null space projections using a generalized pseudoinverse $(\cdot)^{\#}$ [Siciliano 1991]:

$$\boldsymbol{\chi}^\star = \boldsymbol{\mathcal{A}}_1^{\#}\boldsymbol{b}_1 + \Big(\boldsymbol{\mathcal{A}}_2(\boldsymbol{I} - \boldsymbol{\mathcal{A}}_1^{\#}\boldsymbol{\mathcal{A}}_1)\Big)^{\#}(\boldsymbol{b}_2 - \boldsymbol{\mathcal{A}}_2\boldsymbol{\mathcal{A}}_1^{\#}\boldsymbol{b}_1) \tag{5.5}$$

An arbitrary number of priority levels can be handled with iterative projections.

Projections have a number of disadvantages, in particular, they do not account for inequalities. This drawback was addressed in [Kanoun 2011] using cascades of QP's. For example, a hierarchy of two inequality objectives

Hierarchy (5.3)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2$

can be solved with two QP's:

1. First, we find minimal violation of the first objective $\boldsymbol{v}_1^\star$ using

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}_1}{\text{minimize}} \quad & \|\boldsymbol{v}_1\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} - \boldsymbol{v}_1 \leq \bar{\boldsymbol{b}}_1. \end{aligned} \tag{5.6}$$

2. Then the optimal violation of the second objective is determined with

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}_2}{\text{minimize}} \quad & \|\boldsymbol{v}_2\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} - \boldsymbol{v}_1^\star \leq \bar{\boldsymbol{b}}_1 \\ & \underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} - \boldsymbol{v}_2 \leq \bar{\boldsymbol{b}}_2. \end{aligned} \tag{5.7}$$

A similar approach was proposed in the field of *Model Predictive Control* (MPC) to cope with infeasibilities of constraints [Vada 1999].

Recent developments in numerical methods, however, allow for solution of hierarchies with equalities and inequalities in much more efficient way than null space projections and cascades of QP's [Escande 2014, Dimitrov 2015]. Efficiency is achieved using dedicated active set strategies, one of which was implemented in our research group in the software package `LexLS` [Dimitrov 2015]. `LexLS` is the primary optimization tool employed in this thesis.

5.2.2 Singularities and regularization

It is recognized that solvers for hierarchies may experience numerical difficulties near singularities [Siciliano 1991, Deo 1995, Kanoun 2011]. In robotic applications this leads to control inputs taking unacceptably high values, if unconstrained, or flipping between the bounds at high frequency otherwise.

One way to cope with deteriorated behavior near singularities is to introduce regularization. Standard *Tikhonov regularization* is implemented by extending the ℓ-th objective as follows

$$\begin{bmatrix} \underline{\boldsymbol{b}}_\ell \\ \boldsymbol{0} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{\mathcal{A}}_\ell \\ \gamma_{r,\ell}\boldsymbol{I} \end{bmatrix} \boldsymbol{\chi} \leq \begin{bmatrix} \bar{\boldsymbol{b}}_\ell \\ \boldsymbol{0} \end{bmatrix}, \tag{5.8}$$

where $\gamma_{r,\ell} \in \mathbb{R}_{\geq 0}$ is a *damping factor*, which is used to trade-off accuracy of the solution with the norm of $\boldsymbol{\chi}^\star$. The best results are achieved, when the value of $\gamma_{r,\ell}$ is chosen automatically and is fading to zero far from singularities. The objective (5.8) corresponds to the QP

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}_\ell}{\text{minimize}} \quad & \|\boldsymbol{v}_\ell\|_2^2 + \gamma_{r,\ell}^2\|\boldsymbol{\chi}\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}}_j \leq \boldsymbol{\mathcal{A}}_j\boldsymbol{\chi} - \boldsymbol{v}_j^\star \leq \bar{\boldsymbol{b}}_j, \quad j \in \{1, ..., \ell-1\}, \\ & \underline{\boldsymbol{b}}_\ell \leq \boldsymbol{\mathcal{A}}_\ell\boldsymbol{\chi} - \boldsymbol{v}_\ell \leq \bar{\boldsymbol{b}}_\ell. \end{aligned} \tag{5.9}$$

Note that this optimization problem has a unique solution $\boldsymbol{\chi}$. When a cascade of QP's is used to solve the hierarchy, this means that the regularizing task $\gamma_{r,\ell}\boldsymbol{I}\boldsymbol{\chi} = \boldsymbol{0}$ must be omitted in $\ell + 1$ QP in order to be able to execute objectives of lower importance [Kanoun 2009, Chapter 3]

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}_{\ell+1}}{\text{minimize}} \quad & \|\boldsymbol{v}_{\ell+1}\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}}_j \leq \boldsymbol{\mathcal{A}}_j\boldsymbol{\chi} - \boldsymbol{v}_j^\star \leq \bar{\boldsymbol{b}}_j, \qquad j \in \{1, ..., \ell\}, \\ & \underline{\boldsymbol{b}}_{\ell+1} \leq \boldsymbol{\mathcal{A}}_{\ell+1}\boldsymbol{\chi} - \boldsymbol{v}_{\ell+1} \leq \bar{\boldsymbol{b}}_{\ell+1}. \end{aligned} \tag{5.10}$$

Similarly, dedicated solvers like those proposed in [Escande 2014, Dimitrov 2015] should implement special logic to handle regularization tasks.

We believe that Tikhonov regularization is not the best choice in general. For example, if $\boldsymbol{\mathcal{A}}_\ell = \boldsymbol{I}$ it is more reasonable to bias $\boldsymbol{\chi}$ towards $\boldsymbol{b}_{r,\ell} = \underline{\boldsymbol{b}}_\ell + (\bar{\boldsymbol{b}}_\ell - \underline{\boldsymbol{b}}_\ell)/2$ rather than $\boldsymbol{0}$. Hence, we advocate for regularization using a general matrix $\boldsymbol{\mathcal{A}}_{r,\ell}$ and vector $\boldsymbol{b}_{r,\ell}$ [Sherikov 2015]

$$\begin{bmatrix} \underline{\boldsymbol{b}}_\ell \\ \gamma_{r,\ell}\boldsymbol{b}_{r,\ell} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{\mathcal{A}}_\ell \\ \gamma_{r,\ell}\boldsymbol{\mathcal{A}}_{r,\ell} \end{bmatrix} \boldsymbol{\chi} \leq \begin{bmatrix} \bar{\boldsymbol{b}}_\ell \\ \gamma_{r,\ell}\boldsymbol{b}_{r,\ell} \end{bmatrix}, \tag{5.11}$$

which corresponds to the least-squares problem

$$\begin{aligned} \underset{\boldsymbol{\chi},\boldsymbol{v}_\ell}{\text{minimize}} \quad & \|\boldsymbol{v}_\ell\|_2^2 + \gamma_{r,\ell}^2 \|\boldsymbol{\mathcal{A}}_{r,\ell}\boldsymbol{\chi} - \boldsymbol{b}_{r,\ell}\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}}_j \leq \boldsymbol{\mathcal{A}}_j\boldsymbol{\chi} - \boldsymbol{v}_j^\star \leq \bar{\boldsymbol{b}}_j, \qquad j \in \{1, ..., \ell - 1\}, \\ & \underline{\boldsymbol{b}}_\ell \leq \boldsymbol{\mathcal{A}}_\ell\boldsymbol{\chi} - \boldsymbol{v}_\ell \leq \bar{\boldsymbol{b}}_\ell. \end{aligned} \tag{5.12}$$

Regularization can be tuned by changing the damping factor $\gamma_{r,\ell}$. The choice of $\boldsymbol{\mathcal{A}}_{r,\ell}$ and $\boldsymbol{b}_{r,\ell}$ may appear to be nontrivial, but in most practical applications these terms are already defined and imposed on the very last level of the hierarchy to resolve any remaining redundancy. Our practical experience also suggests, that regularization with the last objective of a hierarchy is much easier to tune than Tikhonov regularization.

5.3 Examples and applications

A hierarchy of objectives is not a new concept, but it has experienced a significant growth of interest in the recent years due to its spreading in the control of humanoid robots [Kanoun 2009, Saab 2013, Sentis 2007]. We believe that the reason for this is not only the power of prioritization, but also the clearness and conciseness of representation of robot control problems in the form of a hierarchy [Dimitrov 2014]. The second reason, however, led to what we believe to be a misuse of hierarchies and to a certain disappointment in them.

We would like to stress that posing optimization problems as hierarchies is not always meaningful, but should be considered for the following purposes

- Relaxation of objectives and resolution of conflicts between them by introducing strict priorities. Note that in the literature it is common to prioritize objectives even if they are compatible [Sentis 2007, Chapter 5], [Dietrich 2015, Saab 2013]. Consequently, many proposed hierarchies can be reformulated as QP's and solved with off-the-shelf software without qualitative changes in the robot behavior.
- Increase of the computational performance due to variable eliminations. Prioritization of the objectives is equivalent to variable eliminations, which are commonly performed in robot control as preliminary steps [Dimitrov 2014], and, likewise, allows for faster solution of optimization problems [Dimitrov 2015]. This point is explained in more detail using an example in Subsection 5.3.1.

In this section we brought together several examples to illustrate situations where hierarchies are useful. The ideas behind these examples are general and are not limited to humanoid robot control.

5.3.1 Variable elimination

Variable elimination, which is ubiquitously performed in robotics, can be perceived as prioritization of constraints. We illustrate this with a simple hierarchy corresponding to an MPC problem

Hierarchy (5.4)

1: $\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k$
$\boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1}$
$\boldsymbol{u}_k \in \mathbb{U}_k$
$\boldsymbol{x}_N \in \mathbb{T}$
2: $\boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{x}_{k+1} = \boldsymbol{0}$
$\boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{u}_k = \boldsymbol{0}$

Decision variables: $\boldsymbol{\chi} = (\boldsymbol{x}_{k+1}, \boldsymbol{u}_k)$ with $k \in \{0, ..., N-1\}$

where $\boldsymbol{x}_N \in \mathbb{T}$ is a terminal or capturability constraint, $\boldsymbol{\Gamma}_{\boldsymbol{x}}$ and $\boldsymbol{\Gamma}_{\boldsymbol{u}}$ are some weighting matrices.

In practice it is common to perform so called *condensing* of the considered MPC problem [Bock 1984], Appendix E. The idea of condensing consists in

- expressing states $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ through the current state $\boldsymbol{x}_0$ and control inputs $(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1})$ using the equation of dynamics of the system and
- elimination of $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ from the problem.

This procedure can be interpreted as satisfying the dynamics first and the remaining objectives later and, thus, can be posed as the hierarchy

Hierarchy (5.5)

1: $\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k$
2: $\boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1}$
$\boldsymbol{u}_k \in \mathbb{U}_k$
$\boldsymbol{x}_N \in \mathbb{T}$
3: $\boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{x}_{k+1} = \boldsymbol{0}$
$\boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{u}_k = \boldsymbol{0}$

Decision variables: $\boldsymbol{\chi} = (\boldsymbol{x}_{k+1}, \boldsymbol{u}_k)$ with $k \in \{0, ..., N-1\}$

The null space of the objective 1, where objectives 2 and 3 are satisfied, is of lower dimension than the space of all vectors $\boldsymbol{\chi}$ [Kanoun 2011, de Lasa 2010, Dimitrov 2015]. A PLLS solver can exploit this fact to automatically eliminate variables and avoid unnecessary computations [Dimitrov 2015].

Representation of variable eliminations with hierarchies is appealing, since it emphasizes the essence of the problem rather than implementation details. Therefore, it encourages development of general approaches to solving hierarchies instead of *ad-hoc* methods for solving particular optimization and control problems. Note that generality does not imply lower performance, since the solvers can take advantage of the structure of the problem as it is done during manual elimination of variables (see Section 5.4).

The above conclusion is not limited to condensing in MPC and applies to many problems in robotics, for example, elimination of the external forces in whole body motion controllers [Sentis 2007, Chapter 2] [Mansard 2012], or elimination of joint torques [Herzog 2015].

5.3.2 Relaxation of capturability and terminal constraints

Consider Hierarchy 5.4 corresponding to a general MPC problem. It is recognized that the terminal (or capturability) constraint in it may be the source of infeasibility [Rossiter 2003, Chapter 8]. If, for various reasons, neither the length of the preview horizon nor the terminal set $\mathbb{T}$ can be adjusted to avoid infeasibility, a reasonable option is relaxation. In this case we have to take into account two goals: (**i**) relaxation of the terminal constraint should not interfere with other high priority tasks, (**ii**) low priority objectives should not interfere with

satisfaction of the terminal constraint. We achieve both goals by adding a separate priority level for the terminal or capturability constraint [Sherikov 2014]

Hierarchy (5.6)

$$
\begin{aligned}
&\textbf{1:}\ \boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k \\
&\quad\ \boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1} \\
&\quad\ \boldsymbol{u}_k \in \mathbb{U}_k \\
&\textbf{2:}\ \boldsymbol{x}_N \in \mathbb{T} \\
&\textbf{3:}\ \boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{x}_{k+1} = \boldsymbol{0} \\
&\quad\ \boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{u}_k = \boldsymbol{0}
\end{aligned}
$$

Decision variables: $\boldsymbol{\chi} = (\boldsymbol{x}_{k+1}, \boldsymbol{u}_k)$ with $k \in \{0, ..., N-1\}$

5.3.3 Time optimal Model Predictive Control

In some situations, hierarchies can be used for expressing backup control goals, for example, hierarchy

Hierarchy (5.7)

$$
\begin{aligned}
&\textbf{1:}\ \underline{\boldsymbol{b}} \leq \boldsymbol{\chi} \leq \bar{\boldsymbol{b}} \\
&\ldots\ \ \ldots \\
&\boldsymbol{\ell}\textbf{:}\ \boldsymbol{\chi} = \underline{\boldsymbol{b}} + \frac{1}{2}(\bar{\boldsymbol{b}} - \underline{\boldsymbol{b}})
\end{aligned}
$$

can be interpreted as follows: if satisfaction of the equality task on $\boldsymbol{\chi}$ on level ℓ is impossible, fall back to bounding of $\boldsymbol{\chi}$. We combined this idea with the idea of hierarchical relaxation of the terminal constraint to obtain a time optimal MPC scheme hinted in [Kerrigan 2000, Chapter 8] and implemented in [Homsi 2016b]. The goal of a time optimal controller is to reach the desired state $\boldsymbol{x}^{\heartsuit}$ in minimal time. Hence, we can first try to achieve the goal within the first sampling interval of the preview horizon, if not impossible – within 2 intervals, and so on until the end of preview horizon is reached. We can formalize this process with a single hierarchy, where reaching $\boldsymbol{x}^{\heartsuit}$ in $k+1$ sampling intervals in the preview horizon is infinitely more important than reaching it in k intervals:

Hierarchy (5.8)

$$
\begin{aligned}
&\textbf{1:}\ \boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k \\
&\quad\ \boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1} \\
&\quad\ \boldsymbol{u}_k \in \mathbb{U}_k \\
&\textbf{2:}\ \boldsymbol{x}_N = \boldsymbol{x}^{\heartsuit} \\
&\ldots\ \ \ldots \\
&\boldsymbol{N+1}\textbf{:}\ \boldsymbol{x}_1 = \boldsymbol{x}^{\heartsuit} \\
&\boldsymbol{N+2}\textbf{:}\ \boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{x}_{k+1} = \boldsymbol{0} \\
&\quad\ \boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{u}_k = \boldsymbol{0}
\end{aligned}
$$

Decision variables: $\boldsymbol{\chi} = (\boldsymbol{x}_{k+1}, \boldsymbol{u}_k)$ with $k \in \{0, ..., N-1\}$

More information on implementation of this time optimal controller and results of its evaluation can be found in [Homsi 2016b].

5.3.4 Mixed Model Predictive Control

A sequence of actions can often be interpreted as a strict hierarchy between them. We use this insight to present our *Mixed Model Predictive Control* (MMPC) approach from a different perspective.

The traditional approach to control of humanoid robots consists of two sequential stages (see Section 4.5)

1. Anticipation, which can be performed using MPC in the form of Hierarchy 5.6. Assuming that the anticipation is performed at the same rate as whole body motion control, we extract the initial control $\boldsymbol{u}_0^\star$ from the solution of the MPC to feed it to the whole body motion controller.
2. Execution of the desired values $\boldsymbol{u}_0^\star$ obtained from the solution of Hierarchy 5.6 by a whole body motion controller. The controller is also based on a hierarchy, which incorporates the whole body WB model presented in Subsection 3.1.3:

Hierarchy (5.9)

1: Dynamics of the robot

- $\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{h} = \boldsymbol{I}_{\boldsymbol{\tau}}\boldsymbol{\tau} + m\boldsymbol{J}_{com}^\top \boldsymbol{g} + \sum_{i=1}^{M} \boldsymbol{J}_i^\top \begin{bmatrix} \boldsymbol{V}_i \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix}$

Fixed contact positions

- $\boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} = \boldsymbol{0}$

Mechanical joint constraints

- $\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}}$
- $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$

Constraints on the contact wrenches

- $\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}$
- $\boldsymbol{\lambda}_i \geq \boldsymbol{0}$

A task with some $\boldsymbol{\mathcal{A}}_t$ and $\boldsymbol{b}_t$ for tracking of the desired $\boldsymbol{u}_0^\star$

- $\boldsymbol{\mathcal{A}}_t \begin{bmatrix} \boldsymbol{\chi} \\ \boldsymbol{u}_0^\star \end{bmatrix} = \boldsymbol{b}_t$

2: Arbitrary whole body tasks (see Subsection 3.1.5).

Decision variables: $\boldsymbol{\chi} = (\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$ with $i \in \{1, ..., M\}$

After concatenation of Hierarchies 5.6 and 5.9 and shuffling of the tasks in the resulting hierarchy we obtain an MMPC controller [Sherikov 2014, Sherikov 2015]

Hierarchy (5.10)

1: Tasks of the whole body motion controller

- $\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{h} = \boldsymbol{I}_{\boldsymbol{\tau}}\boldsymbol{\tau} + m\boldsymbol{J}_{com}^\top \boldsymbol{g} + \sum_{i=1}^{M} \boldsymbol{J}_i^\top \begin{bmatrix} \boldsymbol{V}_i \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix}$
- $\boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} = \boldsymbol{0}$
- $\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}}$
- $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$
- $\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}$
- $\boldsymbol{\lambda}_i \geq \boldsymbol{0}$

Tasks of the MPC controller

- $\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k$

- $\boldsymbol{x}_{k+1} \in \mathbb{X}_{k+1}$
- $\boldsymbol{u}_k \in \mathbb{U}_k$

The coupling task

- $\boldsymbol{\mathcal{A}}_t \boldsymbol{\chi} = \boldsymbol{b}_t$

2: The capturability constraint

- $\boldsymbol{x}_N \in \mathbb{T}$

3: Arbitrary whole body tasks
Tasks of the MPC controller

- $\boldsymbol{\Gamma}_{\boldsymbol{x}} \boldsymbol{x}_{k+1} = \boldsymbol{0}$
- $\boldsymbol{\Gamma}_{\boldsymbol{u}} \boldsymbol{u}_k = \boldsymbol{0}$

Decision variables: $\boldsymbol{\chi} = (\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \boldsymbol{x}_{k+1}, \boldsymbol{u}_k)$
with $i \in \{1, ..., M\}$ and $k \in \{0, ..., N-1\}$

More detailed examples of this hierarchy are given in Chapter 6.

5.3.5 Minimization of an optional contact force

We extended Hierarchy 5.10 in [Sherikov 2015] in such a way, that the controller applies non-zero contact force at a certain contact only when it is necessary to maintain balance or execute a whole body task. This is achieved with a hierarchy, which can roughly be stated as

Hierarchy (5.11)

1: maintain balance
2: execute whole body task
3: minimize optional contact force

and is considered in detail in Section 6.2.

5.4 Solving hierarchies efficiently

Computational efficiency is one of the most important factors in real time controllers. Hence, we aim at efficient resolution of hierarchies used in our controllers. In order to achieve this, we employ a number of techniques, which are reviewed in this section and are supported by `LexLS` to various degrees. The techniques are of general nature and are shared with conventional active set QP solvers, whose performance is studied extensively in the literature [Herceg 2015, Wang 2010, Ferreau 2008].

5.4.1 Exploitation of the problem structure

One of the ways to improve performance of the solver is to shape the optimization problem in a beneficial manner and to inform the solver about the structure of the problem.

5.4.1.1 Two-sided inequalities

We formulate all objectives in such a way that they have both upper and lower bounds:

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}} \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}. \tag{5.13}$$

The reason for this is that bounds $\underline{\boldsymbol{b}} < \bar{\boldsymbol{b}}$ cannot be activated simultaneously, and an active set solver can exploit this fact to reduce computational load. If one of the bounds is undefined

it can be replaced by some "very large" number. When $\underline{\boldsymbol{b}} = \bar{\boldsymbol{b}} = \boldsymbol{b}$ the constraints are treated as equalities, which are always active.

5.4.1.2 Simple bounds

In some situations it is possible to express general inequalities (5.13) as *simple bounds* or *box constraints* on the decision variables:

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}. \tag{5.14}$$

Handling of such constraints can be implemented in a very efficient way [Gill 1984, Ferreau 2008, Dimitrov 2015]. For this reason, we reformulate MPC problems to convert general inequalities to simple bounds, *e.g.*, [Dimitrov 2011a], Subsection 4.4.2.

5.4.1.3 Sparsity

We call a task *sparse* if the corresponding matrix $\boldsymbol{\mathcal{A}}$ contains a large number of zeros. A task with simple bounds (5.14) is a typical example of a sparse task. Other examples are the tasks on the first level of Hierarchy 5.4 corresponding to MPC constraints. This type of sparsity is sometimes exploited in QP solvers designed for MPC problems [Wang 2010, Dimitrov 2011b]. `LexLS` currently can take advantage of the sparsity only in the case of simple bounds on the first level of a hierarchy.

5.4.2 Early termination

Control problems are typically resolved with a constant rate, which means that there is an upper limit on the time available for computations. Hence, the solvers should provide a mechanism for early termination in order to fit within the limits [Ferreau 2008, Wang 2010]. Termination can be triggered by a timer or the number of iterations of the solver. One iteration of an active set solver typically corresponds to activation or deactivation of a single constraint.

Early termination is potentially dangerous since the solution returned by the solver is suboptimal. In the PLLS framework it implies that violations of objectives can be unacceptably large, *i.e.*, high priority constraints may not be satisfied even if they are feasible. This is not an issue, if a solver is provided with an initial guess of the solution, which is feasible with respect to the primary tasks, and the solver does not increase violations of the objectives while searching for the solution.

5.4.3 Warm start

Since the hierarchies for control of robots are resolved at high frequency, the solutions usually do not change much from one control iteration to another [Kuindersma 2014, Escande 2014, Sherikov 2014]. This allows to use the solution $\boldsymbol{\chi}^\star$ and the active set obtained at i-th iteration to warm start the solver on $i+1$ iteration.

5.4.3.1 Guessing the solution

Ideally, a solution guess should be feasible with respect to the high priority objectives, in this case it is safe to terminate the solver before the solution is found. However, determination of such a guess for hierarchies employed in this thesis is not trivial, and we simply reuse the solution $\boldsymbol{\chi}^\star$ from the previous control iteration. When the dimension of $\boldsymbol{\chi}$ changes, for example, due to a switch from a single to a double support, it is necessary to drop parts of

the previous solution or provide a guess for the missing parts of $\boldsymbol{\chi}$. Our main heuristic for assigning the missing parts is to avoid activation of the corresponding inequality constraints.

An alternative approach to generate a guess is to solve an auxiliary hierarchy, which has roughly the following structure

Hierarchy (5.12)

1: Important equality tasks.

2: Important inequality tasks $\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$ converted to equalities $\boldsymbol{\mathcal{A}}\boldsymbol{\chi} = \underline{\boldsymbol{b}} + \frac{1}{2}(\bar{\boldsymbol{b}} - \underline{\boldsymbol{b}})$ and weighted with respect to each other.

and guarantees satisfaction of primary equality constraints if the solver is terminated prematurely. Solution of this hierarchy may result in a better guess, but it is time consuming and the weights have to be tuned carefully depending on the setting.

5.4.3.2 Guessing the active set

In addition to the solution guess we provide the solver with a guess of the active set, *i.e.*, the set of active inequality constraints at the solution [Ferreau 2008, Escande 2014, Kuindersma 2014]. Since the number of constraints also changes from one control iteration to another, we employ a number of task specific heuristics for modifying the active set when such changes occur.

5.4.4 Preprocessing

Since `LexLS` has rather limited capabilities for exploiting the problem structure (see Subsection 5.4.1), we perform several preprocessing steps to improve performance.

5.4.4.1 Variable elimination

During the preprocessing, we eliminate some of the variables in the hierarchies to reduce computational load. In particular, we eliminate the whole body joint torques $\boldsymbol{\tau}$ as described in Subsection 3.2.1 and perform condensing of MPC problems [Bock 1984], Appendix E.

5.4.4.2 Removing excessive simple bounds

Simple bounds on the joint accelerations $\ddot{\boldsymbol{q}}'$ are overdetermined as explained in Appendix C. Due to simplicity of these constraints, it is possible to reduce their number using a trivial procedure.

5.4.4.3 Removing excessive equality constraints

In some cases the matrix $\boldsymbol{\mathcal{A}}$ in an equality task $\boldsymbol{\mathcal{A}}\boldsymbol{\chi} = \boldsymbol{b}$ includes linearly dependent constraints. If this task does not change on each control iteration, it is beneficial to remove excessive constraints using a *QR-decomposition* of $\boldsymbol{\mathcal{A}}$ with column pivoting [Golub 1996, Chapter 5].

5.5 Conclusion

We introduced the *Prioritized Linear Least-Squares* (PLLS) framework and illustrated it with several optimization problems proposed in our works [Sherikov 2014, Sherikov 2014, Homsi 2016b]. In particular, we presented an interpretation of *Mixed Model Predictive Control* (MMPC) problem as a result of merging and reprioritization of motion anticipation and

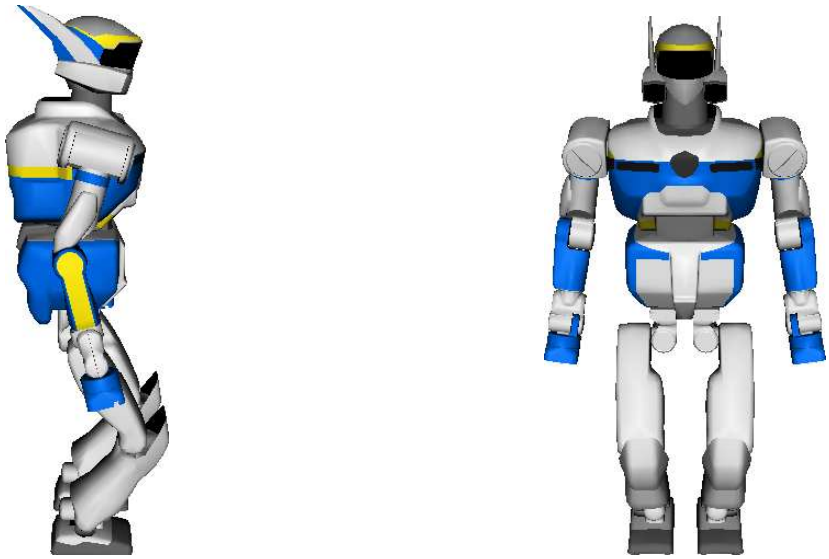
whole body motion controllers. In addition to that we outlined several techniques, which we employ to reduce computational load when solving PLLS problems.

Chapter 6

Simulations and experiments

In the present chapter we compile and extend simulation results presented in [Sherikov 2014, Sherikov 2015] and overview results obtained in [Brasseur 2015b, Agravante 2016a, Homsi 2016b]. Section 6.1 is based on [Sherikov 2014] and is focused on the interplay between whole body tasks and walking motions in a *Mixed Model Predictive Control* (MMPC) problem. The next Section 6.2 describes a more sophisticated MMPC controller, where we introduce prioritization in contact force distribution. Both sections highlight the importance of the capturability constraints for balance preservation. The final Section 6.3 discusses results of the collaborative works [Brasseur 2015b, Agravante 2016a, Homsi 2016b]. Videos illustrating the presented works can be found on the web page of the author [SHE 2016].

Controllers considered in Sections 6.1 and 6.2 are formulated as *Prioritized Linear Least-Squares* (PLLS) problems, which are solved using a dedicated solver – `LexLS` [Dimitrov 2015]. `LexLS` is implemented in `C++` using the `Eigen` template library [Guennebaud 2010]. The solver is compiled to a binary module for the `Octave` environment [OCT 2016], where we implemented all functionalities required for simulations.

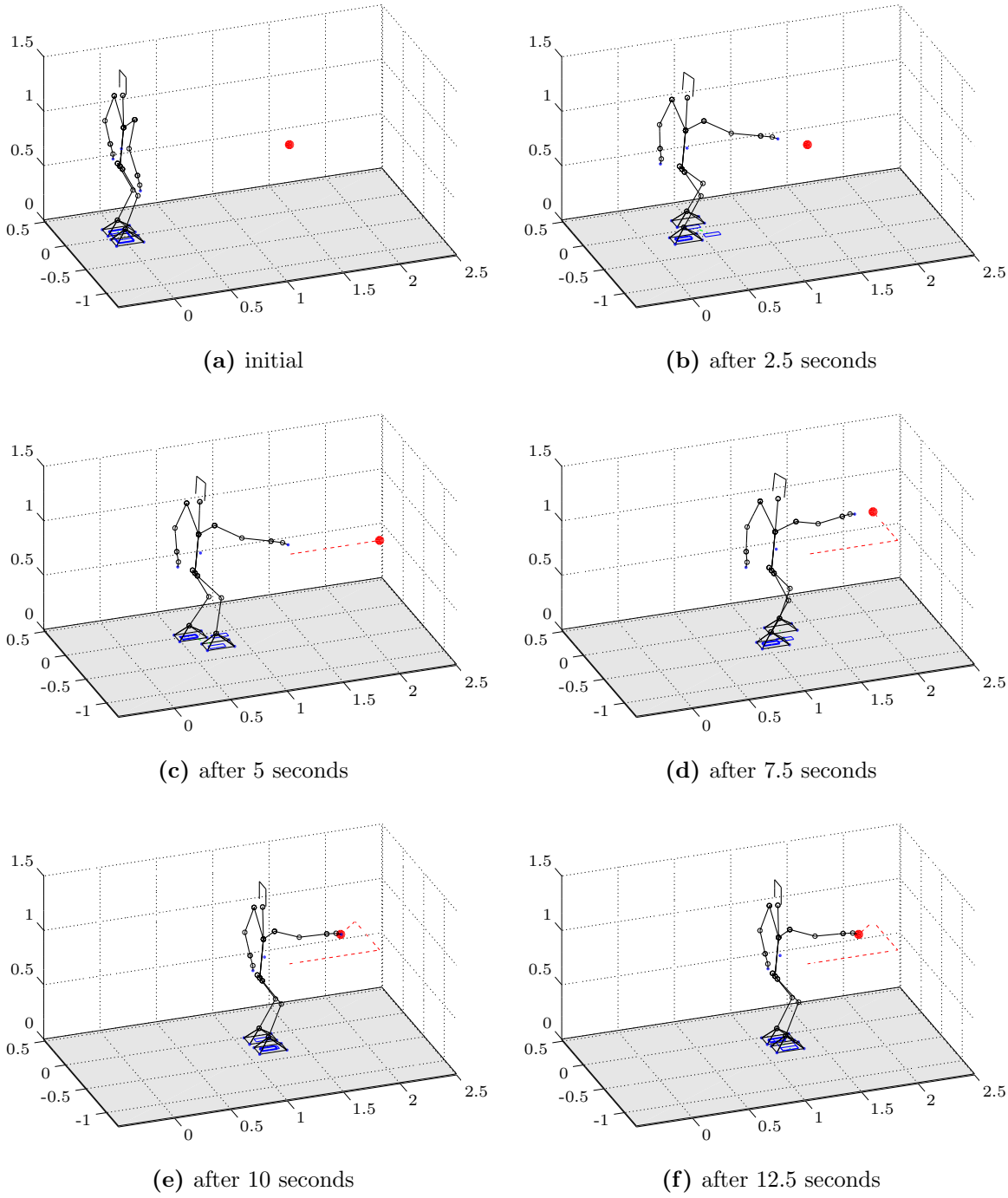


(a) side view **(b)** front view

Figure 6.1: `HRP-2` robot.

Unless stated otherwise, we control and simulate an `HRP-2` robot depicted in Figure 6.1 [Kaneko 2004]. In these cases we assume a perfect model and perfect inertial measurement unit. The robot has 30 actuated joints, its total weight is around 57 $[kg]$, and the height is around 1.5 $[m]$. The control sampling interval is chosen to be 5 $[ms]$ (200 $[Hz]$).

6.1 Task-driven walking



(a) initial **(b)** after 2.5 seconds

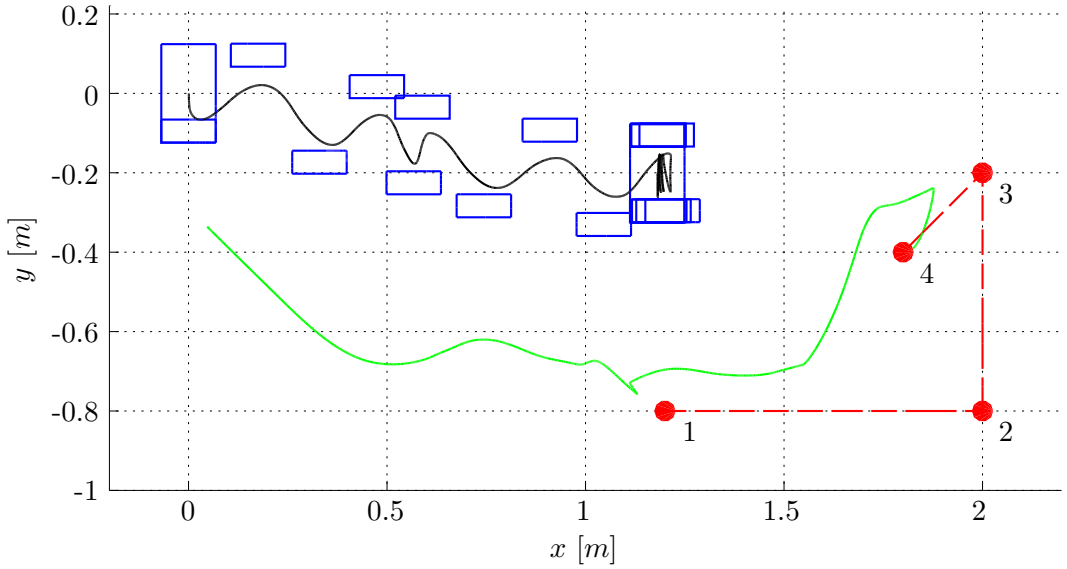
(c) after 5 seconds **(d)** after 7.5 seconds

(e) after 10 seconds **(f)** after 12.5 seconds

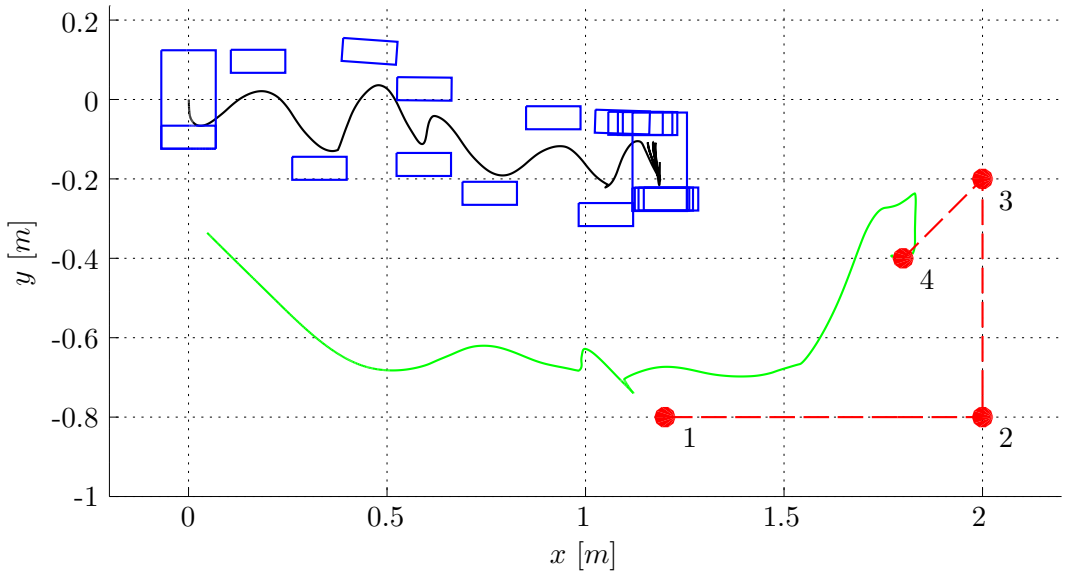
Figure 6.2: Task-driven walk: configurations of the robot during the simulation. Red point indicates position of the target, blue rectangles – current and anticipated positions of the feet.

It is typical to express high level goals of humanoid robot control using whole body tasks, execution of which may require locomotion. For example, it might be necessary to approach an object before grasping it. In such situations it is necessary to anticipate walking motions taking the whole body tasks into account. This, however, may not be straightforward when anticipation with an approximate model is performed, due to simplifications made during the construction of the model (see Section 4.5). For example, in the case of point-mass models it is common to map whole body tasks to motions of the *Center of Mass* (CoM) or feet

[Yoshida 2006, Nishiwaki 2003, Fukumoto 2004, Herdt 2010]. Such mappings, however, are often task- and model- specific and their development requires human involvement. *Mixed Model Predictive Control* (MMPC) addresses this problem by mixing approximate and whole body models to allow their automatic interaction without any additional mapping procedures. It also allows to account for multiple tasks simultaneously in a straightforward way.



(a) without disturbances



(b) with disturbances

Figure 6.3: Task-driven walk: top view. Footsteps are represented by rectangles, trajectory of the CoM is in black, trajectory of the hand is in green, while the trajectory of the target is in dashed red. Numbers indicate ordering of the target positions.

The interplay between the models in MMPC was shown in [Sherikov 2014], where a whole body task induces walking motion. The task is to reach a varying target point with the right hand of the robot (see Figures 6.2 and 6.3). We also demonstrated that this controller performs automatic repositioning of the feet in order to compensate for disturbances applied to the robot. We describe this controller and simulation setting in Subsections 6.1.1 and 6.1.2. The obtained results are discussed in Subsection 6.1.3.

6.1.1 Setting

We evaluate the proposed controller in a simulation, which lasts for approximately 15 second. During this time the robot has to reach a target with its right hand. Initially, the target cannot be reached by the robot while standing. During the simulation the target is repositioned at 4.6, 6.25, and 8 second in an unpredictable way (see Figure 6.3). In order to further complicate the task for the controller, we apply disturbances to the robot at 2.5 second from the right ($\rho_d^y = 15\ [Ns]$) and at 7 second from the front ($\rho_d^x = -15\ [Ns]$). The disturbances are simulated as described in Appendix B.

Thus, the goal of the controller is to automatically choose positions of the feet on the ground in order to reach the target and compensate for disturbances.

6.1.2 Design of the controller

We construct a hierarchy corresponding to MMPC controller based on the whole body (WB) and point-mass (PPMZ) models as proposed in Subsection 5.3.4. In order to reduce the size of the PLLS problem, we condense PPMZ and eliminate torques from WB in advance (see Subsection 5.4.4). This results in the following hierarchy:

Hierarchy (6.1)

1: Simple bounds

- $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$ — 30 joint limits
- $\boldsymbol{\lambda}_i \geq \boldsymbol{0}$ — $3M$ constraints due to friction cones
- $\underline{\boldsymbol{b}}_{\boldsymbol{p},j} \leq \Delta\hat{\boldsymbol{p}}_j \leq \bar{\boldsymbol{b}}_{\boldsymbol{p},j}$ — $2K$ constraints on foot positions
- $\underline{\boldsymbol{z}} \leq \hat{\boldsymbol{z}}_{k+1} \leq \bar{\boldsymbol{z}}$ — $2N$ constraints on the CoP positions

2: Tasks of the whole body motion controller

- $\begin{bmatrix}\boldsymbol{H}_2\\ \boldsymbol{H}_3\end{bmatrix}\ddot{\boldsymbol{q}} + \begin{bmatrix}\boldsymbol{h}_2\\ \boldsymbol{h}_3\end{bmatrix} = m\begin{bmatrix}\boldsymbol{J}_{com,2}^\top\\ \boldsymbol{J}_{com,3}^\top\end{bmatrix}\boldsymbol{g} + \sum_{i=1}^{M}\begin{bmatrix}\boldsymbol{J}_{i,2}^\top\\ \boldsymbol{J}_{i,3}^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_i\boldsymbol{\lambda}_i\\ \boldsymbol{\mu}_i\end{bmatrix}$ — 6 equalities due to Newton-Euler equations
- $\underline{\boldsymbol{\tau}} \leq \boldsymbol{H}_1\ddot{\boldsymbol{q}} + \boldsymbol{h}_1 - m\boldsymbol{J}_{com,1}^\top\boldsymbol{g} - \sum_{i=1}^{M}\boldsymbol{J}_{i,1}^\top\begin{bmatrix}\boldsymbol{V}_i\boldsymbol{\lambda}_i\\ \boldsymbol{\mu}_i\end{bmatrix} \leq \bar{\boldsymbol{\tau}}$ — 30 bounds on the joint torques
- $\boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} = \boldsymbol{0}$ — $6M$ equalities due to fixed contacts
- $\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},i}\begin{bmatrix}\boldsymbol{\lambda}_i\\ \boldsymbol{\mu}_i\end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},i}$ — $6M$ constraints on the contact moments
- $\boldsymbol{I}_z\left(\boldsymbol{J}_{com}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{com}\dot{\boldsymbol{q}}\right) = \pi_{cz}$ — 1 equality to maintain the CoM height

Coupling with PPMZ model

- $\boldsymbol{I}_{xy}\left(\boldsymbol{J}_{com}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{com}\dot{\boldsymbol{q}}\right) = \ddot{\boldsymbol{c}}_0^{xy}$ — 2 equalities due to CoM motion
- $\boldsymbol{J}_{\uparrow,s}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,s}\dot{\boldsymbol{q}} = \ddot{\boldsymbol{s}}_0$ — $3(2-M)$ equalities due to foot motion

3: Capturability constraint (4.24)

- $\dot{\boldsymbol{c}}_N^{xy} + \sqrt{\zeta}\ddot{\boldsymbol{c}}_N^{xy} = \boldsymbol{0}$ — 2 equalities

4: Orientations

- $\boldsymbol{J}_{\circlearrowleft,t}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\circlearrowleft,t}\dot{\boldsymbol{q}} = \boldsymbol{\pi}_t$ — 3 equalities to maintain torso orientation
- $\boldsymbol{J}_{\circlearrowleft,s}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\circlearrowleft,s}\dot{\boldsymbol{q}} = \boldsymbol{\pi}_s$ — $3(2-M)$ equalities to maintain foot orientation

5: Whole body tasks

- $\boldsymbol{J}_{\uparrow,rh}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,rh}\dot{\boldsymbol{q}} = \boldsymbol{\pi}_{rh}$ — 3 equalities due to the right hand task
- $\ddot{\boldsymbol{q}}' = \mathcal{K}_p(\boldsymbol{q}'^{\heartsuit} - \boldsymbol{q}') - \mathcal{K}_d\dot{\boldsymbol{q}}'$ — 30 equalities to control the joints

Anticipation tasks

- $\dddot{\boldsymbol{s}}^{xy} = \boldsymbol{0}$ — $2(2-M)$ equalities to minimize foot jerk
- $\hat{\boldsymbol{z}}_{k+1} = \boldsymbol{0}$ — $2N$ equalities to center CoP positions
- $\dot{\boldsymbol{z}}_k = \boldsymbol{0}$ — $2N$ equalities to minimize CoP velocities

Decision variables: $\boldsymbol{\chi} = (\ddot{\boldsymbol{q}}, \boldsymbol{\lambda}_i, \boldsymbol{\mu}_i, \ddot{\boldsymbol{c}}_0^{xy}, \hat{\boldsymbol{z}}_{k+1}, \Delta\hat{\boldsymbol{p}}_j)$
with $i \in \{1, ..., M\}$, $k \in \{0, ..., N-1\}$, $j \in \{0, ..., K\}$

where $M \in \{1, 2\}$ is the number of foot contacts, K is the number of varying footstep positions in the preview horizon, N is the length of the preview horizon. Notation in the hierarchy is the same as in the preceding chapters with a few additions:

$\boldsymbol{J}_s = (\boldsymbol{J}_{\uparrow,s}, \boldsymbol{J}_{\circlearrowleft,s})$	Jacobian of the foot in the air,
$\boldsymbol{J}_{\circlearrowleft,t}$	rotational Jacobian of the torso,
$\boldsymbol{J}_{\uparrow,rh}$	translational Jacobian of the right hand,
π_{cz}, $\boldsymbol{\pi}_t$, $\boldsymbol{\pi}_s$, $\boldsymbol{\pi}_{rh}$	appropriately defined PD-controllers,
$\boldsymbol{q}'^{\heartsuit}$	desired joint angles.

The decision variables are

- the generalized accelerations $\ddot{\boldsymbol{q}}$,
- contact forces represented with $\boldsymbol{\lambda}_i$ as described in Subsection 3.1.3,
- contact moments $\boldsymbol{\mu}_i$,
- current CoM acceleration $\ddot{\boldsymbol{c}}_0^{xy}$ (the reason for this is given in Subsection 4.5.1),
- anticipated CoP positions $\hat{\boldsymbol{z}}_{k+1}$ expressed in frames fixed to the feet (see Section 4.4),
- distances $\Delta\hat{\boldsymbol{p}}_j$ between the j-th and $j+1$ steps in the preview horizon (see Section 4.4).

Current acceleration and jerk of the foot in the air $\ddot{\boldsymbol{s}}_0$, $\dddot{\boldsymbol{s}}^{xy}$; anticipated velocities of the CoP $\dot{\boldsymbol{z}}_k$; and parts of the final state of the approximate model ($\dot{\boldsymbol{c}}_N^{xy}$, $\ddot{\boldsymbol{c}}_N^{xy}$) are kept in the hierarchy to simplify presentation. In the actual controller they are expressed using variables in $\boldsymbol{\chi}$ as explained in Subsection 4.4.2 and Appendix E.

The considered hierarchy contains two more levels compared to the abstract hierarchy proposed in Subsection 5.3.4. The simple bounds are collected on a separate level since it is necessary for `LexLS` to be able to exploit their structure. Also, we have chosen to prioritize the tasks controlling orientations of the torso and foot in the air over the tasks of the last level. Otherwise, when disturbances are applied, the controller may not be able to restore correct orientation of the foot before it touches the ground.

Tasks on the final 5-th level of the hierarchy are incompatible and are weighted with respect to each other. Some of the choices of the weights are discussed later in this section. All PD-controllers used in the hierarchy are critically damped, their gains are task specific. $\mathcal{K}_p$ gain in the joint level PD-controller is set to zero for the joints of the legs and right hand.

Approximate PPMZ model cannot automatically choose durations of steps. For this reason, the duration is fixed for all steps to 0.8 $[s]$, which includes transitional double support of 0.1 $[s]$ as in [Herdt 2010]. In accordance with the same paper, sequence of steps is produced by a simple state machine, and the preview horizon is sampled using $T_k = 0.1$ $[s]$.

6.1.3 Results and discussion

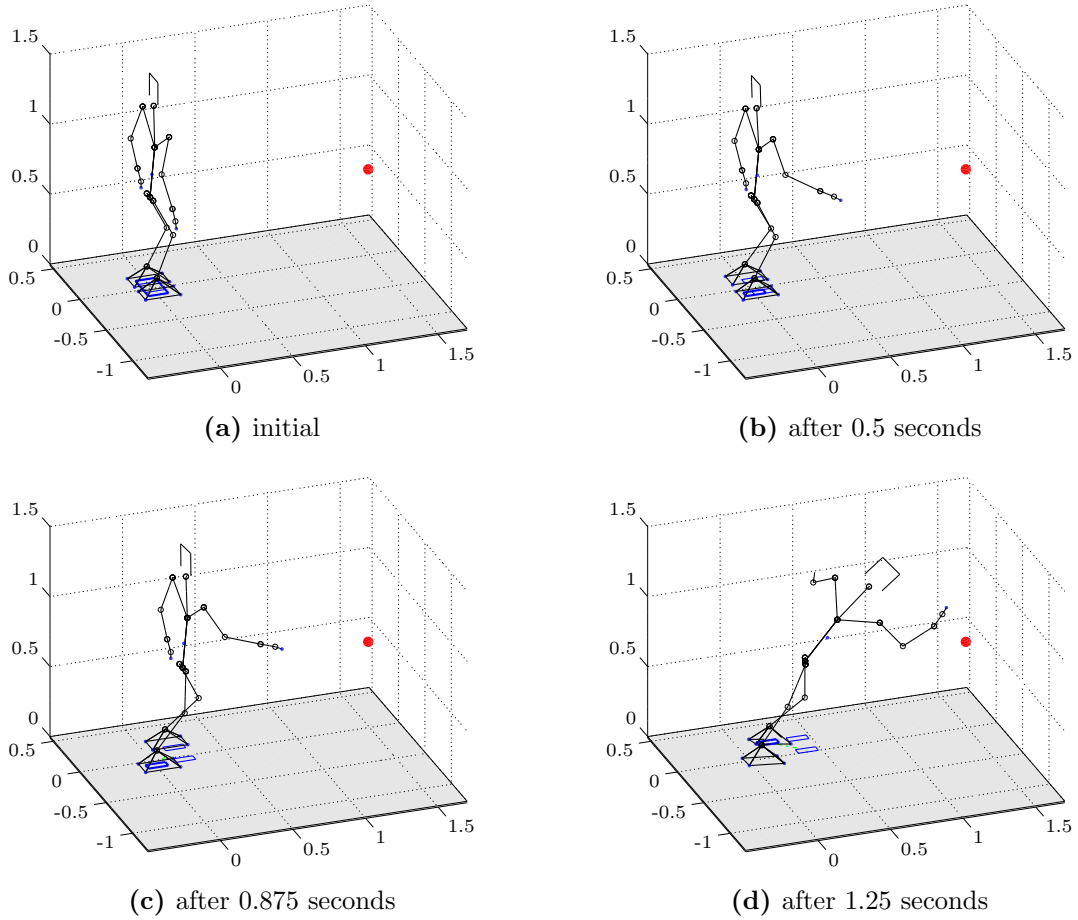
Main results were obtained using Hierarchy 6.1 with $N = 16$, which implies that $K \in \{1, 2\}$. In other words the controller anticipates for approximately two steps into the future.

Figures 6.3, 6.5 and 6.6 illustrate the ability of the basic version of the controller to

automatically position feet of the robot in order to execute whole body tasks and compensate for disturbances. In the beginning of the simulation the robot starts walking since the target is unreachable, and continues to walk until the target is reached, around 4 second. However, due to a change in the x position of the target, the walk is resumed. Lateral motion of the target influences the walk in the same way. Moreover, we can see that the controller immediately reacts to disturbances and successfully compensates for them by adjusting footsteps in mid-air (Figures 6.7 and 6.8). Once the target is reached the robot continues to walk in place.

In the following subsections we discuss computational performance of `LexLS`, behavior of the controller with a few minor modifications, and the quality of the generated motions.

6.1.3.1 Capturability constraint



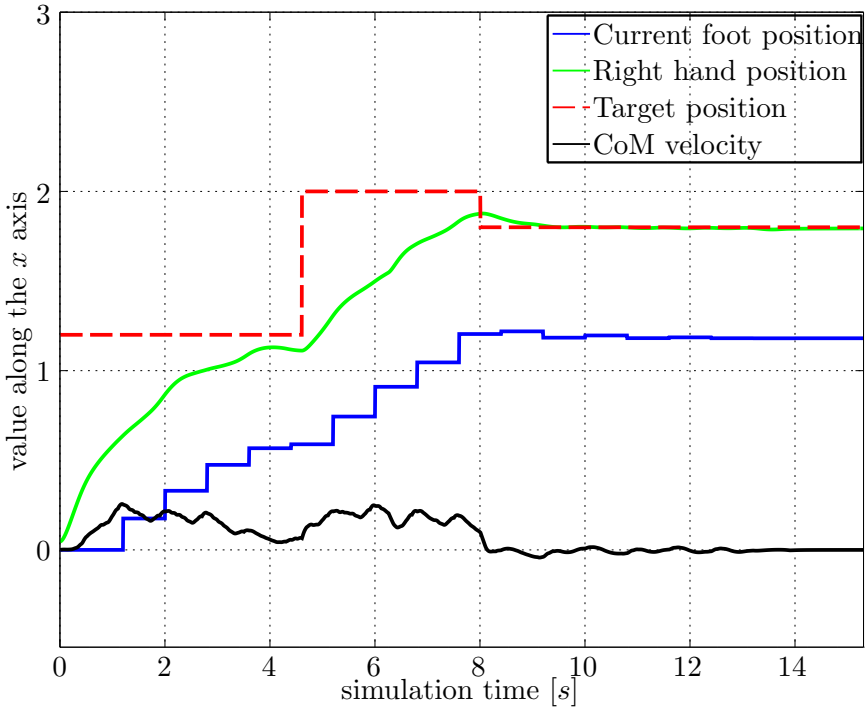
(a) initial **(b)** after 0.5 seconds

(c) after 0.875 seconds **(d)** after 1.25 seconds

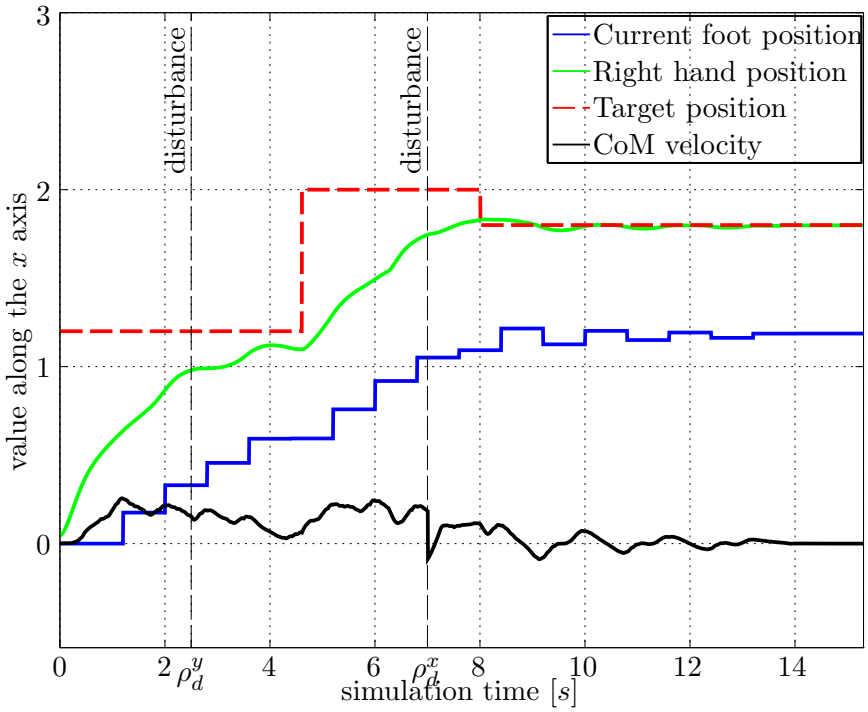
Figure 6.4: A fall due to removal of the capturability constraint.

Balanced walking motions can be obtained without a capturability constraint provided that the weights of the objectives are properly tuned [Wieber 2008, Herdt 2010]. However, addition of such constraint makes controller less sensitive to the weights. For example, the considered MMPC controller makes the robot fall in the very beginning of the simulation, when the capturability constraint is omitted (see Figure 6.4). Though, it is possible to adjust objectives on the last level of the hierarchy and their weights to avoid this, it is unnecessary due to the capturability constraint.

Satisfaction of the capturability constraint, however, does not guarantee that the balance is always preserved. We observed that introduction of an additional level in the hierarchy in order to prioritize the hand task over other tasks of the last level leads to violent motions of the upper body and, eventually, to a fall. We believe that the reason for this is that the

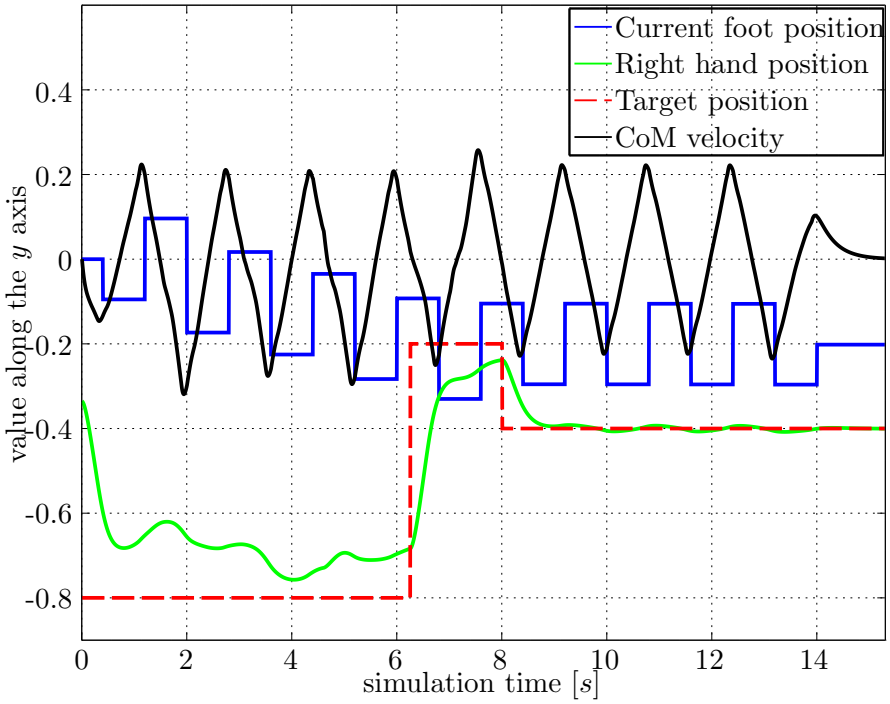


(a) $N = 16$, without disturbances

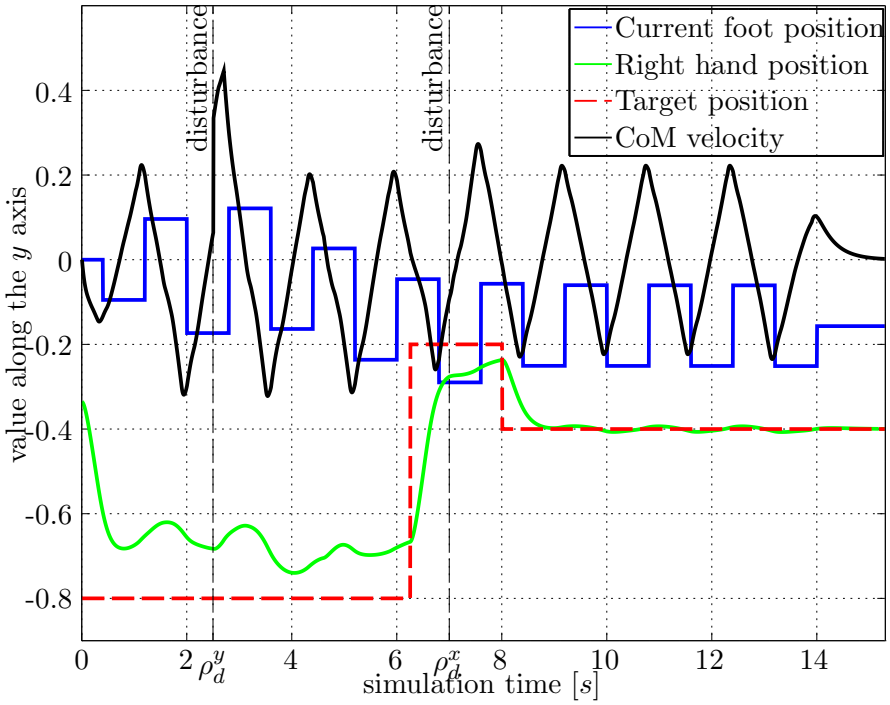


(b) $N = 16$, with disturbances

Figure 6.5: Evolution of the x components of the target, hand, and current support positions and CoM velocity with time. The time instants, when disturbances are applied, are indicated with vertical dashed black lines.

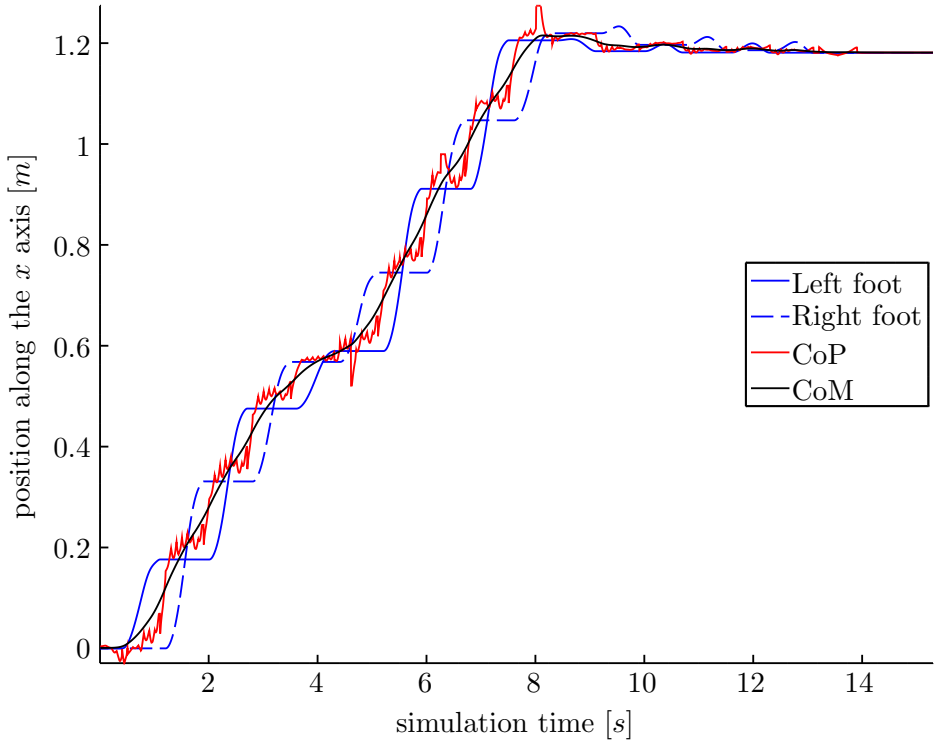


(a) $N = 16$, without disturbances

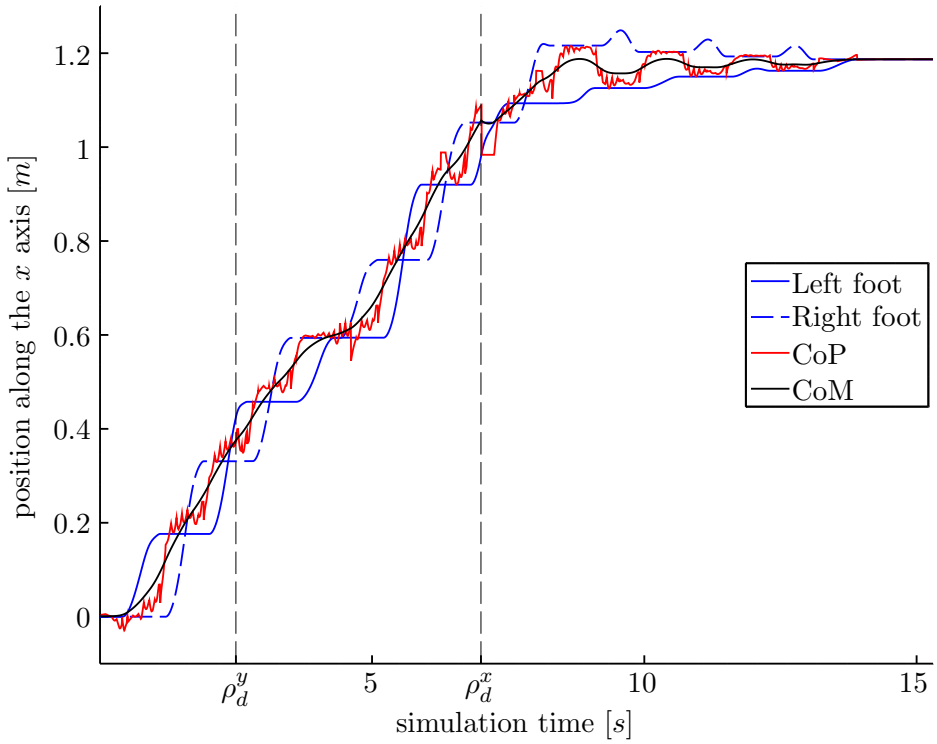


(b) $N = 16$, with disturbances

Figure 6.6: Evolution of the y components of the target, hand, and current support positions and CoM velocity with time. The time instants, when disturbances are applied, are indicated with vertical dashed black lines.

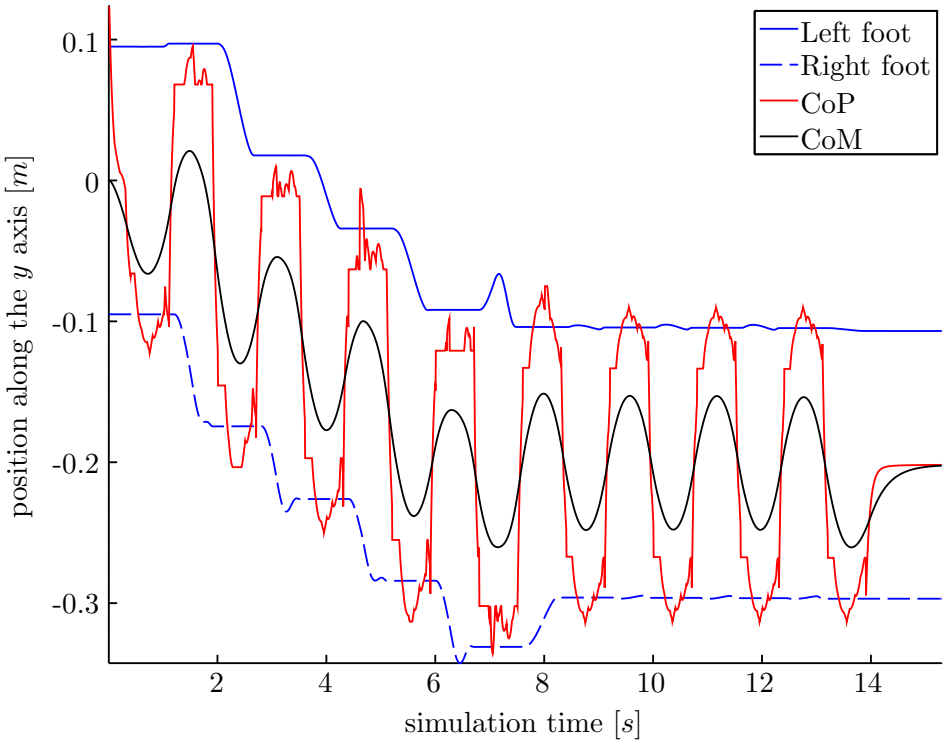


(a) $N = 16$, without disturbances

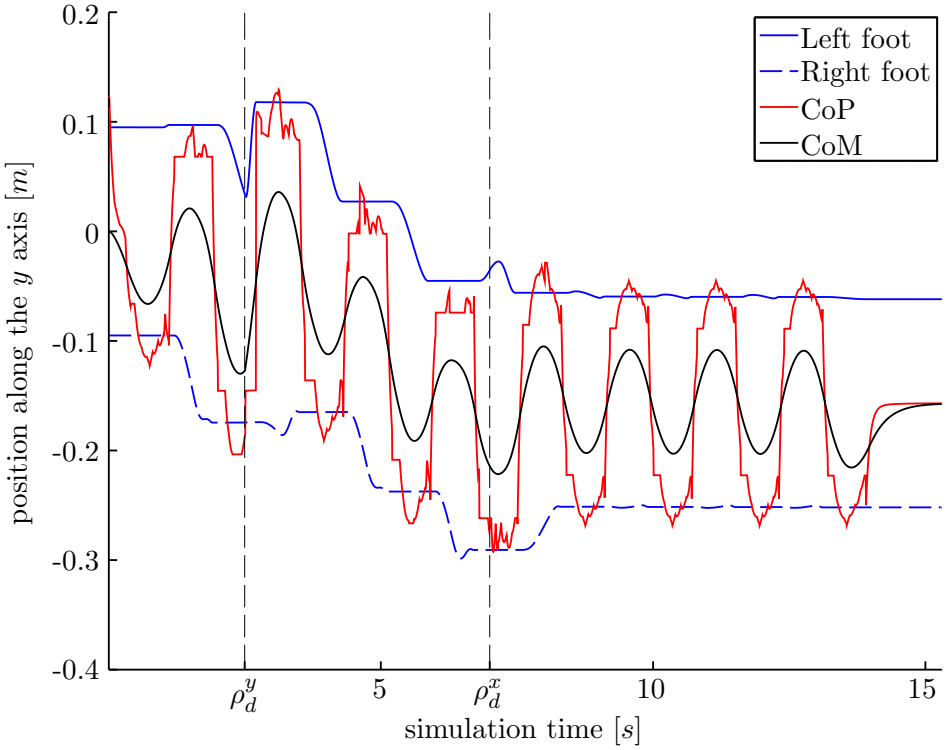


(b) $N = 16$, with disturbances

Figure 6.7: Evolution of the positions of feet, CoM, and CoP with time along the x axis. The time instants, when disturbances are applied, are indicated with vertical dashed black lines.



(a) $N = 16$, without disturbances

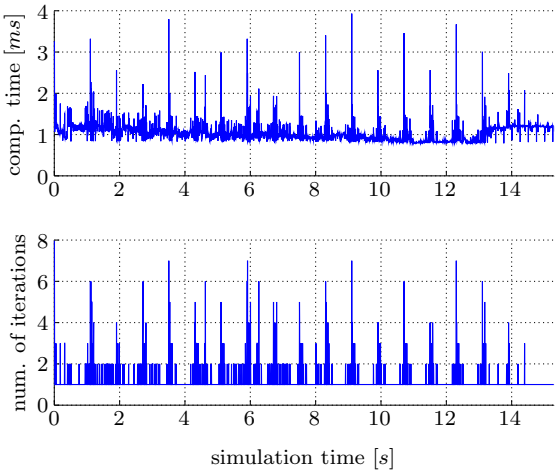


(b) $N = 16$, with disturbances

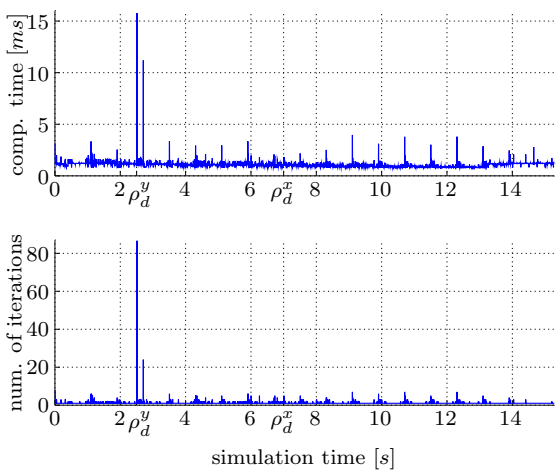
Figure 6.8: Evolution of the positions of feet, CoM, and CoP with time along the y axis. The time instants, when disturbances are applied, are indicated with vertical dashed black lines.

point-mass approximation does not reflect the complex dynamics of the robot to a necessary extent. Hence, approximate models including angular momentum may be more appropriate for the considered setting.

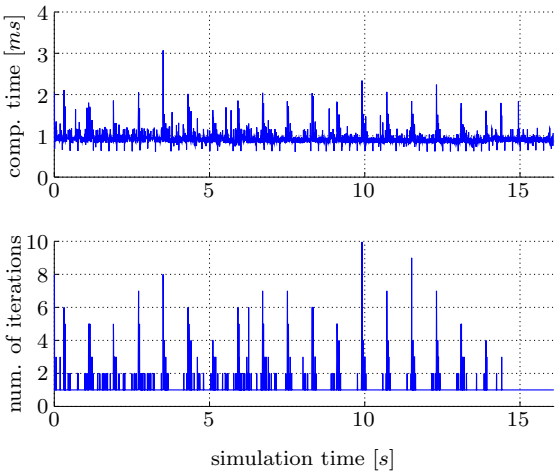
6.1.3.2 Computational performance of `LexLS`



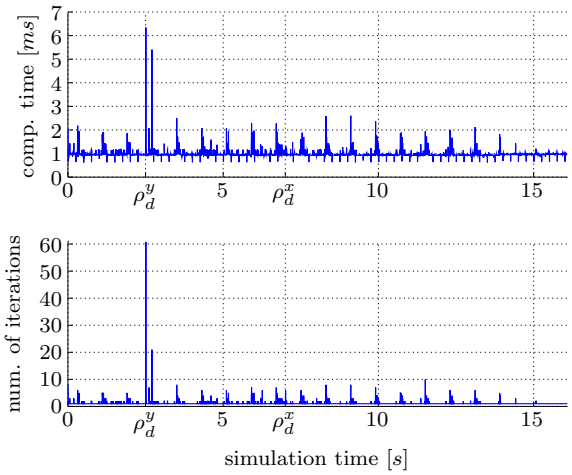
(a) $N = 16$, without disturbances.
Mean computational time: 1.04 $[ms]$.
Time measurements > 1 $[ms]$: 49%



(b) $N = 16$, with disturbances.
Mean computational time: 1.09 $[ms]$.
Time measurements > 1 $[ms]$: 62%



(c) $N = 8$, without disturbances.
Mean computational time: 9.4 $[ms]$.
Time measurements > 1 $[ms]$: 11%



(d) $N = 8$, with disturbances.
Mean computational time: 9.8 $[ms]$.
Time measurements > 1 $[ms]$: 13%

Figure 6.9: Computation time and number of iterations of `LexLS`. Time instants, when disturbances are applied, are indicated with ρ_d^y and ρ_d^x.

Hierarchy 6.1 is supposed to be solved in order of milliseconds to control a robot in real time. This is a challenging problem, since the hierarchy has around 85 decision variables and includes more than 100 inequality and 120 equality constraints. In order to demonstrate that this is possible we measured the time required for `LexLS` to solve this PLLS problem (see Figures 6.9a and 6.9b). The time measurement at each control instant is averaged over three simulation runs to suppress outliers. All measurements were performed on a laptop with Intel Core i5-3360M (2.80 $[GHz]$) CPU.

When disturbances are not applied, the time required to solve the hierarchy is less than

5 [ms], the number of iterations of the solver does not exceed 8. However, disturbances lead to increase in the number of iterations of the solver, which, in turn, leads to significant increase in the computational time. In order to alleviate this issue it might be necessary to employ early termination of the solver (see Subsection 5.4.2). It is, however, important to note, that the current implementation of `LexLS` adds and removes constraints from the active set in an inefficient way [Dimitrov 2015]. Hence, further development of the solver is expected to give a performance boost for the controller. In an attempt to reduce the computational time, we also tried to shorten the preview horizon from $N = 16$ to $N = 8$ sampling intervals. This modification reduces the number of decision variables by 16-18, the number of equality and inequality constraints by 32 and 16-18 respectively. The problem with shorter preview horizon can be solved slightly faster on average and 2 times faster, when disturbance is applied (see Figures 6.9c and 6.9d). One can also observe that in almost 90% of the cases the problem is solved faster than 1 [ms]. At the same time, we did not observe qualitative changes in behavior of the robot.

6.1.3.3 Quality of the motion

Although, the controller produces the desired behavior, the generated motion is not completely satisfactory:

- Figures 6.5 and 6.6 demonstrate that the right hand position oscillates near the target in the end of the simulation due to the sway motion of the robot. This may be caused by several reasons: (**i**) compromise between satisfaction of the hand task and other tasks on the last level of the hierarchy, (**ii**) lack of anticipation for the hand position and respective kinematic constraints.
- We can see in Figure 6.10 (as well as in Figures 6.7 and 6.8) periodic variations of the CoP position with period of 0.1 [s] caused by the discrepancy between the control interval of 0.005 [s] and preview sampling interval of 0.1 [s]. This issue was discussed in Section 4.6.
- The controller is designed in such a way, that it always trades off between two strategies for reaching the target: moving the hand and walking. Consequently, near singularities of the elbow, the controller prefers walking to bending the arm, which is undesirable in some situations.
- We tuned the weights on the last level of the hierarchy so that the CoP centering task dominates all others. The reason for this is that the hand task "pulls" the CoP to the front edges of the support areas, which, to some extend, corresponds to walking on tiptoes. This negatively impacts controller's ability to cope with disturbances and, therefore, is potentially unsafe [Lafaye 2014]. Moreover, it leads to a larger number of active inequality constraints and larger number of iterations of the solver.

6.1.4 Conclusion

We demonstrated that MMPC allows to account for the whole body tasks while generating walking motions without relying on time-demanding planning and nonlinear optimization procedures [Escande 2009, Kanoun 2010, Tassa 2014]. The major limitation of the approach is the fact that durations and sequence of the steps must still be decided outside of the controller. There is also a number of technical difficulties discussed in Subsection 6.1.3.3, which should be addressed in the future works.

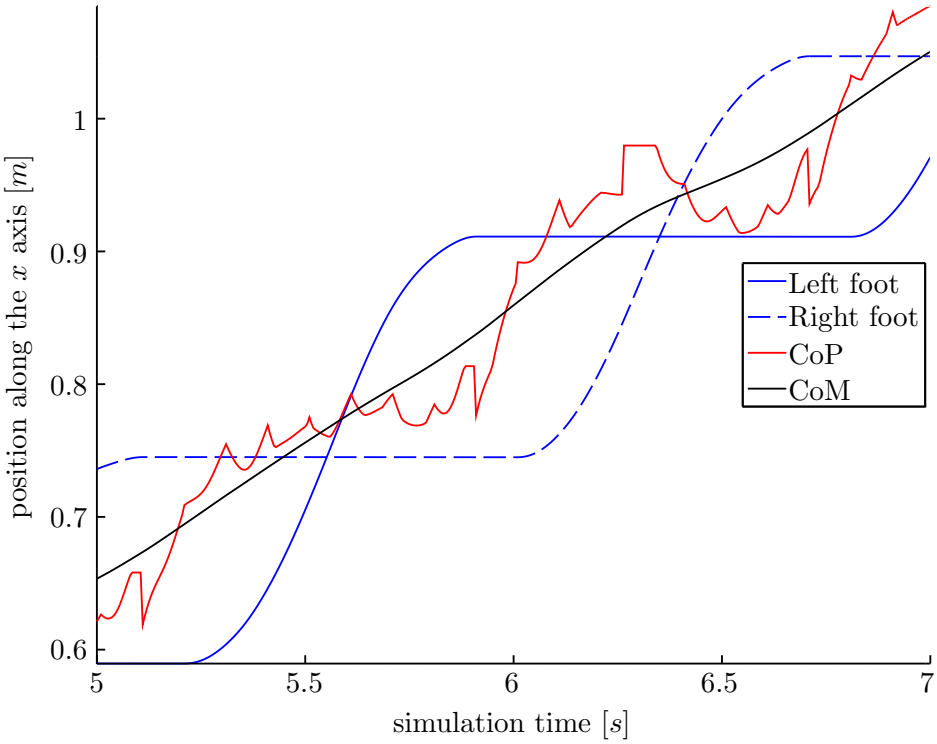


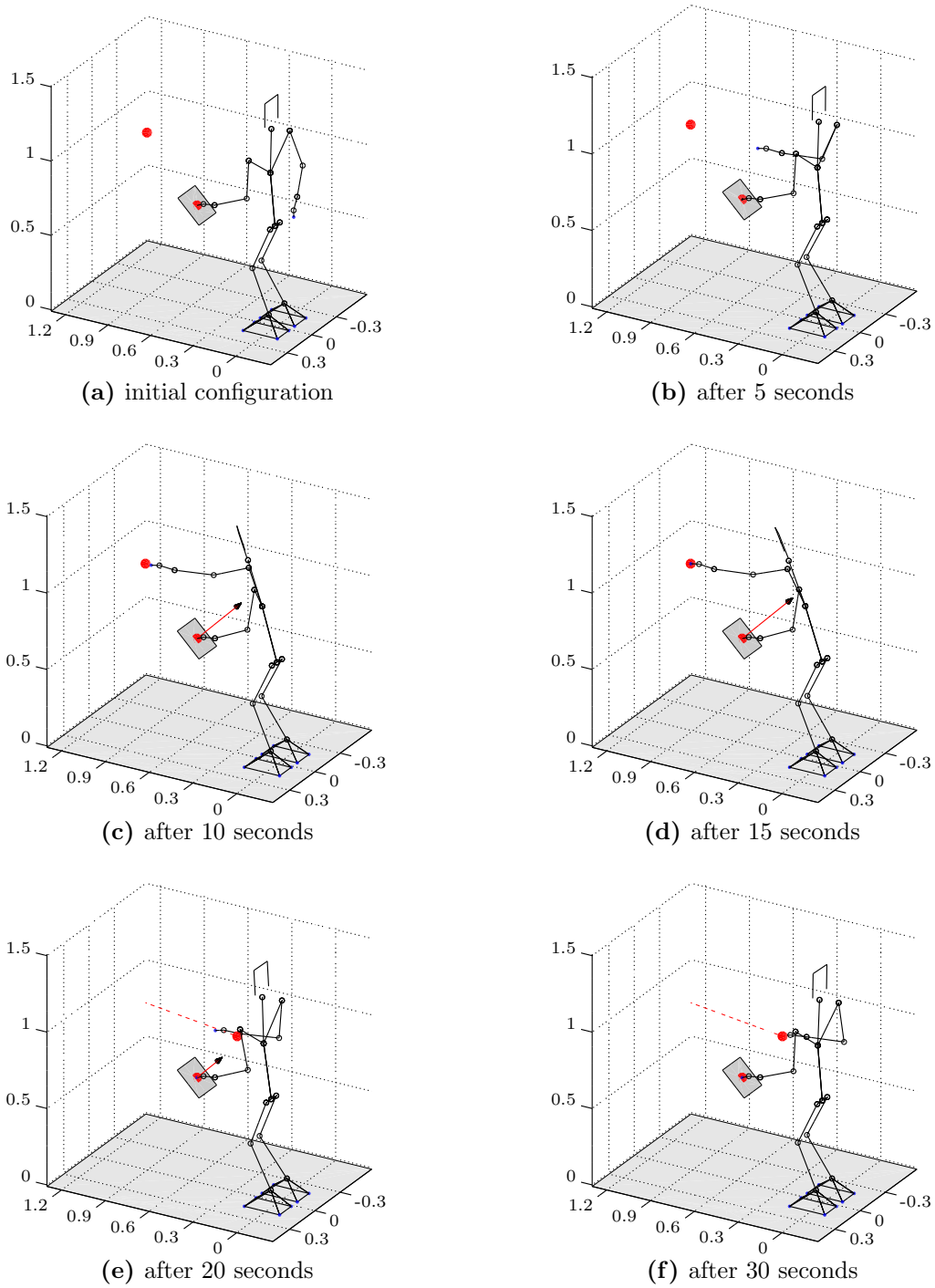
Figure 6.10: Magnified part of Figure 6.7a: evolution of the feet, CoM, and CoP positions with time along the x axis. Periodic variations of the CoP position with period 0.1 $[s]$ can be clearly seen.

6.2 Prioritization in the contact force distribution

We continued to develop the idea of MMPC in [Sherikov 2015], where we proposed a controller for balancing in a multicontact setting with prioritized contact force distribution. In most settings with multiple contacts there exists an infinite number of force distributions that achieve the same base motion. The typical approach to resolve this ambiguity is to make contacts as robust as possible, by keeping each contact force far from the bounds of the respective friction cone, and distribute the forces evenly between all the contacts [Saab 2013, Ott 2011, Herzog 2015, Hyon 2007]. There are situations, however, such as when a contact area is fragile, when it is preferable to avoid using it unless strictly necessary for balance. In this case, distributing forces evenly between all possible contacts should be avoided. We propose therefore to introduce a prioritized distribution of the contact forces, with the help of hierarchical optimization [Saab 2013, Escande 2014, Kanoun 2011]. We demonstrate our idea in a setting, where a humanoid robot can optionally exploit a hand contact with an additional support to maintain balance and execute certain task with the free hand.

6.2.1 Setting

We use the following setting: the robot is standing with its left hand positioned on an additional support, while the right hand executes certain task. Hence, the number of contacts $M = 3$ (two feet and the hand) is constant during the simulations. We define two different hand tasks. In the first case the robot has to reach a target, which is initially unreachable without using the additional support and later moved closer to the robot (see Figures 6.11 and 6.12). The second task is to maintain position of the right hand while an external



(a) initial configuration **(b)** after 5 seconds

(c) after 10 seconds **(d)** after 15 seconds

(e) after 20 seconds **(f)** after 30 seconds

Figure 6.11: Configurations of the robot while it is trying to reach a target indicated by the red point. Grey areas represent contact surfaces. Length of the arrow indicates magnitude of the contact force applied by the left hand.

disturbing force is acting on it (see Figure 6.13). In other words, the robot holds a heavy object, such as a filled bucket [Stephens 2010]. The external force is varying with time and is assumed to be measured. In all tests the controller is expected to apply a force on the additional support only if necessary to preserve balance and execute the hand task.

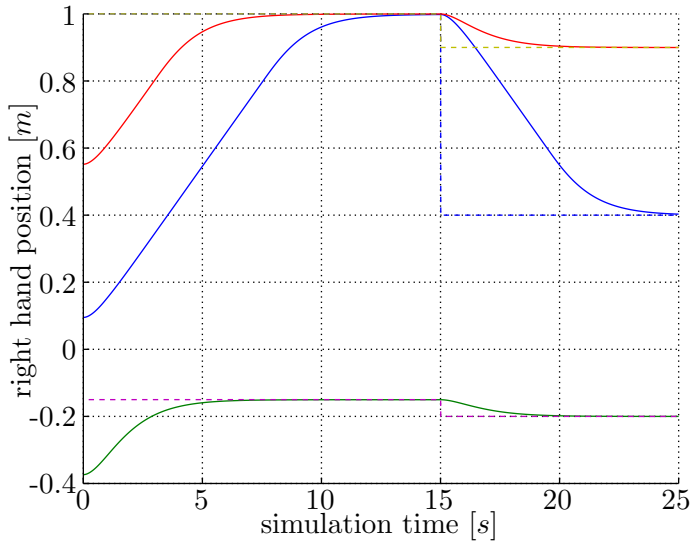
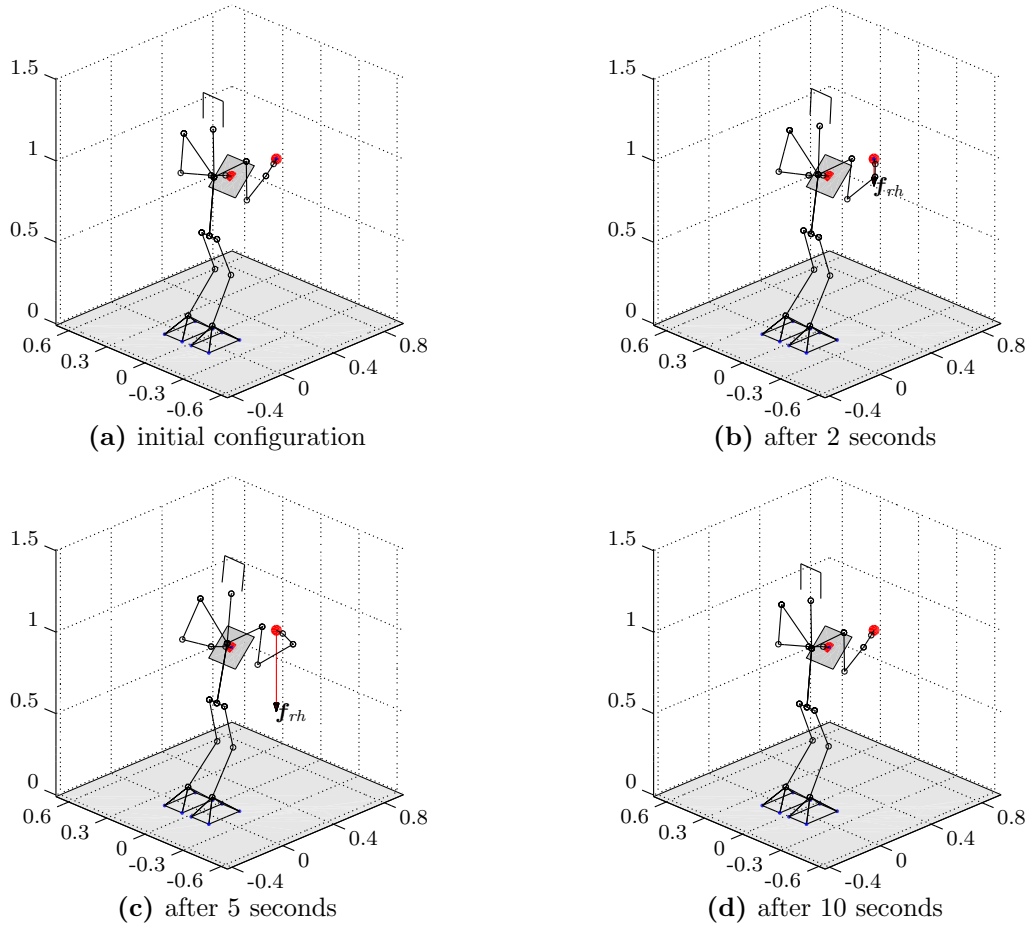


Figure 6.12: Position of the right hand during execution of the reaching task. x, y, and z coordinates are shown in solid blue, green, and red respectively. The desired positions are shown in dashed lines.



(a) initial configuration

(b) after 2 seconds

(c) after 5 seconds

(d) after 10 seconds

Figure 6.13: Configurations of the robot in presence of varying disturbing force $\boldsymbol{f}_{rh}$. Grey areas represent contact surfaces. Length of the arrow indicates magnitude of the external force.

6.2.2 Design of the controller

The point-mass model exploited in Hierarchy 6.1 is not suitable for multicontact scenarios. For this reason, we adopt the momenta-based (MB) model and couple it with the whole body (WB) model through the current contact forces. In order to achieve high computational performance we condense MB and eliminate torques from WB in advance (see Subsection 5.4.4). Once this is done, the controller is formulated as follows

Hierarchy (6.2)

1: Simple bounds

- $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$ — 30 joint limits
- $\boldsymbol{\lambda}_{k,i} \geq \boldsymbol{0}$ — $3NM$ constraints due to friction cones

2: Whole body tasks and coupling

- $\begin{bmatrix}\boldsymbol{H}_2\\ \boldsymbol{H}_3\end{bmatrix}\ddot{\boldsymbol{q}} + \begin{bmatrix}\boldsymbol{h}_2\\ \boldsymbol{h}_3\end{bmatrix} = m\begin{bmatrix}\boldsymbol{J}_{com,2}^\top\\ \boldsymbol{J}_{com,3}^\top\end{bmatrix}\boldsymbol{g} + \begin{bmatrix}\boldsymbol{J}_{\uparrow,lh,2}^\top\\ \boldsymbol{J}_{\uparrow,lh,3}^\top\end{bmatrix}\boldsymbol{V}_{0,lh}\boldsymbol{\lambda}_{0,lh} + \sum_{i=1}^{M-1}\begin{bmatrix}\boldsymbol{J}_{i,2}^\top\\ \boldsymbol{J}_{i,3}^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{0,i}\boldsymbol{\lambda}_{0,i}\\ \boldsymbol{\mu}_{0,i}\end{bmatrix}$
  6 equalities due to Newton-Euler equations
- $\underline{\boldsymbol{\tau}} \leq \boldsymbol{H}_1\ddot{\boldsymbol{q}} + \boldsymbol{h}_1 - m\boldsymbol{J}_{com,1}^\top\boldsymbol{g} - \boldsymbol{J}_{\uparrow,lh,1}^\top\boldsymbol{f}_{lh,0} - \sum_{i=1}^{M-1}\boldsymbol{J}_{i,1}^\top\begin{bmatrix}\boldsymbol{V}_{0,i}\boldsymbol{\lambda}_{0,i}\\ \boldsymbol{\mu}_{0,i}\end{bmatrix} \leq \bar{\boldsymbol{\tau}}$
  30 bounds on the joint torques
- $\boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} = \boldsymbol{0}$ — $6M$ equalities due to fixed contacts

Anticipation tasks

- $\sum_{i=1}^{M} f_{k,i}^z = -mg^z$ — N equalities due to fixed CoM height
- $\boldsymbol{\mathcal{A}}_{\boldsymbol{\mu},k,i}\begin{bmatrix}\boldsymbol{\lambda}_{k,i}\\ \boldsymbol{\mu}_{k,i}\end{bmatrix} \geq \underline{\boldsymbol{b}}_{\boldsymbol{\mu},k,i}$ — $6(M-1)$ constraints on the moments

3: Capturability constraint (4.18)

- $\begin{matrix}\boldsymbol{\mathcal{P}}_N^{xy} = \boldsymbol{0}, & {}^{\mathrm{c}}\boldsymbol{\mathcal{L}}_N^{xy} = \boldsymbol{0},\\ \dot{\boldsymbol{\mathcal{P}}}_N^{xy} = \boldsymbol{0}, & {}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}}_N = \boldsymbol{0}\end{matrix}$ — 9 equalities

4: Right hand task

- $\boldsymbol{J}_{\uparrow,rh}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,rh}\dot{\boldsymbol{q}} = \boldsymbol{\pi}_{rh}$ — 3 equalities

5: Minimization of the normal left hand contact force

- $f_{k,lh}^n = 0$ — N equalities

6: Whole body tasks

- $\boldsymbol{L}\ddot{\boldsymbol{q}}' = \boldsymbol{L}\big(\mathcal{K}_p(\boldsymbol{q}'^{\heartsuit} - \boldsymbol{q}') - \mathcal{K}_d\dot{\boldsymbol{q}}'\big)$ — 30 equalities to control the joints

Anticipation tasks

- $\boldsymbol{\lambda}_{k,i} = \boldsymbol{0}$ — $3NM$ equalities to minimize forces
- $\boldsymbol{\mu}_{k,\{1,\dots,M-1\}} = \boldsymbol{0}$ — $3N(M-1)$ equalities to minimize moments

Decision variables: $\boldsymbol{\chi} = \big(\ddot{\boldsymbol{q}}, \boldsymbol{\lambda}_{k,i}, \boldsymbol{\mu}_{k,\{1,\dots,M-1\}}\big)$
with $\quad i \in \{1,\dots,M\}, \quad k \in \{0,\dots,N-1\}$

where

$\boldsymbol{J}_{lh} = (\boldsymbol{J}_{\uparrow,lh}, \boldsymbol{J}_{\circlearrowleft,lh})$ Jacobian of the left hand,

$\boldsymbol{J}_{\uparrow,rh}$	translational Jacobian of the right hand,
$\boldsymbol{\pi}_{rh}$	appropriately defined PD-controller,
$\boldsymbol{L}$	Cholesky factor of $\boldsymbol{H}$: $\boldsymbol{L}^\top \boldsymbol{L} = \boldsymbol{H}$.

Whenever the external force $\boldsymbol{f}_{rh}$ is applied to the right hand, we account for its contribution in the equation of dynamics and joint torque constraints.

The decision variables are

- the generalized accelerations $\ddot{\boldsymbol{q}}$,
- current and anticipated contact forces represented with $\boldsymbol{\lambda}_{k,i}$,
- current and anticipated contact moments $\boldsymbol{\mu}_{k,\{1,\dots,M-1\}}$.

Note that the left hand contact moment with index $i = M$ is omitted, since this contact is chosen to be a point contact. Final momenta and their rates $\boldsymbol{\mathcal{P}}_N^{xy}$, ${}^{\mathrm{c}}\boldsymbol{\mathcal{L}}_N^{xy}$, $\dot{\boldsymbol{\mathcal{P}}}_N^{xy}$, ${}^{\mathrm{c}}\dot{\boldsymbol{\mathcal{L}}}_N$ are expressed through the decision variables $\boldsymbol{\chi}$ as explained in Subsection 4.3.1. The normal components of the optional contact forces $f_{k,lh}^n$ are also expressed with $\boldsymbol{\lambda}_{k,lh}$.

We achieve force prioritization using a separate priority level for minimization of the optional contact force. This level is of lower priority than the capturability constraints, which ensure the balance, and the right hand task. At the same time, this level is more important than minimization of all non-optional contact forces.

Length of the preview horizon is the same in all considered scenarios $N = 6$, and the sampling interval of the preview horizon is set to $T_k = 0.1$ $[s]$.

6.2.3 Results and discussion

We observed that in both considered scenarios the controller produces the desired behavior, *i.e.*, it applies the hand contact force if necessary, and stops applying it, when the need is gone (see Figures 6.14 and 6.15).

Note that if the capturability constraint is omitted nothing prevents the controller from sacrificing the balance in order to execute the hand task as demonstrated in Figure 6.14. Contrary to the case of the walking controller discussed in Section 6.1, the capturability constraint appears to be sufficient for preservation of balance.

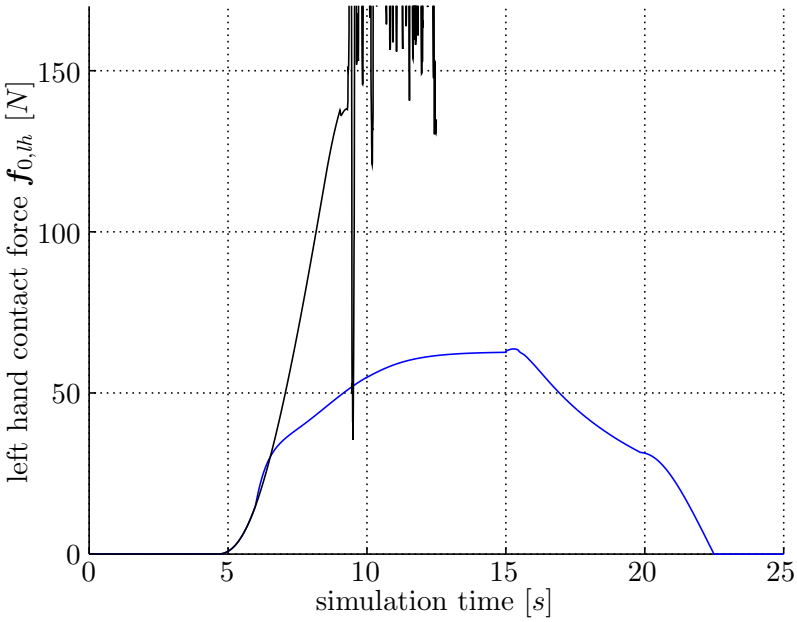
During construction and tuning of the controller one should keep in mind the following aspects:

- When the hand contact force minimization task starts to conflict with the higher priority objectives, the problem becomes singular. This results in violent accelerations of the joints illustrated in Figure 6.16. In order to avoid this we employ regularization of the 4-th and 5-th levels of the hierarchy using the objective on the final level as described in Subsection 5.2.2. Due to lacking support of regularization in `LexLS` we have to solve a sequence of PLLS problems to obtain the solution (see Appendix G).
- We have indicated earlier in Section 4.6, that the choice of sampling of the preview horizon may have a negative impact on the performance of the controller. This problem is illustrated in Figure 6.17. Periodic variations in the norm of the hand contact force are produced by the controller, if we choose to reduce the first sampling interval by 5 $[ms]$ instead of shifting the whole preview horizon by 5 $[ms]$. These two approaches are discussed in detail in Section 4.6.

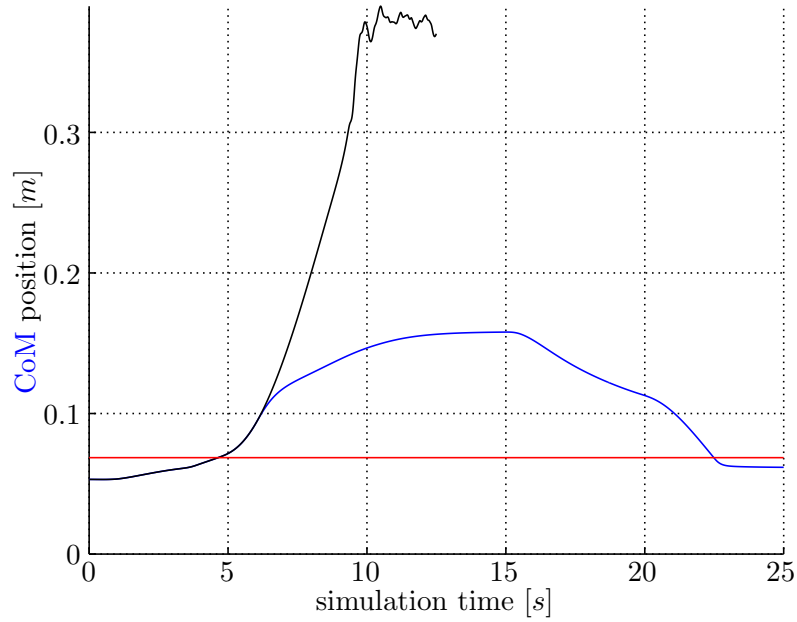
Though we have to solve the hierarchy in an inefficient way in order to implement regularization as described in Appendix G, the average computation time is below 100 $[ms]$.

6.2.4 Conclusion

An interesting property of the constructed controller is that the decision to apply the optional contact force is taken automatically based on the current state and tasks. Hence, there is

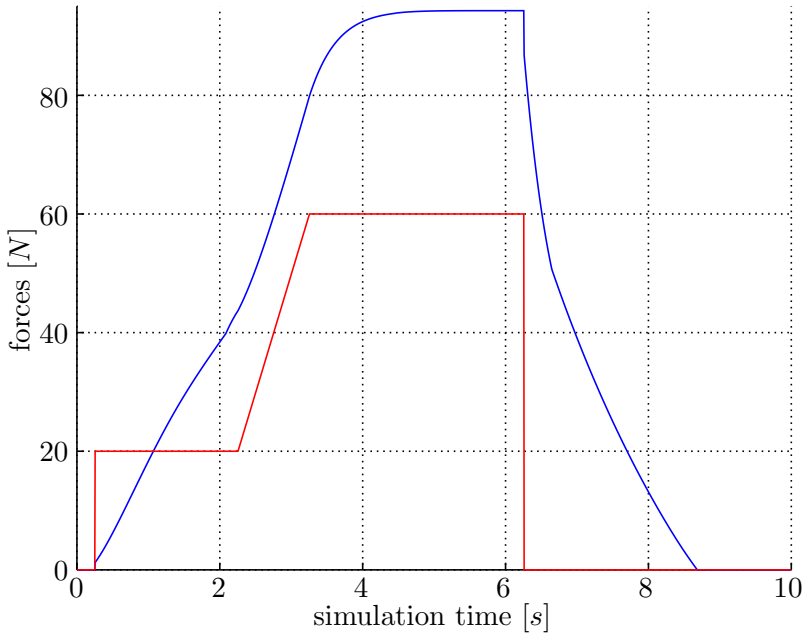


(a) Norms of the optional left hand contact forces. Force shown in solid blue line corresponds to the controller with the capturability constraint. The controller without the capturability constraint produces force shown in solid black.

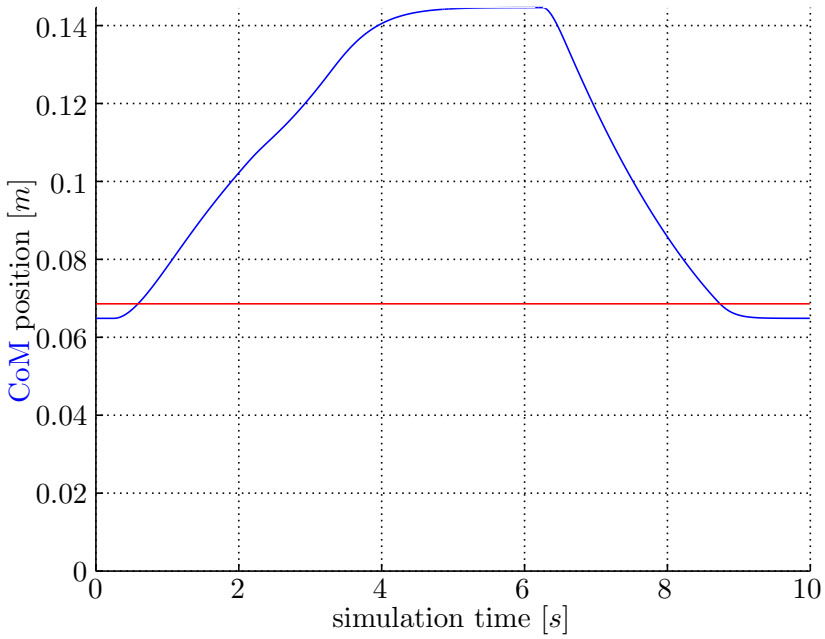


(b) Position of the CoM along the x axis. Position indicated with solid blue line corresponds to the controller with the capturability constraint. Solid black line indicates position of the CoM produced by the controller without the capturability constraint.

Figure 6.14: Exploitation of the optional hand contact while performing a right hand task.



(a) Norms of the optional left hand contact force $\boldsymbol{f}_{0,lh}$ (solid blue) and the disturbing external force $\boldsymbol{f}_{rh}$ (solid red).



(b) Position of the CoM along the x axis (solid blue) and the bound of the foot support area (solid red).

Figure 6.15: Exploitation of the optional hand contact in presence of external disturbing force.

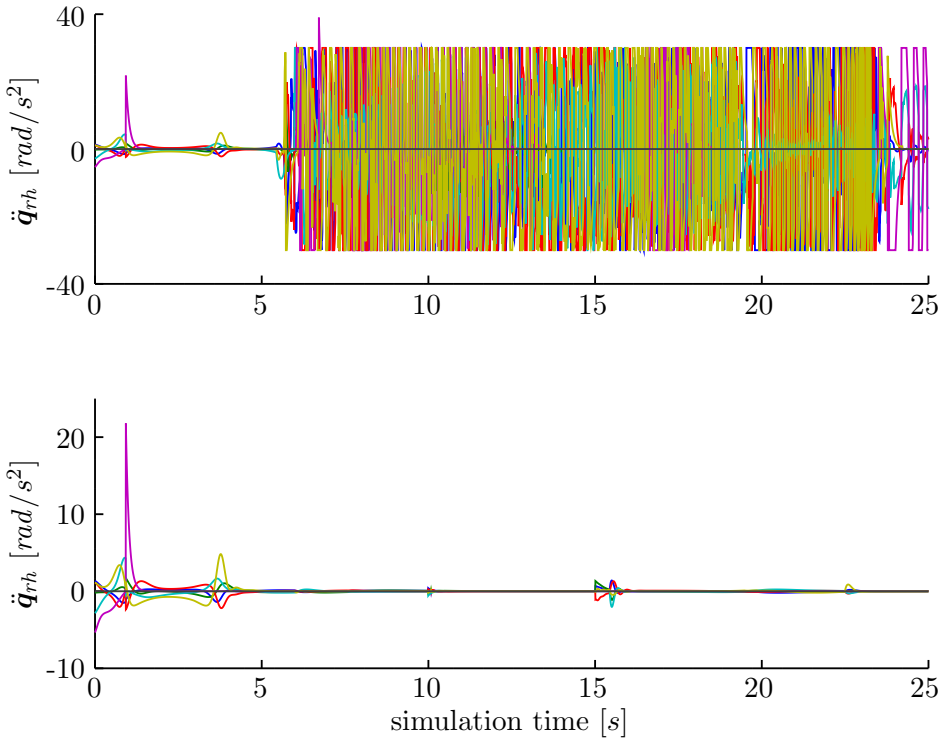


Figure 6.16: Accelerations of the right arm joints without regularization (top) and with it (bottom).

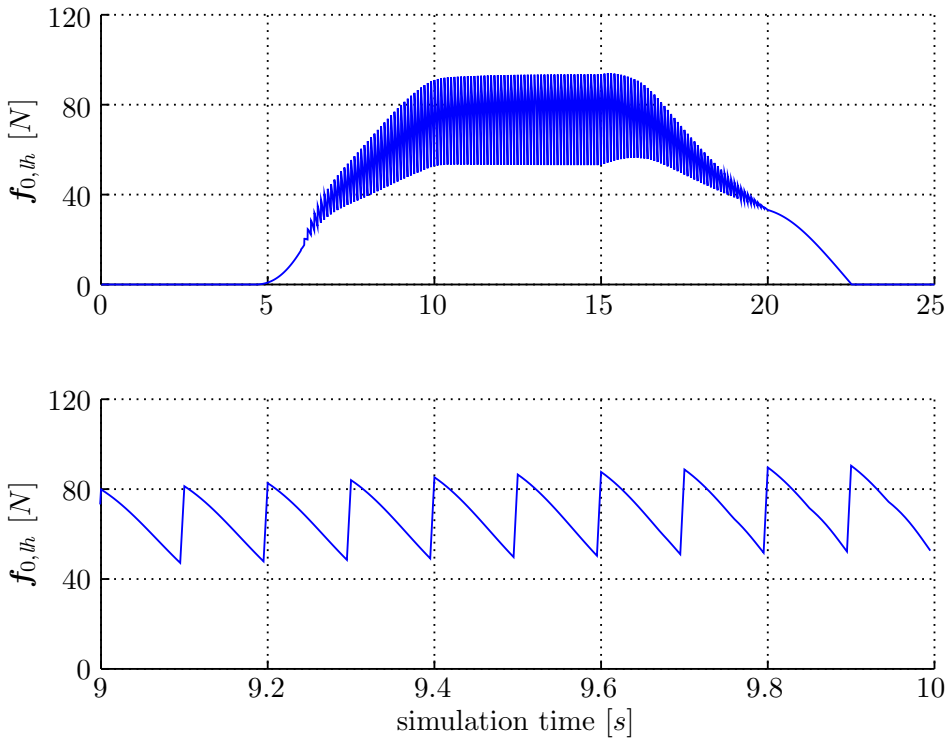


Figure 6.17: Norm of the optional left hand contact force when the first sampling interval is reduced at each control iteration.

no need for planning and tuning of contact timings, which are recognized as a significant drawback of approximate linear models [Audren 2014]. On the other hand, the considered setting is very limited: the contact positions are known and fixed in advance. Extensibility of the approach to more general situations is to be investigated.

One potential application of the controller could be evaluation of necessity of an optional contact for balance preservation in the future. For example, when the robot is pushed, the controller may decide if using an additional hand contact with a wall at some time t in the future would be necessary to avoid a fall. If so, an appropriate task can be triggered to move the hand towards the wall.

We believe that the current primary obstacle for adoption of the proposed approach is the lack of numerical tools, which can solve hierarchies both efficiently and accurately. The size of Hierarchy 6.2 makes its solution with a sequence of QPs impractical. At the same time, the existing specialized solvers for hierarchies, `LexLS` [Dimitrov 2015] and `SOTH` [Escande 2014, SOT 2016], provide very limited mechanisms for coping with the numerical problems near singularities discussed in Subsection 6.2.3. Since singularities commonly arise due to conflicts in hierarchies, it is necessary to develop and extend these mechanisms in order to maximally benefit from prioritization. For example, it is desirable to have regularization with general equality objective with automatic selection of the regularization factor (see Subsection 5.2.2).

6.3 Collaborations

A substantial part of the work covered by the present thesis was performed in collaboration with other doctoral and master students from different research institutes. The author's contribution to the collaborative works is briefly outlined in the following subsections.

6.3.1 Walking with nonplanar CoM motion

One of the recurring assumptions made in the construction of linear approximate models of humanoid robots is that the CoM motion is constrained to a plane (see Assumption 3.17 in Section 3.3). This limitation is particularly inconvenient when the robot has to walk on uneven terrain, *i.e.*, when walking up and down stairs. Moreover, even walking on a flat ground with planar CoM motion is unnatural and energy inefficient in comparison with humans. This issue was addressed in the Master's thesis of Camille Brasseur, who was, to some extend, co-advised by the author.

Camille's work resulted in the development of an approximate model [Brasseur 2015a], which enables nonplanar motion of the CoM at the price of a larger number of constraints, as indicated in Subsection 3.3.3 of this thesis. The obtained model was employed by Camille in the implementation of an *Model Predictive Control* (MPC) scheme for the generation of walking motions on flat and uneven ground. Since this MPC scheme was not integrated in an MMPC controller, the author developed a simple whole body inverse dynamics controller based on Hierarchy 5.9 for tracking the generated CoM and foot trajectories. The whole body controller was used to demonstrate the validity of the approach using a simulated `HRP-2` robot. Snapshots of a simulated walk on stairs are given in Figure 6.18. This joint work led to a publication in a conference proceedings [Brasseur 2015b]. Also, the results with detailed derivations were reported in the corresponding Master's thesis [Brasseur 2015a].

6.3.2 Time-optimal control of industrial manipulators

The idea of time-optimal control using the temporal task prioritization in an MPC problem, as described in Subsection 5.3.3, was proposed by the author to another doctoral student,

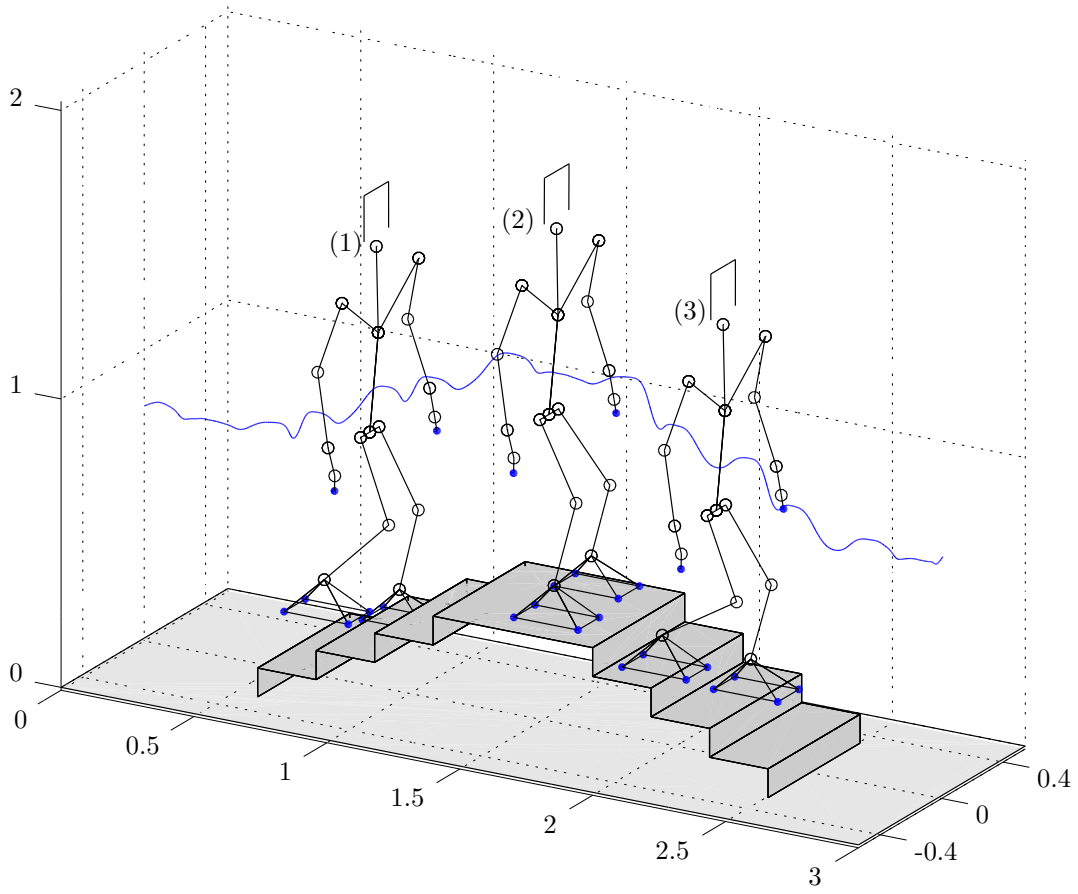


Figure 6.18: Robot walking up and down stairs. Sequence of the snapshots is indicated by the numbers. Blue curve indicates nonplanar trajectory of the CoM.

Saed Al Homsi, who performed research in time-optimal control of industrial manipulators. One of the goals of Saed's work was to reduce involvement of humans in deployment of industrial robots operating in a shared environment. One of the problems, which must be faced in such setups, is collision avoidance between the robots (see Figure 6.19). This can be achieved by designing an MPC scheme with appropriate constraints on positions of the links of the manipulator. At the same time, addition of task prioritization in this MPC problem allows for time-optimal behavior, which is valuable in industrial applications. The obtained PLLS optimization problem can be solved using `LexLS` or another specialized solver.

Based on the proposed approach to time-optimal control, Saed developed a real-time controller for Adept Cobra SCARA manipulators [ADE 2016] and validated it on real robots in the experimental setup visualized in Figure 6.19. Results of the experiments with additional simulations and overview of the concept were published in [Homsi 2016b], an interested reader can find further details in Saed's dissertation [Homsi 2016a].

6.3.3 Physical human-robot collaboration for carrying

The execution of certain tasks may require physical collaboration of robots with each other or with humans [Caccavale 2008, Bicchi 2008]. Humanoid robots are not an exception [Agravante 2016a]. For example, they can assist humans in carrying heavy or bulky objects, as shown in Figures 6.20 and 6.21. The realization of such collaboration was the goal

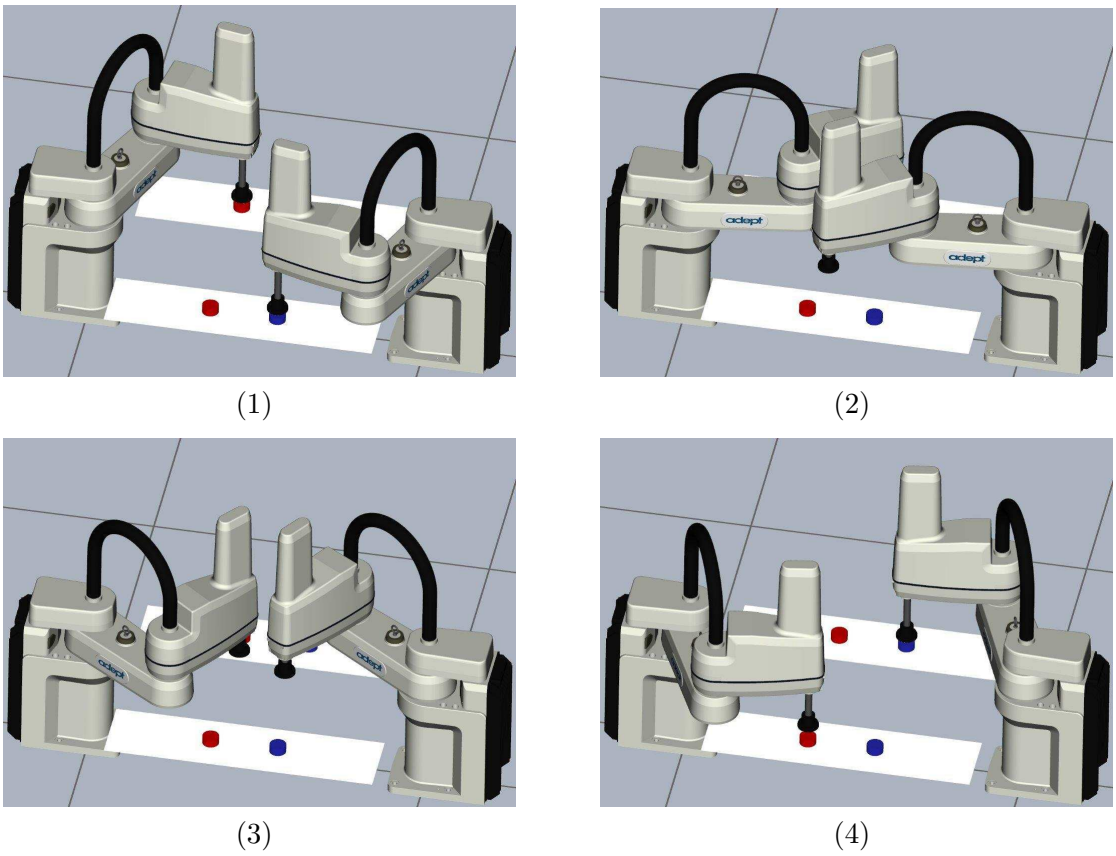


Figure 6.19: Two Adept Cobra SCARA robots [ADE 2016] operating in a shared environment. The robots have to reach targets, indicated by colored cylinders, on two conveyors. A target for a particular robot may appear on either of the conveyors. In order to realize the tasks, the robots have to avoid collisions with each other.

of Joven Agravante during his Doctoral research at LIRMM in Montpellier.

The transportation of an object requires walking and, hence, motion anticipation, which must take into account two important aspects of physical collaboration between the human and the robot:

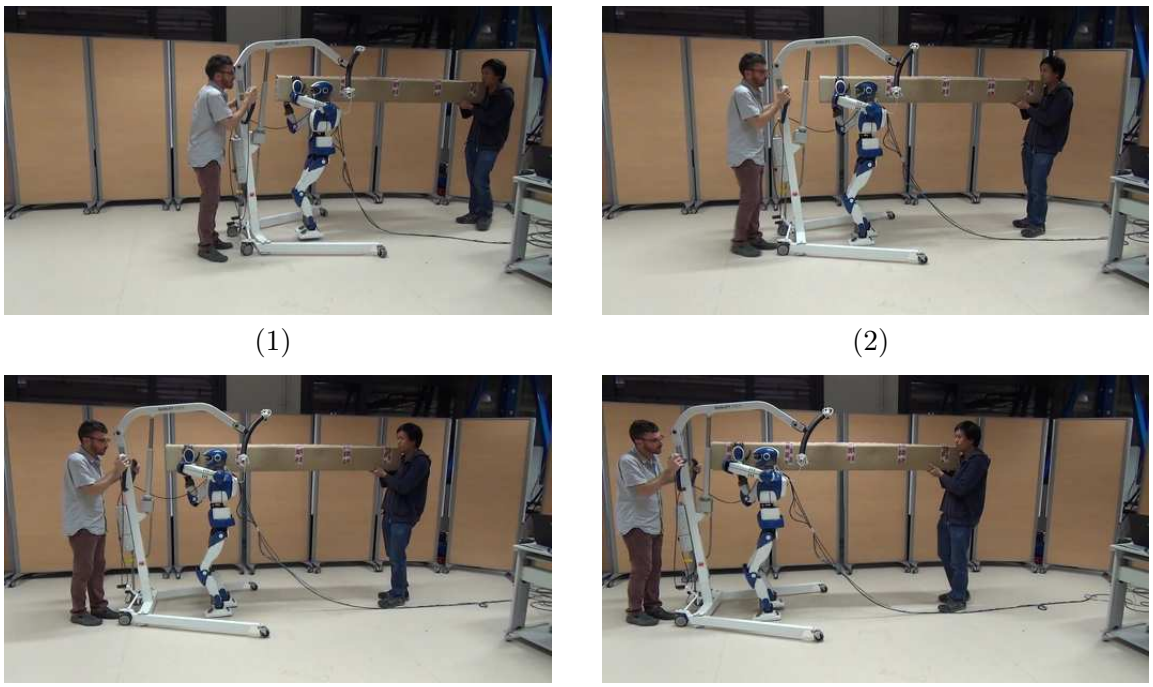
- it is necessary to account for the interaction forces exerted on the robot in order to maintain balance;
- it is necessary to react to human intentions conveyed through the interaction forces by adjusting motions of the robot appropriately, in particular, by adjusting its waking speed.

Joven addressed both of these aspects by developing the approximate CPPMJ model, which incorporates the external wrench $(\boldsymbol{f}_{ext}, \boldsymbol{\mu}_{ext})$ acting on the CoM of the robot. Thus, the estimated interaction force applied to the CoM is taken into account in the balance preservation constraints and can influence the motion of the CoM through an impedance task of the following form

$$\boldsymbol{f}_{ext}^{xy} = \boldsymbol{\Gamma}_{\boldsymbol{x}} \boldsymbol{x}, \tag{6.1}$$

where $\boldsymbol{\Gamma}_{\boldsymbol{x}}$ is an appropriately chosen matrix and the state of the model is $\boldsymbol{x} = (c^x, \dot{c}^x, \ddot{c}^x, c^y, \dot{c}^y, \ddot{c}^y)$. The obtained model was employed in an MPC scheme for the generation of walking motions of an `HRP-4` robot [Kaneko 2011], as illustrated in Figures 6.20 and 6.21. In order to attain real-time performance of this scheme it was implemented by Joven in `C++` using the software framework `HuMoTo`, which was designed and partially imple-

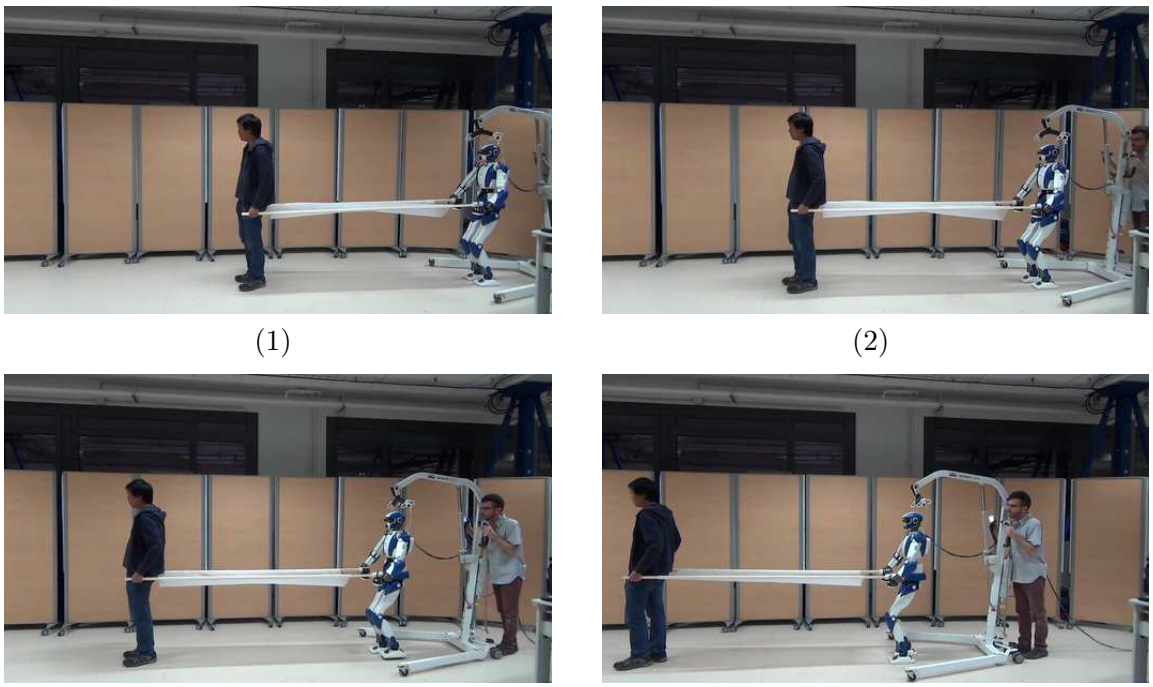
mented by the author. Experimental and theoretical results of this work were reported in [Agravante 2016a, Agravante 2016b] and Joven's thesis [Agravante 2015].



(1) (2)

(3) (4)

Figure 6.20: An HRP-4 [Kaneko 2011] carries a box in collaboration with a human.



(1) (2)

(3) (4)

Figure 6.21: An HRP-4 [Kaneko 2011] carries a stretcher in collaboration with a human.

Chapter 7

Conclusion

7.1 Summary

This thesis aims at understanding and developing motion control of humanoid robots, with an ultimate goal to make motion capabilities of humans and robots comparable. One of the key aspects of human motion is preservation of balance. An abstract discussion in Chapter 2 led us to the conclusion that preservation of balance is equivalent to maintaining capturability, *i.e.*, the ability to stop. The most practical way to ensure capturability is to anticipate motions which are constrained to end in statically balanced states. In order to account for non-deterministic changes in the environment, anticipation must be performed in real time, which makes anticipation of whole body motions a particularly challenging problem. A common approach is to sacrifice quality and completeness of anticipated motions by employing approximate models of humanoid robots. Approximate models lack the expressiveness of whole body models (Chapter 3) and, hence, cannot accurately reflect whole body tasks and constraints. We address this drawback by the integration of an instantaneous whole body motion controller with anticipation based on an approximate model. We call this approach *Mixed Model Predictive Control* (MMPC), since we mix models of different accuracy within a single predictive controller (Chapter 4). While this idea applies to all kinds of models, we limit the present work to linear models, which allows us to formulate our MMPC controllers as *Prioritized Linear Least-Squares* (PLLS) optimization problems (Chapter 5). We evaluate our approach in simulations using two different MMPC controllers (Chapter 6). One of them is capable of adjusting steps in response to disturbances and under the influence of whole body tasks. The second one includes a strict prioritization of the objectives in the controller in order to exploit additional hand contact with the environment only when it is necessary for balance preservation or execution of a particular whole body task. The results obtained with both controllers support the validity of our approach. There is, of course, a number of topics for further investigation in the considered controllers, in MMPC, modeling of humanoid robots, and solution of PLLS problems. We give a brief outlook of these topics in the following section.

7.2 Perspectives

7.2.1 Mixed Model Predictive Control

It appears that the discrepancy between the whole body motion control sampling time and the anticipation sampling time results in periodic variations in commands generated by MMPC controllers. We discuss this issue in Section 4.6, Subsection 6.1.3.3, and Subsection 6.2.3, but so far could not propose a satisfactory solution. Therefore, it is an important topic for

further research.

7.2.2 Walking using an approximate model

The MMPC controller for walking considered in Section 6.1 requires further development to make it interesting for practical applications. Though such limitations as fixed duration and sequence of steps are unlikely to be lifted without adoption of nonlinear approximate models, there is still room for improvement with linear models:

- It is possible to enable walking with varying *Center of Mass* (CoM) height using the recent proposal from [Brasseur 2015b].
- We can abandon the point-mass model and employ a model including angular momentum. Incorporation of angular momentum in the capturability constraint may help to avoid situations when the currently used capturability constraint is not sufficient for preservation of balance (see Subsection 6.1.3.1). Furthermore, angular momentum allows to account for motions of heavy end-effectors, *e.g.*, legs, in the anticipated motions (see Section 3.4).
- So far we used simple cubic polynomials for generation of foot trajectories. Instead, we can adopt a triple integrator for this purpose, in the same way as we do for the CoM motion in CPPMJ model. This modification introduces additional freedom in trajectory generation and allows for variation of the step height or simple obstacle avoidance.
- In this work we did not realize rotations of the robot and its feet, which is a significant drawback. In general, rotations result in nonlinear constraints on the *Center of Pressure* (CoP) positions and positions of the feet. We can address the first problem by shrinking the CoP constraints as proposed in Appendix D.2. The second problem can be avoided by reducing the length of the preview horizon from 2 to 1 step, which we have already demonstrated to be possible in Subsection 6.1.3.2.

Last, but not least is the experimental evaluation of the controller on a real robot.

7.2.3 Prioritization in contact force distribution

Further development of the idea of prioritization in contact force distribution for partial contact planning as explained in Subsection 6.2.4 is appealing, but may require significant improvements in numerical tools to make it applicable in practice.

7.2.4 Solvers for optimization problems with prioritization

We see two primary directions for improvement of the existing solvers of *Prioritized Linear Least-Squares* (PLLS) problems:

- We would like the solvers to be able to exploit general sparsity patterns in the objectives for improvement of performance and not only simple bounds. Ideally, we would like to avoid any manual variable elimination steps before solution of a PLLS problem.
- The current mechanisms for coping with issues near singularities appear to be insufficient and require further development (see Subsection 6.2.4).

Appendix A

Dynamics of a multibody system

In this appendix we derive equation of dynamics of a floating base multibody system with unilateral constraints. The presentation is based on [Wieber 2006a], and likewise aims at exposing the structure of this equation, rather than at fast computation of its components. In addition to that, we demonstrate how momenta of the whole system are computed.

A.1 Preliminaries

A.1.1 Notation

We consider a system of $n+1$ interconnected rigid bodies. We associate the following variables with each body $k \in \{1, ..., n+1\}$

m_k	$\in \mathbb{R}_{>0}$	mass of k-th body
$\boldsymbol{c}_k$	$\in \mathbb{R}^3$	position of *Center of Mass* (CoM) of k-th body in the global frame
$\ddot{\boldsymbol{x}}_k = (\ddot{\boldsymbol{c}}_k, \dot{\boldsymbol{\omega}}_k)$	$\in \mathbb{R}^6$	constrained spatial acceleration of k-th body in frame k
$\ddot{\boldsymbol{x}}_{k,u}$	$\in \mathbb{R}^6$	unconstrained spatial acceleration of k-th body in frame k
$\boldsymbol{\mathcal{I}}_k$	$\in \mathbb{R}^{3\times3}$	inertia matrix of k-th body in frame k
$\boldsymbol{H}_k = \begin{bmatrix} m_k\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{I}}_k \end{bmatrix}$	$\in \mathbb{R}^{6\times6}$	spatial inertia matrix of k-th body in frame k
$(\boldsymbol{f}_k, \boldsymbol{\mu}_k)$	$\in \mathbb{R}^{6\times6}$	wrench acting on k-th body in frame k
$\boldsymbol{J}_k = (\boldsymbol{J}_{\uparrow,k}, \boldsymbol{J}_{\circlearrowleft,k})$	$\in \mathbb{R}^{6\times(n+6)}$	Jacobian of k-th body

and fix frame k to the CoM of k-th body. All frames k, as well as frame r fixed to the base have the same orientation as the global frame. Also, we reuse some of the variables defined in Subsection 3.1.1

$\boldsymbol{q} = (\boldsymbol{q}', \boldsymbol{r}, \boldsymbol{\mathcal{E}})$	$\in \mathbb{R}^{n+6}$	vector of generalized coordinates
$\boldsymbol{H}(\boldsymbol{q})$	$\in \mathbb{R}^{(n+6)\times(n+6)}$	inertia matrix of the whole system
$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$	$\in \mathbb{R}^{n+6}$	vector of Coriolis and centrifugal terms of the whole system
$\boldsymbol{J}_{com}$	$\in \mathbb{R}^{3\times(n+6)}$	Jacobian of the CoM
$\boldsymbol{I}_{\boldsymbol{\tau}}$	$\in \mathbb{R}^{(n+6)\times n}$	torque selection matrix
$\boldsymbol{g}$	$\in \mathbb{R}^3$	vector of gravitational acceleration
m	$\in \mathbb{R}_{>0}$	total mass of the system

A.1.2 Structure of Jacobians

Vector of generalized coordinates $\boldsymbol{q}$ contains three parts corresponding to joint angles, position, and orientation of the base. Consequently, the Jacobians, which map generalized velocities to the twist of the k-th body

$$\begin{bmatrix}\dot{\boldsymbol{c}}_k \\ \boldsymbol{\omega}_k\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\circlearrowleft,k}\end{bmatrix}\dot{\boldsymbol{q}} = \boldsymbol{J}_k\dot{\boldsymbol{q}}, \tag{A.1}$$

have corresponding structure:

$$\boldsymbol{J}_k = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\circlearrowleft,k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{J}_{\uparrow,k,2} & \boldsymbol{J}_{\uparrow,k,3} \\ \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{J}_{\circlearrowleft,k,2} & \boldsymbol{J}_{\circlearrowleft,k,3}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{I}_3 & -(\boldsymbol{c}_k - \boldsymbol{r})^\times \\ \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{0}_{3,3} & \boldsymbol{I}_3\end{bmatrix} \tag{A.2}$$

In situations when orientation of the base is represented by Euler angles $\boldsymbol{\mathcal{E}} \in \mathbb{R}^3$ and its angular velocity and acceleration are replaced with $\dot{\boldsymbol{\mathcal{E}}}$ and $\ddot{\boldsymbol{\mathcal{E}}}$, it is necessary to introduce matrix $\boldsymbol{T}_{\mathcal{E}}$, which transforms derivatives of the Euler angles to angular velocities:

$$\boldsymbol{J}_k = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{I}_3 & -(\boldsymbol{c}_k - \boldsymbol{r})^\times\boldsymbol{T}_{\mathcal{E}} \\ \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{I}_3 & ((\boldsymbol{c}_k - \boldsymbol{r})^\times)^\top\boldsymbol{T}_{\mathcal{E}} \\ \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}\end{bmatrix} \tag{A.3}$$

Henceforth, we use this version of the Jacobian, since we represent orientation, angular velocity and acceleration of the base using Euler angles in our simulations.

A.2 Gauss' principle

The Gauss' principle states that constrained motions of rigid bodies in a system are as close as possible to the unconstrained motions in least-squares sense [Wieber 2006a, Moreau 1966]. Based on this principle we define the following optimization problem:

$$\begin{aligned}
\underset{\ddot{\boldsymbol{x}}_1,\ldots,\ddot{\boldsymbol{x}}_n,\ddot{\boldsymbol{q}}}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=1}^{n+1}(\ddot{\boldsymbol{x}}_k - \ddot{\boldsymbol{x}}_{k,u})^\top\boldsymbol{H}_k(\ddot{\boldsymbol{x}}_k - \ddot{\boldsymbol{x}}_{k,u}) \\
\text{subject to} \quad & \ddot{\boldsymbol{x}}_k = \begin{bmatrix}\ddot{\boldsymbol{c}}_k \\ \dot{\boldsymbol{\omega}}_k\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\circlearrowleft,k}\end{bmatrix}\ddot{\boldsymbol{q}} + \begin{bmatrix}\dot{\boldsymbol{J}}_{\uparrow,k} \\ \dot{\boldsymbol{J}}_{\circlearrowleft,k}\end{bmatrix}\dot{\boldsymbol{q}} = \boldsymbol{J}_k\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} \\
& \boldsymbol{H}_k\ddot{\boldsymbol{x}}_{k,u} = \begin{bmatrix}\boldsymbol{f}_k \\ \boldsymbol{\mu}_k - \boldsymbol{\omega}_k \times \boldsymbol{\mathcal{I}}_k\boldsymbol{\omega}_k\end{bmatrix} \\
& \ddot{\boldsymbol{c}}_j = \boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} \geq \boldsymbol{b}_j,
\end{aligned} \tag{A.4}$$

where inequality constraint on $\ddot{\boldsymbol{c}}_j$ with $j \in \{1, \ldots, n+1\}$ is added for illustration. Handling of this constraint is easily generalized to arbitrary inequalities, such as contact constraints in Section 3.1.

First, we expand multiplication in the objective function and eliminate constant terms to obtain:

$$\begin{aligned}
\underset{\ddot{\boldsymbol{x}}_1,\ldots,\ddot{\boldsymbol{x}}_n,\ddot{\boldsymbol{q}}}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=1}^{n+1}\left(\ddot{\boldsymbol{x}}_k^\top\boldsymbol{H}_k\ddot{\boldsymbol{x}}_k - 2\ddot{\boldsymbol{x}}_k^\top\boldsymbol{H}_k\ddot{\boldsymbol{x}}_{k,u}\right) \\
\text{subject to} \quad & \ddot{\boldsymbol{x}}_k = \boldsymbol{J}_k\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} \\
& \boldsymbol{H}_k\ddot{\boldsymbol{x}}_{k,u} = \begin{bmatrix}\boldsymbol{f}_k \\ \boldsymbol{\mu}_k - \boldsymbol{\omega}_k \times \boldsymbol{\mathcal{I}}_k\boldsymbol{\omega}_k\end{bmatrix} \\
& \boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} \geq \boldsymbol{b}_j.
\end{aligned} \tag{A.5}$$

Then we eliminate all equality constraints

$$\begin{aligned} \underset{\ddot{\boldsymbol{q}}}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=1}^{n+1}\Bigg((\boldsymbol{J}_k\ddot{\boldsymbol{q}})^\top \boldsymbol{H}_k\boldsymbol{J}_k\ddot{\boldsymbol{q}} + 2(\boldsymbol{J}_k\ddot{\boldsymbol{q}})^\top \boldsymbol{H}_k\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} + (\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}})^\top \boldsymbol{H}_k\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} \\ & - 2(\boldsymbol{J}_k\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}})^\top \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{\mu}_k - \boldsymbol{\omega}_k \times \boldsymbol{\mathcal{I}}_k\boldsymbol{\omega}_k \end{bmatrix}\Bigg) \\ \text{subject to} \quad & \boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} \geq \boldsymbol{b}_j, \end{aligned} \tag{A.6}$$

and drop the constant terms

$$\begin{aligned} \underset{\ddot{\boldsymbol{q}}}{\text{minimize}} \quad & \sum_{k=1}^{n+1}\left(\frac{1}{2}(\boldsymbol{J}_k\ddot{\boldsymbol{q}})^\top \boldsymbol{H}_k\boldsymbol{J}_k\ddot{\boldsymbol{q}} + (\boldsymbol{J}_k\ddot{\boldsymbol{q}})^\top \boldsymbol{H}_k\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - (\boldsymbol{J}_k\ddot{\boldsymbol{q}})^\top \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{\mu}_k - \boldsymbol{\omega}_k \times \boldsymbol{\mathcal{I}}_k\boldsymbol{\omega}_k \end{bmatrix}\right) \\ \text{subject to} \quad & \boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} \geq \boldsymbol{b} \end{aligned} \tag{A.7}$$

Karush-Kuhn-Tucker (KKT) optimality conditions for the considered problem are stated as a complementarity system [Nocedal 2006]

$$\begin{cases} \displaystyle\sum_{k=1}^{n+1}\left(\boldsymbol{J}_k^\top \boldsymbol{H}_k\boldsymbol{J}_k\ddot{\boldsymbol{q}} + \boldsymbol{J}_k^\top \boldsymbol{H}_k\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} - \boldsymbol{J}_k^\top \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{\mu}_k - \boldsymbol{\omega}_k \times \boldsymbol{\mathcal{I}}_k\boldsymbol{\omega}_k \end{bmatrix}\right) - \boldsymbol{J}_{\uparrow,j}^\top\boldsymbol{\Lambda} = 0, & \text{(A.8a)} \\ \boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} \geq \boldsymbol{b}, & \text{(A.8b)} \\ \boldsymbol{\Lambda} \geq \boldsymbol{0}, & \text{(A.8c)} \\ \boldsymbol{\Lambda}^\top\left(\boldsymbol{J}_{\uparrow,j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\uparrow,j}\dot{\boldsymbol{q}} - \boldsymbol{b}\right) = \boldsymbol{0}, & \text{(A.8d)} \end{cases}$$

where $\boldsymbol{\Lambda}$ is the vector of Lagrange multipliers. System (A.8) corresponds to the complementarity system presented in Section 3.1: the first line is the equation of dynamics of the whole system and $\boldsymbol{\Lambda}$ correspond to the contact forces.

We further modify (A.8a) by substituting $\boldsymbol{\omega}_k = \boldsymbol{J}_{\circlearrowleft,k}\dot{\boldsymbol{q}}$ and moving all forces to the right side:

$$\underbrace{\sum_{k=1}^{n+1}\left(\boldsymbol{J}_k^\top \boldsymbol{H}_k\boldsymbol{J}_k\right)}_{\boldsymbol{H}}\ddot{\boldsymbol{q}} + \underbrace{\sum_{k=1}^{n+1}\left(\boldsymbol{J}_k^\top \boldsymbol{H}_k\dot{\boldsymbol{J}}_k\dot{\boldsymbol{q}} + \boldsymbol{J}_{\circlearrowleft,k}^\top((\boldsymbol{J}_{\circlearrowleft,k}\dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k\boldsymbol{J}_{\circlearrowleft,k}\dot{\boldsymbol{q}})\right)}_{\boldsymbol{h}} = \sum_{k=1}^{n+1}\boldsymbol{J}_k^\top\begin{bmatrix}\boldsymbol{f}_k \\ \boldsymbol{\mu}_k\end{bmatrix} + \boldsymbol{J}_{\uparrow,j}^\top\boldsymbol{\Lambda}. \tag{A.9}$$

A.3 Structure of the equation of dynamics

In this section we employ our knowledge of the structure of Jacobians to explore the structure of components of Equation (A.9).

A.3.1 Inertia matrix

Expansion of the Jacobians exposes the structure of inertia matrix as shown below

$$\boldsymbol{H} = \sum_{k=1}^{n+1} \boldsymbol{J}_k^\top \boldsymbol{H}_k \boldsymbol{J}_k \tag{A.10a}$$

$$= \sum_{k=1}^{n+1} m_k \boldsymbol{J}_{\uparrow,k}^\top \boldsymbol{J}_{\uparrow,k} + \sum_{k=1}^{n+1} \boldsymbol{J}_{\circlearrowleft,k}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \tag{A.10b}$$

$$= \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k,1}^\top \boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\uparrow,k,2}^\top \boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\uparrow,k,3}^\top \boldsymbol{J}_{\uparrow,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{\circlearrowleft,k,1}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \\ \boldsymbol{J}_{\circlearrowleft,k,2}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \\ \boldsymbol{J}_{\circlearrowleft,k,3}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \end{bmatrix} \right) \tag{A.10c}$$

$$\begin{aligned} = \sum_{k=1}^{n+1} \Bigg(& m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k,1}^\top \boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{J}_{\uparrow,k,1}^\top & -\boldsymbol{J}_{\uparrow,k,1}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times \boldsymbol{T}_\mathcal{E} \\ \boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{I}_3 & -(\boldsymbol{c}_k - \boldsymbol{r})^\times \boldsymbol{T}_\mathcal{E} \\ \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times \boldsymbol{J}_{\uparrow,k,1} & \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times & -\boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times (\boldsymbol{c}_k - \boldsymbol{r})^\times \boldsymbol{T}_\mathcal{E} \end{bmatrix} \\ & + \begin{bmatrix} \boldsymbol{J}_{\circlearrowleft,k,1}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{0}_{n,3} & \boldsymbol{J}_{\circlearrowleft,k,1}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{T}_\mathcal{E} \\ \boldsymbol{0}_{3,n} & \boldsymbol{0}_{3,3} & \boldsymbol{0}_{3,3} \\ \boldsymbol{T}_\mathcal{E}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k,1} & \boldsymbol{0}_{3,3} & \boldsymbol{T}_\mathcal{E}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{T}_\mathcal{E} \end{bmatrix} \Bigg) \end{aligned} \tag{A.10d}$$

$$= \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k,1}^\top \boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{J}_{\uparrow,k} \\ \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times \boldsymbol{J}_{\uparrow,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{\circlearrowleft,k,1}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \\ \boldsymbol{0}_{n+6,3} \\ \boldsymbol{T}_\mathcal{E}^\top \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \end{bmatrix} \right) = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \end{bmatrix} \tag{A.10e}$$

It is worth noting that the structure of the inertia matrix simplifies when the base $\boldsymbol{r}$ is chosen to coincide with the CoM $\boldsymbol{c}$. In this case some of the components of $\boldsymbol{H}$ are equal to zero due to

$$\sum_{k=1}^{n+1} m_k (\boldsymbol{c}_k - \boldsymbol{r})^\times = \sum_{k=1}^{n+1} (m_k \boldsymbol{c}_k - m_k \boldsymbol{c})^\times = \left(m\boldsymbol{c} - \sum_{k=1}^{n+1} (m_k \boldsymbol{c}) \right)^\times = (\boldsymbol{0}_{3,1})^\times = \boldsymbol{0}_{3,3}. \tag{A.11}$$

A.3.2 Nonlinear term

$$\begin{aligned} \boldsymbol{h} &= \sum_{k=1}^{n+1} \Big(\boldsymbol{J}_k^T \boldsymbol{H}_k \dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}} + \boldsymbol{J}_{\circlearrowleft,k}^T ((\boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \Big) \\ &= \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k,1}^T \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \\ \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{\circlearrowleft,k,1}^T \boldsymbol{\mathcal{I}}_k \dot{\boldsymbol{J}}_{\circlearrowleft,k} \dot{\boldsymbol{q}} + \boldsymbol{J}_{\circlearrowleft,k,1}^T ((\boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \\ \boldsymbol{0}_{3,1} \\ \boldsymbol{T}_\mathcal{E}^\top \boldsymbol{\mathcal{I}}_k \dot{\boldsymbol{J}}_{\circlearrowleft,k} \dot{\boldsymbol{q}} + \boldsymbol{T}_\mathcal{E}^\top ((\boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \end{bmatrix} \right) \\ &= \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \boldsymbol{h}_3 \end{bmatrix} \end{aligned} \tag{A.12}$$

A.3.3 Forces

Assuming that the wrenches $(\boldsymbol{f}_k, \boldsymbol{\mu}_k)$ are the result of gravity field and joint torques we obtain

$$\sum_{k=1}^{n+1} \boldsymbol{J}_k^\top \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{\mu}_k \end{bmatrix} + \boldsymbol{J}_{\uparrow,j}^\top \boldsymbol{\Lambda} = \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k,1}^T \\ \boldsymbol{I}_3 \\ \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_k - \boldsymbol{r})^\times \end{bmatrix} \right) \boldsymbol{g} + \boldsymbol{I}_{\boldsymbol{\tau}} \boldsymbol{\tau} + \begin{bmatrix} \boldsymbol{J}_{\uparrow,j,1}^T \\ \boldsymbol{I}_3 \\ \boldsymbol{T}_\mathcal{E}^\top (\boldsymbol{c}_j - \boldsymbol{r})^\times \end{bmatrix} \boldsymbol{\Lambda}. \tag{A.13}$$

A.4 Momenta of the system

Consider the lower 6 lines of the equation of dynamics:

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}^{\top} \end{bmatrix} \left(\sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k} \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \boldsymbol{J}_{\uparrow,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,n+6} \\ \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \end{bmatrix} \right) \ddot{\boldsymbol{q}} \right. \\
&\left. + \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,1} \\ \boldsymbol{\mathcal{I}}_k \dot{\boldsymbol{J}}_{\circlearrowleft,k} \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,1} \\ ((\boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \end{bmatrix} \right) \right) \\
&= \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}^{\top} \end{bmatrix} \left(\sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{I}_3 \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \end{bmatrix} \right) \boldsymbol{g} + \begin{bmatrix} \boldsymbol{I}_3 \\ (\boldsymbol{c}_j - \boldsymbol{r})^{\times} \end{bmatrix} \boldsymbol{\Lambda} \right).
\end{aligned} \tag{A.14}$$

This is the Newton-Euler equation of the whole system multiplied by the constant matrix $\operatorname{diag}(\boldsymbol{I}_3, \boldsymbol{T}_{\mathcal{E}}^{\top})$ [Wieber 2006a]. The right hand side of the equation gives the total wrench acting at the reference point $\boldsymbol{r}$ of the system, hence

$$\begin{aligned}
&\sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k} \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \boldsymbol{J}_{\uparrow,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,n+6} \\ \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \end{bmatrix} \right) \ddot{\boldsymbol{q}} \\
&+ \sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \dot{\boldsymbol{J}}_{\uparrow,k} \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,1} \\ \boldsymbol{\mathcal{I}}_k \dot{\boldsymbol{J}}_{\circlearrowleft,k} \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,1} \\ ((\boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \times \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \dot{\boldsymbol{q}}) \end{bmatrix} \right) \\
&= \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}^{-\top} \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{h}_2 \\ \boldsymbol{h}_3 \end{bmatrix} \right) = \begin{bmatrix} {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}} \\ {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}} \end{bmatrix},
\end{aligned} \tag{A.15}$$

where ${}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}}$ and ${}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}}$ are the rates of linear and angular momenta of the system expressed in frame r. Integration of this equation allows to compute the momenta as

$$\sum_{k=1}^{n+1} \left(m_k \begin{bmatrix} \boldsymbol{J}_{\uparrow,k} \\ (\boldsymbol{c}_k - \boldsymbol{r})^{\times} \boldsymbol{J}_{\uparrow,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{3,n+6} \\ \boldsymbol{\mathcal{I}}_k \boldsymbol{J}_{\circlearrowleft,k} \end{bmatrix} \right) \dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{T}_{\mathcal{E}}^{-\top} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \end{bmatrix} \dot{\boldsymbol{q}} = \begin{bmatrix} {}^{\mathsf{r}}\boldsymbol{\mathcal{P}} \\ {}^{\mathsf{r}}\boldsymbol{\mathcal{L}} \end{bmatrix}. \tag{A.16}$$

Then, assuming that frame r has the same orientation as the global frame

$${}^{\mathsf{r}}\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{P}} = m\dot{\boldsymbol{c}} = \sum_{k=1}^{n+1} (m_k \boldsymbol{J}_{\uparrow,k}) \dot{\boldsymbol{q}}, \quad \dot{\boldsymbol{c}} = \left(\frac{1}{m} \sum_{k=1}^{n+1} m_k \boldsymbol{J}_{\uparrow,k} \right) \dot{\boldsymbol{q}} = \boldsymbol{J}_{com} \dot{\boldsymbol{q}}, \tag{A.17}$$

where $\boldsymbol{J}_{com}$ is Jacobian of the CoM in accordance with [Sugihara 2002, Espiau 1998].

Often, it is more convenient to work with momenta $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{L}}$ expressed in a frame c fixed to the CoM of the system [Orin 2013]. If this frame has the same orientation as frame r transformation of the linear momentum is not needed ${}^{\mathsf{c}}\boldsymbol{\mathcal{P}} = {}^{\mathsf{r}}\boldsymbol{\mathcal{P}}$. In order to transform angular momentum we use ${}^{\mathsf{r}}\boldsymbol{\mathcal{L}} = {}^{\mathsf{c}}\boldsymbol{\mathcal{L}} + m\,{}^{\mathsf{r}}\boldsymbol{c} \times {}^{\mathsf{r}}\dot{\boldsymbol{c}}$, where ${}^{\mathsf{r}}\boldsymbol{c} = \boldsymbol{c} - \boldsymbol{r}$ is the vector from the base to the CoM:

$${}^{\mathsf{c}}\boldsymbol{\mathcal{L}} = {}^{\mathsf{r}}\boldsymbol{\mathcal{L}} - m\,{}^{\mathsf{r}}\boldsymbol{c} \times {}^{\mathsf{r}}\dot{\boldsymbol{c}} = {}^{\mathsf{r}}\boldsymbol{\mathcal{L}} - m\,{}^{\mathsf{r}}\boldsymbol{c} \times (\boldsymbol{J}_{com} \dot{\boldsymbol{q}}) = {}^{\mathsf{r}}\boldsymbol{\mathcal{L}} - {}^{\mathsf{r}}\boldsymbol{c} \times {}^{\mathsf{r}}\boldsymbol{\mathcal{P}}. \tag{A.18}$$

Rate of the angular momentum ${}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}}$ is computed by differentiating (A.18):

$${}^{\mathsf{c}}\dot{\boldsymbol{\mathcal{L}}} = {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{L}}} - \dot{\boldsymbol{c}} \times {}^{\mathsf{r}}\boldsymbol{\mathcal{P}} - {}^{\mathsf{r}}\boldsymbol{c} \times {}^{\mathsf{r}}\dot{\boldsymbol{\mathcal{P}}}. \tag{A.19}$$

Note, that computation of momenta with respect to the CoM as suggested in [Orin 2013] is equivalent to computation of the equation of dynamics with $\boldsymbol{r} = \boldsymbol{c}$, and it is unnecessary if equation of dynamics with $\boldsymbol{r} \neq \boldsymbol{c}$ is already constructed.

Appendix B

Impact law

Definition of impact law for the complementarity system (3.1) is an intricate problem [Glocker 2006]. In the present work we do not aim for accurate modeling, and, therefore, adopt two approximate versions of the law to simulate disturbances due to foot touchdowns and pushes of the robot by given impulsive forces. Any disturbance results in a discontinuous change of velocities of the system at the collision instant. We denote the right and left limits of velocities with respect to this instant with superscripts $(\cdot)^+$ and $(\cdot)^-$.

B.1 Foot touchdown

When modeling disturbance due to a foot touchdown we assume that

AS-B.1 the contacts never detach as a result of an impact;

AS-B.2 the tangential impulsive contact forces are not limited due to friction;

AS-B.3 mechanical constraints of the system can be safely ignored, so that modeling of impacts in the joints can be avoided.

Under these assumptions the impact law is defined as

$$\begin{cases} \boldsymbol{H}(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-) = \boldsymbol{J}_{\uparrow,p}^\top \boldsymbol{\rho}, & \text{(B.1a)} \\ \boldsymbol{J}_{\uparrow,i}\dot{\boldsymbol{q}}^+ = \boldsymbol{0}, \qquad i \in \{1, ..., M\}, & \text{(B.1b)} \\ \rho_i^n \geq 0, & \text{(B.1c)} \end{cases}$$

where

$$(\dot{\boldsymbol{p}}_1^\pm, \ldots, \dot{\boldsymbol{p}}_M^\pm) = \boldsymbol{J}_{\uparrow,p}\dot{\boldsymbol{q}}^\pm, \quad \text{(B.2a)}$$

$$(\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_M) = \boldsymbol{\rho}, \quad \text{(B.2b)}$$

$$\begin{bmatrix} \dot{\boldsymbol{p}}_i^{t,\pm} \\ \dot{p}_i^{n,\pm} \end{bmatrix} = {}^{\mathrm{i}}\boldsymbol{R}\dot{\boldsymbol{p}}_i^\pm, \quad \begin{bmatrix} \boldsymbol{\rho}_i^t \\ \rho_i^n \end{bmatrix} = {}^{\mathrm{i}}\boldsymbol{R}\boldsymbol{\rho}_i, \quad \text{(B.2c)}$$

$\boldsymbol{\rho}$ denotes impulsive contact forces, and other terms are defined in the same way as in Subsection 3.1.1.

We obtain post-impact generalized velocities $\dot{\boldsymbol{q}}^+$ by solving an optimization problem based on the impact law (B.1)

Hierarchy (B.1)

1: $\rho_i^n \geq 0$
$\boldsymbol{J}_{\uparrow,i}\dot{\boldsymbol{q}}^+ = \boldsymbol{0}$

2: $\boldsymbol{H}(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-) = \boldsymbol{J}_{\uparrow,p}^\top \boldsymbol{\rho}$

Decision variables: $\dot{\boldsymbol{q}}^+$, $\boldsymbol{\rho}$

Exact satisfaction of all objectives is not always possible. Hence, in order to prevent drift of the feet during simulations, we have chosen to assign higher priority to the objective, which prevents motion of the feet.

B.2 Push

We represent pushes of the robot by impulsive forces $\boldsymbol{\rho}_d$ applied at certain points on the robot body. For this reason, the impact law (B.1) is changed to:

$$\begin{cases} \boldsymbol{H}(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-) = \boldsymbol{J}_{\uparrow,p}^\top \boldsymbol{\rho} + \boldsymbol{J}_d^\top \boldsymbol{\rho}_d, & \text{(B.3a)} \\ \boldsymbol{J}_{\uparrow,i} \dot{\boldsymbol{q}}^+ = \boldsymbol{0}, \qquad i \in \{1, ..., M\}, & \text{(B.3b)} \\ \rho_i^n \geq 0. & \text{(B.3c)} \end{cases}$$

The generalized velocities $\dot{\boldsymbol{q}}^+$ after a push are found using appropriately modified Hierarchy B.1.

Appendix C

Joint limits

C.1 Bounds on the joint angular accelerations

Joint angles $\boldsymbol{q}'$ and angular velocities $\dot{\boldsymbol{q}}'$ are parts of the state of the whole body model considered in Section 3.1, while joint angular accelerations $\ddot{\boldsymbol{q}}'$ are controlled variables. When the whole body model is used for instantaneous control of the robot, it is not possible to constrain $\boldsymbol{q}'$ and $\dot{\boldsymbol{q}}'$ directly, but we can constrain $\boldsymbol{q}'_T$ and $\dot{\boldsymbol{q}}'_T$

$$\boldsymbol{q}'_T = \boldsymbol{q}' + T\dot{\boldsymbol{q}}' + \frac{T^2}{2}\ddot{\boldsymbol{q}}', \tag{C.1}$$

$$\dot{\boldsymbol{q}}'_T = \dot{\boldsymbol{q}}' + T\ddot{\boldsymbol{q}}', \tag{C.2}$$

which will be reached in time T starting from the current time instant assuming constant $\ddot{\boldsymbol{q}}'$ on $[0, T]$ [Rubrecht 2012, Saab 2013]. Moreover, the joint bounds may conflict with each other: when a joint hits its mechanical limit, there is no guarantee that bounds on angular velocities and accelerations are respected. We resolve the conflicts by introducing prioritization (see Chapter 5) of the bounds

Hierarchy (C.1)

1: $\underline{\boldsymbol{q}}' \leq \boldsymbol{q}'_T \leq \bar{\boldsymbol{q}}'$
2: $\underline{\dot{\boldsymbol{q}}}' \leq \dot{\boldsymbol{q}}'_T \leq \bar{\dot{\boldsymbol{q}}}'$
3: $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$

so that the position limits take over the velocity and acceleration limits.

All constraints in Hierarchy C.1 are expressed as bounds on $\ddot{\boldsymbol{q}}'$

Hierarchy (C.2)

1: $\underline{\boldsymbol{q}}'_p \leq \ddot{\boldsymbol{q}}' \leq \bar{\boldsymbol{q}}'_p$
2: $\underline{\boldsymbol{q}}'_v \leq \ddot{\boldsymbol{q}}' \leq \bar{\boldsymbol{q}}'_v$
3: $\underline{\ddot{\boldsymbol{q}}}' \leq \ddot{\boldsymbol{q}}' \leq \bar{\ddot{\boldsymbol{q}}}'$

where

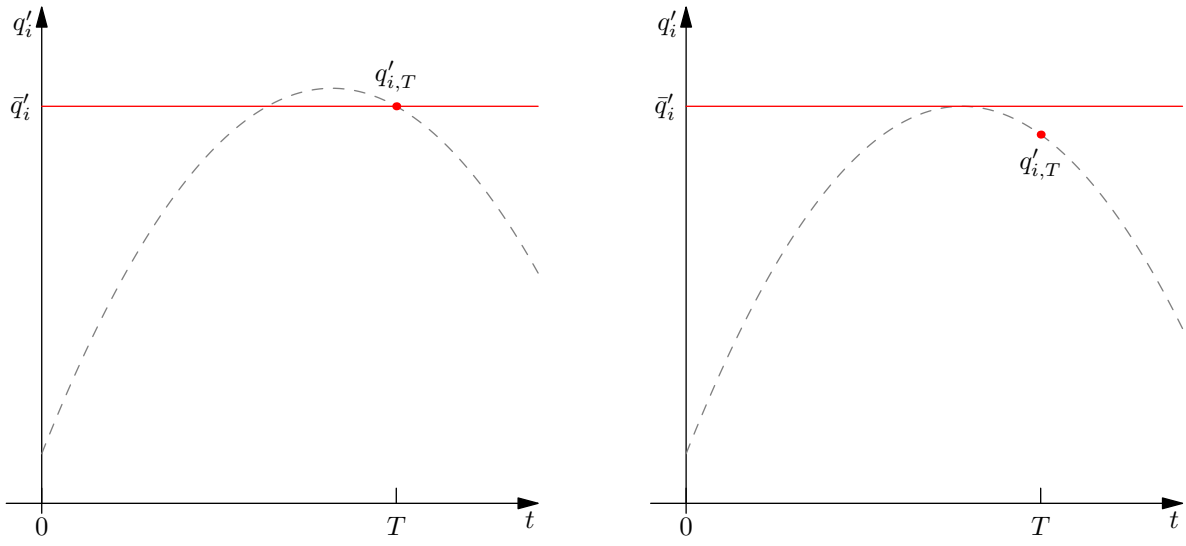
$$\underline{\boldsymbol{q}}'_p = \frac{2(\underline{\boldsymbol{q}}' - \boldsymbol{q}')}{T^2} - \frac{2\dot{\boldsymbol{q}}'}{T}, \quad \bar{\boldsymbol{q}}'_p = \frac{2(\bar{\boldsymbol{q}}' - \boldsymbol{q}')}{T^2} - \frac{2\dot{\boldsymbol{q}}'}{T}, \tag{C.3a}$$

$$\underline{\boldsymbol{q}}'_v = \frac{(\underline{\dot{\boldsymbol{q}}}' - \dot{\boldsymbol{q}}')}{T}, \quad \bar{\boldsymbol{q}}'_v = \frac{(\bar{\dot{\boldsymbol{q}}}' - \dot{\boldsymbol{q}}')}{T}, \tag{C.3b}$$

and division by T is safe, since it is always greater than zero. Constraints in Hierarchy C.2 are redundant and are preprocessed to collapse it to a single level and reduce the number of constraints 3 times.

Computation of $\underline{\boldsymbol{q}}'_p$ and $\bar{\boldsymbol{q}}'_p$ as suggested in (C.3a) is potentially unsafe and we construct them using a slightly different approach described in the following section.

C.2 Bounds on the joint angles



(a) Violation of the upper bound of i-th joint angle. Solid red line and dashed grey curve indicate the bound the joint angle trajectory.

(b) All values of i-th joint angle in the time interval $[0, T]$ lie below the upper bound.

Figure C.1: Respecting the upper bound of the i-th joint angle.

Note that the bounds (C.3a) are quadratic in $1/T$, which means that satisfaction of $\underline{\boldsymbol{q}}' \leq \boldsymbol{q}'_T \leq \bar{\boldsymbol{q}}'$ does not imply satisfaction $\underline{\boldsymbol{q}}' \leq \boldsymbol{q}'_t \leq \bar{\boldsymbol{q}}'$ for $t \in (0, T)$ as illustrated in Figure C.1a. In practice, imposing only $\underline{\boldsymbol{q}}' \leq \boldsymbol{q}'_T \leq \bar{\boldsymbol{q}}'$ results in violations of the joint bounds in the model and collisions in the joints of the robot. However, this issue can be avoided by exploiting the quadratic nature of the constraint.

Let us substitute variable $\nu = \frac{1}{t}$ with $t \in (0, T]$ into the upper bound on acceleration of i-th joint $\ddot{q}'_i$

$$\bar{q}'_{p,i} = 2\big(\bar{q}'_i - q'_i\big)\nu^2 - 2\dot{q}'_i\nu. \tag{C.4}$$

Then in order to make sure that the joint angle constraint is not violated for all $t \in (0, T]$ as shown in Figure C.1b, we have to make sure that $\ddot{q}'_i$ does not exceed the minimal value of (C.4). The minimal value is obtained by solving the following *Quadratic Program* (QP)

$$\begin{aligned} \underset{\nu}{\text{minimize}} \quad & 2\big(\bar{q}'_i - q'_i\big)\nu^2 - 2\dot{q}'_i\nu \\ \text{subject to} \quad & \nu \geq \frac{1}{T}. \end{aligned} \tag{C.5}$$

Solution $\nu^\star$ of this QP is unbounded in two cases

- when $\bar{q}'_i < q'_i$, *i.e.*, the bound is already violated;
- when $\bar{q}'_i = q'_i$ and $\dot{q}'_i > 0$, *i.e.*, the bound is reached with non-zero velocity.

The first case corresponds to a situation when there is a mismatch between the state of the model and the robot, since the physical constraint cannot be violated. In the second case there is an inevitable collision in the joint. Such collisions can be simulated using an impact law similar to the one described in Appendix B, but validity of such simulation is questionable without an accurate joint model. Thus, in both cases there is no $\ddot{q}'_i$, which prevents violation

of the bound or collision with it. One possible approach to recover from such situations is to assume $\nu^\star = \frac{1}{T}$. Neither of these cases, however, was observed in our simulations.

When the solution $\nu^\star$ is bounded, it coincides with extremum of the objective function or the bound $1/T$:

$$\nu^\star = \begin{cases} \dfrac{\dot{q}'_i}{2(\bar{q}'_i - q'_i)}, & \text{if} \quad \bar{q}'_i > q'_i \quad \text{and} \quad \dfrac{\dot{q}'_i}{2(\bar{q}'_i - q'_i)} \geq \dfrac{1}{T} \\ \dfrac{1}{T}, & \text{otherwise.} \end{cases} \tag{C.6}$$

The upper bound on $\ddot{q}'_i$ is obtained by substitution of $\nu^\star$ in the objective function and is equal to

$$\bar{\ddot{q}}'_{p,i} = \begin{cases} -\dfrac{(\dot{q}'_i)^2}{2(\bar{q}'_i - q'_i)}, & \text{if} \quad \bar{q}'_i > q'_i \quad \text{and} \quad \dfrac{\dot{q}'_i}{2(\bar{q}'_i - q'_i)} > \dfrac{1}{T} \\ \dfrac{2(\bar{q}'_i - q'_i)}{T^2} - \dfrac{2\dot{q}'_i}{T}, & \text{otherwise.} \end{cases} \tag{C.7}$$

The lower bound on acceleration is determined equivalently by solving a maximization problem. There exist a possibility of conflict between the lower and upper bounds, *i.e.*, $\bar{\ddot{q}}'_{p,i} < \underline{\ddot{q}}'_{p,i}$, if the joint angular velocity is high, but we did not observe such situations in our simulations.

C.3 Imposing position constraints through accelerations

So far we were considering bounds on the joint angles, but sometimes general constraints on position such as

$$\underline{b}_{\boldsymbol{y}} \leq \boldsymbol{A}\boldsymbol{y} \leq \bar{b}_{\boldsymbol{y}} \tag{C.8}$$

with $\boldsymbol{y} \in \mathbb{R}^p$ and $\boldsymbol{A} \in \mathbb{R}^{1\times p}$ must also be ensured using constraints on acceleration

$$\underline{b}_{\ddot{\boldsymbol{y}}} \leq \boldsymbol{A}\ddot{\boldsymbol{y}} \leq \bar{b}_{\ddot{\boldsymbol{y}}}. \tag{C.9}$$

For example, we might be interested in constraining projection of the *Center of Mass* (CoM) position to support area using acceleration of the CoM. In this case, the upper bound $\bar{b}_{\ddot{\boldsymbol{y}}}$ is

$$\bar{b}_{\ddot{\boldsymbol{y}}} = \begin{cases} -\dfrac{(\boldsymbol{A}\dot{\boldsymbol{y}})^2}{2\big(\bar{b}_{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{y}\big)} & \text{if} \quad \underline{b}_{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{y} > 0 \quad \text{and} \quad \dfrac{\boldsymbol{A}\dot{\boldsymbol{y}}}{2\big(\bar{b}_{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{y}\big)} > \dfrac{1}{T} \\ \dfrac{2\big(\bar{b}_{\boldsymbol{y}} - \boldsymbol{A}\boldsymbol{y}\big)}{T^2} - \dfrac{2\boldsymbol{A}\dot{\boldsymbol{y}}}{T}, & \text{otherwise.} \end{cases} \tag{C.10}$$

$\underline{b}_{\ddot{\boldsymbol{y}}}$ is computed in a similar way.

Appendix D

Constraints of surface contacts

D.1 Rectangular foot contact

Foot contact with the ground is the most common type of contact realized by humanoid robots. A foot typically consists of one or two, if the robot has toes, rigid bodies with flat soles. Contact of a sole with the ground is often characterized by 3-dimensional contact forces applied at N vertices of the support polygon as shown in Figure D.1a [Kuindersma 2014, Abe 2007, Ott 2011, Hauser 2008]. In this case, the interaction with the support surface is described by $3N$ force variables, which are redundant, since the number of degrees of freedom of a rigid body does not exceed 6. The redundancy leads to excessive number of constraints and variables, which have negative impact on computational performance and may cause numerical problems in solvers. For this reason, here we consider a more concise representation of interaction between a rectangular sole and support by a single wrench. The wrench is computed with respect to the origin of a frame fixed to the middle of the sole as illustrated in Figure D.1b. All variables used below in the derivations are expressed in the said frame, therefore, appropriate transformations must be performed to switch to the global frame.

Let us integrate over the support area $\mathbb{S}$ to obtain the total contact force [Howe 1996]

$$\begin{bmatrix} f^x \\ f^y \end{bmatrix} = \int_{\mathbb{S}} \tilde{\kappa}(x,y)\boldsymbol{v}(x,y)\, df^z(x,y), \tag{D.1a}$$

$$f^z = \int_{\mathbb{S}} df^z(x,y), \tag{D.1b}$$

where $\boldsymbol{v}(x,y)$ is a unit vector of the tangential contact force at a given point, $\tilde{\kappa}(x,y) \in [0,\kappa]$ relates the norms of the tangential and the normal forces, and integration is performed with respect to $f^z(x,y) \geq 0$ to account for potential discontinuities in pressure distribution. Norm of the tangential contact force reaches maximal value when tangential forces at each point of $\mathbb{S}$ are collinear $\boldsymbol{v}(x,y) = \boldsymbol{v}$ and parameter $\tilde{\kappa}(x,y)$ is equal to the friction coefficient κ, hence

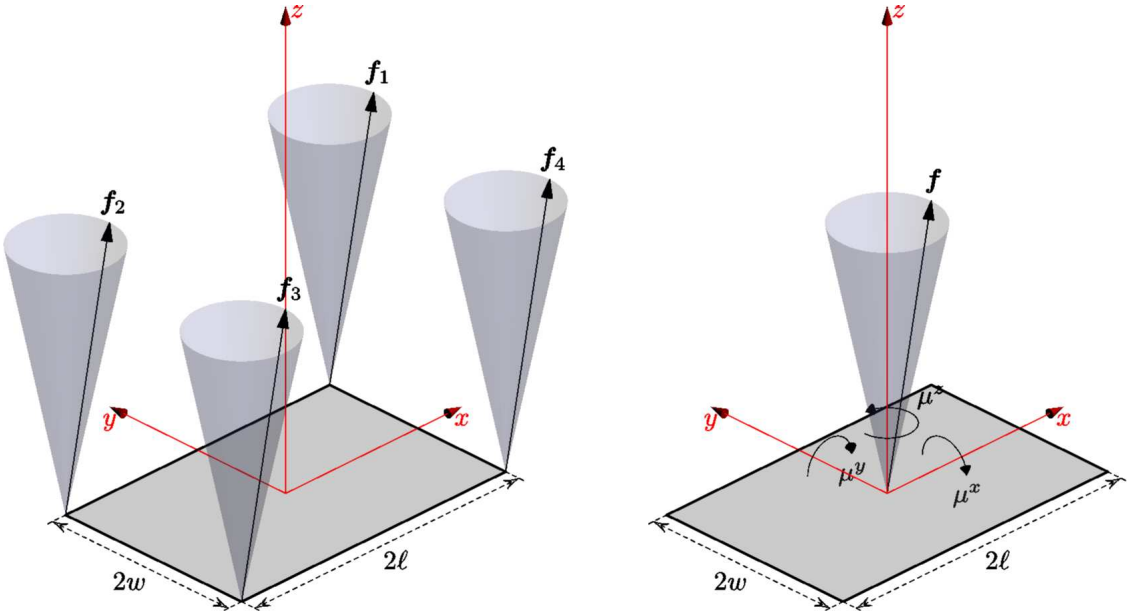
$$\boldsymbol{f}^{xy}_{max} = \int_{\mathbb{S}} \kappa \boldsymbol{v}\, df^z(x,y) = \kappa f^z \boldsymbol{v}. \tag{D.2}$$

This implies that the total tangential contact force is limited to the friction cone, which can be linearized as described in Subsection 3.1.3.

We continue by obtaining the total contact moment

$$\begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix} = \int_{\mathbb{S}} \begin{bmatrix} y \\ -x \end{bmatrix} df^z(x,y), \tag{D.3a}$$

$$\mu^z = \int_{\mathbb{S}} \tilde{\kappa}(x,y) \begin{bmatrix} -y & x \end{bmatrix} \boldsymbol{v}(x,y)\, df^z(x,y). \tag{D.3b}$$



(a) Contact forces are comprised of four 3-dimensional force vectors $(\boldsymbol{f}_1, ..., \boldsymbol{f}_4)$ applied at the corners of the support area.

(b) Contact forces are comprised of a single 6-dimensional wrench $(\boldsymbol{f}, \boldsymbol{\mu})$ applied at the center of the support area.

Figure D.1: Representation of foot contact forces. Grey rectangles are rectangular foot support areas of length 2ℓ and width $2w$, cones are Coulomb's friction constraints.

First we focus on Equation (D.3a). It is straightforward that the moments about the x and y axes take their maximal values, when the normal contact force is applied at the respective edges of the support area, *i.e.*, when $x = \pm\ell$ or $y = \pm w$, consequently,

$$-\begin{bmatrix} w \\ \ell \end{bmatrix} f^z \leq \begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix} \leq \begin{bmatrix} w \\ \ell \end{bmatrix} f^z. \tag{D.4}$$

Significance of these inequalities becomes clear after division of (D.3a) by (D.1b):

$$\frac{1}{f^z}\begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix} = \frac{\int_{\mathbb{S}} \begin{bmatrix} y \\ -x \end{bmatrix} df^z(x,y)}{\int_{\mathbb{S}} df^z(x,y)} = \begin{bmatrix} z^y \\ -z^x \end{bmatrix}, \tag{D.5}$$

where $\boldsymbol{z} = (z^x, z^y) = (-\mu^y, \mu^x)/f^z$ is position of *Center of Pressure* (CoP) of the contact pressure field. Thus, constraints (D.4) imply that $\boldsymbol{z} \in \mathbb{S}$, but they also remain valid when CoP is not defined, *i.e.*, when $f^z = 0$. Furthermore, (D.4) can be generalized to arbitrary convex polygonal support areas, when the constraints take the following form [Nagasaka 2012]

$$a\mu^x + b\mu^y \leq cf^z, \tag{D.6}$$

where a, b, c are parameters of a line defining a face of the support polygon.

Analysis of Equation (D.3b) is more intricate. In order to illustrate this let us assume that $\tilde{\kappa}(x,y) = \tilde{\kappa}$ and $\boldsymbol{v}(x,y) = \boldsymbol{v}$. Then

$$\mu^z = \int_{\mathbb{S}} \tilde{\kappa}\begin{bmatrix} -y & x \end{bmatrix} \boldsymbol{v}\, df^z(x,y) = \tilde{\kappa}\boldsymbol{v}^\top \int_{\mathbb{S}} \begin{bmatrix} -y \\ x \end{bmatrix} df^z(x,y) = -\tilde{\kappa}\boldsymbol{v}^\top \begin{bmatrix} \mu^x \\ \mu^y \end{bmatrix}, \tag{D.7}$$

which can be multiplied by f^z to obtain

$$\mu^z f^z = -f^x\mu^x - f^y\mu^y. \tag{D.8}$$

Thus, under the assumption that all tangential forces are collinear, the moment about the z axis is nonlinearly determined by other components of the wrench. In fact, under this assumption the total contact moment is computed by the cross product of the CoP position and the total contact force. In general, however, we cannot make such assumption and should avoid nonlinear constraints. Instead, we compute the absolute minimum and maximum of μ^z, which are achieved when the origin of the frame is the center of rotation

$$\boldsymbol{v} = \pm \frac{1}{\sqrt{x^2+y^2}} \begin{bmatrix} -y \\ x \end{bmatrix} \tag{D.9}$$

and $\tilde{\kappa}(x,y) = \kappa$. Hence, the bounds on μ^z are

$$\underline{\mu}^z = -\kappa \int_{\mathbb{S}} \sqrt{x^2+y^2}\, df^z(x,y), \quad \bar{\mu}^z = \kappa \int_{\mathbb{S}} \sqrt{x^2+y^2}\, df^z(x,y). \tag{D.10}$$

They reach the minimal (maximal) values when the normal contact force is concentrated in the corners of $\boldsymbol{S}$ with coordinates $(\pm\ell, \pm w)$, so that

$$\underline{\mu}^z = -\kappa\sqrt{\ell^2+w^2} f^z, \quad \bar{\mu}^z = \kappa\sqrt{\ell^2+w^2} f^z. \tag{D.11}$$

Note that these bounds must be tighter if any of the following holds true:

- $\tilde{\kappa}(x,y) < \kappa$;
- the center of rotation does not coincide with the center of the foot;
- the pressure is not concentrated in the corners of $\mathbb{S}$.

However, constraints (D.11) have found their application in contact modeling due to their simplicity and linearity [Fujimoto 1996, Bouchard 2015]. Constraints of a similar form

$$-\tilde{\kappa} f^z \le \mu^z \le \tilde{\kappa} f^z, \tag{D.12}$$

where $\tilde{\kappa}$ is a certain torsional friction coefficient, are also applied [Mansour 2013, Nagasaka 2012], especially in the works using the so called *soft finger* contact model for grasping [Murray 1994, Chapter 5]. Other approaches to limitation of μ^z include

- Complete omission of the bounds [Stephens 2010, Audren 2014, Henze 2014, Herzog 2015]. This, however, is more dangerous than imposing (D.11).
- Construction of more elaborate bounds depending on $\boldsymbol{\mu}^{xy}$ and $\boldsymbol{f}^{xy}$ [Caron 2015]. Unfortunately, obscurity of the derivations makes it difficult to analyze implications of these constraints.
- Derivation of the bounds based on certain assumptions about distribution of the contact pressure [Zhu 2006, Zhou 2013].
- Use nonlinear bounds with some additional assumptions [Yamane 2004, Chapter 5].

We draw two conclusions from the present discussion: the total tangential contact force is subject to Coulomb's friction constraints in the same way as in the case of a point contact; the contact moments can and should be bounded using (D.4) and (D.11), which are summarized as

$$f^n \underbrace{\begin{bmatrix} -w \\ -\ell \\ -\sqrt{\ell^2+w^2} \end{bmatrix}}_{\underline{\boldsymbol{\mu}}} \le \boldsymbol{\mu} \le f^n \underbrace{\begin{bmatrix} w \\ \ell \\ \sqrt{\ell^2+w^2} \end{bmatrix}}_{\bar{\boldsymbol{\mu}}}. \tag{D.13}$$

D.2 General surface contacts

Surface contacts are not necessarily rectangular: multiple contacts with the same surface may lead to a support area of a rather complex shape. For example, when the robot walks on a flat ground, it may have two foot contacts with the ground at the same time. The support area in this case is the convex hull $\mathbb{S}(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$ of two rectangular feet with positions $\boldsymbol{p}_1^{xy}$ and $\boldsymbol{p}_2^{xy}$ (see Figure D.2). Walking with coplanar contacts is so common that some approximate models were specifically designed for this task, in particular, the point-mass models considered in Subsections 3.3.2 and 3.3.3. For this reason, we discuss general surface contacts in more detail here.

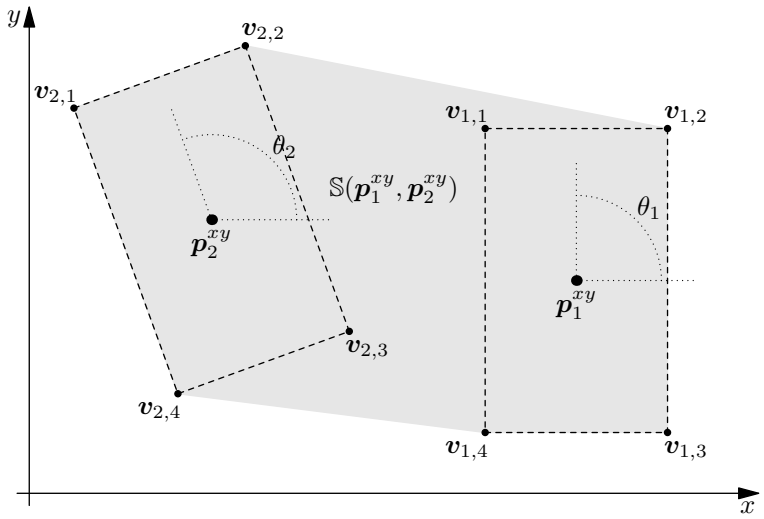


Figure D.2: Grey area represents support area $\mathbb{S}(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$ of two rectangular foot contacts.

From the preceding section it is clear that bounding of the tangential contact forces is easy (see also Subsection 3.2.2). At the same time, derivation of the bounds on the contact moment about the z axis is nontrivial even in the case of a simple rectangular shape. Therefore, we entirely focus on the constraints on moments about the x and y axes. Provided that the normal force is nonzero, these constraints are equivalent to constraints on the CoP position: $\boldsymbol{z} \in \mathbb{S}(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$. One way to express these constraints is to find faces of the support polygon and force $\boldsymbol{z}$ to stay on their inner side with inequality constraints. These inequalities, however, are difficult to parametrize with respect to positions $(\boldsymbol{p}_1^{xy}, \boldsymbol{p}_2^{xy})$ and orientations of the feet (θ_1, θ_2). Therefore, we consider constraints on the CoP using a different approach: we exploit the fact that as long as the CoP lies within the support area it is a convex combination of vertices $\boldsymbol{v}_{i,j} \in \mathbb{R}^2$ with $i \in \{1, 2\}$ and $j \in \{1, 2, 3, 4\}$ of the rectangular foot contacts shown in Figure D.2.

Since the CoP is a convex combination of vertices $\boldsymbol{v}_{i,j}$, we express it as

$$\boldsymbol{z} = \sum_{j=1}^{4} \sum_{i=1}^{2} \eta_{i,j} \boldsymbol{v}_{i,j}\,, \quad \sum_{j=1}^{4} \sum_{i=1}^{2} \eta_{i,j} = 1, \quad \eta_{i,j} \geq 0. \tag{D.14}$$

Let us denote positions of vertices of a given foot expressed in a frame fixed to its center as

$\boldsymbol{v}_{\{1,2,3,4\}}$. Then $\boldsymbol{v}_{i,j} = \hat{\boldsymbol{R}}_{\mathsf{i}}\boldsymbol{v}_j + \boldsymbol{p}_i^{xy}$, where $\hat{\boldsymbol{R}}_{\mathsf{i}}$ is a rotation matrix depending on θ_i. Hence,

$$\begin{aligned} \boldsymbol{z} &= \sum_{j=1}^{4}\left(\left(\sum_{i=1}^{2}\eta_{i,j}\hat{\boldsymbol{R}}_{\mathsf{i}}\right)\boldsymbol{v}_j\right) + \left(\sum_{j=1}^{4}\eta_{1,j}\right)\hat{\boldsymbol{R}}_1\boldsymbol{p}_1^{xy} + \left(\sum_{j=1}^{4}\eta_{2,j}\right)\hat{\boldsymbol{R}}_2\boldsymbol{p}_2^{xy}, \\ \sum_{j=1}^{4}\sum_{i=1}^{2}\eta_{i,j} &= 1, \quad \eta_{i,j} \geq 0. \end{aligned} \tag{D.15}$$

Since $\boldsymbol{v}_j$ are constant, the CoP position is determined by coefficients $\eta_{i,j}$, foot positions $\boldsymbol{p}_i^{xy}$, and foot orientations θ_i. This relation is linear in several cases:

- Positions and orientations of the feet are fixed. Then variables $\eta_{i,j}$ determine position of the CoP.
- Orientations of the feet are fixed and distribution of the pressure between the feet is known, *i.e.*,

$$\sum_{j=1}^{4}\eta_{1,j} = const, \quad \sum_{j=1}^{4}\eta_{2,j} = const. \tag{D.16}$$

  In this case positions of the feet and, to some extent, position of the CoP can vary.
- There is only one contact with the surface and orientation of the foot is fixed. Then, in addition to position of the CoP within the foot, the position of the foot can vary. This allows to anticipate walking motions without predetermined contact positions [Herdt 2010, Sherikov 2014].

Though nonlinear *Model Predictive Control* (MPC) allows to handle nonlinear constraints on the CoP position [Naveau 2016], in practice it is common to employ approximations, for example,

- Support area in a double support can be approximated using support area of a single support, which is translated and oriented appropriately as demonstrated in Figure D.3 [Dimitrov 2011a], [Sherikov 2012, Chapter 3].

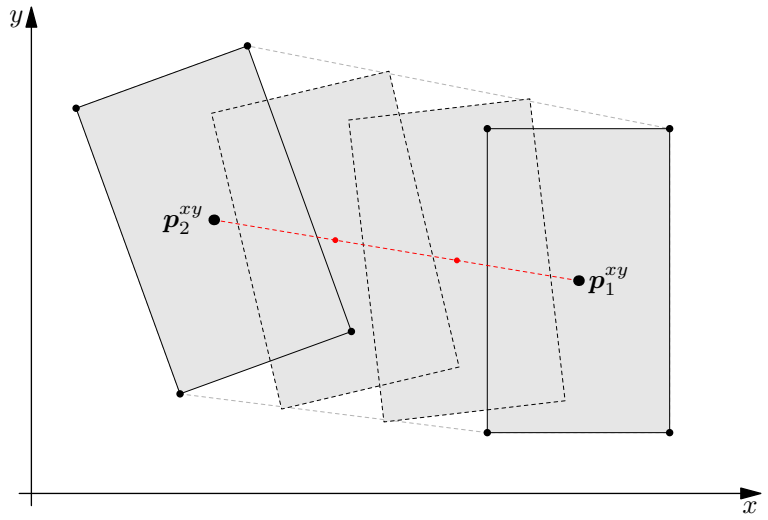


Figure D.3: Approximation of the support area in a double support with support area of a single support.

- The support area does not depend on foot orientations in the case of point or round feet. Therefore, we can constrain the CoP positions to conservative round support

areas as illustrated in Figure D.4 [Lafaye 2014]. This guarantees that the CoP position always stays within the real support area. Alternatively, orientations of the feet can be precomputed using a dedicated MPC scheme as proposed in [Herdt 2012, Chapter 2],

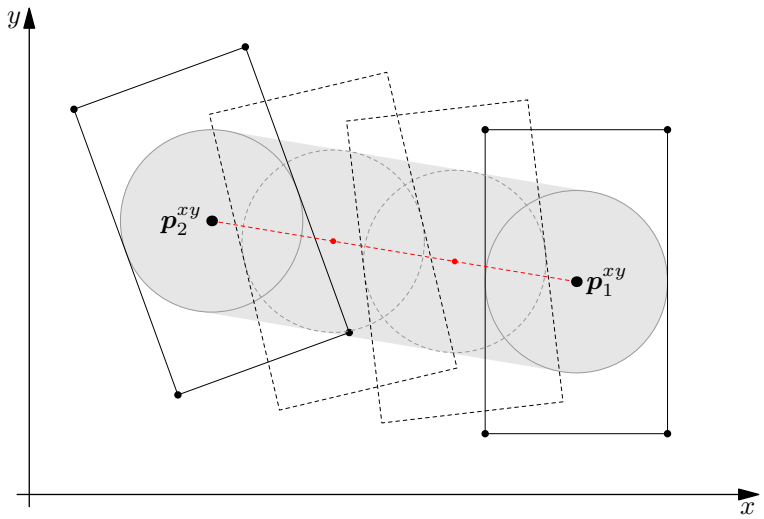


Figure D.4: Grey area demonstrates support area, which is independent of foot orientations.

Appendix E

Condensing of a Model Predictive Control problem

In this appendix we illustrate the condensing procedure employed in the *Model Predictive Control* (MPC) field in order to eliminate some of the decision variables [Bock 1984].

Consider a linear model

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k \in \{0, ..., N-1\} \tag{E.1}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ are k-th state and control vectors respectively, and N is the length of preview horizon.

Condensing amounts to finding such matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_u$ that

$$\hat{\boldsymbol{x}} = \boldsymbol{U}_x \boldsymbol{x}_0 + \boldsymbol{U}_u \hat{\boldsymbol{u}}, \tag{E.2}$$

where

$$\hat{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N), \quad \hat{\boldsymbol{u}} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}). \tag{E.3}$$

Matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_u$ are obtained by evaluating equation of the dynamics recursively

$$\begin{aligned}
&\boldsymbol{x}_1 = \boldsymbol{A}_0 \boldsymbol{x}_0 + \boldsymbol{B}_0 \boldsymbol{u}_0, \\
&\boldsymbol{x}_2 = \boldsymbol{A}_1 \boldsymbol{x}_1 + \boldsymbol{B}_1 \boldsymbol{u}_1 = \boldsymbol{A}_1(\boldsymbol{A}_0 \boldsymbol{x}_0 + \boldsymbol{B}_0 \boldsymbol{u}_0) + \boldsymbol{B}_1 \boldsymbol{u}_1 = \boldsymbol{A}_1 \boldsymbol{A}_0 \boldsymbol{x}_0 + \begin{bmatrix} \boldsymbol{A}_1 \boldsymbol{B}_0 & \boldsymbol{B}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \end{bmatrix}, \\
&\ldots,
\end{aligned} \tag{E.4}$$

which results in

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}_1 \boldsymbol{A}_0 \\ \vdots \\ \boldsymbol{A}_{N-1} \ldots \boldsymbol{A}_0 \end{bmatrix}, \quad \boldsymbol{U}_u = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{A}_1 \boldsymbol{B}_0 & \boldsymbol{B}_1 & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{N-1} \ldots \boldsymbol{A}_1 \boldsymbol{B}_0 & \boldsymbol{A}_{N-1} \ldots \boldsymbol{A}_2 \boldsymbol{B}_1 & \ldots & \boldsymbol{B}_{N-1} \end{bmatrix}. \tag{E.5}$$

In the case of time invariant system, matrices take a simpler form

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix}, \quad \boldsymbol{U}_u = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \ldots & \boldsymbol{B} \end{bmatrix}, \tag{E.6}$$

After condensing, all state $\hat{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ variables are expressed through the initial state $\boldsymbol{x}_0$ and control variables $\hat{\boldsymbol{u}} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1})$. Therefore, $\hat{\boldsymbol{x}}$ and can be eliminated from constraints and objectives of the MPC problem.

Appendix F

Trajectory of a foot in the air

Trajectory of a foot in the air (*swing foot*) is generated using cubic polynomial of the form

$$a_\alpha t^3 + b_\alpha t^2 + c_\alpha t + d_\alpha = s_t^\alpha, \tag{F.1}$$

where $\alpha \in \{x, y, z\}$ denotes coordinate axis, t is time instant, $a_\alpha, b_\alpha, c_\alpha, d_\alpha$ are coefficients, and s_t^α is position of the swing foot at time t. Derivatives of the cubic polynomial are

$$\begin{aligned} 3a_\alpha t^2 + 2b_\alpha t + c_\alpha &= \dot{s}_t^\alpha, \\ 6a_\alpha t + 2b_\alpha &= \ddot{s}_t^\alpha, \\ 6a_\alpha &= \dddot{s}_t^\alpha. \end{aligned} \tag{F.2}$$

There are four boundary conditions for the polynomial, the first two are defined for the current swing foot state at $t = 0$:

$$\begin{aligned} d_\alpha &= s_0^\alpha, \\ c_\alpha &= \dot{s}_0^\alpha, \end{aligned} \tag{F.3}$$

where s_0^α and $\dot{s}_0^\alpha$ are initial position and velocity. The remaining two conditions for landing time instant $t = l$ are:

$$\begin{aligned} a_\alpha l^3 + b_\alpha l^2 + \dot{s}_0^\alpha l + s_0^\alpha &= s_l^\alpha, \\ 3a_\alpha l^2 + 2b_\alpha l + \dot{s}_0^\alpha &= \dot{s}_l^\alpha. \end{aligned} \tag{F.4}$$

Final position s_l^z for trajectory along the z axis is set to the step height during the first half of the step and to 0 during the second half:

$$s_l^z = \begin{cases} h & t \leq \frac{T_{step}}{2}; \\ 0 & t > \frac{T_{step}}{2}. \end{cases} \tag{F.5}$$

The final x, y positions are set to the next landing position (see the PPMZ model)

$$\begin{bmatrix} s_l^x \\ s_l^y \end{bmatrix} = \hat{\boldsymbol{p}}_1 = \hat{\boldsymbol{p}}_0 + \hat{\boldsymbol{B}}_0 \Delta\hat{\boldsymbol{p}}_0. \tag{F.6}$$

Velocity at the end of trajectory is set to zero $\dot{s}_l^\alpha = 0$.

F.1 Computation of the foot acceleration

Whole body control requires current acceleration (at time 0) of the swing foot, which can be found as

$$2b_\alpha = \ddot{s}_0^\alpha. \tag{F.7}$$

Hence it is necessary to find coefficients b_α from equations (F.3) and (F.4).

Trivial algebraic operations on (F.4) and (F.7) lead to the following equation:

$$\ddot{s}_0^\alpha = \frac{6(s_l^\alpha - s_0^\alpha)}{l^2} - \frac{4\dot{s}_0^\alpha}{l} = \frac{6}{l^2} s_l^\alpha - \frac{6}{l^2} s_0^\alpha - \frac{4}{l}\dot{s}_0^\alpha. \tag{F.8}$$

Consequently,

$$\ddot{\boldsymbol{s}}_0 = \begin{bmatrix} \ddot{\boldsymbol{s}}_0^{xy} \\ \ddot{s}_0^z \end{bmatrix} = \underbrace{\frac{6}{l^2}\begin{bmatrix} \hat{\boldsymbol{B}}_0 \\ 0 \end{bmatrix}}_{\boldsymbol{\mathcal{A}}_{sa}} \Delta\hat{\boldsymbol{p}}_0 + \underbrace{\frac{6}{l^2}\begin{bmatrix} \hat{\boldsymbol{p}}_0 \\ s_l^z \end{bmatrix} - \frac{6}{l^2}\boldsymbol{s}_0 - \frac{4}{l}\dot{\boldsymbol{s}}_0}_{\boldsymbol{b}_{sa}}, \tag{F.9}$$

F.2 Computation of the foot jerk

The footstep adjustments are allowed during the whole duration of a step, which may result in violent position changes near the end of a step. A straightforward workaround is to block footstep adaptation when the foot approaches the ground. Instead, we have chosen to minimize the x, y components of the foot jerk.

Using (F.4) and (F.2) we find the jerk as

$$\dddot{s}^\alpha = -\frac{12}{l^3} s_l^\alpha + \frac{12}{l^3} s_0^\alpha + \frac{6}{l}\dot{s}_0^\alpha. \tag{F.10}$$

Therefore, the x, y components of the foot jerk are expressed as

$$\dddot{\boldsymbol{s}}^{xy} = \underbrace{-\frac{12}{l^3}\hat{\boldsymbol{B}}_0}_{\boldsymbol{\mathcal{A}}_{sj}} \Delta\hat{\boldsymbol{p}}_0 + \underbrace{-\frac{12}{l^3}\hat{\boldsymbol{p}}_0 + \frac{12}{l^3} s_0^{xy} + \frac{6}{l^2}\dot{s}_0^{xy}}_{\boldsymbol{b}_{sj}}. \tag{F.11}$$

Appendix G

Solving a hierarchy with regularization

Consider a simple hierarchy

Hierarchy (G.1)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2$
3: $\boldsymbol{\mathcal{A}}_3\boldsymbol{\chi} = \boldsymbol{b}_3$

which must be solved with regularization of the objective on the second level using the equality objective on the last level. We can solve this hierarchy in two steps using a solver, which does not support regularization.

The first step is to solve the following hierarchy consisting of two levels

Hierarchy (G.2)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2$
$\gamma_r\boldsymbol{\mathcal{A}}_3\boldsymbol{\chi} = \gamma_r\boldsymbol{b}_3$

where γ_r is regularization factor. Solution of Hierarchy G.2 yields the vector of violations $\boldsymbol{v}_2^\star$ of the second objective:

$$\underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} - \boldsymbol{v}_2^\star \leq \bar{\boldsymbol{b}}_2. \tag{G.1}$$

We decompose $\boldsymbol{v}_2^\star$ into three components

$$\boldsymbol{v}_2^\star = \underline{\boldsymbol{v}}_2^\star + \tilde{\boldsymbol{v}}_2^\star + \bar{\boldsymbol{v}}_2^\star \tag{G.2}$$

Vector $\underline{\boldsymbol{v}}_2^\star$ contains violations of the lower bounds and zeros for the constraints, where lower bounds are satisfied. Vectors $\tilde{\boldsymbol{v}}_2^\star$ and $\bar{\boldsymbol{v}}_2^\star$ are formed in the same way for equality and inequality constraints.

The second step is to solve a hierarchy with relaxed bounds on the second level:

Hierarchy (G.3)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\underline{\boldsymbol{b}}_2 + \underline{\boldsymbol{v}}_2^\star + \tilde{\boldsymbol{v}}_2^\star \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2 + \bar{\boldsymbol{v}}_2^\star + \tilde{\boldsymbol{v}}_2^\star$
3: $\boldsymbol{\mathcal{A}}_3\boldsymbol{\chi} = \boldsymbol{b}_3$

The procedure is easily generalized to a larger number of levels with and without regularization. The number of hierarchies, which must be solved, is equal to the number of regularized levels plus one.

Bibliography

[Abe 2007] Y. Abe, M. da Silva and J. Popović. *Multiobjective Control with Frictional Contacts*. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '07, pages 249–258, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. (Cited on page 92.)

[ADE 2016] *Omron Adept Technologies, Inc.* Available: http://www.adept.com/, May 2016. (Cited on pages 77 and 78.)

[Agravante 2015] D. J. Agravante. *Human-humanoid collaborative object transportation*. PhD thesis, University of Montpellier, 2015. (Cited on page 79.)

[Agravante 2016a] D. J. Agravante, A. Cherubini, A. Sherikov, P.-B. Wieber and A. Kheddar. Human-Humanoid Collaborative Carrying. working paper or preprint, 2016. (Cited on pages 56, 77, and 79.)

[Agravante 2016b] D. J. Agravante, A. Sherikov, P.-B. Wieber, A. Cherubini and A. Kheddar. *Walking pattern generators designed for physical collaboration*. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, May 2016. (Cited on pages 2, 20, 23, 24, 25, 28, and 79.)

[Aubin 2011] J.-P. Aubin, A. Bayen and P. Saint-Pierre. Viability Theory: New Directions. Springer, 2011. (Cited on page 5.)

[Audren 2014] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko and E. Yoshida. *Model preview control in multi-contact motion-application to a humanoid robot*. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 4030–4035, Sept 2014. (Cited on pages 9, 22, 23, 33, 43, 76, and 94.)

[Azevedo 2002] C. Azevedo, P. Poignet and B. Espiau. *Moving horizon control for biped robots without reference trajectory*. In Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, volume 3, pages 2762–2767, 2002. (Cited on page 10.)

[Bao-Cang 2010] D. Bao-Cang. Modern predictive control. CRC Press, 2010. (Cited on page 30.)

[Bemporad 2000] A. Bemporad, G. Ferrari-Trecate and M. Morari. *Observability and controllability of piecewise affine and hybrid systems*. Automatic Control, IEEE Transactions on, vol. 45, no. 10, pages 1864–1876, Oct 2000. (Cited on page 7.)

[Bicchi 2008] A. Bicchi, M. A. Peshkin and J. E. Colgate. *Safety for Physical Human–Robot Interaction*. In B. Siciliano and O. Khatib, editors, Springer Handbook of Robotics, pages 1335–1348. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. (Cited on page 77.)

[Blanchini 2008] F. Blanchini and S. Miani. Set-theoretic methods in control. Springer International Publishing, 2008. (Cited on page 7.)

[Bock 1984] H. G. Bock and K. J. Plitt. *A multiple shooting algorithm for direct solution of optimal control problems.* In Proceedings 9th IFAC World Congress Budapest, pages 243–247. Pergamon Press, 1984. (Cited on pages 49, 54, and 98.)

[Borrelli 2015] F. Borrelli, A. Bemporad and M. M. Predictive control for linear and hybrid systems. 2015. (Cited on page 7.)

[Bouchard 2015] C. Bouchard, M. Nesme, M. Tournier, B. Wang, F. Faure and P. G. Kry. *6D Frictional Contact for Rigid Bodies.* In Proceedings of the 41st Graphics Interface Conference, GI '15, pages 105–114, Toronto, Ont., Canada, Canada, 2015. Canadian Information Processing Society. (Cited on page 94.)

[Bouyarmane 2012] K. Bouyarmane and A. Kheddar. *Humanoid Robot Locomotion and Manipulation Step Planning.* Advanced Robotics, vol. 26, no. 10, pages 1099–1126, 2012. (Cited on page 9.)

[Boyd 2004] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge University Press, New York, NY, USA, 2004. (Cited on pages 29 and 46.)

[Bramley 1994] R. Bramley and B. Winnicka. *Solving Linear Inequalities In A Least Squares Sense.* SIAM J. Sci. Comp, vol. 17, pages 17–275, 1994. (Cited on page 45.)

[Brasseur 2015a] C. Brasseur. Robust predictive control for a humanoid robot walking on stairs. Master's thesis, Joseph Fourier University, 2015. (Cited on page 76.)

[Brasseur 2015b] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov and P.-B. Wieber. *A robust linear MPC approach to online generation of 3D biped walking motion.* In Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on, Nov 2015. (Cited on pages 2, 26, 27, 28, 33, 56, 76, and 81.)

[Bretl 2006] T. Bretl. *Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem.* Int. J. Rob. Res., vol. 25, no. 4, pages 317–342, April 2006. (Cited on page 9.)

[Bretl 2008] T. Bretl and S. Lall. *Testing Static Equilibrium for Legged Robots.* Robotics, IEEE Transactions on, vol. 24, no. 4, pages 794–807, Aug 2008. (Cited on page 8.)

[Brogliato 2003] B. Brogliato. *Some perspectives on the analysis and control of complementarity systems.* Automatic Control, IEEE Transactions on, vol. 48, no. 6, pages 918–935, June 2003. (Cited on pages 12 and 14.)

[Caccavale 2008] F. Caccavale and M. Uchiyama. *Cooperative Manipulators.* In B. Siciliano and O. Khatib, editors, Springer Handbook of Robotics, pages 701–718. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. (Cited on page 77.)

[Cagienard 2007] R. Cagienard, P. Grieder, E. Kerrigan and M. Morari. *Move blocking strategies in receding horizon control.* Journal of Process Control, vol. 17, no. 6, pages 563 – 570, 2007. (Cited on page 43.)

[Caron 2015] S. Caron, Q.-C. Pham and Y. Nakamura. *Stability of surface contacts for humanoid robots: Closed-form formulae of the Contact Wrench Cone for rectangular support areas.* In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 5107–5112, May 2015. (Cited on page 94.)

[Carver 2009] S. G. Carver, N. J. Cowan and J. M. Guckenheimer. *Lateral stability of the spring-mass hopper suggests a two-step control strategy for running.* Chaos, vol. 19, no. 2, 2009. (Cited on pages 7 and 10.)

[Chevallereau 2003] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-de Wit and J. Grizzle. *RABBIT: a testbed for advanced control theory.* Control Systems, IEEE, vol. 23, no. 5, pages 57–79, Oct 2003. (Cited on page 9.)

[Chevallereau 2009] C. Chevallereau, J. Grizzle and C.-L. Shih. *Asymptotically Stable Walking of a Five-Link Underactuated 3-D Bipedal Robot.* Robotics, IEEE Transactions on, vol. 25, no. 1, pages 37–50, Feb 2009. (Cited on page 9.)

[Christensen 2008] H. I. Christensen and G. D. Hager. *Sensing and Estimation.* In B. Siciliano and O. Khatib, editors, Springer Handbook of Robotics, pages 87–107. Springer Berlin Heidelberg, 2008. (Cited on page 12.)

[Collette 2007] C. Collette, A. Micaelli, C. Andriot and P. Lemerle. *Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts.* In Humanoid Robots, 2007 7th IEEE-RAS International Conference on, pages 81–88, Nov 2007. (Cited on page 8.)

[Collins 2005] S. Collins, A. Ruina, R. Tedrake and M. Wisse. *Efficient Bipedal Robots Based on Passive-Dynamic Walkers.* Science, vol. 307, no. 5712, pages 1082–1085, 2005. (Cited on page 8.)

[COM 2016] *Multi-contact Collaborative Humanoids in Aircraft Manufacturing.* Available: http://comanoid.cnrs.fr/project-overview, January 2016. (Cited on page 9.)

[Dai 2014] H. Dai, A. Valenzuela and R. Tedrake. *Whole-body motion planning with centroidal dynamics and full kinematics.* In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 295–302, Nov 2014. (Cited on pages 9, 28, 33, and 41.)

[Dalibard 2013] S. Dalibard, A. El Khoury, F. Lamiraux, A. Nakhaei, M. Taïx and J.-P. Laumond. *Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach.* The International Journal of Robotics Research, vol. 32, no. 9-10, pages 1089–1103, 2013. (Cited on page 9.)

[de Lasa 2010] M. de Lasa, I. Mordatch and A. Hertzmann. *Feature-based Locomotion Controllers.* ACM Trans. Graph., vol. 29, no. 4, pages 131:1–131:10, July 2010. (Cited on page 49.)

[de Santos 2005] P. de Santos, J. Estremera, E. Garcia and M. Armada. *Including Joint Torques and Power Consumption in the Stability Margin of Walking Robots.* Autonomous Robots, vol. 18, no. 1, pages 43–57, 2005. (Cited on page 8.)

[Dellin 2012] C. Dellin and S. Srinivasa. *A Framework for Extreme Locomotion Planning.* In IEEE International Conference on Robotics and Automation (ICRA '12), May 2012. (Cited on page 27.)

[Deo 1995] A. S. Deo and I. D. Walker. *Overview of damped least-squares methods for inverse kinematics of robot manipulators.* Journal of Intelligent and Robotic Systems, vol. 14, no. 1, pages 43–68, 1995. (Cited on page 47.)

[Diedam 2008] H. Diedam, D. Dimitrov, P. Wieber, K. Mombaur and M. Diehl. *Online walking gait generation with adaptive foot positioning through linear model predictive control.* In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 1121–1126. IEEE, 2008. (Cited on page 23.)

[Dietrich 2015] A. Dietrich, C. Ott and A. Albu-Schäffer. *An overview of null space projections for redundant, torque-controlled robots.* The International Journal of Robotics Research, 2015. (Cited on page 48.)

[Dimitrov 2011a] D. Dimitrov, A. Paolillo and P. Wieber. *Walking motion generation with online foot position adaptation based on ℓ_1-and ℓ_∞-norm penalty formulations.* In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 3523–3529. IEEE, 2011. (Cited on pages 53 and 96.)

[Dimitrov 2011b] D. Dimitrov, A. Sherikov and P. Wieber. *A sparse model predictive control formulation for walking motion generation.* In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 2292–2299. IEEE, 2011. (Cited on page 53.)

[Dimitrov 2014] D. Dimitrov, P.-B. Wieber and A. Escande. *Multi-objective control of robots.* Journal of the Robotics Society of Japan, vol. 32, no. 6, pages 512–518, 2014. (Cited on page 48.)

[Dimitrov 2015] D. Dimitrov, A. Sherikov and P.-B. Wieber. Efficient resolution of potentially conflicting linear constraints in robotics. Preprint, August 2015. (Cited on pages 2, 47, 48, 49, 53, 56, 67, and 76.)

[DRC 2015] *DARPA Robotics Challenge (DRC).* Available: http://www.theroboticschallenge.org/overview, September 2015. (Cited on pages 1 and 9.)

[Englsberger 2011] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer and G. Hirzinger. *Bipedal walking control based on Capture Point dynamics.* In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4420 –4427, sept. 2011. (Cited on pages 24 and 37.)

[Englsberger 2014] J. Englsberger, A. Werner, C. Ott, B. Henze, M. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid and A. Albu-Schaffer. *Overview of the torque-controlled humanoid robot TORO.* In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 916–923, Nov 2014. (Cited on page 24.)

[Escande 2009] A. Escande and A. Kheddar. *Contact planning for acyclic motion with tasks constraints.* In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 435–440, Oct 2009. (Cited on page 67.)

[Escande 2014] A. Escande, N. Mansard and P.-B. Wieber. *Hierarchical quadratic programming: Fast online humanoid-robot motion generation.* The International Journal of Robotics Research, 2014. (Cited on pages 8, 45, 47, 48, 53, 54, 68, and 76.)

[Espiau 1998] B. Espiau and R. Boulic. *On the Computation and Control of the Mass Center of Articulated Chains.* Rapport technique RR-3479, INRIA, August 1998. (Cited on page 86.)

[Feng 2013] S. Feng, X. Xinjilefu, W. Huang and C. G. Atkeson. *3D walking based on online optimization*. In Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on, pages 21–27, Oct 2013. (Cited on page 28.)

[Ferreau 2008] H. J. Ferreau, H. G. Bock and M. Diehl. *An online active set strategy to overcome the limitations of explicit MPC*. International Journal of Robust and Nonlinear Control, vol. 18, no. 8, pages 816–830, 2008. (Cited on pages 52, 53, and 54.)

[Ferreau 2014] H. Ferreau, C. Kirches, A. Potschka, H. Bock and M. Diehl. *qpOASES: A parametric active-set algorithm for quadratic programming*. Mathematical Programming Computation, vol. 6, no. 4, pages 327–363, 2014. (Cited on page 29.)

[Fraichard 2003] T. Fraichard and H. Asama. *Inevitable collision states. A step towards safer robots?* In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, volume 1, pages 388–393 vol.1, Oct 2003. (Cited on page 7.)

[Fujimoto 1996] Y. Fujimoto and A. Kawamura. *Proposal of biped walking control based on robust hybrid position/force control*. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, volume 3, pages 2724–2730 vol.3, Apr 1996. (Cited on page 94.)

[Fukumoto 2004] Y. Fukumoto, K. Nishiwaki, M. Inaba and H. Inoue. *Hand-centered whole-body motion control for a humanoid robot*. In Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, volume 2, pages 1186–1191 vol.2, Sept 2004. (Cited on pages 27, 41, and 58.)

[Fukuoka 2003] Y. Fukuoka, H. Kimura and A. H. Cohen. *Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts*. The International Journal of Robotics Research, vol. 22, no. 3-4, pages 187–202, 2003. (Cited on page 9.)

[Gill 1984] P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright. *Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints*. ACM Trans. Math. Softw., vol. 10, no. 3, pages 282–298, August 1984. (Cited on page 53.)

[Glocker 2006] C. Glocker. *An Introduction to Impacts*. In J. Haslinger and G. E. Stavroulakis, editors, Nonsmooth Mechanics of Solids, volume 485 of *CISM International Centre for Mechanical Sciences*, pages 45–101. Springer Vienna, 2006. (Cited on page 87.)

[Golub 1996] G. Golub and C. Van Loan. Matrix computations. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996. (Cited on page 54.)

[Grizzle 2006] J. W. Grizzle. *Remarks on Event-Based Stabilization of Periodic Orbits in Systems with Impulse Effects*. In Second International Symposium on Communication, Control and Signal Processing, 2006. (Cited on page 9.)

[Guennebaud 2010] G. Guennebaud, B. Jacob*et al.* *Eigen v3*. Available: http://eigen.tuxfamily.org, 2010. (Cited on page 56.)

[Hauser 2008] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada and B. Wilcox. *Motion Planning for Legged Robots on Varied Terrain*. The International Journal of Robotics Research, vol. 27, no. 11-12, pages 1325–1349, 2008. (Cited on pages 9 and 92.)

[Henze 2014] B. Henze, C. Ott and M. Roa. *Posture and balance control for humanoid robots in multi-contact scenarios based on Model Predictive Control.* In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 3253–3258, Sept 2014. (Cited on pages 8, 10, 44, and 94.)

[Herceg 2015] M. Herceg, C. N. Jones and M. Morari. *Dominant speed factors of active set methods for fast MPC.* Optimal Control Applications and Methods, vol. 36, no. 5, pages 608–627, 2015. (Cited on page 52.)

[Herdt 2010] A. Herdt, H. Diedam, P. Wieber, D. Dimitrov, K. Mombaur and M. Diehl. *Online Walking Motion Generation with Automatic Footstep Placement.* Advanced Robotics, 24, vol. 5, no. 6, pages 719–737, 2010. (Cited on pages 10, 23, 24, 25, 28, 33, 34, 38, 39, 41, 43, 58, 60, 61, and 96.)

[Herdt 2012] A. Herdt. *Model predictive control of a humanoid robots.* PhD thesis, Paris Institute of Technology, 2012. (Cited on pages 27, 34, and 97.)

[Herzog 2015] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti. *Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid.* Autonomous Robots, vol. 40, no. 3, pages 473–491, 2015. (Cited on pages 43, 49, 68, and 94.)

[Hof 2005] A. Hof, M. Gazendam and W. Sinke. *The condition for dynamic stability.* Journal of Biomechanics, vol. 38, no. 1, pages 1 – 8, 2005. (Cited on page 37.)

[Homsi 2016a] S. A. Homsi. *Online Generation of Time-Optimal Trajectories for Industrial Robots in Dynamic Environments.* PhD thesis, Communauté Université Grenoble Alpes, 2016. (Cited on page 77.)

[Homsi 2016b] S. A. Homsi, A. Sherikov, D. Dimitrov and P.-B. Wieber. *A hierarchical approach to minimum-time control of industrial robots.* In Robotics and Automation (ICRA), 2016 IEEE International Conference on, May 2016. (Cited on pages 2, 50, 54, 56, and 77.)

[Howe 1996] R. D. Howe and M. R. Cutkosky. *Practical Force-Motion Models for Sliding Manipulation.* The International Journal of Robotics Research, vol. 15, no. 6, pages 557–572, 1996. (Cited on pages 14 and 92.)

[Hurmuzlu 2004] Y. Hurmuzlu, F. Génot and B. Brogliato. *Modeling, stability and control of biped robots—a general framework.* Automatica, vol. 40, no. 10, pages 1647 – 1664, 2004. (Cited on page 12.)

[Hyon 2007] S. Hyon, J. Hale and G. Cheng. *Full-Body Compliant Human Humanoid Interaction: Balancing in the Presence of Unknown External Forces.* Robotics, IEEE Transactions on, vol. 23, no. 5, pages 884–898, Oct 2007. (Cited on pages 8 and 68.)

[Ijspeert 2008] A. J. Ijspeert. *Central pattern generators for locomotion control in animals and robots: A review.* Neural Networks, vol. 21, no. 4, pages 642 – 653, 2008. Robotics and Neuroscience. (Cited on page 9.)

[Kajita 2001] S. Kajita, O. Matsumoto and M. Saigo. *Real-time 3D walking pattern generation for a biped robot with telescopic legs.* In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, volume 3, pages 2299–2306 vol.3, 2001. (Cited on page 23.)

[Kajita 2003] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. *Biped walking pattern generation by using preview control of zero-moment point.* In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 2, pages 1620–1626 vol.2, Sept 2003. (Cited on pages 9, 10, 23, 24, 25, 34, 41, and 43.)

[Kajita 2010] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi. *Biped walking stabilization based on linear inverted pendulum tracking.* In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 4489 –4496, oct. 2010. (Cited on page 24.)

[Kanehiro 2009] F. Kanehiro, W. Suleiman, K. Miura, M. Morisawa and E. Yoshida. *Feasible pattern generation method for humanoid robots.* In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pages 542–548, Dec 2009. (Cited on page 41.)

[Kaneko 2004] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi and T. Isozumi. *Humanoid robot HRP-2.* In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, volume 2, pages 1083–1090 Vol.2, April 2004. (Cited on pages 12, 24, and 56.)

[Kaneko 2009] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka and S. Kajita. *Cybernetic human HRP-4C.* In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pages 7–14, Dec 2009. (Cited on page 24.)

[Kaneko 2011] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi and N. Kanehira. *Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body.* In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4400–4407, Sept 2011. (Cited on pages 78 and 79.)

[Kanoun 2009] O. Kanoun. *Task-driven motion control for humanoid robots.* PhD thesis, University of Toulouse III Paul Sabatier, 2009. (Cited on pages 1, 16, 17, 47, and 48.)

[Kanoun 2010] O. Kanoun, J.-P. Laumond and E. Yoshida. *Planning Foot Placements for a Humanoid Robot: A Problem of Inverse Kinematics.* The International Journal of Robotics Research, 2010. (Cited on pages 9, 41, and 67.)

[Kanoun 2011] O. Kanoun, F. Lamiraux and P.-B. Wieber. *Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task.* Robotics, IEEE Transactions on, vol. 27, no. 4, pages 785–792, Aug 2011. (Cited on pages 46, 47, 49, and 68.)

[Kemp 2008] C. Kemp, P. Fitzpatrick, H. Hirukawa, K. Yokoi, K. Harada and Y. Matsumoto. *Humanoids.* In B. Siciliano and O. Khatib, editors, Springer Handbook of Robotics, pages 361–389. Springer Berlin Heidelberg, 2008. (Cited on page 1.)

[Kerrigan 2000] E. C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control.* PhD thesis, University of Cambridge, 2000. (Cited on pages 7 and 50.)

[Khatib 1987] O. Khatib. *A unified approach for motion and force control of robot manipulators: The operational space formulation.* Robotics and Automation, IEEE Journal of, vol. 3, no. 1, pages 43–53, February 1987. (Cited on page 16.)

[Koenemann 2015] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz and N. Mansard. *Whole-body model-predictive control applied to the HRP-2 humanoid*. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 3346–3351, Sept 2015. (Cited on pages 22 and 42.)

[Koolen 2012] T. Koolen, T. de Boer, J. Rebula, A. Goswami and J. Pratt. *Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models*. The International Journal of Robotics Research, vol. 31, no. 9, pages 1094–1113, 2012. (Cited on pages 5, 7, 10, and 37.)

[Krause 2012] M. Krause, J. Englsberger, P.-B. Wieber and C. Ott. *Stabilization of the Capture Point Dynamics for Bipedal Walking based on Model Predictive Control*. In Proceedings of the IFAC Symposium on Robot Control, 2012. (Cited on page 24.)

[Kuffner Jr. 2002] J. J. Kuffner Jr., S. Kagami, K. Nishiwaki, M. Inaba and H. Inoue. *Dynamically-Stable Motion Planning for Humanoid Robots*. Autonomous Robots, vol. 12, no. 1, pages 105–118, 2002. (Cited on page 9.)

[Kuindersma 2014] S. Kuindersma, F. Permenter and R. Tedrake. *An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion*. In Proceedings of the International Conference on Robotics and Automation (ICRA), Hong Kong, China, May 2014. (Cited on pages 15, 43, 53, 54, and 92.)

[Lafaye 2014] J. Lafaye, D. Gouaillier and P.-B. Wieber. *Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 336–341, Nov 2014. (Cited on pages 27, 67, and 97.)

[LaValle 2006] S. LaValle. Planning algorithms. Cambridge University Press, 2006. (Cited on pages 7 and 9.)

[Lee 2012] J. Lee, N. Mansard and J. Park. *Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control*. Robotics, IEEE Transactions on, vol. 28, no. 6, pages 1260–1277, Dec 2012. (Cited on page 17.)

[Lewis 1999] A. D. Lewis and R. M. Murray. *Configuration Controllability of Simple Mechanical Control Systems*. SIAM Review, vol. 41, no. 3, pages 555–574, 1999. (Cited on page 7.)

[Liégeois 1977] A. Liégeois. *Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms*. Systems, Man and Cybernetics, IEEE Transactions on, vol. 7, no. 12, pages 868–871, Dec 1977. (Cited on page 46.)

[Maciejowski 2002] J. M. Maciejowski. Predictive control with constraints. Prentice-Hall, 2002. (Cited on page 29.)

[Mansard 2009] N. Mansard, O. Khatib and A. Kheddar. *A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks*. IEEE Transactions on Robotics, vol. 25, no. 3, pages 670–685, June 2009. (Cited on page 46.)

[Mansard 2012] N. Mansard. *A dedicated solver for fast operational-space inverse dynamics*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 4943–4949, May 2012. (Cited on page 49.)

[Mansour 2011] D. Mansour, A. Micaelli, A. Escande and P. Lemerle. *A new optimization based approach for push recovery in case of multiple noncoplanar contacts*. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 331–338, Oct 2011. (Cited on page 10.)

[Mansour 2013] D. Mansour, A. Micaelli and P. Lemerle. *Humanoid push recovery control in case of multiple non-coplanar contacts*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 4137–4144, Nov 2013. (Cited on page 94.)

[MAX 2016] *Maxima, a Computer Algebra System*. Available: http://maxima.sourceforge.net/, January 2016. (Cited on page 30.)

[Mayne 2000] D. Mayne, J. Rawlings, C. Rao and P. Scokaert. *Constrained model predictive control: Stability and optimality*. Automatica, vol. 36, no. 6, pages 789 – 814, 2000. (Cited on pages 34 and 35.)

[McGeer 1990] T. McGeer. *Passive Dynamic Walking*. The International Journal of Robotics Research, vol. 9, no. 2, pages 62–82, 1990. (Cited on page 8.)

[Miura 2013] K. Miura, F. Kanehiro, K. Kaneko, S. Kajita and K. Yokoi. *Slip-Turn for Biped Robots*. Robotics, IEEE Transactions on, vol. 29, no. 4, pages 875–887, Aug 2013. (Cited on page 14.)

[Moreau 1966] J. J. Moreau. *Quadratic Programming in Mechanics: Dynamics of One-Sided Constraints*. SIAM Journal on Control, vol. 4, no. 1, pages 153–158, 1966. (Cited on page 83.)

[Morisawa 2007] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko and H. Hirukawa. *Experimentation of Humanoid Walking Allowing Immediate Modification of Foot Place Based on Analytical Solution*. In Robotics and Automation, 2007 IEEE International Conference on, pages 3989–3994, April 2007. (Cited on pages 10 and 41.)

[Murray 1994] R. Murray, Z. Li and S. Sastry. A mathematical introduction to robotic manipulation. Taylor & Francis, 1994. (Cited on page 94.)

[Muske 1993] K. R. Muske and J. B. Rawlings. *Model predictive control with linear models*. AIChE Journal, vol. 39, no. 2, pages 262–287, 1993. (Cited on page 36.)

[Nagasaka 2012] K. Nagasaka, T. Fukushima and H. Shimomura. *Whole-body Control of a Humanoid Robot Based on Generalized Inverse Dynamics and Multi-contact Stabilizer that can Take Account of Contact Constraints*. In the Proceedings of the 17th Robotics Symposia, pages 134–141, 2012. (Cited on pages 9, 22, 33, 93, and 94.)

[Nakanishi 2004] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal and M. Kawato. *Learning from demonstration and adaptation of biped locomotion*. Robotics and Autonomous Systems, vol. 47, no. 2–3, pages 79 – 91, 2004. Robot Learning from Demonstration. (Cited on page 9.)

[Naveau 2016] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Soueres. *A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control*. Robotics and Automation Letters, IEEE, vol. PP, no. 99, pages 1–1, 2016. (Cited on page 96.)

[Nemoto 2015] T. Nemoto and A. Yamamoto. *Thermobot: A Bipedal Walker Driven by Constant Heating*. In Intelligent Robots and Systems (IROS 2015), 2015 IEEE/RSJ International Conference on, Hamburg, Germany, 2015. (Cited on page 8.)

[Nishiwaki 2003] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba and H. Inoue. *Online humanoid walking control system and a moving goal tracking experiment*. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 1, pages 911–916 vol.1, Sept 2003. (Cited on pages 27, 41, and 58.)

[Nishiwaki 2009] K. Nishiwaki and S. Kagami. *Online Walking Control System for Humanoids with Short Cycle Pattern Generation*. The International Journal of Robotics Research, vol. 28, no. 6, pages 729–742, 2009. (Cited on pages 9, 40, 41, and 42.)

[Nishiwaki 2011] K. Nishiwaki and S. Kagami. *Online design of torso height trajectories for walking patterns that takes future kinematic limits into consideration*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 2029–2034, May 2011. (Cited on page 28.)

[Nocedal 2006] J. Nocedal and S. Wright. Numerical optimization. Springer Series in Operations Research and Financial Engineering. Springer, second édition, 2006. (Cited on pages 29, 45, and 84.)

[OCT 2016] *GNU Octave*. Available: https://gnu.org/software/octave/, January 2016. (Cited on page 56.)

[Orin 2013] D. E. Orin, A. Goswami and S.-H. Lee. *Centroidal dynamics of a humanoid robot*. Autonomous Robots, vol. 35, no. 2-3, pages 161–176, 2013. (Cited on page 86.)

[Ott 2011] C. Ott, M. Roa and G. Hirzinger. *Posture and balance control for biped robots based on contact force optimization*. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 26–33, Oct 2011. (Cited on pages 8, 68, and 92.)

[Popov 2010] V. Popov. Contact mechanics and friction: Physical principles and applications. Springer Berlin Heidelberg, 2010. (Cited on page 11.)

[Pratt 2006] J. Pratt and R. Tedrake. *Velocity-Based Stability Margins for Fast Bipedal Walking*. In M. Diehl and K. Mombaur, editors, Fast Motions in Biomechanics and Robotics, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 299–324. Springer Berlin Heidelberg, 2006. (Cited on page 7.)

[Rawlings 2009] J. B. Rawlings and D. Q. Mayne. Model predictive control: Theory and design. Nob Hill Publishing, 2009. (Cited on pages 7, 29, and 41.)

[Rossiter 2003] J. Rossiter. Model-based predictive control: A practical approach. CRC Press control series. Taylor & Francis, 2003. (Cited on pages 7, 34, 41, 42, and 49.)

[Rubrecht 2012] S. Rubrecht, V. Padois, P. Bidaud, M. de Broissia and M. Da Silva Simoes. *Motion safety and constraints compatibility for multibody robots*. Autonomous Robots, vol. 32, no. 3, pages 333–349, 2012. (Cited on pages 7, 14, and 89.)

[Saab 2013] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères and J.-Y. Fourquet. *Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints*. IEEE Transactions on Robotics, vol. 29, no. 2, pages 346–362, 2013. (Cited on pages 1, 43, 48, 68, and 89.)

[Samson 1991] C. Samson, M. Borgne and B. Espiau. Robot control: the task function approach. Oxford engineering science series. Clarendon Press, 1991. (Cited on pages 16 and 17.)

[Scheinerman 1996] E. Scheinerman. Invitation to dynamical systems. Prentice Hall, 1996. (Cited on pages 35 and 36.)

[Schouwenaars 2006] T. Schouwenaars. *Safe trajectory planning of autonomous vehicles*. PhD thesis, Massachusetts Institute of Technology, 2006. (Cited on pages 7, 10, and 34.)

[Sentis 2007] L. Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford University, 2007. (Cited on pages 1, 48, and 49.)

[SHE 2016] Available: http://asherikov.github.com/, May 2016. (Cited on page 56.)

[Sherikov 2012] A. Sherikov. Model predictive control of a walking bipedal robot using online optimization. Master's thesis, Örebro University, 2012. (Cited on page 96.)

[Sherikov 2014] A. Sherikov, D. Dimitrov and P.-B. Wieber. *Whole body motion controller with long-term balance constraints*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 444–450, Nov 2014. (Cited on pages 2, 3, 10, 24, 25, 27, 40, 41, 44, 50, 51, 53, 54, 56, 58, and 96.)

[Sherikov 2015] A. Sherikov, D. Dimitrov and P.-B. Wieber. *Balancing a humanoid robot with a prioritized contact force distribution*. In Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on, Nov 2015. (Cited on pages 2, 3, 10, 22, 34, 35, 41, 44, 48, 51, 52, 56, and 68.)

[Shimmyo 2013] S. Shimmyo, T. Sato and K. Ohnishi. *Biped Walking Pattern Generation by Using Preview Control Based on Three-Mass Model*. Industrial Electronics, IEEE Transactions on, vol. 60, no. 11, pages 5137–5147, Nov 2013. (Cited on page 27.)

[Shkolnik 2011] A. Shkolnik, M. Levashov, I. R. Manchester and R. Tedrake. *Bounding on rough terrain with the LittleDog robot*. The International Journal of Robotics Research, vol. 30, no. 2, pages 192–215, 2011. (Cited on page 9.)

[Siciliano 1991] B. Siciliano and J. J. E. Slotine. *A general framework for managing multiple tasks in highly redundant robotic systems*. In Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on, pages 1211–1216 vol.2, June 1991. (Cited on pages 46 and 47.)

[Siciliano 2009] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo. Robotics: Modelling, planning and control. Advanced Textbooks in Control and Signal Processing. Springer London, 2009. (Cited on page 12.)

[Sontag 1998] E. D. Sontag. Mathematical control theory: deterministic finite dimensional systems, volume 6. Springer Science & Business Media, 1998. (Cited on page 7.)

[SOT 2016] *SOT Hierarchical solver*. Available: https://github.com/stack-of-tasks/soth, January 2016. (Cited on page 76.)

[Sparrow 2005] W. Sparrow and O. Tirosh. *Gait termination: a review of experimental methods and the effects of ageing and gait pathologies*. Gait & Posture, vol. 22, no. 4, pages 362 – 371, 2005. (Cited on page 10.)

[Sreenath 2010] K. Sreenath, H.-W. Park, I. Poulakakis and J. Grizzle. *A Compliant Hybrid Zero Dynamcis Controller for Stable, Efficient and Fast Bipedal Walking on MABEL.* The International Journal of Robotics Research, 2010. (Cited on page 9.)

[Stasse 2009] O. Stasse, P. Evrard, N. Perrin, N. Mansard and A. Kheddar. *Fast foot prints re-planning and motion generation during walking in physical human-humanoid interaction.* In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pages 284–289, Dec 2009. (Cited on page 38.)

[Stephens 2010] B. Stephens and C. Atkeson. *Dynamic Balance Force Control for compliant humanoid robots.* In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 1248–1255, Oct 2010. (Cited on pages 35, 69, and 94.)

[Sugihara 2002] T. Sugihara and Y. Nakamura. *Whole-body cooperative balancing of humanoid robot using COG Jacobian.* In Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, volume 3, pages 2575–2580 vol.3, 2002. (Cited on page 86.)

[Takenaka 2009] T. Takenaka, T. Matsumoto and T. Yoshiike. *Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-.* In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 1084 –1091, oct. 2009. (Cited on pages 10, 24, 27, and 37.)

[Tassa 2014] Y. Tassa, N. Mansard and E. Todorov. *Control-limited differential dynamic programming.* In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 1168–1175, May 2014. (Cited on pages 10, 28, and 67.)

[Tomlin 2003] C. Tomlin, I. Mitchell, A. Bayen and M. Oishi. *Computational techniques for the verification of hybrid systems.* Proceedings of the IEEE, vol. 91, no. 7, pages 986–1001, July 2003. (Cited on page 7.)

[Trinkle 1997] J. C. Trinkle, J.-S. Pang, S. Sudarsky and G. Lo. *On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction.* ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, vol. 77, no. 4, pages 267–279, 1997. (Cited on page 12.)

[Vada 1999] J. Vada, O. Slupphaug and B. A. Foss. *Infeasibility Handling In Linear Mpc Subject To Prioritized Constraints.* In In Preprints IFAC'99 14th World Congress, pages 163–168, 1999. (Cited on page 47.)

[Van De Panne 1997] M. Van De Panne. *From Footprints to Animation.* Computer Graphics Forum, vol. 16, no. 4, pages 211–223, 1997. (Cited on page 1.)

[Vukobratović 1970] M. Vukobratović, A. Frank and D. Juricić. *On the Stability of Biped Locomotion.* Biomedical Engineering, IEEE Transactions on, vol. BME-17, no. 1, pages 25–36, Jan 1970. (Cited on page 10.)

[Vukobratović 2004] M. Vukobratović and B. Borovać. *Zero-Moment Point — thirty five years of its life.* International Journal of Humanoid Robotics, vol. 01, no. 01, pages 157–173, 2004. (Cited on page 21.)

[Wang 2010] Y. Wang and S. Boyd. *Fast Model Predictive Control Using Online Optimization.* Control Systems Technology, IEEE Transactions on, vol. 18, no. 2, pages 267 –278, march 2010. (Cited on pages 52 and 53.)

[Westervelt 2003] E. Westervelt, J. Grizzle and D. Koditschek. *Hybrid zero dynamics of planar biped walkers*. Automatic Control, IEEE Transactions on, vol. 48, no. 1, pages 42–56, Jan 2003. (Cited on page 9.)

[Westervelt 2007] E. R. Westervelt, C. Chevallereau, B. Morris, J. W. Grizzle and J. H. Choi. Feedback control of dynamic bipedal robot locomotion. CRC Press, 2007. (Cited on page 9.)

[Wieber 2002] P.-B. Wieber. *On the stability of walking systems*. In Proceedings of the International Workshop on Humanoid and Human Friendly Robotics, Tsukuba, Japon, 2002. (Cited on pages 5, 7, and 8.)

[Wieber 2006a] P.-B. Wieber. *Holonomy and Nonholonomy in the Dynamics of Articulated Motion*. In M. Diehl and K. Mombaur, editors, Fast Motions in Biomechanics and Robotics, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 411–425. Springer Berlin Heidelberg, 2006. (Cited on pages 18, 19, 82, 83, and 86.)

[Wieber 2006b] P.-B. Wieber. *Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations*. In IEEE-RAS International Conference on Humanoid Robots, Genova, Italie, 2006. (Cited on page 10.)

[Wieber 2008] P.-B. Wieber. *Viability and Predictive Control for Safe Locomotion*. In IEEE-RSJ International Conference on Intelligent Robots & Systems, Nice, France, 2008. (Cited on pages 34 and 61.)

[Wieber 2015] P.-B. Wieber, R. Tedrake and S. Kuindersma. Modeling and control of legged robots. (to appear), 2015. (Cited on pages 7, 21, and 32.)

[Wikipedia 2015a] Wikipedia. *Falling Cat — Wikipedia, The Free Encyclopedia*. Available: https://en.wikipedia.org/wiki/Falling_Cat, December 2015. (Cited on page 19.)

[Wikipedia 2015b] Wikipedia. *Three Laws of Robotics — Wikipedia, The Free Encyclopedia*. Available: https://en.wikipedia.org/wiki/Three_Laws_of_Robotics, September 2015. (Cited on page 46.)

[Yamane 2004] K. Yamane. Simulating and generating motions of human figures (springer tracts in advanced robotics, v. 9). SpringerVerlag, 2004. (Cited on page 94.)

[Yamashita 1974] T. Yamashita and H. Yamada. *A Study on Stability of Bipedal Locomotion*. In On Theory and Practice of Robots and Manipulators, volume 201 of *International Centre for Mechanical Sciences*, pages 41–53. Springer Vienna, 1974. (Cited on page 7.)

[Yoshida 2006] E. Yoshida, O. Kanoun, C. Esteves and J.-P. Laumond. *Task-driven Support Polygon Reshaping for Humanoids*. In Humanoid Robots, 2006 6th IEEE-RAS International Conference on, pages 208–213, Dec 2006. (Cited on pages 27, 41, and 58.)

[Zaytsev 2015a] P. Zaytsev, S. J. Hasaneini and A. Ruina. *Two steps is enough: no need to plan far ahead for walking balance*. In Proceedings of the International Conference on Robotics and Automation (ICRA), May 2015. (Cited on pages 5, 7, and 10.)

[Zaytsev 2015b] P. Zaytsev. *Using Controllability of Simple Models to Generate Maximally Robust Walking-Robot Controllers*. PhD thesis, Cornell University, 2015. (Cited on pages 6, 7, and 18.)

[Zhou 2013] X. Zhou, Y. Guan, L. Jiang, H. Zhu, C. Cai, W. Wu and H. Zhang. *Stability of biped robotic walking with frictional constraints.* Robotica, vol. 31, pages 573–588, 7 2013. (Cited on page 94.)

[Zhu 2006] C. Zhu and A. Kawamura. *What Is the Real Frictional Constraint in Biped Walking? - Discussion on Frictional Slip with Rotation.* In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 5762–5768, Oct 2006. (Cited on page 94.)

[Zucker 2011] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson and J. Kuffner. *Optimization and learning for rough terrain legged locomotion.* The International Journal of Robotics Research, vol. 30, no. 2, pages 175–191, 2011. (Cited on page 9.)