



HAL
open science

Design of safe control laws for the locomotion of biped robots

Nestor Bohorquez Dorante

► **To cite this version:**

Nestor Bohorquez Dorante. Design of safe control laws for the locomotion of biped robots. Automatic. Université Grenoble Alpes, 2018. English. NNT : 2018GREAT110 . tel-02057022v2

HAL Id: tel-02057022

<https://theses.hal.science/tel-02057022v2>

Submitted on 29 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

**DOCTEUR DE LA
COMMUNAUTÉ UNIVERSITÉ GRENOBLE
ALPES**

Spécialité : **Automatique-Productique**

Arrêté ministériel : 25 mai 2016

Présentée par

Néstor BOHÓRQUEZ DORANTE

Thèse dirigée par **Bernard BROGLIATO**
et co-encadrée par **Pierre-Brice WIEBER**

préparée au sein de l'**Inria Grenoble Rhône-Alpes**
dans l'**École Doctorale EEATS**

Design of Safe Control Laws for the Locomotion of Biped Robots

Thèse soutenue publiquement le **14 décembre 2018**,
devant le jury composé de :

Olivier STASSE

Directeur de Recherche LAAS Toulouse, Rapporteur

Nacim RAMDANI

Professeur à l'Université d'Orléans, Président

Julien PETTRÉ

Chargé de Recherche Inria Rennes, Examineur

Bernard BROGLIATO

Directeur de Recherche Inria Grenoble, Directeur de thèse

Pierre-Brice WIEBER

Chargé de Recherche Inria Grenoble, Co-encadrant de thèse



Abstract

We want a biped robot to walk safely in a crowd. This involves two aspects: balance and collision avoidance. The first implies avoiding kinematic and dynamical failures of the unstable walking dynamics of the robot; the second refers to avoiding collisions with people. We want to be able to solve both problems not only now but also in the future. We can ensure balance indefinitely by entering in a cyclic walk or by making the robot stop after a couple of steps. Nonetheless, we cannot give a comparable guarantee in collision avoidance for many reasons: impossibility of having absolute knowledge of where people are moving, kinematic/dynamical limitations of the robot, adversarial crowd motion, *etc.* We address this limitation with a standard strategy for crowd navigation, known as *passive safety*, that allows us to formulate a unified *Model Predictive Control* approach for balance and collision avoidance in which we require the robot to stop safely in finite time. In addition, we define a novel safe navigation strategy based on the premise of avoiding collisions for *as long as possible* that minimizes their occurrence and severity. We propose a lexicographic formulation that produces motions that comply with such premise.

We increase the degrees of freedom of the locomotion of a biped robot by allowing the duration and orientation of its steps to vary online. This introduces nonlinearities in the constraints of the optimization problems we solve. We approximate these nonlinear constraints with *safe linear constraints* so that satisfying the latter implies satisfying the former. We propose a novel method (*Safe Sequential Quadratic Programming*) that ensures feasible Newton iterates in the solution of nonlinear problems based on this redefinition of constraints.

We make a series of simulations of a biped robot walking in a crowd to evaluate the performance of our proposed controllers. We are able to attest the reduction in the number and in the severity of collisions with our proposed navigation strategy in comparison with passive safety, specially when there is uncertainty in the motion of people. We show typical behaviors of the robot that arise when we allow the online variation of the duration and orientation of the steps and how it further improves collision avoidance. We report the computational cost of our proposed numerical method for nonlinear problems in comparison with a standard method. We show that we only need one Newton iteration to arrive to a feasible solution but that the CPU time is dependent on the amount of active set factorizations needed to arrive to the optimal active set.

Résumé

Un robot bipède doit pouvoir marcher en toute sécurité dans une foule. Pour cela, il faut prendre en compte deux aspects : l'équilibre et l'évitement des collisions. Maintenir l'équilibre implique d'éviter les défaillances dynamiques et cinématiques de la dynamique instable du robot. Pour ce qui est de l'évitement des collisions, il s'agit d'éviter le contact entre le robot et des individus. Nous voulons être capables de satisfaire ces deux contraintes simultanément, à l'instant présent mais aussi dans le futur. Nous pouvons assurer l'équilibre du robot indéfiniment en le faisant entrer dans un cycle limite de marche ou en le faisant s'arrêter après quelques pas. Néanmoins, une telle garantie pour l'évitement d'obstacle n'est pas possible pour plusieurs raisons : impossibilité de connaître de manière absolue la direction vers laquelle les individus se dirigent, limitations cinématiques et dynamiques du robot, mouvement non-coopératif de la foule, *etc.* Nous traitons ces limitations avec une stratégie standard de navigation dans une foule, appelée *passive safety*, qui nous permet de formuler une loi de commande prédictive avec laquelle nous assurons l'équilibre et l'évitement des collisions, de manière unifiée, en faisant s'arrêter le robot de manière sécurisée et en temps fini. De plus, nous définissons une nouvelle stratégie de navigation sûre basée sur le principe d'évitement des collisions aussi longtemps que possible, qui a la propriété de minimiser leur apparition et sévérité. Nous proposons une formulation lexicographique qui synthétise des mouvements conformes à ce principe.

Nous augmentons les degrés de liberté de la locomotion d'un robot bipède en permettant la variation de l'orientation et de la durée des pas en ligne. Cependant, cela introduit des non-linéarités dans les contraintes de nos problèmes d'optimisation. Nous faisons des approximations de ces contraintes non-linéaires avec des contraintes linéaires sûres de sorte que la satisfaction des secondes implique la satisfaction des premières. Nous proposons une nouvelle méthode de résolution des problèmes non-linéaires (Optimisation Quadratique Successive Sûre) qui assure la faisabilité des itérations de Newton en utilisant cette redéfinition des contraintes.

Nous simulons la marche d'un robot bipède dans une foule pour évaluer la performance de nos lois des commandes. D'une part, nous réussissons à réduire (statistiquement) la quantité et la sévérité des collisions en comparaison avec la méthode de *passive safety*, spécialement dans les conditions d'incertitude de la marche du robot dans une foule. D'autre part, nous montrons des exemples de comportements typiques du robot, qui découlent de la liberté de choisir l'orientation et la durée des pas. Nous rapportons le coût de calcul de notre méthode de résolution des problèmes non-linéaires en comparaison avec une méthode standard. Nous montrons qu'une seule itération de Newton est nécessaire pour arriver à une solution faisable, mais que le coût de calcul dépend du nombre de factorisations de *l'active set* dont nous avons besoin pour arriver à *l'active set* optimal.

Acknowledgements

First and foremost, I thank Jesus, my Lord and Saviour. My life was changed when I met Him and He is the one who has made all of this possible. Let all the glory be to Him.

I would like to give very special thanks to Pierre-Brice Wieber, my supervisor. He is the person who opened me the doors to the world of research four years ago and with him I realized how much I love doing this. He is a wise and brilliant man. May God bless him, his family and his work.

I thank my dad, my mom, Karem (la Gorda) and Gustavo (el Feo). They are my family, joy and home. I thank my sweet, princess' hands Eva (la Beba) and my second mom Eleyda, I cannot be grateful enough with you. I thank everyone at the church *Le Refuge*, specially Mark and Dalene. They have become my second family and my new home here in France. To my friend Valentin.

To my colleagues: Luciano (Gerva), Nahuel (Tucu) and Matteo (Ciocca). To Jose Grimaldo (da Silva Filho). You are not getting away from me that easily! I will stay around for some months here! To everyone at Inria and at Grenoble who has made it an excellent adventure for the last four years.

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Summary of contributions	2
1.2.1	List of publications	2
1.3	Outline	2
1.4	Notation	3
1.4.1	Functions	3
1.4.2	Sets	3
1.4.3	Vectors and matrices	3
1.4.4	Software	3
I	Design aspects	4
2	Modeling a biped robot	5
2.1	Introduction	5
2.2	Mechanical model of a biped robot	5
2.3	Walking motion control	6
2.3.1	Center of Pressure	7
2.4	Conclusion	10
3	Control design	11
3.1	Introduction	11
3.2	Specifications of the walking motion	11
3.2.1	Objectives	11
3.2.2	Constraints	12
3.2.3	Classical approach	13
3.3	Walking motion generation as a control problem	13
3.4	Reasoning on an infinite horizon	14
3.5	Reasoning on a finite horizon	14
3.5.1	Avoiding failures	15
3.5.2	Avoiding collisions	15
3.6	Model Predictive Control	15
3.6.1	Dynamics	17
3.6.2	Cost function	17
3.6.3	Constraints	18
3.7	Conclusion	18

4	Safe navigation strategies	20
4.1	Introduction	20
4.2	Setting	20
4.2.1	Motion of the robot	20
4.2.2	Motion of the crowd	20
4.3	Collision mitigation	23
4.4	Control design for collision mitigation	23
4.4.1	Maximal Collision Delay	23
4.4.2	Minimal Collision Velocity	24
4.5	Hierarchy of time for failures and collision avoidance	25
4.5.1	Passive safety	25
4.5.2	Variations of collision mitigation	25
4.6	Ethical considerations	26
4.7	Conclusion	28
II	Numerical aspects	29
5	Lexicographic Programming	30
5.1	Introduction	30
5.2	Lexicographic least-squares programming	31
5.2.1	Null-space projections	31
5.2.2	Cascade of QPs	32
5.2.3	Advanced decompositions	32
5.3	Solutions near singularities	32
5.4	Conclusion	34
6	Safe Sequential Quadratic Programming	35
6.1	Introduction	35
6.2	Sequential Quadratic Programming	35
6.2.1	Active set methods	36
6.2.2	Warm-start	37
6.2.3	The Hessian of the Lagrangian	37
6.2.4	Merit functions	38
6.2.5	Maratos effect and second-order correction	39
6.2.6	A line search algorithm	39
6.3	Feasible Sequential Quadratic Programming	40
6.4	Convexification of constraints	41
6.4.1	Convexification by transformation	41
6.4.2	Convexification by polytopic approximation	42
6.5	Safe Sequential Quadratic Programming	43
6.5.1	A trust-region algorithm	43
6.5.2	Termination and convergence	44
6.6	Conclusion	46
7	Nonlinear Model Predictive Control for biped walking	47
7.1	Introduction	47
7.2	Motion of the CoM	47
7.2.1	Preview horizon	48
7.3	Position, orientation and duration of the steps	48
7.4	Constraints of walking	49

7.4.1	Kinematics and dynamics	49
7.4.2	Collision avoidance with the crowd	56
7.4.3	Terminal constraint for stopping	56
7.5	Objectives of the walk	57
7.6	Optimal Control Problem	58
7.6.1	NLP formulation	58
7.6.2	Safe SQP approach	59
7.7	Conclusion	59
8	Simulations	60
8.1	Introduction	60
8.2	Safe navigation strategies	60
8.2.1	No uncertainty in the motion of the crowd	61
8.2.2	Uncertainty in the motion of the crowd	63
8.3	Safe Sequential Quadratic Programming	66
8.3.1	Adapting the duration of the steps	66
8.3.2	Adapting the rotation of the steps	67
8.3.3	Comparison with SQP	69
8.4	Conclusion	75
9	Conclusion	76
9.1	Summary	76
9.2	Perspectives	76
9.2.1	Safe navigation strategies	77
9.2.2	Lexicographic programming	77
9.2.3	Safe SQP	77
9.2.4	Nonlinear MPC for biped walking	77
9.2.5	Simulations	78
Appendix A Bernstein polynomials		79
Appendix B Concepts of convex analysis		80
B.1	Sets and polyhedra	80
B.2	Functions	81
Appendix C Warm start of MPC		82
C.1	MPC as a sequence of neighboring SQPs	82
C.1.1	Linearization of the collision avoidance constraint	82

List of Figures

2.1	Schematic bipedal gait cycle [Sherikov 2016, Chapter 2]. Reprinted with permission.	7
3.1	Shift of the preview horizon in MPC [Sherikov 2016, Chapter 4]. Adapted with permission.	16
4.1	Walking against a crowd.	21
4.2	Two different predictions of the motion of a person: without uncertainty (left) and with uncertainty in velocity (right).	23
7.1	Correspondence between samples and steps in the preview horizon with $N_s = 8$	49
7.2	Nonlinear kinematic and dynamic constraints of the robot.	51
7.3	Safe polytopic approximations of the dynamic and kinematic constraints that depend on θ	52
7.4	Boundaries of $\mathbf{x}_{(i k)}(\tau)$ defined as $\text{conv}\{\beta_{i0}, \beta_{i1}, \beta_{i2}\}$	54
7.5	Future trajectories of the CoP for different τ	55
8.1	Failure rate and median anticipation time for different speeds of the crowd and different durations of the horizon.	62
8.2	Adapting the step duration while walking in a crowd.	67
8.3	Adapting the rotation of the feet of the robot while walking in a crowd.	68
8.4	The robot walking with no crowd under a Safe SQP scheme.	71
8.5	The robot walking with no crowd under a SQP scheme.	72
8.6	The robot walking in a crowd under a Safe SQP scheme.	73
8.7	The robot walking in a crowd under a SQP scheme.	74

List of Tables

8.1	Parameters of biped robot and the persons in the crowd	60
8.2	Common parameters	61
8.3	Parameters of simulations with no uncertainty	61
8.4	Parameters of simulations with uncertainty	63
8.5	Collisions of passive safety strategies	64
8.6	Collisions of collision mitigation strategies	65
8.7	Parameters of simulation with adaptation of the step duration	66
8.8	Parameters of simulation with adaptation of step rotation	69
8.9	Parameters of simulation for comparison with SQP	69

List of Algorithms

1	Cascade of QPs	33
2	Line-search SQP	39
3	Trust-region Safe SQP	45

List of Acronyms

CoM	Center of Mass
CoP	Center of Pressure
CP	Capture Point
CPU	Central Processing Unit
FoV	Field of View
FSQP	Feasible Sequential Quadratic Programming
HRM	Human Robot Motion
ICS	Inevitable Collision States
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LTI	Linear Time Invariant
MCD	Maximal Collision Delay
MCV	Minimal Collision Velocity
MPC	Model Predictive Control
NLP	Nonlinear Programming
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
QP	Quadratic Program
RHC	Receding Horizon Control
SQP	Sequential Quadratic Program
Safe SQP	Safe Sequential Quadratic Program
ZMP	Zero Moment Point

Chapter 1

Introduction

1.1 Motivations

Humanoids are the ultimate robots. We have dreamed for long time in science fiction works of creating machines that look, move and behave like us. We give them arms and hands so they can grab, lift, pinch, twist and manipulate tools. We give them a head with eyes and ears so they can perceive in ways similar to ours. And a pair of legs and feet should allow them to walk, run, jump and hop. It is a challenge to make all the different subsystems work individually and as a whole, as we have seen in the demonstrations of the latest humanoids (DRC 2015, Toro 2017, Atlas 2017). In this thesis we are interested in the bipedal locomotion problem.

A humanoid robot can remain in static balance indefinitely by standing still. But as soon as it starts walking, balance is no longer trivially achieved. The first problem that we need to solve is, therefore, to walk without falling. As anyone who has felt the sensation of “losing balance” with every step before an inevitable fall, just being able to avoid falling in the next step is not enough. We need to preview a balanced motion, ideally, on an infinite horizon of time. We resort to techniques from *viability theory* in order to make this problem tractable by previewing, after a couple of steps, to: 1) go back to static balance or 2) enter a cyclic walk.

In contrast to industrial robots (which are confined to closed spaces and out of the reach of people), we want humanoids to cohabit with us. We want them to collaborate with us at work, help us take care of patients at hospitals, clean our houses and maybe cook dinner. In these environments collisions can pose a real threat for both the people and the robot. The second problem we need to solve is to walk without colliding with the crowd. One limitation arises immediately: we cannot predict accurately where people are moving in the future and, therefore, we cannot avoid reaching all Inevitable Collision States (ICS). We can only preview a couple of seconds in the future and we need to make choices regarding what to do in case collisions seem inevitable. A standard approach is *passive safety*: in case of collision the robot is guaranteed to be at rest. A common criticism to this approach is, precisely, its passivity: the burden of avoiding a collision that seems inevitable is left to the other party. We believe that these situations can be, at least, partially mitigated by actively avoiding them for *as long as possible*. We propose a novel safe navigation strategy based on this premise.

In bipedal robotics, balance and collision avoidance are usually addressed separately resulting in systems that are less responsive to perturbations and sudden changes in the environment. We solve them simultaneously in a fast Model Predictive Control (MPC) approach using direct methods of optimal control.

In order to improve the mobility of the robot, we increase the number of degrees of freedom of locomotion by considering: the ability to reorient itself to face its direction of

walk and to adapt the duration of its steps. But by doing so, the optimization problems we need to solve become nonlinear. Traditional nonlinear methods, such as Sequential Quadratic Programming (SQP), can only guarantee the feasibility of the solution at convergence. More specialized methods, such as Feasible Sequential Quadratic Programming (FSQP), do ensure feasible iterates at the expense of increased computation costs. The third problem we want to solve is to improve the feasibility of the Newton iterates while keeping the computation costs low. We propose to linearize the constraints in a *safe* way such that the solution of the resulting problem is a solution of the original nonlinear problem.

1.2 Summary of contributions

These are three major contributions of this thesis:

- We defined *collision mitigation* as a safe navigation strategy in a crowd. Furthermore, we designed a controller that synthesizes motion trajectories that satisfy this definition.
- We proposed a new algorithm, Safe SQP, to solve nonlinear optimization problems featuring feasible iterates. It is based on the redefinition of the nonlinear constraints of the problem by its safe polytopic approximations.
- We defined safe polytopic approximations to some of the nonlinear problems that arise in biped walking: adaptive step rotation and duration.

1.2.1 List of publications

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

- N. Bohórquez, A. Sherikov, D. Dimitrov and P.-B. Wieber. *Safe navigation strategies for a biped robot walking in a crowd*. In IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 379–386, Cancun, Mexico, November 2016. IEEE.
- N. Bohórquez and P.-B. Wieber. *Adaptive step duration in biped walking: a robust approach to nonlinear constraints*. In IEEE RAS International Conference on Humanoid Robots 2017, Birmingham, United Kingdom, November 2017.
- N. Bohórquez and P.-B. Wieber. *Adaptive step rotation in biped walking*. In IEEE/RSJ International Conference on Intelligent Robots and Systems 2018, Madrid, Spain, October 2018.

1.3 Outline

This thesis is divided in two parts: Part I, composed of Chapters 2-4, where we discuss the design aspects of making the robot walk in a crowd and their mathematical formulations; Part II, composed of Chapters 6-8, where we describe the numerical schemes we use in order to solve the problems we formulate in the first part.

We start the first part in Chapter 2 introducing the dynamical model of the robot. Then, in Chapter 3, we state the problem of generating walking trajectories in a crowd for this model as an Optimal Control Problem of finite horizon. We present a novel strategy for crowd navigation, in Chapter 4, that makes emphasis on safety with respect to collisions.

At the beginning of the second part, in Chapter 5, we introduce lexicographic programming as an important foundation of our work, both at the design and at the numerical levels. In Chapter 6 we describe a novel method to solve Nonlinear Programming problems that is

based on continuously making safe linear approximations of the constraints. In Chapter 7 we characterize the control scheme we use for biped walking in a crowd. We present results on the numerical evaluation of different aspects of this controller on Chapter 8. We conclude with Chapter 9 with reflections on the work done and on what remains to be investigated.

1.4 Notation

1.4.1 Functions

- Scalar-valued and vector-valued functions are written f, g, h .
- Vector-valued functions are written $\mathbf{f}, \mathbf{g}, \mathbf{h}$.
- Set-valued functions are written in calligraphic $\mathcal{F}, \mathcal{G}, \mathcal{H}$.
- Special functions are written in plain text: $\text{conv}\{\cdot\}$, $\text{diag}(\cdot)$, $\text{centroid}\{\cdot\}$.

$$\text{diag}(\mathbf{A}) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}.$$

- $\|\cdot\|_p$ refers to the L_p norm, for $p \in \{1, \dots, \infty\}$. When the subscript is not specified, *i.e.* $\|\cdot\|$, then it should be understood as the L_2 norm.

1.4.2 Sets

- Sets are written in calligraphic \mathcal{S}, \mathcal{P} .
- Special sets are written in blackboard bold $\mathbb{R}, \mathbb{N}, \mathbb{Z}$.

1.4.3 Vectors and matrices

- Matrices are written in uppercase and in bold: \mathbf{A}, \mathbf{M} .
- Vectors are written in lowercase and in bold: \mathbf{x}, \mathbf{y} .
- Scalars and constants are written in lowercase in regular font: $x, y, k, \epsilon, \delta$.
- Column vectors are written as:

$$\mathbf{x} = (x_1, \dots, x_n) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

- Row vectors are written as:

$$\mathbf{x} = (x_1, \dots, x_n)^T = [x_1 \ \cdots \ x_n].$$

1.4.4 Software

We write the name of programs and libraries with monospace font, *e.g.* LexLS, Octave.

Part I

Design aspects

Chapter 2

Modeling a biped robot

2.1 Introduction

The goal of this chapter is to introduce the dynamical model that we employ to generate walking trajectories for the robot.

We start by describing the generalized mechanical model of a biped robot, revealing its hybrid, complex and underactuated nature in Section 2.2. We then characterize the type of motion that we want to generate with it and do a small overview of the different methods in the literature that have been developed for that purpose in Section 2.3. We explain that, by making standard assumptions on the dynamics of the Center of Mass (CoM), we can linearly relate the acceleration of the CoM with the forces applied on the ground. This resulting dynamics are central in this thesis.

2.2 Mechanical model of a biped robot

Given generalized coordinates

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}, \quad (2.1)$$

where $\mathbf{q}_1 = (\alpha_1, \dots, \alpha_p) \in \mathbb{R}^p$ are joint angles and $\mathbf{q}_2 = (\mathbf{c}, \boldsymbol{\gamma}) \in \mathbb{R}^{3+3}$ is the position of the CoM and orientation of the trunk in $SO(3)$, we can write the generalized model of a biped robot with K contacts [Hurmuzlu 2004, Sherikov 2016, Wieber 2015] as follows:

$$\begin{cases} \begin{bmatrix} \mathbf{M}_1(\mathbf{q}) \\ \mathbf{M}_2(\mathbf{q}) \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{n}_1(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{n}_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1(\mathbf{q}) \\ \mathbf{g}_2(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta} \\ \mathbf{0} \end{bmatrix} + \sum_{i=1}^K \begin{bmatrix} \mathbf{J}_{i,1}^T \\ \mathbf{J}_{i,2}^T \end{bmatrix} \boldsymbol{\lambda}_i, & (2.2a) \\ r_i^n, \lambda_i^n \geq 0 \quad r_i^n \lambda_i^n = 0, & (2.2b) \\ \boldsymbol{\phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\eta}) \leq \mathbf{b}, & (2.2c) \\ (\mathbf{q}^+, \dot{\mathbf{q}}^+) = \Delta(\mathbf{q}^-, \dot{\mathbf{q}}^-), & (2.2d) \\ \kappa_i \lambda_i^n \geq \|\boldsymbol{\lambda}_i^t\|. & (2.2e) \end{cases}$$

where $\boldsymbol{\eta} \in \mathbb{R}^p$ is the vector of joint torques, $\boldsymbol{\lambda}_i \in \mathbb{R}^3$ is the force at the i^{th} contact and $\mathbf{r}_i \in \mathbb{R}^3$ is the position of the i^{th} contact. Superscripts $(\cdot)^t$ and $(\cdot)^n$ denote tangential and normal components of vectors with respect to the contact surfaces. Equations (2.2a) to (2.2e) read as follows:

(2.2a) Lagrangian mechanics. In here, each term represents

$$\left\{ \begin{array}{l} \mathbf{M}(\mathbf{q}) = \begin{bmatrix} \mathbf{M}_1(\mathbf{q}) \\ \mathbf{M}_2(\mathbf{q}) \end{bmatrix} \in \mathbb{R}^{(p+6) \times (p+6)}, \quad \text{matrix of inertia} \\ \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{n}_1(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{n}_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \in \mathbb{R}^{p+6}, \quad \text{vector of Coriolis and centrifugal effects} \\ \mathbf{g}(\mathbf{q}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{q}) \\ \mathbf{g}_2(\mathbf{q}) \end{bmatrix} \in \mathbb{R}^{p+6}, \quad \text{vector of gravity} \\ \mathbf{J}_i^T = \begin{bmatrix} \mathbf{J}_{i,1}^T \\ \mathbf{J}_{i,2}^T \end{bmatrix} \in \mathbb{R}^{(p+6) \times 3}, \quad \text{translational Jacobian of the } i^{\text{th}} \text{ contact point} \end{array} \right.$$

(2.2b) Complementarity of contacts and forces. It ensures that forces can only be applied to contacts already made.

(2.2c) Joint and torque limits. It enforces kinematic and task-related limits on the posture of the robot such as: collisions with the environment, self-collisions and joint motor limitations.

(2.2d) Impact map. It describes the relationship between the pre and post impact velocities of the system.

(2.2e) Friction constraints. It limits the forces that can be applied at the contacts. In here, κ_i is the friction coefficient of the i^{th} contact.

A biped robot is a hybrid, complex, underactuated dynamical system:

- Hybrid because the continuous dynamics (2.2a) are suddenly interrupted every time the robot makes a step (2.2d).
- Complex because of the many degrees of freedom of the kinematic chain. There can be many ways to accomplish objectives of the form

$$\mathbf{J}_{\text{part}}^T \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\text{part}}^T \dot{\mathbf{q}} = \boldsymbol{\pi}_{\text{ref}}, \quad (2.3)$$

where \mathbf{J}_{part} is the Jacobian of the intended part of the robot to be controlled, since not all degrees of freedom are constrained all the time.

- Underactuated because we cannot exert arbitrary accelerations to the CoM. They can only be achieved through forces that appear from contacts with the environment (2.2b). These forces are, moreover, constrained by friction (2.2e).

2.3 Walking motion control

Walking is a discrete sequence of alternating steps in which there is, at least, one foot on the ground at all times. A bipedal walking *gait* is presented schematically in Figure 2.1. A biped walker starts in *double support*, that is, with both feet in contact with the ground. Then it lifts the left leg and swings it in the air to a new position on the ground. While one of the legs is in the air the walker is in *single support*. The moment the swinging leg hits the ground produces an *impact* and the walker returns to the double support position again. Subsequently, the walker lifts the right leg from the ground and swings in the air to a new position coming back again to double support where the cycle is repeated.

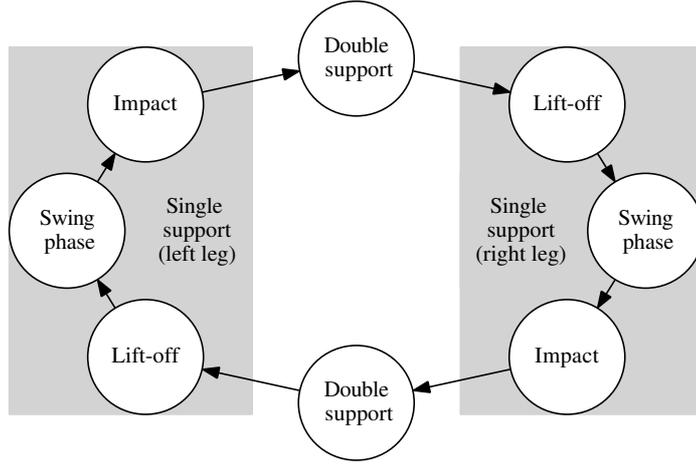


Figure 2.1: Schematic bipedal gait cycle [Sherikov 2016, Chapter 2]. Reprinted with permission.

In order to move the whole configuration \mathbf{q} of the robot in a walking motion, we want to affect the posture \mathbf{q}_1 (using joint torques $\boldsymbol{\eta}$) and produce contact forces $\{\boldsymbol{\lambda}_i\}$ with the contacts $\{\mathbf{r}_i\}$ with ground. In the literature there are different ways in which this can be done:

- Passive dynamic walking. McGeer started a line of research in which robots are designed with dynamics that approximately realize simple models of motion [McGeer 1990b, McGeer 1990a]. As a result, we can reason about \mathbf{q}_1 directly. The dynamic stability of the walking motion comes from the mechanical design (with little to no actuation) and not from feedback control, achieving remarkable “human-like” motion. However, applications are limited since their velocity cannot be changed at will. Recent results include the work of Collins *et al.* [Collins 2005].
- Hybrid-zero dynamics. Like in passive walking, we reason at the level of the posture \mathbf{q}_1 of the robot [Grizzle 2014], that is, we describe (at least partially) the path followed in the configuration space of the robot. However, in this case all joints are actuated except the feet. We define *virtual constraints* of the form

$$\mathbf{y} = \mathbf{q}_1 - \mathbf{h}(\theta), \quad (2.4)$$

where $\mathbf{h} : \mathbb{R} \rightarrow \mathbb{R}^p$ is the desired configuration of the robot as a function of the angle θ between the stance foot and the leg. We make the robot follow this path by doing a characterization of the zero dynamics $\mathbf{y} = \mathbf{0}$ during the swing phase. Then we develop an invariance condition at every impact of the swinging foot. 2D walkers following this scheme offer impressive results [Sreenath 2010], however, 3D walking is still under research [Chevallereau 2009].

2.3.1 Center of Pressure

Due to the structure of the Lagrangian dynamics (2.2), the global position \mathbf{c} and orientation γ are underactuated if no external forces $\boldsymbol{\lambda}_i$ are applied:

$$M_2(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{n}_2(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_2(\mathbf{q}) = \sum_i \mathbf{J}_{i,2}^T \boldsymbol{\lambda}_i, \quad (2.5)$$

We can reason about this part of the dynamics (\mathbf{q}_2), which can be described as a function of the Center of Pressure (CoP) of the ground reaction forces, and then make the rest of the

robot (\mathbf{q}_1) follow it. This decoupling is an effective method that has been used in a wide variety of robots [Takenaka 2009, Tajima 2009, Morisawa 2007] and is the method we employ in this thesis.

Forward kinematics

This subsection is largely based on the work of Wieber [Wieber 2015].

Equation (2.5) boils down to the Newton and Euler equations of motion of the robot taken as a whole [Wieber 2006]:

$$m(\ddot{\mathbf{c}} + \mathbf{g}) = \sum_i \boldsymbol{\lambda}_i, \quad (2.6a)$$

$$\dot{\mathbf{L}} = \sum_i (\mathbf{r}_i - \mathbf{c}) \times \boldsymbol{\lambda}_i, \quad (2.6b)$$

where m is the total mass of the robot and

$$\mathbf{L} = \sum_k (\mathbf{x}_k - \mathbf{c}) \times m_k \dot{\mathbf{x}}_k + \mathbf{I}_k \boldsymbol{\omega}_k, \quad (2.7)$$

is the angular momentum with respect to the whole robot. In this equation, \mathbf{x}_k , $\dot{\mathbf{x}}_k$ and $\boldsymbol{\omega}_k$ are the position, linear and angular velocities of the k^{th} part of the robot, while m_k and \mathbf{I}_k represent its mass and inertia tensor matrix (in global coordinates), respectively.

We derive the CoP from (2.6) by summing the Euler equation with the cross product of \mathbf{c} and the Newton equation

$$m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}} = \sum_i \mathbf{r}_i \times \boldsymbol{\lambda}_i, \quad (2.8)$$

and then dividing it by the z component of the Newton equation to obtain:

$$\frac{m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathbf{L}}}{m(\ddot{c}^z - g^z)} = \frac{\sum_i \mathbf{r}_i \times \boldsymbol{\lambda}_i}{\sum_i \lambda_i^z}. \quad (2.9)$$

Superscripts $(\cdot)^x, (\cdot)^y, (\cdot)^z$ represent the x, y and z components of a vector, respectively, in a given frame of reference.

Assumption 1. All contact points $\{\mathbf{r}_i\}$ lie on the same plane, the walking plane.

We set the world frame such that the normal vector of this plane is the z axis. For simplicity, this plane is located at zero height, therefore, $r_i^z = 0$ for all i . This allows us to simplify the cross products in (2.9) in the following way

$$\mathbf{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z} (\ddot{\mathbf{c}}^{x,y} + \mathbf{g}^{x,y}) + \frac{1}{m(\ddot{c}^z + g^z)} \mathbf{S} \dot{\mathbf{L}}^{x,y} = \frac{\sum_i \lambda_i^z \mathbf{r}_i^{x,y}}{\sum_i \lambda_i^z}, \quad (2.10)$$

where $\mathbf{S} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is a 90° rotation matrix. On the RHS we have the CoP $\mathbf{p}^{x,y}$ of the contact forces $\boldsymbol{\lambda}_i$

$$\mathbf{p}^{x,y} = \frac{\sum_i \lambda_i^z \mathbf{r}_i^{x,y}}{\sum_i \lambda_i^z}. \quad (2.11)$$

Due to the unilaterality of the contact forces with the ground (*i.e.* $\lambda_i^z \geq 0$), the CoP can only reside inside the *support polygon*

$$\mathbf{p}^{x,y} \in \text{conv} \{\mathbf{r}_i\}. \quad (2.12)$$

It can be shown that the horizontal momenta of the contact forces λ_i with respect to the CoP $\mathbf{p}^{x,y}$ are equal to zero:

$$\left[\sum_i (\mathbf{r}_i - \mathbf{p}) \times \lambda_i \right]^{x,y} = \sum_i (\mathbf{r}_i^{x,y} - \mathbf{p}^{x,y}) \times \lambda_i^z = 0. \quad (2.13)$$

This is why, under some conditions, the CoP is also called the Zero Moment Point (ZMP) [Vukobratović 2004].

Assumption 2. *The walking plane is horizontal.*

This assumption implies that $\mathbf{g}^{x,y} = \mathbf{0}$ and the dynamics of the CoM result in:

$$\mathbf{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z} \ddot{\mathbf{c}}^{x,y} + \frac{1}{m(\ddot{c}^z + g^z)} \mathbf{S} \dot{\mathbf{L}}^{x,y} = \mathbf{p}^{x,y}. \quad (2.14)$$

We can allow the vertical motion of the CoM in order to, for example, make the robot climb up and down stairs [Brasseur 2015]. However, in this thesis we consider that:

Assumption 3. *The CoM moves strictly horizontally above the ground.*

Therefore, c^z is constant and $\ddot{c}^z = 0$ and we obtain:

$$\mathbf{c}^{x,y} - \frac{c^z}{g^z} \ddot{\mathbf{c}}^{x,y} + \frac{1}{mg^z} \mathbf{S} \dot{\mathbf{L}}^{x,y} = \mathbf{p}^{x,y}. \quad (2.15)$$

Similarly, we could bound the variation of the angular momentum to, for example, account for possible external forces [Serra 2016], but we choose to set:

Assumption 4. *Variations of the angular momentum are equal to zero.*

We end up with a linear differential equation for the dynamics of the CoM:

$$\mathbf{c}^{x,y} - \frac{1}{\omega^2} \ddot{\mathbf{c}}^{x,y} = \mathbf{p}^{x,y}. \quad (2.16)$$

where

$$\omega = \sqrt{\frac{g^z}{c^z}}. \quad (2.17)$$

Inverse dynamics

This subsection is largely based on the work of Sherikov et al. [Sherikov 2014] and it is presented here only for completeness of our walking motion control, i.e. it is not necessary for the understanding of the rest of this thesis.

We use inverse dynamics to make the robot track the positions of the CoM $\mathbf{c}^{x,y}$ and the CoP $\mathbf{p}^{x,y}$ we computed with (2.16) [Fujimoto 1996].

Given any $\ddot{\mathbf{q}}$ and λ , we can find joint torques $\boldsymbol{\eta}$ satisfying the first part of the Lagrangian dynamics in (2.2). Therefore, we only need to find $\ddot{\mathbf{q}}$ and λ that satisfy (2.5) while requiring the joint angles to stay within their mechanical limits

$$\underline{\mathbf{q}}_1 \leq \mathbf{q}_1 + \tau \dot{\mathbf{q}}_1 + \frac{\tau^2}{2} \ddot{\mathbf{q}}_1 \leq \bar{\mathbf{q}}_1, \quad (2.18)$$

where $\tau > 0$ is the sampling time of the system.

In the following we consider that the robot is in single support. Let us consider the contact forces λ_i on the foot as a single wrench

$$\lambda = \mathbf{T} \begin{bmatrix} \lambda_s \\ \mu_s \end{bmatrix}, \quad (2.19)$$

with force λ_s and moment μ_s applied at the center of the foot $s \in \mathbb{R}^3$ on the ground and expressed in the local frame (with frame transformation matrix \mathbf{T}). In order to satisfy the Coulomb friction constraints, the force λ_s should be contained in the conic hull of the N unit vectors that approximate the Coulomb friction cone

$$\lambda_s \in \text{cone} \{v_1, \dots, v_N\}. \quad (2.20)$$

The moment about the z axis is limited as follows

$$|\mu_s^z| \leq \zeta \lambda_s^z, \quad (2.21)$$

where ζ is a torsional friction coefficient. The CoP is kept inside the support polygon with a constraint of the form

$$\mathbf{Q} \begin{bmatrix} \lambda_s \\ \mu_s \end{bmatrix} \in \text{conv} \{r_i\}. \quad (2.22)$$

We also keep fixed contacts with the ground by requiring the translational and rotational accelerations to be zero

$$\mathbf{J}_s \ddot{\mathbf{q}} + \dot{\mathbf{J}}_s \dot{\mathbf{q}} = \mathbf{0}. \quad (2.23)$$

Impacts that occur when the swinging foot of the robot lands on the ground are resolved by integrating the Lagrangian dynamics

$$\mathbf{M}(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-) = \mathbf{J}_s^T \lambda_{\text{impulse}}, \quad (2.24)$$

where λ_{impulse} is an impulsive force. The post-impact behavior is assumed to be

$$\mathbf{J}_s \dot{\mathbf{q}}^+ = \mathbf{0}. \quad (2.25)$$

Finally, the coupling between the motion of the CoM and the motion of the whole body is made through the x, y accelerations of the CoM $\ddot{\mathbf{c}}^{x,y}$ and the swinging foot $\ddot{\mathbf{f}} \in \mathbb{R}^3$

$$\mathbf{J}_{\text{com}} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\text{com}} \dot{\mathbf{q}} = \ddot{\mathbf{c}}^{x,y}, \quad (2.26a)$$

$$\mathbf{J}_{\text{foot}} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\text{foot}} \dot{\mathbf{q}} = \ddot{\mathbf{f}}. \quad (2.26b)$$

2.4 Conclusion

We presented in this chapter the ordinary differential equation (2.16) that relates the acceleration of the CoM with the forces applied to the ground. We obtained this equation after presenting the generalized mechanical model of a biped robot (2.2) and making standard assumptions on the motion of the CoM. For the rest of this thesis, we refer to the x, y components of the CoM $\mathbf{c}^{x,y}$ and the CoP $\mathbf{p}^{x,y}$ simply as \mathbf{c} and \mathbf{p} , respectively.

Chapter 3

Control design

3.1 Introduction

In this chapter we state the problem of a biped robot walking in a crowd as an Optimal Control Problem (OCP) of finite horizon. More specifically, we state it in the framework of Model Predictive Control (MPC), as it was first proposed by Kajita *et al.* [Kajita 2003].

In Section 3.2 we define the objectives and constraints of the motion of the CoM. We also do an overview of the approaches in the literature to biped robot navigation in environments filled with static and dynamic elements. Then, in Sections 3.3, 3.4 and 3.5, we frame our work in the context of Optimal Control making emphasis on avoiding failures and collisions in the future. Finally, we define an MPC controller for biped walking and describe some of its most important aspects to consider in Section 3.6.

3.2 Specifications of the walking motion

In order to make the biped robot walk we generate: a trajectory for the CoM $\mathbf{c} \in \mathbb{R}^2$, a trajectory for the CoP $\mathbf{p} \in \mathbb{R}^2$ (following the CoP dynamics we introduced in Section 2.3) and a sequence of steps with given position $\mathbf{s} \in \mathbb{R}^2$, orientation $\theta \in \mathbb{R}$ and duration. Then, as mentioned in Section 2.3, we compute a motion for the whole body of the robot that tracks the motion of the CoM and the sequence of steps and, at the same time, generates the desired forces on the ground (CoP).

3.2.1 Objectives

Our desired behavior for the robot is to:

- move the CoM smoothly at a reference speed,
- walk at a comfortable distance from each person in the crowd,
- align the feet with the direction the CoM is moving,
- keep the duration of the steps at energy-efficient value,
- keep the CoP at the center of the foot for improved balance.

These objectives will be discussed in depth in Section 7.5.

3.2.2 Constraints

While the objectives express the desired characteristics of the walk, the constraints enforce the conditions for the motion to be feasible. In this regard, two stability criteria for the CoP dynamics are pertinent [Wieber 2002]:

- A biped robot is *statically stable* on a flat ground if the CoM \mathbf{c} projects vertically inside the support polygon $\mathcal{P}(\mathbf{s}, \boldsymbol{\theta})$, that is

$$\mathbf{c} = \mathbf{p} \in \text{conv}\{\mathbf{r}_i\} \triangleq \mathcal{P}(\mathbf{s}, \boldsymbol{\theta}), \quad (3.1)$$

while $\dot{\mathbf{c}} = \mathbf{0}$. The support polygon is defined as the convex hull of all the contact points of the foot with the ground and its shape depends on the position \mathbf{s} and orientation $\boldsymbol{\theta}$ of the foot. In this regime, the positions of the CoM and the CoP remain bounded over time but it is of no interest for walking since the CoM does not move.

- A biped robot is *dynamically stable* on a flat ground if the CoP lies inside the support polygon

$$\mathbf{c} - \frac{1}{\omega^2} \ddot{\mathbf{c}} = \mathbf{p} \in \mathcal{P}(\mathbf{s}, \boldsymbol{\theta}), \quad (3.2)$$

which is necessarily true since the contact forces with the ground are unilateral (the robot can only push on the ground, not pull from it).

However, this last condition alone does not impede the CoM from diverging. A situation where the CoP is inside the support polygon but the CoM is accelerating away, *i.e.* $\int_{t_0}^{\infty} \|\mathbf{c}^{(n)}\|_p dt \rightarrow \infty$ for any $n > 0$, is a solution to (3.2). But before the CoM goes too far, the legs of the robot would be stretched beyond its limits. Therefore, the robot also needs to be *kinematically feasible*: the CoM must remain within a permissible region that depends on the position and orientation of the foot on the ground

$$\mathbf{c} \in \mathcal{C}(\mathbf{s}, \boldsymbol{\theta}). \quad (3.3)$$

The sequence of steps itself must not lead to self-collisions and must also comply with the kinematic structure of the robot. These two conditions can be represented as the possible places where the left foot can be placed with respect to the position and orientation of the right foot

$$\mathbf{s}_{\text{left}} \in \mathcal{S}(\mathbf{s}_{\text{right}}, \boldsymbol{\theta}_{\text{right}}), \quad (3.4)$$

and vice versa. We say that a biped robot *fails* if it is not dynamically stable and kinematically feasible.

We represent a crowd as a collection of M individuals. By restricting the physical interactions between the robot and the crowd to simply avoiding contact, we can allow abstractions in their physical representation: the occupied space of the j^{th} person is a ball of radius D_{person} . Let \mathcal{B}^j be the space occupied by the j^{th} person when it is at position \mathbf{m}^j

$$\mathcal{B}^j \triangleq \text{ball}(\mathbf{m}^j, D_{\text{person}}). \quad (3.5)$$

Let \mathcal{B} be the occupied space by all M persons in a crowd

$$\mathcal{B} = \bigcup_{j \in \{1, \dots, M\}} \mathcal{B}^j. \quad (3.6)$$

Let \mathcal{A} be the space occupied by a robot when the CoM is at position \mathbf{c}

$$\mathcal{A} \triangleq \text{ball}(\mathbf{c}, D_{\text{robot}}). \quad (3.7)$$

We say that a biped robot is in *collision* if

$$\mathcal{A} \cap \mathcal{B} \neq \emptyset. \quad (3.8)$$

3.2.3 Classical approach

In humanoid robotics, balance and collision avoidance are usually achieved in two separate steps:

1. *Plan kinematic sequence of steps.* We generate a feasible collision-free trajectory towards the goal. We usually start with a discretization of the state space centered around the position/orientation of the footsteps. Then a finite set of pre-computed state transitions of the robot between steps is stored in a library along with the heuristics to determine the costs of such transitions. The trajectories for traversing the graph from A to B without colliding can then be found by performing searches [Chestnutt 2003, Chestnutt 2005, Chestnutt 2007b], using sampling methods [Perrin 2012a, Perrin 2012b, Liu 2012] or by being exhaustive when possible [Ayaz 2009]. Others have formulated the problem as a mixed-integer optimization problem [Deits 2014, Kuindersma 2015]. Some of these approaches have been extended to account for dynamic environments [Chestnutt 2007a].

In order to tackle the complexity of the problem, some authors have developed tiered strategies and/or adaptive levels of planning detail [Hornung 2012, Liu 2012, Chestnutt 2004]. This is customary in classical motion planning [Fraichard 2012, Macek 2009] where we combine two levels of decision: global planning, which is coarse and approximate and usually only includes static elements in the environment [Latombe 1991, LaValle 2006]; and local planning, for higher fidelity and more expressive trajectories that also account for changing conditions in the environment [Khatib 1986, Fox 1997, Quinlan 1993].

2. *Plan dynamic trajectory.* We take the previously calculated sequence of steps and solve the equations of motion (2.2) to obtain a trajectory of the body that follows those steps while maintaining balance.

All of these methods provide impressive results under controlled conditions, but by separating the planning stages of steps and motion of the body, the reactivity of the robot to perturbations and to unforeseen persons is reduced. More recently, dynamical constraints have been taken into account at the step-planning phase using kinodynamical planners. These make sure that every two adjacent states of the system, and the transition between them, complies with dynamic and collision-avoidance constraints [Pham 2014, Fernbach 2017, Harada 2009, Ahn 2018]. However, these planners cannot yet be operated online and usually do not produce very dynamic motions.

3.3 Walking motion generation as a control problem

In this thesis we assume that:

Assumption 5. *The walking environment is convex or a global plan has been previously computed and the robot has to follow it.*

This presents us with a local problem where a control approach seems more appropriate. Let us introduce the state vector $\mathbf{x} = (\mathbf{c}, \dot{\mathbf{c}})$ and the control vector $\mathbf{u} = \mathbf{p}$. We can rewrite the dynamics (2.16) as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{c}} \\ \ddot{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{c}} \\ \omega^2(\mathbf{c} - \mathbf{u}) \end{bmatrix}. \quad (3.9)$$

The motion the robot realizes has to satisfy dynamical (3.2), kinematical (3.3),(3.4) and collision avoidance (3.8) constraints. We obtain the canonical nonlinear system:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t), \\ \mathbf{x} \in \mathcal{X}(t) \subset \mathbb{R}^n, \\ \mathbf{u} \in \mathcal{U}(t) \subset \mathbb{R}^m. \end{cases} \quad (3.10)$$

A crucial requirement is that not only we want to solve the problem now but also be able to solve the problem in the future. This is an observation that has been done in both planning and control: the need to anticipate future events to avoid putting the robot in situations where no action can be taken to avoid failures [Wieber 2002, Wieber 2008] and collisions [Fiorini 1998, Seder 2007, Large 2005].

3.4 Reasoning on an infinite horizon

A given state $\mathbf{x}(t_0)$ is said to be *viable* [Aubin 1991] if there exists at least one infinite evolution of control actions $\mathbf{u}(t) \in \mathcal{U}(t)$ that makes $\mathbf{x}(t) \in \mathcal{X}(t)$ for all $t > t_0$. The set $\mathcal{X}(t)$ describes the *viability constraints*. The state $\mathbf{x}(t_0)$ is *non-viable* if no trajectory of control actions exists that makes $\mathbf{x}(t) \in \mathcal{X}(t)$ for all $t > t_0$.

Viability is concerned about the existence of control laws that make invariant a subset of $\mathcal{X}(t)$, the *viability kernel* $\mathcal{V} \subset \mathcal{X}(t)$

$$\mathcal{V} \triangleq \{\mathbf{x}(t_0) : \forall t > t_0, \exists \mathbf{u}(t) \in \mathcal{U}(t), \mathbf{x}(t_0) + \int_{t_0}^t f(\mathbf{x}, \mathbf{u}, s) ds \in \mathcal{V}\}. \quad (3.11)$$

This is closely related to the concept of *Inevitable Collision States (ICS)* [Asama 2004]. It describes a set of states $\{\mathbf{x}(t_0)\}$ starting from which no infinite trajectory of control actions $\mathbf{u}(t) \in \mathcal{U}(t)$ exists that avoids collisions for all $t > t_0$.

The problem is that the system described by (2.2) is too complex to be able to define the viability kernel [Wieber 2008]. In general, such task is computationally intractable in all but the simplest systems. On the other hand, avoiding all ICS requires us to have absolute knowledge of the motion of people until infinity.

3.5 Reasoning on a finite horizon

In order to make the problem tractable, we focus instead on the long-term implications of what we can guarantee in the short term.

Let $\mathcal{T} \subseteq \mathcal{V}$ be a viable *target set* that is made invariant under a known control law $\mathbf{u}_{\mathcal{T}}(\cdot)$, that is,

$$\mathcal{T} \triangleq \{\mathbf{x}(t_f) : \forall t > t_f, \mathbf{x}(t_f) + \int_{t_f}^t f(\mathbf{x}, \mathbf{u}_{\mathcal{T}}, s) ds \in \mathcal{T}\}. \quad (3.12)$$

The problem is then reduced to reaching \mathcal{T} in finite time. A state $\mathbf{x}(t_0)$ is said to be *capturable* if there exists a time $t_f > t_0$ and a finite trajectory of control actions $\mathbf{u}(t) \in \mathcal{U}(t)$ such that $\mathbf{x}(t_f) \in \mathcal{T}$. The set of all capturable states is called the *capture basin* of \mathcal{T} , $\mathcal{C}_{\mathcal{T}} \subseteq \mathcal{V}$

$$\mathcal{C}_{\mathcal{T}} \triangleq \{\mathbf{x}(t_0) : \forall t \in (t_0, t_f], \exists \mathbf{u}(t) \in \mathcal{U}(t), \mathbf{x}(t_0) + \int_{t_0}^t f(\mathbf{x}, \mathbf{u}, s) ds \in \mathcal{X}(t), \mathbf{x}(t_f) \in \mathcal{T}\}. \quad (3.13)$$

We can compute a single capturable motion, instead of the whole capture basin, from the initial state $\mathbf{x}(t_0)$ solving the **OCP**

$$J(\mathbf{u}) = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, s) ds + V(\mathbf{x}(t_f)), \quad (3.14)$$

bounded by system and capturability constraints

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad (3.15a)$$

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{u}, t), \quad (3.15b)$$

$$\mathbf{x}(t) \in \mathcal{X}(t) \subset \mathbb{R}^n, \quad (3.15c)$$

$$\mathbf{u}(t) \in \mathcal{U}(t) \subset \mathbb{R}^m, \quad (3.15d)$$

$$\mathbf{x}(t_f) \in \mathcal{T}. \quad (3.15e)$$

3.5.1 Avoiding failures

Cyclic walking

We can define a target set \mathcal{T}_Π for cyclic walking [Scianca 2016, Takenaka 2009]. This set is made invariant under a cyclic control law, that is a control law $\mathbf{u}_\Pi(\cdot)$ that satisfies

$$\exists T > 0 \quad \text{such that} \quad \mathbf{u}_\Pi(t+T) = \mathbf{u}_\Pi(t), \quad \mathbf{x}_\Pi(t+T) = \mathbf{x}_\Pi(t), \quad \forall t \geq 0. \quad (3.16)$$

Let $\eta = (\mathbf{x}_\Pi(\cdot), \mathbf{u}_\Pi(\cdot))$ be a periodic solution to (3.10). The *orbit* η is *stable* if trajectories starting near η stay near η . The orbit η is *asymptotically stable* if trajectories starting near η converge to η . The orbit is *exponentially stable* if it is asymptotically stable and the convergence is exponential [Hauser 1994].

Stopping

We can guarantee long-term feasibility by making the robot reach the target set \mathcal{T}_Ω defined as all states where the robot has stopped, *i.e.* where the robot can maintain balance without making an extra step. This set is made invariant under a feedback law $\mathbf{u}_\Omega(\cdot)$ of the form

$$k(\mathbf{x}). \quad (3.17)$$

3.5.2 Avoiding collisions

We want to avoid collisions all the time but it might not be possible for many reasons: impossibility of having absolute knowledge of where people are moving, kinematic/dynamical limitations of the robot, adversarial crowd motion, *etc.* Instead, we try to avoid collisions and, when we cannot, we ensure that we can stop (*i.e.* reach \mathcal{T}_Ω) before it happens. This is called *passive safety* [Macek 2009, Bouraine 2012]. It is safe in the sense that, if collisions happen, they will not be the fault of the robot. This corresponds to the *emergency stop* procedures required for industrial and personal care robots [ISO 2011, ISO 2014].

3.6 Model Predictive Control

We can observe that the capacity to stop is integral to both feasibility and collision avoidance, thanks to this, passive safety can be combined effortlessly with capturability.

In order to have an uninterrupted motion (instead of a stopping motion), we continuously postpone the time at which the robot should reach the stopped state \mathcal{T}_Ω by regularly recomputing (3.14) as we execute the previous optimal state and control trajectories $(\mathbf{x}(\cdot), \mathbf{u}(\cdot))$. In this way, we have a continuous motion of the robot where it is able to stop, at any moment, within no more than a time $|t_f - t_0|$. This is called Receding Horizon Control (RHC) or Model Predictive Control (MPC) [Rawlings 2016].

Due to the time-changing nature and complexity of the sets $\mathcal{X}(t)$ and $\mathcal{U}(t)$ we use direct methods to solve (3.14). We parametrize the control inputs $\mathbf{u}(t)$ as \mathbf{u}_k and the state $\mathbf{x}(t)$ as

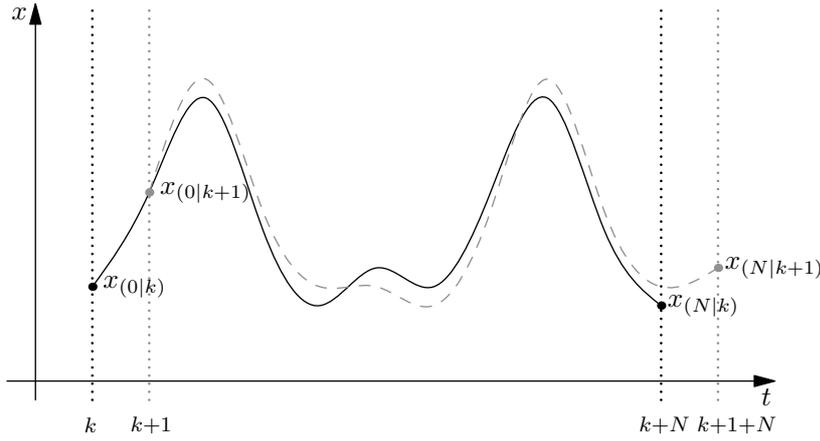


Figure 3.1: Shift of the preview horizon in MPC [Sherikov 2016, Chapter 4]. Adapted with permission.

\mathbf{x}_k , we approximate the infinite-dimensional cost functional $J(\mathbf{u})$ with a finite-dimensional cost function $J(\tilde{\mathbf{u}}_k)$ and the optimal control problem (3.14) is translated into the Nonlinear Programming (NLP) problem

$$\underset{\tilde{\mathbf{u}}_k}{\text{minimize}} \quad J(\tilde{\mathbf{u}}_k) = \sum_{i=0}^{N-1} l(\mathbf{x}_{(i|k)}, \mathbf{u}_{(i|k)}) + v(\mathbf{x}_{(N|k)}) \quad (3.18a)$$

$$\text{subject to} \quad \mathbf{x}_{(0|k)} = \mathbf{x}_0, \quad (3.18b)$$

$$\mathbf{x}_{(i+1|k)} = \hat{f}(\mathbf{x}_{(i|k)}, \mathbf{u}_{(i|k)}), \quad (3.18c)$$

$$\mathbf{x}_{(i|k)} \in \mathcal{X}(k+i), \quad (3.18d)$$

$$\mathbf{u}_{(i|k)} \in \mathcal{U}(k+i), \quad (3.18e)$$

$$\mathbf{x}_{(N|k)} \in \mathcal{T}_\Omega, \quad (3.18f)$$

written in a sequential approach. As shown in Figure 3.1, at the k^{th} MPC problem we compute a feedforward sequence of N control actions

$$\tilde{\mathbf{u}}_k = (\mathbf{u}_{(0|k)}, \dots, \mathbf{u}_{(N-1|k)}) \quad (3.19)$$

that steer the initial state $\mathbf{x}_{(0|k)}$ in a trajectory of states

$$\tilde{\mathbf{x}}_k = (\mathbf{x}_{(1|k)}, \dots, \mathbf{x}_{(N|k)}). \quad (3.20)$$

towards the target set \mathcal{T}_Ω following the dynamics of the system \hat{f} , respecting state \mathcal{X} and control \mathcal{U} constraints and minimizing a cost function J . We then apply the first control action $\mathbf{u}_{(0|k)}$ to the system

$$\mathbf{x}_{(1|k)} = \hat{f}(\mathbf{x}_{(0|k)}, \mathbf{u}_{(0|k)}), \quad (3.21)$$

and we feedback the next state to the controller

$$\mathbf{x}_{(0|k+1)} = \mathbf{x}_{(1|k)}, \quad (3.22)$$

in order to repeat the process at the $(k+1)^{\text{th}}$ MPC problem.

3.6.1 Dynamics

We discretize the motion of the CoM, given by the ODE (3.9), into a Linear Time Invariant (LTI) system of the form

$$\mathbf{x}_{(i+1|k)} = \mathbf{A}\mathbf{x}_{(i|k)} + \mathbf{B}\mathbf{u}_{(i|k)}, \quad (3.23)$$

using some of the methods described by Sherikov [Sherikov 2016, Chapter 4.2]. We discuss this aspect in detail in Section 7.2.

Uncertainties

We can also consider a more complete formulation

$$\mathbf{x}_{(i+1|k)} = \mathbf{A}\mathbf{x}_{(i|k)} + \mathbf{B}\mathbf{u}_{(i|k)} + \mathbf{v}_{(i|k)}, \quad (3.24)$$

where we introduce the uncertainty vector $\mathbf{v}_{(i|k)}$ in order to account for noisy sensors, imperfect actuators, unmodeled dynamics, inaccurate models of the environment, *etc.* Villa *et al.* discuss an MPC scheme for biped walking where these uncertainties are assumed to be bounded, $\mathbf{v} \in \mathcal{W}$, and a linear feedback law, $\mathbf{u} = \mathbf{u}_{\text{ref}} + K(\mathbf{x} - \mathbf{x}_{\text{ref}})$, is used to compensate for tracking errors. In here, $(\cdot)_{\text{ref}}$ refers to the nominal, undisturbed linear dynamics (3.23). The analysis of this system requires carefully constructed constraints and feedback gains [Villa 2017].

3.6.2 Cost function

The cost function (3.18a) is divided into two parts: a finite and an infinite horizon costs.

Finite horizon cost

Represented by the term $\sum_{i=0}^{N-1} l(\mathbf{x}_{(i|k)}, \mathbf{u}_{(i|k)})$. It measures how good we accomplish objectives during the preview horizon such as: to follow a desired velocity, to maintain a desired distance from people, *etc.*

The length of the preview horizon is a subject worth studying carefully. The longer it is the more expensive it is to solve the optimization problem. The shorter it is the more discrepancy there is between open and closed-loop operations. This discrepancy can even make the system unstable [Cannon 2018]. In biped walking it is more convenient to talk about the length of the horizon in terms of the physical steps the robot makes [Zaytsev 2015].

Infinite horizon cost

Represented by the term $v(\mathbf{x}_{(N|k)})$. It is used to make sure the close-loop operation matches the open-loop prediction by accounting for the costs after the end of the preview horizon.

With this scheme of two costs the control vector is, correspondingly, composed of two parts:

$$\text{Dual mode prediction} \begin{cases} \mathbf{u}_{(i|k)}, & i \in \{0, \dots, N-1\} \quad \text{optimization variables,} \\ \mathbf{K}\mathbf{x}_{(i|k)}, & i \in \{N, N+1, \dots\} \quad \text{feedback control.} \end{cases} \quad (3.25)$$

In order to have a finite number of variables in an infinite horizon, the mode 2 control vector is designed to be a feedback control.

Example 1. If we define the cost to be quadratic

$$J(\tilde{\mathbf{u}}_k) = \sum_{i=0}^{N-1} \left[\mathbf{x}_{(i|k)}^T \mathbf{Q} \mathbf{x}_{(i|k)} + \mathbf{u}_{(i|k)}^T \mathbf{R} \mathbf{u}_{(i|k)} \right] + \mathbf{x}_{(N|k)}^T \bar{\mathbf{Q}} \mathbf{x}_{(N|k)}, \quad (3.26)$$

and the system to be *LTI*, then the terminal weighting matrix $\bar{\mathbf{Q}}$, so that the infinite horizon cost is equal to the cost over the mode 2 prediction horizon, is the solution of the Lyapunov equation:

$$\bar{\mathbf{Q}} - (\mathbf{A} + \mathbf{BK})^T \bar{\mathbf{Q}} (\mathbf{A} + \mathbf{BK}) = \mathbf{Q} + \mathbf{K}^T \mathbf{R} \mathbf{K}. \quad (3.27)$$

3.6.3 Constraints

The ability to handle constraints (3.18d) (3.18e) is an important reason for the industrial interest of MPC. However, constraints can make the open and closed loop operations differ even when we consider an infinite horizon cost in the formulation. This is because the control sequence given by the feedback law in mode 2 might be infeasible and, therefore, the infinite optimal trajectory might not be realizable/viable.

Terminal constraint

The target set \mathcal{T}_Ω ensures the viability of the computed control trajectory or, in MPC terminology, *recursive feasibility*. If (3.18) can be solved then there is one feasible infinite trajectory that the robot can follow to reach \mathcal{T}_Ω and stay there.

Strong recursive feasibility

When we choose \mathcal{T}_Ω as the target set, we have a plan that allows the robot to stop. However, in MPC for biped walking we postpone, at every sample, the moment at which the robot stops in order to continue moving. The added difficulty is not only the target set \mathcal{T}_Ω being postponed but being changed from location every time a new step enters the preview horizon. We want to know when does solving (3.18) one time imply solving it all the time.

Let the set of all solutions of (3.18) starting from $\mathbf{x}_{(0|k)}$ be

$$\mathcal{W}_k \triangleq \{(\mathbf{x}_{(0|k)}, \tilde{\mathbf{u}}_k) : \tilde{\mathbf{u}}_k \text{ solves (3.18)}\}, \quad (3.28)$$

and its projections

$$(\mathcal{W}_0)_k \triangleq \{(\mathbf{x}_{(0|k)}, \boldsymbol{\kappa}) : \exists \tilde{\mathbf{u}}_k, (\mathbf{x}_{(0|k)}, \tilde{\mathbf{u}}_k) \in \mathcal{W}_k, \mathbf{u}_{(0|k)} = \boldsymbol{\kappa}\}, \quad (3.29a)$$

$$\mathcal{Z}_k \triangleq \{\mathbf{x}_{(0|k)} : \exists \tilde{\mathbf{u}}_k, (\mathbf{x}_{(0|k)}, \tilde{\mathbf{u}}_k) \in \mathcal{W}_k\}. \quad (3.29b)$$

The MPC is *strong recursive feasible* [Kerrigan 2001] if and only if

$$\forall k, \forall (\mathbf{x}_{(0|k)}, \boldsymbol{\kappa}) \in (\mathcal{W}_0)_k \implies \mathbf{x}_{(0|k+1)} = \mathbf{x}_{(1|k)} = \hat{\mathbf{f}}(\mathbf{x}_{(0|k)}, \boldsymbol{\kappa}) \in \mathcal{Z}_{k+1}. \quad (3.30)$$

Ciocca *et al.* provides a numerical proof of the computation of \mathcal{W} for a specific biped system [Ciocca 2017], which is an approximation of the capture basin $\mathcal{W} \subseteq \mathcal{C}_\mathcal{T}$.

3.7 Conclusion

We described the problem of avoiding failures and collisions in biped walking not only now but also in the future. We explored the relevant concepts of viability and ICS (in infinite horizon) and capturability and passive safety (in finite horizon). We unified the requirements

of balance preservation and collision avoidance by forcing the robot to stop in finite time. We presented [MPC](#) as a method to generate desired motions while respecting constraints. We described some key aspects to take into account in such control schemes and how they relate to biped walking.

Chapter 4

Safe navigation strategies

4.1 Introduction

This chapter contains preliminary work done in collaboration with Matteo Ciocca.

We introduce, in this chapter, *collision mitigation* strategies for crowd navigation. We address a common criticism over passive safety, which is precisely its “passivity”: in case of inevitable collision the responsibility of avoiding it is transferred to the other party. We show that collisions can be further mitigated by actively avoiding them, specially when there is uncertainty in the motion of people.

We begin by describing the dynamics of the crowd that we consider throughout this thesis in Section 4.2. We then define the two concepts that constitute collision mitigation in Section 4.3. In Section 4.4 and 4.5 we design a controller that generates motions that comply with these definitions. Finally, we briefly explore some ethical considerations of the choices we make in this chapter in Section 4.6. Part of the work presented here has been published by Bohórquez *et al.* [Bohórquez 2016].

4.2 Setting

4.2.1 Motion of the robot

We consider the LTI system $\mathbf{x}_i = \mathbf{A}\mathbf{x}_{i-1} + \mathbf{B}\mathbf{u}_{i-1}$ where the state \mathbf{x}_i is composed of the position \mathbf{c}_i and velocity $\dot{\mathbf{c}}_i$ of the CoM. At sampling time i , $\mathcal{A}(i)$ represents the space occupied by the robot. Let $\mathcal{X}(i)$ and $\mathcal{U}(i-1)$ be the state and control constraints. We define $\mathcal{X}(i)$ as the intersection of two sets

$$\mathcal{X}(i) \triangleq \mathcal{X}_1(i) \cap \mathcal{X}_2(i), \quad (4.1)$$

where $\mathcal{X}_1(i)$ represents kinematically and dynamically feasible states and $\mathcal{X}_2(i)$ represents an approximation of collision-free states in a finite horizon of N samples.

Definition 1 (Reachable set of the robot). *The reachable set of the system at sampling time $i \in \{1, \dots, N\}$ from initial state \mathbf{x}_0 is*

$$\mathcal{R}_{\text{robot}}(i) \triangleq \{\mathbf{x}_i : \forall j < i, \exists \mathbf{u}_j \in \mathcal{U}(j), \mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j + \mathbf{B}\mathbf{u}_j \in \mathcal{X}(j+1)\}. \quad (4.2)$$

4.2.2 Motion of the crowd

In order to compute the set $\mathcal{X}_2(i)$, we make a prediction of the motion of the people inside the *Field of View* (FoV) of the robot during the preview horizon.

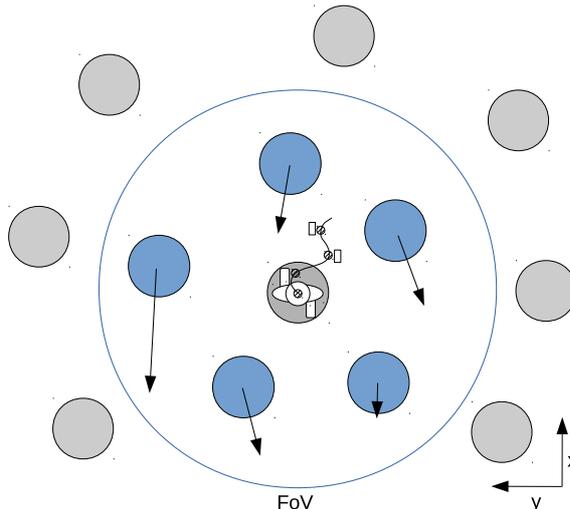


Figure 4.1: Walking against a crowd.

Definition 2 (Field of View). *The Field of View of the robot at sampling time i is the circle of radius D_{fov} centered at \mathbf{c}_i where persons can be perceived. The j^{th} person in the crowd is inside the FoV if*

$$\mathcal{B}^j(i) \cap \text{ball}(\mathbf{c}_i, D_{\text{fov}}) \neq \emptyset, \quad (4.3)$$

where $\mathcal{B}^j(i)$ represents the space occupied by the person.

As implied by (4.3), we do not consider occlusions in perception, making the robot aware of everybody within the FoV (see Figure 4.1). Let the set $\mathcal{F}(i)$ contain the indices of all the persons inside the FoV at the i^{th} sampling time.

The prediction of the future position of the crowd depends on its expected behavior towards the robot. We can consider different behaviors such as:

1. People are aware of their surroundings and actively try to collide with the robot.
2. People are aware of their surroundings and actively try to avoid collisions with the robot.
3. People are not aware of their surroundings and do not actively avoid collisions with the robot.

The first behavior represents a hostile or adversarial environment to the robot, as would be the case of a team game such as football. Techniques for navigation relying on game theory would be more appropriate in this case. The second behavior is what would be more reasonable to assume in most robotic applications. We are concerned on how the robot should move around and interact with people where collaboration is expected. This is the field of study of Human Robot Motion (HRM), where the main interest is the social acceptability and legibility of the motions. The crowd is assumed to be a dynamical system that can be, at least partially, controlled or affected by the motion of the robot. For example, we can study how effort is shared between an individual and a robot when their walking paths cross each other [Da Silva Filho 2017]. The third behavior describes a crowd that is neither collaborative nor adversarial. It is assumed that no action can be taken by the robot to make it change its motion.

The first behavior is not interesting since it is far from the description of our use case. The second behavior is closer to our use case but it seems to add unnecessary complexity

when determining a safe path to follow. This is because such crowd can avoid collisions even without the collaboration of the robot [Fraichard 2007]. The third behavior is more neutral and it appears to be better suited to test the safety of a walking controller. With a crowd that behaves in this way, the responsibility of avoiding collisions lies entirely on the robot. This “inattentive” crowd does not change its velocity in response to the motion of the robot.

Assumption 6. *People are not aware of their surroundings and do not actively avoid collisions with the robot or with each other.*

Let us consider the state vector $\boldsymbol{\eta}^j \in \mathbb{R}^4$ of the j^{th} person in the crowd as its position and velocity in the xy plane

$$\boldsymbol{\eta}^j = (m^{j,x}, \dot{m}^{j,x}, m^{j,y}, \dot{m}^{j,y}). \quad (4.4)$$

We model its motion with a double integrator

$$\dot{\boldsymbol{\eta}}^j(t) = \hat{\mathbf{\Gamma}}_1 \boldsymbol{\eta}^j(t) + \hat{\mathbf{\Gamma}}_2 \ddot{\boldsymbol{m}}^j(t). \quad (4.5)$$

where

$$\hat{\mathbf{\Gamma}}_1 = \text{diag}_2 \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right), \quad \hat{\mathbf{\Gamma}}_2 = \text{diag}_2 \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right). \quad (4.6)$$

We keep $\ddot{\boldsymbol{m}}^j(t)$ piecewise constant over intervals of time τ giving a solution of the form:

$$\boldsymbol{\eta}_i^j = \mathbf{\Gamma}_1 \boldsymbol{\eta}_{i-1}^j + \mathbf{\Gamma}_2 \ddot{\boldsymbol{m}}_{i-1}^j. \quad (4.7)$$

where

$$\mathbf{\Gamma}_1 = e^{\hat{\mathbf{\Gamma}}_1 \tau} = \text{diag}_2 \left(\begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \right), \quad \mathbf{\Gamma}_2 = \left(\int_0^\tau e^{\hat{\mathbf{\Gamma}}_1 t} dt \right) \hat{\mathbf{\Gamma}}_2 = \text{diag}_2 \left(\begin{bmatrix} \frac{\tau^2}{2} \\ \tau \end{bmatrix} \right). \quad (4.8)$$

The reachable sets of the crowd are commonly obtained by assuming that it moves with bounded velocities and/or bounded accelerations [LaValle 2006, Chapter 14.2] [Liu 2017]. We propose a slightly different approach: we assume that people maintain their velocity constant (*i.e.* accelerations equal to zero following Assumption 6) at least during the preview horizon

$$\forall i \in \{1, \dots, N\}, \quad \ddot{\boldsymbol{m}}_i^j = \mathbf{0}, \quad (4.9)$$

that is, for about one or two seconds. However, we acknowledge that the measurement of the initial state of the crowd has a quantifiable error that is reflected in the predictions. This means that the real initial state $\boldsymbol{\eta}_0^j$ (position and velocity) is in the vicinity of the measured initial state $\hat{\boldsymbol{\eta}}_0^j$. Given uncertainties in position D_p and in velocity D_v , we could have that $\boldsymbol{\eta}_0^j$ belongs to the ellipsoid:

$$\mathcal{E}(\hat{\boldsymbol{\eta}}_0^j) \triangleq \{ \boldsymbol{\eta} : (\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}_0^j)^T \mathbf{P}^{-1} (\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}_0^j) \leq 1 \}, \quad (4.10)$$

whose four semi-axes have lengths: D_p (for position in x, y) and D_v (for speed in x, y). The matrix \mathbf{P} is symmetric, positive definite and has eigenvalues D_p^2 and D_v^2 .

Definition 3 (Reachable set of the crowd). *The reachable set of the j^{th} person in the crowd at the sampling time i from the measured initial state $\hat{\boldsymbol{\eta}}_0^j$ is*

$$\mathcal{R}_{\text{person}}^j(i) \triangleq \{ \boldsymbol{\eta}_i^j : \exists \boldsymbol{\eta}_s^j \in \mathcal{E}(\hat{\boldsymbol{\eta}}_0^j), \forall s \leq i, \boldsymbol{\eta}_{s+1}^j = \mathbf{\Gamma}_1 \boldsymbol{\eta}_s^j \}. \quad (4.11)$$

The reachable set $\mathcal{R}_{\text{fov}}(i)$ of all individuals in the FoV at the sampling time i is

$$\mathcal{R}_{\text{fov}}(i) = \bigcup_{j \in \mathcal{F}(i)} \mathcal{R}_{\text{person}}^j(i). \quad (4.12)$$

The resulting reachable sets are shown in Figure 4.2. The gray areas are the predicted future positions of the person. When there is uncertainty in velocity, the set of possible positions where the person could be in the future increases in size with time. The set $\mathcal{X}_2(i)$ can be then defined as:

$$\mathcal{X}_2(i) \triangleq \{ \boldsymbol{x}_i : \mathcal{A}(i) \cap \mathcal{R}_{\text{fov}}(i) = \emptyset \}. \quad (4.13)$$

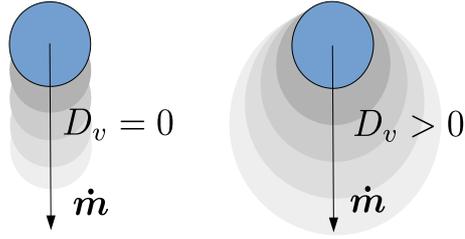


Figure 4.2: Two different predictions of the motion of a person: without uncertainty (left) and with uncertainty in velocity (right).

4.3 Collision mitigation

Definition 4 (Maximal Collision Delay). *If there exists $n \in \{1, \dots, N\}$ such that $\mathcal{R}_{\text{robot}}(n-1) \neq \emptyset$ and $\mathcal{R}_{\text{robot}}(n) = \emptyset$ then it is the Maximal Collision Delay (MCD).*

Remark 1. *In this formulation, colliding is a condition that we want to postpone as late in the future as possible. This is opposite to the minimal-time control problem [Van den Broeck 2011] in which we want to reach a target set \mathcal{T} as soon as possible, that is, we check if there exists an $n \in \{1, \dots, N\}$ such that $\forall i < n, \mathcal{R}_{\text{robot}}(i) \cap \mathcal{T} = \emptyset$ and $\mathcal{R}_{\text{robot}}(n) \cap \mathcal{T} \neq \emptyset$.*

The MCD does not exist if no collision can happen (e.g. there is no crowd) or if the robot can always avoid collisions. Let us suppose that the MCD exists and the collision happens. However, Definition 4 does not tell us how the collision is going to happen, that is, it does not tell us how to choose a specific trajectory until collision. We will choose a trajectory that minimizes the velocities at collision.

Definition 5 (Minimal Collision Velocity). *Let there be a collision at sampling time $n \in \{1, \dots, N\}$ with the j^{th} person in the crowd \mathbf{m}_n^j . Let $\hat{\mathbf{b}}$ be the normalized vector $(\mathbf{c}_n - \mathbf{m}_n^j) / \|\mathbf{c}_n - \mathbf{m}_n^j\|$. The trajectory $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ that solves*

$$\text{minimize } \hat{\mathbf{b}}^T (\dot{\mathbf{c}}_n - \dot{\mathbf{m}}_n^j), \quad (4.14)$$

produces the Minimal Collision Velocity (MCV).

4.4 Control design for collision mitigation

Problem 1. *Design a controller that computes a finite trajectory of states $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ that solves for the maximal collision delay and the minimal collision velocity problems.*

4.4.1 Maximal Collision Delay

If the maximal collision delay is $n \in \{1, \dots, N\}$ then, by definition, there exists a trajectory of states $(\mathbf{x}_1, \dots, \mathbf{x}_{n-1})$ that can avoid collisions all the time until then

$$\forall i \in \{1, \dots, n-1\}, \mathbf{x}_i \in \mathcal{X}_1(i) \cap \mathcal{X}_2(i), \quad (4.15)$$

because the intersection of the sets is not empty. If we define the metric

$$\delta_i = \{\|\mathbf{x}_i - \mathbf{q}\| : \mathbf{x}_i \in \mathcal{X}_1(i), \mathbf{q} \in \mathcal{X}_2(i)\}, \quad (4.16)$$

then we can rewrite (4.15) as

$$\forall i \in \{1, \dots, n-1\}, \delta_i = 0. \quad (4.17)$$

Being able to avoid collisions at all times until $n-1$ is a necessary condition for the maximal collision delay to be n or greater. A MCD of n also implies that at sampling time $k+n$ the intersection of the sets is empty, that is

$$\delta_n > 0. \quad (4.18)$$

We would like to maximize the value of the first i for which we cannot make $\delta_i = 0$. We do this by solving equations (4.17) (4.18) in a hierarchical way: for every sampling time $i \in \{1, \dots, N\}$ in the preview horizon we try to satisfy firstly $\delta_i = 0$ and secondly $\delta_{i+1} = 0$ without interfering with the first. That is, we want to

$$\text{lex minimize } (\delta_1, \dots, \delta_N), \quad (4.19)$$

which means that the vector $\boldsymbol{\delta} = (\delta_1, \dots, \delta_N)$ is to be minimized in a lexicographic order [Isermann 1982]. For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ the *strict lexicographic inequality* $\mathbf{a} \succ \mathbf{b}$ holds, if and only if, $a_i \succ b_i$ for $i = \min\{k \in \{1, \dots, n\} : a_k \neq b_k\}$. The *weak lexicographic inequality* $\mathbf{a} \succeq \mathbf{b}$ holds, if and only if, $\mathbf{a} \succ \mathbf{b}$ or $\mathbf{a} = \mathbf{b}$.

Example 2. The vectors $\mathbf{a} = (0, 0, 3, 0)$ and $\mathbf{b} = (0, 0, 1, 3)$ satisfy $\mathbf{a} \succ \mathbf{b}$.

In the *hierarchy of time* of (4.19), we avoid collisions for as much time as possible by giving higher priority to the present with respect to the future.

Remark 2. A *hierarchy of time similar to (4.19) but with inverted priorities*

$$\text{lex minimize } (\kappa_N, \dots, \kappa_1), \quad (4.20)$$

has been proposed by Al Homsy et al. for the problem of minimal-time control of manipulators [Homsy 2016], in agreement with the observation made in Remark 1. In here, the objective $\kappa_i = 0$ means reaching a target at sampling time i . We want to be able to reach it at sampling time N ; furthermore, if possible, we would like to reach it at sampling time $N-1$; furthermore, if possible, we would like to reach it at sampling time $N-2$, etc.

4.4.2 Minimal Collision Velocity

Let us suppose that a collision occurs between the robot and the j^{th} person in the crowd at sampling time n . Let us define the relative speed between the colliding bodies v_n at that time as

$$v_n = \hat{\mathbf{b}}^T (\dot{\mathbf{c}}_n - \dot{\mathbf{m}}_n^j). \quad (4.21)$$

With a sufficiently small sampling period τ we can approximate $\dot{\mathbf{c}}_n$ and $\dot{\mathbf{m}}_n$

$$\dot{\mathbf{c}}_n \approx \frac{\mathbf{c}_{n+1} - \mathbf{c}_n}{\tau}, \quad \dot{\mathbf{m}}_n \approx \frac{\mathbf{m}_{n+1} - \mathbf{m}_n}{\tau}, \quad (4.22)$$

and (4.21) becomes:

$$v_n \approx \hat{\mathbf{b}}^T \left(\frac{\mathbf{c}_{n+1} - \mathbf{c}_n}{\tau} - \frac{\mathbf{m}_{n+1} - \mathbf{m}_n}{\tau} \right). \quad (4.23)$$

Developing this expression we obtain

$$v_n \approx \tau^{-1} \hat{\mathbf{b}}^T [(\mathbf{c}_{n+1} - \mathbf{c}_n) - (\mathbf{m}_{n+1} - \mathbf{m}_n)], \quad (4.24a)$$

$$v_n \approx \tau^{-1} \hat{\mathbf{b}}^T [(\mathbf{c}_{n+1} - \mathbf{m}_{n+1}) - (\mathbf{c}_n - \mathbf{m}_n)]. \quad (4.24b)$$

The value $\hat{\mathbf{b}}^T(\mathbf{c}_i - \mathbf{m}_i)$ could be considered the *interpenetration* of colliding bodies δ_i if we restrict it to be either positive (in interpenetration) or zero (no contact)

$$v_n \approx \tau^{-1}(\delta_{n+1} - \delta_n). \quad (4.25)$$

Since we assume that the contact happens at sampling time n then $\delta_n = 0$. Furthermore, given that $\tau > 0$ we have that

$$\text{minimize } v_n \propto \text{minimize } \delta_{n+1}. \quad (4.26)$$

This is clearly a subproblem of (4.19), therefore, by solving the **MCD** problem we solve the **MCV** problem as well.

Remark 3. Problem (4.26) complies with standard contact models (e.g. the Hertz model [Johnson 1985]) that describe the impact forces as being proportional to the interpenetration δ of the masses in contact.

4.5 Hierarchy of time for failures and collision avoidance

4.5.1 Passive safety

The solution of Hierarchy 1 provides the robot with a trajectory to walk safely and stop after N sampling periods.

Hierarchy 1: Passive safety

1: Passive safety conditions:

- Feasibility and collision avoidance: $\forall i \in \{1, \dots, N\}, \mathbf{x}_i \in \mathcal{X}_1(i) \cap \mathcal{X}_2(i)$
- Stop at the end: $\mathbf{x}_N \in \mathcal{T}_\Omega$

2: Objectives: $f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$

With an MPC scheme we can continuously postpone the moment the robot eventually stops. If it is not possible to safely postpone the moment the robot stops, it can simply follow the last safe plan and stop before any collision occurs. We can raise an alarm to indicate that the robot is initiating an emergency stop and warn the surrounding crowd of the risk of collision N sampling periods in the future.

During the emergency stop, the robot will steadily approach the moment planned for the stop and the horizon will shrink accordingly. It might be beneficial, however, to continue re-evaluating the situation to check if, based on new measurements of the surrounding crowd, it is possible to postpone the stop of the robot, *as long as possible*. We can go, at each new sampling period, through a simple loop in order to

$$\text{maximize } \{N : N_{min} \leq N \leq N_{max}, \text{ Hierarchy 1 is feasible}\}. \quad (4.27)$$

4.5.2 Variations of collision mitigation

We could consider variations of the metric δ_i in (4.19) of the form

$$\delta_i = \{\|\mathbf{p} - \mathbf{q}\| : \mathbf{p} \in \mathcal{X}_1(i), \mathbf{q} \in \mathcal{X}_2(i)\}, \quad (4.28)$$

where the state \mathbf{x}_i lies on the line that connects \mathbf{p} and \mathbf{q}

$$\mathbf{x}_i = \theta\mathbf{p} + (1 - \theta)\mathbf{q}, \quad (4.29)$$

and is closer to one or the other depending on the value of θ . With $\theta = 1$ we make sure the robot is always feasible while postponing collisions as much as possible, a behavior that can be obtained by solving Hierarchy 3. With $\theta = 0.5$ the robot avoids collisions even at the expense of its own integrity, which can be done by solving Hierarchy 4. With $\theta = 0$ the robot is always collision-free while avoiding infeasibilities as much as possible, like the behavior obtained by solving Hierarchy 5.

4.6 Ethical considerations

Until now we have seen how we can formally define different behaviors of the robot towards the crowd. But why do we choose anyone of them? Under which criteria? Who is responsible for the actions the robot makes? What are our responsibilities towards the robot? These subtle questions seem to escape the domain of robotics and require a more formal grounding on ethics. This section, largely based on preliminary work by Wieber, motivates the discussion of some of these aspects and proposes practical methods to approach them.

Hierarchy 2: Asimov's laws

- 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 robot must be prepared for the conflicts that can happen between the different objectives assigned to it.

The American science-fiction author Isaac Asimov, who unknowingly invented the word *robotics* in 1941, suggested three implicit laws in the creation of all our tools: they must not be dangerous, they must fulfill their function and they must be durable. He proposed formally applying these three laws to the behavior of robots, in a form reminiscent of the biblical Decalogue: a robot will not be dangerous, a robot will obey and a robot will protect itself. But what is probably the most important is that he provided a formal procedure in case of conflict between these three laws, which is to consider that they do not have the same degree of importance: a robot will obey as much as that does not make it dangerous and a robot will protect itself as long as that does not make it dangerous or disobedient. This can be stated as the hierarchy shown in Hierarchy 2. This formal procedure for conflict management can be formulated in a mathematical way, Lexicographic Goal Programming [Escande 2014], which can be used effectively for the control of all kinds of robots, humanoids [Sherikov 2015], industrial [Al Homsy 2016], autonomous vehicles [Blumentals2017RR], with some precautions as some fundamental mathematical difficulties are not solved yet [Wieber 2017]. This ability to take into account multiple potentially incompatible objectives in complete safety is essential to deal with some potentially complex situations. We naturally continue to work actively on the different theoretical and practical aspects of this very promising approach.

A robot must not avoid collisions at all costs.

Following this approach, we can propose that our humanoid robot does everything possible to avoid collisions, even if it means losing its balance. The result is obviously not satisfactory, the robot sacrificing its balance on any occasion, even when facing a simply negligent behavior, not to mention a malicious behavior. We believe an opposite approach is preferable: the robot

Hierarchy 3: Collision Mitigation 1

- 1:** Feasibility: $\forall i \in \{1, \dots, N\}, \mathbf{x}_i \in \mathcal{X}_1(i)$
- 2:** Collision avoidance: $\mathbf{x}_1 \in \mathcal{X}_2(1)$
- \vdots
- N:** Collision avoidance: $\mathbf{x}_{N-1} \in \mathcal{X}_2(N-1)$
- N+1:** Collision avoidance: $\mathbf{x}_N \in \mathcal{X}_2(N)$,
Stop: $\mathbf{x}_N \in \mathcal{T}_\Omega$
- N+2:** Objectives: $f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$

Hierarchy 4: Collision Mitigation 2

- 1:** Feasibility and collision avoidance: $\mathbf{x}_1 \in \mathcal{X}_1(1) \cap \mathcal{X}_2(1)$
- \vdots
- N-1:** Feasibility and collision avoidance: $\mathbf{x}_{N-1} \in \mathcal{X}_1(N-1) \cap \mathcal{X}_2(N-1)$
- N:** Feasibility and collision avoidance: $\mathbf{x}_N \in \mathcal{X}_1(N) \cap \mathcal{X}_2(N)$
Stop: $\mathbf{x}_N \in \mathcal{T}_\Omega$
- N+1:** Objectives: $f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$

Hierarchy 5: Collision Mitigation 3

- 1:** Collision avoidance: $\forall i \in \{1, \dots, N\}, \mathbf{x}_i \in \mathcal{X}_2(i)$
- 2:** Feasibility: $\mathbf{x}_1 \in \mathcal{X}_1(1)$
- \vdots
- N:** Feasibility: $\mathbf{x}_{N-1} \in \mathcal{X}_1(N-1)$
- N+1:** Feasibility: $\mathbf{x}_N \in \mathcal{X}_1(N)$,
Stop: $\mathbf{x}_N \in \mathcal{T}_\Omega$
- N+2:** Objectives: $f(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})$

must postpone as much as possible the risk of collisions, but not to the point of losing its balance, as shown in Hierarchy 3. The programmer of the robot is thus in the unusual position of having to decide and justify that the robot does not avoid a collision even if it would have been formally capable to do so. What elements can support such a position?

We need a normative ethics for machines.

To understand the basis of the relationship that men have with robots, an instructive comparison can be made with animal rights, which can be seen in the absence of reciprocity as simply a duty for men, that of not being cruel, following the analysis proposed by Saint Thomas Aquinas and taken over by John Locke and Immanuel Kant. The notion of cruelty to a machine becomes a reality as soon as it is given a place in our social interactions, what is observed in all kinds of occasions for robots. One example among many others, the public of the quai Branly museum in Paris was careful to say goodbye to the robot Berenson, which did not conceal in any way its mechanical nature and insensitivity to such attention. As a machine to which man can not refrain from giving a social place, the robot must not accept immoral behavior against itself. It is based on this argument that we can argue that our humanoid robot navigating in a crowd must not agree to sacrifice its balance indiscriminately. We can note that the ethics of machines can only be deontological, that is defined by rules that must be followed, their consequences being often difficult to predict (robots having no will of their own, the ethics of virtue is totally off-topic). This imposes a pragmatic approach to ethics, in which this deontology must be evaluated and worked out in the light of the consequences observed in practice. This whole ethical reflection on the place we give to robots in human society and the behavior expected from them is obviously far from complete and we continue to work on it.

If we give to robots a place in our social interactions, we need to equip them for that.

In order for robots to interact in a coherent way, adapted to their human environment, we must provide them with models of human cognition. This can be done with traditional statistical models such as Hidden Markov Models, which can be adjusted to mimic previously recorded human behaviors [Vasquez 2009]. But as discussed previously, the result of such an approach is difficult to validate and, therefore, to use in a general way with full confidence. As a result, it seems necessary once again to work on theoretical models formally established and clearly defined, relying for example on the social comparison theory originally developed in 1954 by the American psycho-sociologist Leon Festinger (who proposed also the theory of cognitive dissonance), which can be used to reproduce faithfully the behavior of individuals in a crowd. It is with the same objective that we are also currently working on a theory of attention [Fraichard 2014].

Understanding under which conditions we can trust the behavior of a robot is a moral prerequisite, but also a practical one.

4.7 Conclusion

Starting from standard assumptions on the motion of the robot and conservative assumptions on the motion of the crowd, we defined *collision mitigation* strategies that make the robot actively avoid collisions as much as possible and reduce the velocities of collision when they are inevitable. We developed a lexicographic approach to generate such motions and we showed that it could also be used to generate passively safe ones. We finished by discussing the ethical implications of having strict priorities among the objectives of the robot.

Part II

Numerical aspects

Chapter 5

Lexicographic Programming

5.1 Introduction

In multi-objective optimization [Boyd 2004, Chapter 4] we want to minimize a vector-valued objective function $\mathbf{f} = (f_1, \dots, f_m)$, composed of m scalar functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, in a feasible set $\mathcal{X} \subseteq \mathbb{R}^n$

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{minimize}} \mathbf{f}(\mathbf{x}). \quad (5.1)$$

If the objectives are *noncompeting* then a solution $\mathbf{x}^* \in \mathcal{X}$ can be found where every f_i is minimized. If the objectives are, at least, partially competing then such solution does not exist and, instead, we are interested in the set of *Pareto minimal* points. A point $\mathbf{x}^{p_0} \in \mathcal{X}$ is a Pareto minimum if there does not exist another point $\mathbf{x} \in \mathcal{X}$ that satisfies

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}^{p_0}), \forall i \in \{1, \dots, m\}, \quad (5.2)$$

and $f_j(\mathbf{x}) < f_j(\mathbf{x}^{p_0})$ for at least one index j . The set of Pareto minimum points gives information about trade-offs among all the objectives that a decision maker can use later. How do we find Pareto minimal points? One standard technique is *scalarization* or *weighting* where we want to

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{minimize}} \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}). \quad (5.3)$$

The vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ is a *weight vector* that we can vary in order to obtain different Pareto optimal points of (5.1).

Theorem 1 ([Miettinen 1998, Theorem 3.1.2]). *The solution of problem (5.3) is Pareto optimal if the components of $\boldsymbol{\lambda}$ are positive, that is, if $\lambda_i > 0$ for all $i \in \{1, \dots, m\}$.*

Another technique is lexicographic programming where we want to

$$\text{lex minimize}_{\mathbf{x} \in \mathcal{X}} \mathbf{f}(\mathbf{x}). \quad (5.4)$$

It is more suitable when a natural ordering (that can be written pre-emptively) exists among the objectives. That is, when they can be solved “sequentially”: we first solve the most important one f_1 , then, among the set of solutions of f_1 , we solve the second objective in importance f_2 , etc.

Theorem 2 ([Miettinen 1998, Theorem 4.2.1]). *The solution of problem (5.4) is Pareto optimal.*

Both techniques have their own domains of applications [Tamiz 1995]. In our case, lexicographic optimization is pertinent in humanoid robotics because of the amount of objectives we want robots to accomplish and the clear natural ordering that appears among them [Dimitrov 2014]. The high number of degrees of freedom available to a humanoid robot provides kinematic redundancy that we can exploit for this purpose. For instance, we can ask the robot to: 1) maintain balance, 2) if possible, additionally, avoid collisions, 3) if possible, additionally, reach an object, 4) if possible, additionally, keep this object within sight. In this thesis we make extensive use of techniques from lexicographic programming, in particular, from lexicographic least-squares programming.

5.2 Lexicographic least-squares programming

The objectives we want our biped robot to complete can be expressed as linear equalities and inequalities. Given a matrix $\mathbf{A}_i \in \mathbb{R}^{m \times n}$ and vectors $\underline{\mathbf{b}}_i, \bar{\mathbf{b}}_i \in \mathbb{R}^m$, we want \mathbf{x} to satisfy the i^{th} objective

$$\underline{\mathbf{b}}_i \leq \mathbf{A}_i \mathbf{x} \leq \bar{\mathbf{b}}_i. \quad (5.5)$$

In this formulation, equalities can be written as $\underline{b}_{i,j} = \bar{b}_{i,j}$, for $j \in \{1, \dots, m\}$. Different objectives might conflict and, in such cases, at least one of them has to be relaxed. We can introduce the violation $\mathbf{v}_i \in \mathbb{R}^m$

$$\underline{\mathbf{b}}_i \leq \mathbf{A}_i \mathbf{x} - \mathbf{v}_i \leq \bar{\mathbf{b}}_i, \quad (5.6)$$

and minimize its norm

$$f_i(\mathbf{x}) = \|\mathbf{v}_i\|^2. \quad (5.7)$$

Note that the objective is satisfied if $f_i(\mathbf{x}) = 0$. The L_2 norm prevents unwanted irregularities in the control of the robot [Kanoun 2011].

Deciding which objectives can be relaxed is problem-dependent. In any case, we can assign a priority to each objective and minimize their violations accordingly, that is, as a lexicographic least squares problem.

Problem 2. *Let there be P levels of priorities, each one described by a triplet $(\mathbf{A}_i, \underline{\mathbf{b}}_i, \bar{\mathbf{b}}_i)$. Solve*

$$\text{lex minimize}_{\mathbf{x}, \tilde{\mathbf{v}}} (\|\mathbf{v}_1\|^2, \dots, \|\mathbf{v}_P\|^2) \quad (5.8a)$$

$$\text{subject to } \begin{bmatrix} \underline{\mathbf{b}}_1 \\ \vdots \\ \underline{\mathbf{b}}_P \end{bmatrix} \leq \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_P \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_P \end{bmatrix} \leq \begin{bmatrix} \bar{\mathbf{b}}_1 \\ \vdots \\ \bar{\mathbf{b}}_P \end{bmatrix}. \quad (5.8b)$$

where $\tilde{\mathbf{v}} = (\mathbf{v}_1, \dots, \mathbf{v}_P)$.

In the following we discuss three different methods that can be used to solve Problem 2: null-space projections, cascade of QPs and advanced decompositions.

5.2.1 Null-space projections

In the case of only equality-constrained systems

$$\text{lex minimize}_{\mathbf{x}, \tilde{\mathbf{r}}} (\|\mathbf{r}_1\|^2, \dots, \|\mathbf{r}_P\|^2) \quad (5.9a)$$

$$\text{subject to } \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_P \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_P \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_P \end{bmatrix}. \quad (5.9b)$$

where $\tilde{\mathbf{r}} = (\mathbf{r}_1, \dots, \mathbf{r}_P)$, the solution can be obtained with null space projections [Siciliano 1991]. For instance, one solution to the first level can be obtained with the pseudoinverse $(\cdot)^+$:

$$\mathbf{x}_1 = \mathbf{A}_1^+ \mathbf{b}_1, \quad (5.10)$$

which is the minimum of the solution set

$$\mathcal{S}_1 = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{r}_1\|^2. \quad (5.11)$$

This set permits the projection of arbitrary vectors \mathbf{z}_1 onto the null-space of \mathbf{A}_1 (with the operator $\mathbf{P}_1 = \mathbf{I} - \mathbf{A}_1^+ \mathbf{A}_1$) as follows

$$\mathcal{S}_1 = \{\mathbf{x}_1 + \mathbf{P}_1 \mathbf{z}_1 : \mathbf{z}_1 \in \mathbb{R}^n\}. \quad (5.12)$$

This vector is what gives freedom to accomplish the next objective in priority. The solution of the second level is constrained by the solution of the first level:

$$\mathcal{S}_2 = \underset{\mathbf{x} \in \mathcal{S}_1}{\operatorname{argmin}} \|\mathbf{r}_2\|^2, \quad (5.13)$$

and its minimum is, then again, obtained with the pseudo-inverse of \mathbf{A}_2 but projected onto the null-space of \mathbf{A}_1

$$\mathbf{x}_2 = \mathbf{x}_1 + (\mathbf{A}_2 \mathbf{P}_1)^+ (\mathbf{b}_2 - \mathbf{A}_2 \mathbf{x}_1). \quad (5.14)$$

We can extend this formulation to account for all objectives in the priority list.

5.2.2 Cascade of QPs

The limitation of the method of null-space projections is that it does not consider inequality objectives. These can be taken into account in Algorithm 1 proposed by Kanoun *et al.* [Kanoun 2011]. In here, we solve the i^{th} level of the hierarchy as a QP and we impose the solution as a constraint to the optimization problem at the $(i+1)^{\text{th}}$ level in a form of *cascade of QPs*.

5.2.3 Advanced decompositions

Solving Algorithm 1 can be expensive as we have to solve a separate QP for each level in the hierarchy. Escande *et al.* solve the inequality-constrained problem (5.8) by repeatedly solving the equality-constrained problem (5.9) with a subset of the constraints that are assumed to be active [Escande 2014]. Each of these subproblems is solved by simultaneously minimizing all violations $(\|\mathbf{r}_1\|^2, \dots, \|\mathbf{r}_P\|^2)$ while preserving the hierarchy with a *hierarchical complete orthogonal decomposition*. Being an active-set method, it has the advantage that problems can be warm-started.

Dimitrov *et al.* propose a faster matrix decomposition, called *lexicographic QR* or ℓ -QR, that efficiently exploits the structure of a hierarchical problem [Dimitrov 2015]. This has led to the development of a specialized solver, called LexLS, which is used throughout this thesis.

5.3 Solutions near singularities

Solvers for prioritized objectives may experience numerical issues near singularities [Siciliano 1991, Kanoun 2011]. In these cases, the Jacobian matrices of the objectives

Algorithm 1 Cascade of QPs**Require:** $(\mathbf{A}_i, \underline{\mathbf{b}}_i, \bar{\mathbf{b}}_i), \forall i \in \{1, \dots, P\}$ **Ensure:** lex minimize $(\|\mathbf{v}_1\|^2, \dots, \|\mathbf{v}_P\|^2)$ 1: **for all** $i \in \{1, \dots, P\}$ **do**2: **if** $i = 1$ **then**

$$\underset{\mathbf{x}, \mathbf{v}}{\text{minimize}} \quad \|\mathbf{v}\|^2 \quad (5.15a)$$

$$\text{subject to} \quad \underline{\mathbf{b}}_1 \leq \mathbf{A}_1 \mathbf{x} - \mathbf{v} \leq \bar{\mathbf{b}}_1 \quad (5.15b)$$

3: **else**

$$\underset{\mathbf{x}, \mathbf{v}}{\text{minimize}} \quad \|\mathbf{v}\|^2 \quad (5.16a)$$

$$\text{subject to} \quad \underline{\mathbf{b}}_1 \leq \mathbf{A}_1 \mathbf{x} - \mathbf{v}_1^* \leq \bar{\mathbf{b}}_1, \quad (5.16b)$$

$$\vdots \quad (5.16c)$$

$$\underline{\mathbf{b}}_{i-1} \leq \mathbf{A}_{i-1} \mathbf{x} - \mathbf{v}_{i-1}^* \leq \bar{\mathbf{b}}_{i-1}, \quad (5.16d)$$

$$\underline{\mathbf{b}}_i \leq \mathbf{A}_i \mathbf{x} - \mathbf{v} \leq \bar{\mathbf{b}}_i \quad (5.16e)$$

4: **end if**5: $\mathbf{v}_i^* \leftarrow \mathbf{v}$ 6: **end for**

become ill-conditioned and solutions can grow unbounded. Better behaved solutions can be obtained by adding standard *regularization* of the form

$$\underset{\mathbf{x}, \mathbf{v}}{\text{minimize}} \quad \|\mathbf{v}\|^2 + \lambda^2 \|\mathbf{x}\|^2 \quad (5.17a)$$

$$\text{subject to} \quad \underline{\mathbf{b}}_i \leq \mathbf{A}_i \mathbf{x} - \mathbf{v} \leq \bar{\mathbf{b}}_i, \quad (5.17b)$$

where $\lambda > 0$ is a *damping factor*. Care must be taken on the value of λ we choose: small values help maintain the numerical stability of the problem but large values can interfere with the expected behavior of the robot. Normally, we would want to adapt λ to be zero everywhere except near singularities.

A different approach is proposed by Wieber *et al.* in which the damping is, instead, chosen to be nonzero far from singularities in order to efficiently steer the Gauss-Newton steps towards a desired solution [Wieber 2017]. In such work, we analyze singularities that appear at the optimum (which are difficult to predict/detect before encountering them) and that delay the convergence to the desired solution. We introduce *artificial constraints*, such as regularization, that interfere “early” with the objectives. Alternatively, we can add a trust region of the form

$$\underset{\mathbf{x}, \mathbf{v}}{\text{minimize}} \quad \|\mathbf{v}\|^2 \quad (5.18a)$$

$$\text{subject to} \quad \underline{\mathbf{b}}_i \leq \mathbf{A}_i \mathbf{x} - \mathbf{v} \leq \bar{\mathbf{b}}_i, \quad (5.18b)$$

$$\|\mathbf{x}\|_\infty \leq \Delta, \quad (5.18c)$$

for some $\Delta > 0$, that forces the steps to go in the “right direction” before arriving to the singularities. Then again, adaptation of the trust region is needed. These artificial constraints methods have been demonstrated numerically to work well on a simple example system, however, they remain to be validated in more complex situations. The handling of singularities in redundancy resolution schemes is still an open area of research.

5.4 Conclusion

We presented lexicographic programming in the general context of multiobjective optimization. We showed that it is interesting and relevant in robotics when resolving redundancy on: kinematics, forces, time, *etc.* We did a small overview of the numerical methods that can be used to solve lexicographic problems and briefly discussed how to mitigate the singularities that can arise.

Chapter 6

Safe Sequential Quadratic Programming

6.1 Introduction

The Nonlinear Programming problem described in (3.18) can be rewritten in its canonical form as:

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \tag{6.1a}$$

$$\text{subject to} \quad c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E}, \tag{6.1b}$$

$$c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I}, \tag{6.1c}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are twice-continuously differentiable functions on \mathbb{R}^n and the sets \mathcal{E}, \mathcal{I} contain the indices of $\mathbf{c} = (c_1, \dots, c_m)$, for $m = |\mathcal{E}| + |\mathcal{I}|$, that are equality and inequality constrained, respectively. We denote the feasible set for (6.1) by

$$\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^n : \forall i \in \mathcal{E}, \forall j \in \mathcal{I}, c_i(\mathbf{x}) = 0, c_j(\mathbf{x}) \geq 0\}, \tag{6.2}$$

which we assume to be nonempty. We want to find a point \mathbf{x}^* that locally minimizes f and satisfies all constraints c_i .

6.2 Sequential Quadratic Programming

This section is largely based on the work of Nocedal et al. chapters 12, 15, 16 and 18 [Nocedal 2006].

First-order necessary conditions of optimality for problem (6.1), the *Karush-Kuhn-Tucker conditions* (KKT conditions), are described in Theorem 3.

Theorem 3 ([Nocedal 2006, Theorem 12.1]). *Suppose that \mathbf{x}^* is a local solution of (6.1), that the functions f and c_i are continuously differentiable, and that the Linear Independence Constraint Qualification (LICQ) holds at \mathbf{x}^* . Then there is a Lagrange multiplier vector $\boldsymbol{\lambda}^*$ such that the following conditions are satisfied at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$:*

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \tag{6.3a}$$

$$c_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E}, \tag{6.3b}$$

$$c_i(\mathbf{x}^*) \geq 0, \quad \forall i \in \mathcal{I}, \tag{6.3c}$$

$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I}, \tag{6.3d}$$

$$\lambda_i^* c_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}. \tag{6.3e}$$

In here, $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$ is the Lagrangian function of (6.1). Sequential Quadratic Programming (SQP) results from applying a Newton method to the KKT conditions of the problem. This is equivalent to iteratively solving the following *inequality-constrained QP* (IQP) subproblem

$$\underset{\mathbf{p}}{\text{minimize}} \quad f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p} \quad (6.4a)$$

$$\text{subject to} \quad \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{E} \quad (6.4b)$$

$$\nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) \geq 0, \quad i \in \mathcal{I}, \quad (6.4c)$$

evaluated at $(\mathbf{x}_k, \boldsymbol{\lambda}_k)$, the k^{th} iteration. The optimal step \mathbf{p}^* and Lagrange multipliers \mathbf{q}^* give us the point

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k + \mathbf{p}^* \\ \mathbf{q}^* \end{bmatrix}, \quad (6.5)$$

where we can create another quadratic subproblem at the $(k+1)^{\text{th}}$ iteration and find a new minimizer. We repeat the process until a convergence test is satisfied.

6.2.1 Active set methods

We denote by *active set* $\mathcal{A}_k(\mathbf{p})$, at a given \mathbf{p} , the indices of the constraints for which equality holds in the k^{th} subproblem

$$\mathcal{A}_k(\mathbf{p}) = \mathcal{E} \cup \{i \in \mathcal{I} : \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0\}. \quad (6.6)$$

A constraint c_i is said to be *active* if it is in the active set.

If we know in advance which constraints are active at the solution of (6.4), we can solve instead the following *equality-constrained QP* (EQP):

$$\underset{\mathbf{p}}{\text{minimize}} \quad f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p} \quad (6.7a)$$

$$\text{subject to} \quad \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{A}_k(\mathbf{p}^*). \quad (6.7b)$$

If we do not know $\mathcal{A}_k(\mathbf{p}^*)$ we need to determine it: we iteratively solve the EQP

$$\underset{\mathbf{p}}{\text{minimize}} \quad f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k) \mathbf{p} \quad (6.8a)$$

$$\text{subject to} \quad \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{W}, \quad (6.8b)$$

while adding and removing constraints from a selection of constraints \mathcal{W} , containing all \mathcal{E} and some $i \in \mathcal{I}$, until the optimality conditions of (6.4) are satisfied [Nocedal 2006, Chapter 16.5]. We can use knowledge of f, c_i and its derivatives to make a series of educated guesses of the *working set* \mathcal{W} avoiding the need to explore exhaustively all the possibilities. This is significant considering that the number of choices for \mathcal{W} may be large, up to $2^{|\mathcal{I}|}$, which accounts for the *combinatorial difficulty* of NLP.

Let the function $\mathbf{c}_{\mathcal{W}} : \mathbb{R}^n \rightarrow \mathbb{R}^{|\mathcal{W}|}$ be all the constraints c_i in the working set. Let the column vectors of the matrix $\mathbf{A}^T(\mathbf{x}_k)$ be the gradients of the constraints in the working set

$$\mathbf{A}^T(\mathbf{x}_k) = [\nabla c_i(\mathbf{x}_k)], \quad \forall i \in \mathcal{W}. \quad (6.9)$$

A solution $(\mathbf{p}^*, \mathbf{q}^*)$ for (6.8) can be found by considering the necessary conditions

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k) & -\mathbf{A}^T(\mathbf{x}_k) \\ \mathbf{A}(\mathbf{x}_k) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}^* \\ \mathbf{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla f(\mathbf{x}_k) \\ -\mathbf{c}_{\mathcal{W}}(\mathbf{x}_k) \end{bmatrix}. \quad (6.10)$$

If this solution also satisfies

$$\nabla c_i(\mathbf{x}_k)^T \mathbf{p}^* + c_i(\mathbf{x}_k) \geq 0, \quad \forall i \in \mathcal{I} \setminus \mathcal{W}, \quad (6.11a)$$

$$q_i^* \geq 0, \quad \forall i \in \mathcal{I} \cap \mathcal{W}, \quad (6.11b)$$

then it is also a solution of (6.4) and \mathcal{W} is equal to $\mathcal{A}(\mathbf{p}^*)$.

6.2.2 Warm-start

An important difficulty in solving an SQP with active set methods is determining which constraints are active at the solution of the original problem (6.1), the optimal active set \mathcal{A}^* . We can take the active set at the solution of the k^{th} subproblem, $\mathcal{A}_k(\mathbf{p}^*)$, as a guess of \mathcal{A}^* since they are both equal under the conditions described in Theorem 4.

Theorem 4 ([Nocedal 2006, Theorem 18.1]). *Suppose that \mathbf{x}^* is a local solution to (6.1) at which the KKT conditions are satisfied for some $\boldsymbol{\lambda}^*$. Suppose, too, that the LICQ qualification, the strict complementarity condition and the second-order sufficient conditions hold at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$. Then if $(\mathbf{x}_k, \boldsymbol{\lambda}_k)$ is sufficiently close to $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, there is a local solution of (6.4) whose active set $\mathcal{A}_k(\mathbf{p}^*)$ is the same as the optimal active set \mathcal{A}^* .*

The generally expensive IQP subproblems become more economical if we use information from the previous iteration as a guess of the solution of the current iteration. This is true as the SQP algorithm converges to the solution.

6.2.3 The Hessian of the Lagrangian

In the exact Newton method, the use of the matrix $\nabla_{xx}^2 \mathcal{L}$ provides a *quadratic convergence rate* in a neighborhood of the solution $\mathbf{z}^* = (\mathbf{x}^*, \boldsymbol{\lambda}^*)$, *i.e.* $\|\mathbf{z}_{k+1} - \mathbf{z}^*\| \leq \frac{\omega}{2} \|\mathbf{z}_k - \mathbf{z}^*\|^2$ when \mathbf{z}_k is sufficiently close to \mathbf{z}^* . However, computing the matrix of second derivatives $\nabla_{xx}^2 \mathcal{L}$ could be expensive. We can substitute it with a general Hessian matrix \mathbf{B}_k at the k^{th} subproblem

$$\underset{\mathbf{p}}{\text{minimize}} \quad f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B}_k \mathbf{p} \quad (6.12a)$$

$$\text{subject to} \quad \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{W}, \quad (6.12b)$$

which we can choose to be any of the following [Diehl 2017]:

- **Quasi-Newton BFGS.** In the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method we approximate \mathbf{B}_{k+1} to $\nabla_{xx}^2 \mathcal{L}(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1})$ in the direction $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ by finding a matrix that satisfies the *secant condition*

$$\mathbf{B}_{k+1} \mathbf{s}_k = \nabla_x \mathcal{L}(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1}) - \nabla_x \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_{k+1}), \quad (6.13)$$

and that minimizes the distance to \mathbf{B}_k in the Frobenious norm sense. This method allows us to obtain a *superlinear convergence rate* in a neighborhood of the solution \mathbf{z}^* , *i.e.* $\|\mathbf{z}_{k+1} - \mathbf{z}^*\| \leq \kappa \|\mathbf{z}_k - \mathbf{z}^*\|$ with $\kappa \rightarrow 0$ and it is less expensive than the exact Hessian since only the first derivatives of the objective function and the constraints are needed.

- **Gauss-Newton.** In the special case of a nonlinear least-squares problem, *i.e.* where $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{F}(\mathbf{x})\|_2^2$, each EQP subproblem has the form

$$\underset{\mathbf{p}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{F}(\mathbf{x}_k) + \nabla \mathbf{F}(\mathbf{x}_k)^T \mathbf{p}\|_2^2 \quad (6.14a)$$

$$\text{subject to} \quad \nabla c_i(\mathbf{x}_k)^T \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{W}. \quad (6.14b)$$

Rewriting the objective function as follows:

$$\frac{1}{2} \underbrace{\mathbf{F}(\mathbf{x}_k)^T \mathbf{F}(\mathbf{x}_k)}_{f(\mathbf{x}_k)} + \underbrace{\mathbf{F}(\mathbf{x}_k)^T \nabla \mathbf{F}(\mathbf{x}_k)^T}_{\nabla f(\mathbf{x}_k)^T} \mathbf{p} + \frac{1}{2} \mathbf{p}^T \underbrace{\nabla \mathbf{F}(\mathbf{x}_k) \nabla \mathbf{F}(\mathbf{x}_k)^T}_{\mathbf{B}_k} \mathbf{p}, \quad (6.15)$$

reveals that the Gauss-Newton Hessian matrix does not depend on the multipliers $\boldsymbol{\lambda}_k$ but only on first derivatives of $\mathbf{F}(\mathbf{x}_k)$. Furthermore, it is symmetric and has only non-zero eigenvalues, therefore, it is convex. It is related to the exact Hessian in the following way:

$$\nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k) = \nabla \mathbf{F}(\mathbf{x}_k) \nabla \mathbf{F}(\mathbf{x}_k)^T + \sum_i \mathbf{F}_i(\mathbf{x}_k) \nabla^2 \mathbf{F}_i(\mathbf{x}_k) + \sum_i \lambda_i \nabla^2 c_i(\mathbf{x}_k). \quad (6.16)$$

The difference between the exact Hessian and the Gauss-Newton Hessian is of order $O(\|\mathbf{F}(\mathbf{x}^*)\|)$. This Hessian allows us to obtain a *linear convergence rate* in a neighborhood of the solution \mathbf{z}^* , *i.e.* $\|\mathbf{z}_{k+1} - \mathbf{z}^*\| \leq \kappa \|\mathbf{z}_k - \mathbf{z}^*\|$ with $\kappa = O(\|\mathbf{F}(\mathbf{x}^*)\|)$; hence, it converges fast if the residuals $\mathbf{F}(\mathbf{x}^*)$ are small.

6.2.4 Merit functions

The solution \mathbf{p}^* that we obtain should be a good compromise between minimizing the objective function and satisfying constraints. We evaluate these (often competing) goals with merit functions. We use an L_1 merit function of the form

$$\phi_1(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu \sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \mu \sum_{i \in \mathcal{I}} [c_i(\mathbf{x})]^{-}, \quad (6.17)$$

despite being non-differentiable, because it is *exact*: there is a positive scalar μ^* such that for any $\mu > \mu^*$ any local solution of (6.1) is a local minimizer of $\phi_1(\mathbf{x}, \mu)$. In here we use the notation $[z]^{-} = \max\{0, -z\}$.

We accept a step if the actual decrease of ϕ_1 is not too small relative to the predicted decrease. In particular, we accept the step $\alpha \mathbf{p}^*$ if the following sufficient decrease condition holds:

$$\phi_1(\mathbf{x}_k + \alpha \mathbf{p}^*, \mu) \leq \phi_1(\mathbf{x}_k, \mu) + \eta \alpha D(\phi_1(\mathbf{x}_k, \mu), \mathbf{p}^*), \quad \eta \in (0, 1), \quad (6.18)$$

where $D(\phi_1(\mathbf{x}_k, \mu), \mathbf{p})$ is the directional derivative of ϕ_1 in the direction \mathbf{p} . This directional derivative exists even if ϕ_1 is non-differentiable.

Theorem 5 ([Nocedal 2006, Theorem 18.2]). *Let $(\mathbf{p}^*, \mathbf{q}^*)$ be a solution of (6.4). Then the directional derivative of ϕ_1 in the direction \mathbf{p}^* satisfies*

$$D(\phi_1(\mathbf{x}_k, \mu), \mathbf{p}^*) = \nabla f(\mathbf{x}_k)^T \mathbf{p}^* - \mu \left(\sum_{i \in \mathcal{E}} |c_i(\mathbf{x}_k)| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x}_k)]^{-} \right). \quad (6.19)$$

We want this directional derivative to be sufficiently negative in the sense that

$$D(\phi_1(\mathbf{x}_k, \mu), \mathbf{p}^*) \leq -\rho \mu \left(\sum_{i \in \mathcal{E}} |c_i(\mathbf{x}_k)| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x}_k)]^{-} \right), \quad (6.20)$$

for some $\rho \in (0, 1)$, which holds if

$$\mu \geq \frac{\nabla f(\mathbf{x}_k)^T \mathbf{p}^*}{(1 - \rho) \left(\sum_{i \in \mathcal{E}} |c_i(\mathbf{x}_k)| + \sum_{i \in \mathcal{I}} [c_i(\mathbf{x}_k)]^{-} \right)}. \quad (6.21)$$

6.2.5 Maratos effect and second-order correction

It is possible that, using merit functions, we reject a good step for the nonlinear problem and we fail to achieve superlinear convergence. This is the case, for example, when the linearization of the constraints is not accurate enough in the vicinity of \mathbf{x}_k . This is known as the *Maratos effect* [Nocedal 2006, Chapter 15.5] and we can avoid it by:

- Adding a second-order correction step $\hat{\mathbf{p}}$ defined to be

$$\hat{\mathbf{p}} = -\mathbf{A}^T(\mathbf{x}_k)(\mathbf{A}(\mathbf{x}_k)\mathbf{A}^T(\mathbf{x}_k))^{-1}\mathbf{c}_{\mathcal{W}}(\mathbf{x}_k + \mathbf{p}). \quad (6.22)$$

This step satisfies the linearization of the constraint $\mathbf{c}_{\mathcal{W}}$ at the point $\mathbf{x}_k + \mathbf{p}$, that is,

$$\mathbf{A}(\mathbf{x}_k)\hat{\mathbf{p}} + \mathbf{c}_{\mathcal{W}}(\mathbf{x}_k + \mathbf{p}) = \mathbf{0}. \quad (6.23)$$

- Allowing the merit function ϕ_1 to sometimes increase from iteration to iteration (adopt a non-monotone technique).

6.2.6 A line search algorithm

We use Algorithm 2, adapted from [Nocedal 2006, Algorithms 18.3, 15.2], to solve the nonlinear optimization problem (6.1). Note that we do not apply the second-order correction at every iteration of the line search to avoid expensive computations.

Algorithm 2 Line-search SQP

Require: $f, c, \mathcal{E}, \mathcal{I}, \epsilon$

Ensure: KKT conditions with ϵ -accuracy

- 1: Choose $\eta \in (0, 0.5)$
 - 2: Choose initial point \mathbf{x}_0
 - 3: $k \leftarrow 0$
 - 4: **repeat**
 - 5: Evaluate $f(\mathbf{x}_k), \nabla f(\mathbf{x}_k), \mathbf{c}(\mathbf{x}_k), \nabla \mathbf{c}(\mathbf{x}_k)$
 - 6: Compute $(\mathbf{p}_k, \mathbf{q}_k)$ from (6.4)
 - 7: Compute μ_k from (6.21)
 - 8: $\alpha_k \leftarrow 1$
 - 9: **if** $\phi_1(\mathbf{x}_k + \mathbf{p}_k, \mu_k) \leq \phi_1(\mathbf{x}_k, \mu_k) + \eta D(\phi_1(\mathbf{x}_k, \mu_k), \mathbf{p}_k)$ **then**
 - 10: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{p}_k, \boldsymbol{\lambda}_{k+1} \leftarrow \mathbf{q}_k$
 - 11: **else**
 - 12: Compute $\hat{\mathbf{p}}_k$ from (6.22)
 - 13: **if** $\phi_1(\mathbf{x}_k + \mathbf{p}_k + \hat{\mathbf{p}}_k, \mu_k) \leq \phi_1(\mathbf{x}_k, \mu_k) + \eta D(\phi_1(\mathbf{x}_k, \mu_k), \mathbf{p}_k)$ **then**
 - 14: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{p}_k + \hat{\mathbf{p}}_k, \boldsymbol{\lambda}_{k+1} \leftarrow \mathbf{q}_k$
 - 15: **else**
 - 16: **while** $\phi_1(\mathbf{x}_k + \alpha_k \mathbf{p}_k, \mu_k) > \phi_1(\mathbf{x}_k, \mu_k) + \eta \alpha_k D(\phi_1(\mathbf{x}_k, \mu_k), \mathbf{p}_k)$ **do**
 - 17: $\alpha_k \leftarrow \alpha_k / 2$
 - 18: **end while**
 - 19: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k, \boldsymbol{\lambda}_{k+1} \leftarrow (1 - \alpha_k) \boldsymbol{\lambda}_k + \alpha_k \mathbf{q}_k$
 - 20: **end if**
 - 21: **end if**
 - 22: $k \leftarrow k + 1$
 - 23: **until** KKT conditions are satisfied with ϵ -accuracy
-

6.3 Feasible Sequential Quadratic Programming

In the line-search SQP presented in Algorithm 2 there is no guarantee of the satisfaction of the constraints until it has converged: an appropriate merit function (and even a second-order correction) is needed in order to minimize the violation of the constraints at every iteration. Feasible SQP (FSQP) refers to methods in which we ensure the feasibility of every iterate \mathbf{x}_k of the SQP. We restrict ourselves to inequality-constrained problems of the form

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \quad (6.24a)$$

$$\text{subject to} \quad g_i(\mathbf{x}) \geq 0, \quad i \in \{1, \dots, m\}, \quad (6.24b)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are twice-continuously differentiable functions. Let the vector function $\mathbf{g} = (g_1, \dots, g_m)$ collect all scalar functions g_i . We denote the feasible set for (6.24) as:

$$\mathcal{D} \triangleq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \geq \mathbf{0}\}. \quad (6.25)$$

In the line-search method presented in [Panier 1993, Lawrence 1996] and [Zhu 2010], feasibility is achieved by “tilting” the search direction towards the feasible set. The new direction \mathbf{d} is written as

$$\mathbf{d} = (1 - \theta)\mathbf{p}^0 + \theta\mathbf{p}^1, \quad \theta \in [0, 1]. \quad (6.26)$$

In here, \mathbf{p}^0 refers to the solution of (6.4) and \mathbf{p}^1 is an arbitrary feasible direction. In order to ensure the quasi-Newton character of \mathbf{p}^0 in \mathbf{d} , θ is selected to be a function of \mathbf{p}^0 that goes to zero fast enough as a solution is approached, that is, θ satisfies $\theta(\mathbf{p}^0) = O(\|\mathbf{p}^0\|^2)$. Furthermore, we add another term \mathbf{d}^C that “bends” the search direction in order to avoid the Maratos effect. A line search is then performed on the arc

$$\mathbf{x}_k + t\mathbf{d} + t^2\mathbf{d}^C \in \mathcal{D}, \quad t \in [0, 1]. \quad (6.27)$$

Note that we need to compute three QPs: one per search direction $\mathbf{p}^0, \mathbf{p}^1$ and \mathbf{d}^C . Although a more efficient algorithm is presented in [Lawrence 2001], this remains a computationally expensive method.

In the trust-region method presented in [Wright 2004] we solve, at each iteration, the quadratic subproblem (6.4) with the extra constraint

$$\|\mathbf{W}\mathbf{p}^0\| \leq \Delta, \quad (6.28)$$

where \mathbf{W} is a diagonal matrix of weights and Δ is a positive constant. Feasibility is achieved by appropriately perturbing the regular step \mathbf{p}^0 to obtain a $\tilde{\mathbf{p}}$ such that

$$\mathbf{x}_k + \tilde{\mathbf{p}} \in \mathcal{D}. \quad (6.29)$$

If this new step satisfies $\|\mathbf{p}^0 - \tilde{\mathbf{p}}\| \leq \phi(\|\mathbf{p}^0\|)\|\mathbf{p}^0\|$, where $\phi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ and $\phi(0) = 0$, then we can ensure superlinear convergence. The vector $\mathbf{x}_k + \tilde{\mathbf{p}}$ can be obtained from the projection of $\mathbf{x}_k + \mathbf{p}^0$ onto the feasible set, however, this is not a trivial problem to solve in general [Boyd 2004, Chapter 8.1]. In problems with more structured constraints, this task could be rather inexpensive. Moreover, in the case where the constraint \mathbf{g} is linear we can simply set $\tilde{\mathbf{p}} = \mathbf{p}$.

In our approach to feasible iterates of the SQP we develop on this last observation: instead of adapting \mathbf{p}^0 , we modify the constraint \mathbf{g} such that the solution of this modified problem is a feasible point of the original problem. Iterations are, therefore, less expensive because we solve a single QP and no extra step corrections have to be made.

Sun *et al.* used FSQP techniques to find stable periodic gaits for the walk of a compass-like robot walker [Sun 2015]. Lebbah *et al.* developed tight and *safe linear relaxations* of

polynomial constraints in the exploration of feasible sets in order to bound the global optima [Lebbah 2005, Lebbah 2007]. They use interval analysis, local consistencies and branch and prune algorithms to approximate the solutions with arbitrary precision. However, in our case we are interested in local optima.

6.4 Convexification of constraints

6.4.1 Convexification by transformation

Let $f : \mathcal{C} \rightarrow \mathbb{R}$ be a function defined in a convex subset $\mathcal{C} \subseteq \mathbb{R}^n$. Let $f(\mathcal{C})$ be the image of \mathcal{C} under f , *i.e.*

$$f(\mathcal{C}) \triangleq \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{C}\}. \quad (6.30)$$

Definition 6 (Convex range transformable [Horst 1984]). *The function f is said to be convex range transformable if there exists a continuous strictly monotone increasing function $F : f(\mathcal{C}) \rightarrow \mathbb{R}$ such that $F(f(\mathbf{x}))$ is convex over \mathcal{C} .*

In the case of the constraint \mathbf{g} in (6.24), the difficulty resides in finding a function F such that $F(g_i(\mathbf{x}))$ is convex over \mathcal{C} for all $i \in \{1, \dots, m\}$. This might prove to be difficult to do by hand except only for highly structured problems¹. A different approach consists in convexifying functions through a change of variables.

Definition 7 (Convex domain transformable [Horst 1984]). *The function f is said to be convex domain transformable if there exists a continuous one-to-one mapping $h : \mathcal{C} \rightarrow \mathbb{R}^n$ such that $f(h^{-1}(\mathbf{y}))$ is convex on $h(\mathcal{C})$.*

Then, again, this could be possible to do when \mathbf{x} and the constraints g_i can be manipulated with ease. Examples of applications include the real-time generation of feasible trajectories in nonlinear systems such as aircrafts [Faiz 1999, Nadim Faiz 2001] and spacecrafts [Louembet 2007].

Example 3. *Khadiiv et al. consider the 1-dimensional modal system of the motion of the CoM [Khadiiv 2016]*

$$\begin{bmatrix} \dot{u} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \omega & 0 \\ 0 & -\omega \end{bmatrix} \begin{bmatrix} u \\ s \end{bmatrix} + \begin{bmatrix} -\omega \\ \omega \end{bmatrix} p,$$

where $u, s \in \mathbb{R}$ represent the unstable and stable components of the dynamics, respectively, $p \in \mathbb{R}$ is the CoP and ω is a positive real constant. The solution to the unstable component of this system can be written, at the i^{th} sampling time, as

$$u_{i+1}(\mathbf{x}) = (u_i - p_i)e^{\omega_i \tau_i} + p_i,$$

where $\mathbf{x} = (u_i, p_i, \omega_i, \tau_i) \in \mathbb{R}^4$ and p_i is maintained constant over an interval $\tau_i > 0$. We want to adapt the duration of the steps by changing τ_i . The relationship between τ_i and u_{i+1} is convex but nonlinear and we want to linearize it. We perform a domain transformation with a function $h_1 : \mathbb{R}^4 \rightarrow \mathbb{R}^2$

$$h_1(\mathbf{x}) = \begin{bmatrix} \delta_i \\ \pi_i \end{bmatrix} = \begin{bmatrix} u_i - p_i \\ e^{\omega_i \tau_i} \end{bmatrix},$$

while constraining $\delta_i = \delta_{i+1} = b$. This results in the reformulated system

$$\begin{aligned} u_{i+1}(h_1(\mathbf{x})) - p_{i+1} &= \delta_{i+1} = b, \\ \delta_i \pi_i + p_i - p_{i+1} &= b, \\ b \pi_i + p_i - b &= p_{i+1}, \end{aligned}$$

¹For problems without an evident structure, neural networks are useful for range convexification.

which is linear with respect to π_i and p_{i+1} . However, by doing so we have relatively “rigid” dynamics fixed to b at every step. We could gain more flexibility by considering instead

$$b_0\pi_i + p_i - b = p_{i+1},$$

where b_0 is a constant and π_i, p_{i+1}, b are variables, as proposed by the authors. In this case, although the system and the constraints are linear, we can only linearly preview a single step in the future.

We propose, instead, to perform a domain transformation with a function $h_2 : \mathbb{R}^5 \rightarrow \mathbb{R}^3$

$$h_2(\mathbf{x}, k_{i+1}) = \begin{bmatrix} \pi_i \\ \delta_i \\ \delta_{i+1}/k_{i+1} \end{bmatrix} = \begin{bmatrix} \omega_i\tau_i \\ u_i - p_i \\ u_{i+1} - p_i \end{bmatrix},$$

which gives

$$\begin{aligned} u_{i+1}(h_2(\mathbf{x}, k_{i+1})) &= \delta_i e^{\pi_i} + p_i, \\ \delta_{i+1}/k_{i+1} &= \delta_i e^{\pi_i}, \\ \delta_{i+1} &= k_{i+1} \delta_i e^{\pi_i}. \end{aligned}$$

This formulation allows us to (indirectly) control the value of $u_i - p_i$ as a factor k_i of $u_i - p_{i-1}$. We can obtain an expression which is linear on the integration time τ_i by restricting $\delta_i, k_{i+1} > 0$ and performing a range transformation with logarithms

$$\ln(\delta_{i+1}) = \ln(k_i) + \ln(\delta_i) + \pi_i.$$

We can preview any number of steps of the robot, each one with potentially different duration. However, the formulation of constraints on the CoM, the steps and the CoP in this new space becomes nonlinear.

6.4.2 Convexification by polytopic approximation

Definition 8 (ϵ -approximation [Jones 2010]). A polytope \mathcal{P} is an ϵ -approximation of a set \mathcal{Q} if the Hausdorff distance between \mathcal{P} and \mathcal{Q} , denoted $d_H(\mathcal{P}, \mathcal{Q})$, is less than or equal to ϵ , for some $\epsilon > 0$. \mathcal{P} is called an outer (resp. inner) ϵ -approximation if $\mathcal{Q} \subseteq \mathcal{P}$ (resp. $\mathcal{Q} \supseteq \mathcal{P}$).

In general, we could define \mathcal{P} to be arbitrarily close to \mathcal{Q} (in the Hausdorff-distance sense) by increasing the number of hyperplanes and halfspaces we use to define it.

Theorem 6 ([Bronshtein 1975]). If \mathcal{Q} is compact and convex then, for every $\epsilon > 0$, there exists a finite number of equalities and inequalities (used to define \mathcal{P}) such that $d_H(\mathcal{P}, \mathcal{Q}) \leq \epsilon$.

If \mathcal{Q} is nonconvex then there is a limit to how close we can approximate \mathcal{Q} with a single polytope [Deits 2015]. In this thesis, however, we prefer the capacity to computing online a safe linear approximation of \mathcal{Q} than to minimize $d_H(\mathcal{P}, \mathcal{Q})$.

Let us consider the constraint of problem (6.24)

$$\mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \tag{6.31}$$

and its feasible set \mathcal{D} . We recall that $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Let there be a set of points $\{\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\}$ that satisfy for some $\mathcal{S} \subseteq \mathbb{R}^n$

$$\forall \mathbf{x} \in \mathcal{S}, \mathbf{g}(\mathbf{x}) \in \text{conv} \{\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\}, \tag{6.32}$$

where $\mathbf{A}_i \in \mathbb{R}^{m \times n}$ and $\mathbf{b}_i \in \mathbb{R}^m$. That is, the set $\{\text{conv}\{\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\} : \mathbf{x} \in \mathcal{S}\}$ is an outer approximation of the set $\mathbf{g}(\mathcal{S})$ and therefore

$$\exists \mathbf{x} \in \mathcal{S}, \forall i, \mathbf{A}_i \mathbf{x} - \mathbf{b}_i \geq \mathbf{0} \implies \mathbf{g}(\mathbf{x}) \geq \mathbf{0}. \quad (6.33)$$

We can then safely approximate the nonlinear constraint $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$ with the linear constraints $\forall i, \mathbf{A}_i \mathbf{x} - \mathbf{b}_i \geq \mathbf{0}$ since the feasible set

$$\hat{\mathcal{D}} \triangleq \{\mathbf{x} \in \mathcal{S} : \forall i, \mathbf{A}_i \mathbf{x} - \mathbf{b}_i \geq \mathbf{0}\}, \quad (6.34)$$

is an inner approximation of \mathcal{D}

$$\hat{\mathcal{D}} \subseteq \mathcal{D}. \quad (6.35)$$

6.5 Safe Sequential Quadratic Programming

Let us reconsider the NLP problem (6.24)

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \quad (6.36a)$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}) \geq \mathbf{0}. \quad (6.36b)$$

We assume that its feasible set \mathcal{D} is nonempty. In a similar fashion to the SQP frameworks presented in Sections 6.2 and 6.3, in this Safe Sequential Quadratic Programming (Safe SQP) framework we design a QP subproblem at the k^{th} iteration of the form:

$$\underset{\mathbf{p}}{\text{minimize}} \quad f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B}_k \mathbf{p} \quad (6.37a)$$

$$\text{subject to} \quad \mathbf{A}_{k,i} \mathbf{p} - \mathbf{b}_{k,i} \geq \mathbf{0}, \quad \forall i, \quad (6.37b)$$

$$\mathbf{x}_k + \mathbf{p} \in \mathcal{S}_k, \quad (6.37c)$$

where constraints (6.37b) are *safe linear approximation* of $\mathbf{g}(\mathbf{x}_k + \mathbf{p}) \geq \mathbf{0}$ in the set \mathcal{S}_k . This implies that, if the feasible set

$$\hat{\mathcal{D}}_k \triangleq \{\mathbf{x}_k + \mathbf{p} \in \mathcal{S}_k : \forall i, \mathbf{A}_{k,i} \mathbf{p} - \mathbf{b}_{k,i} \geq \mathbf{0}\}, \quad (6.38)$$

is not empty then it satisfies

$$\hat{\mathcal{D}}_k \subseteq \mathcal{D}. \quad (6.39)$$

Therefore, as indicated in (6.33), a solution \mathbf{p}^* satisfies

$$\mathbf{x}_k + \mathbf{p}^* \in \hat{\mathcal{D}}_k \subseteq \mathcal{D}. \quad (6.40)$$

We choose \mathcal{S}_k to be a bounded set of the form:

$$\mathcal{S}_k \triangleq \{\mathbf{x}_k + \mathbf{p} \in \mathbb{R}^n : \|\mathbf{W} \mathbf{p}\|_\infty \leq \Delta\}, \quad (6.41)$$

where \mathbf{W} is a diagonal matrix of positive weights.

6.5.1 A trust-region algorithm

The set \mathcal{S}_k also acts as a trust region. It ensures that the iterates are well-defined by restricting the step \mathbf{p}^* to a neighborhood of \mathbf{x}_k where the second-order model of f is considered reliable. Reliability is determined with the ratio ρ between the *actual* and the *predicted reduction* of f related to the step \mathbf{p}^*

$$\rho = -\frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{p}^*)}{\nabla f(\mathbf{x}_k)^T \mathbf{p}^* + \frac{1}{2} \mathbf{p}^{*T} \mathbf{B}_k \mathbf{p}^*}. \quad (6.42)$$

The closer this ratio is to one, the more reliable the step is.

We handle potential infeasibilities, produced by our choice of \mathcal{S}_k and inconsistencies among the linear constraints, by relaxing the problem in the following way:

$$\text{lex minimize}_{\mathbf{p}, v_1, v_2} \quad \mathbf{v} = (\|\mathbf{v}_1\|^2, v_2^2) \quad (6.43a)$$

$$\text{subject to} \quad \mathbf{A}_{k,i}\mathbf{p} - \mathbf{b}_{k,i} - \mathbf{v}_1 \geq \mathbf{0}, \quad (6.43b)$$

$$f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{B}_k \mathbf{p} - v_2 = 0, \quad (6.43c)$$

$$\mathbf{x}_k + \mathbf{p} \in \mathcal{S}_k, \quad (6.43d)$$

The vector \mathbf{v} is to be minimized in a lexicographic order: we relax, firstly, the constraints as little as needed (by minimizing $\|\mathbf{v}_1\|^2$) and then, secondly, the objective function as little as needed (by minimizing v_2^2 without interfering with the optimal value of $\|\mathbf{v}_1\|^2$). This corresponds to the standard relaxation of equality constraints for trust-region methods [Nocedal 2006, Chapter 18.5].

Let $(\mathbf{p}^*, \mathbf{v}_1^*, v_2^*)$ be a solution to the relaxed problem (6.43). If $\|\mathbf{v}_1^*\|^2 = 0$ then the linear constraints are satisfied exactly and $\mathbf{x}_k + \mathbf{p}^*$ is a feasible point of (6.36). Moreover, even if $\|\mathbf{v}_1^*\|^2 > 0$ and the linear constraints are violated, the point $\mathbf{x}_k + \mathbf{p}^*$ could still be a feasible point of the nonlinear problem (as shown statistically in Section 8.3) because the linear constraints are conservative and their violation is minimized in a least-squares sense. We can determine this by simply checking whether

$$\mathbf{g}(\mathbf{x}_k + \mathbf{p}^*) \geq \mathbf{0}. \quad (6.44)$$

Algorithm 3 presents a trust-region **Safe SQP**. It is based on the algorithm for convex constraints presented by Conn *et al.* that computes feasible iterates [Conn 2000, Algorithm 12.2.1]. As in classic trust-region algorithms, we compute ρ_k and use it to both update the size of the trust region (lines 17 to 25) and to accept or reject a step \mathbf{p}_k (lines 26 to 31). Additionally, we check whether a point \mathbf{p}_k from the relaxed problem (6.43) is a feasible point in the original nonlinear problem (6.36) (lines 7 to 15). In any case, whenever the linear constraints are inconsistent we reduce the size of the trust region.

6.5.2 Termination and convergence

For reasons of time, proofs on the termination of Algorithm 3 as well as on its convergence to first-order and second-order critical points are beyond the scope of this thesis. In the following, we sketch some conjectures.

We could anticipate that Algorithm 3 terminates in a finite number of iterations, as shown by Conn *et al.* for a similar algorithm [Conn 2000, Theorem 12.2.1], if we assume: 1) the functions f and g_i are twice-continuously differentiable, 2) the feasible set \mathcal{D} is nonempty, closed and convex, 3) the first-order constraints qualifications hold at the optimum \mathbf{x}^* and 4) our approximated model of f is of second-order.

Convergence to first-order critical points of every limit point of $\{\mathbf{x}_k\}$ can be achieved by ensuring a model decrease at every iteration that is, at least, as large as a fraction obtained at the generalized Cauchy point, that is, by ensuring that ρ is greater than a fraction of 1 [Conn 2000, Chapter 8.1.2]. Once the optimal active set \mathcal{A}^* has been identified after a finite number of iterations, problem (6.36) behaves as an unconstrained problem. Since constraints are linear, the “constrained” space is affine, and one may expect the sequence $\{\mathbf{x}_k\}$ to converge to second-order critical points under conditions similar to those of the unconstrained case [Conn 2000, Theorem 6.5.5].

Algorithm 3 Trust-region Safe SQP

Require: $f, g, \mathcal{H}, \epsilon, |\mathcal{S}_{max}|$ **Ensure:** KKT conditions with ϵ -accuracy

- 1: Choose $\mathbf{x}_0, \mathcal{S}_0$ such that $|\mathcal{S}_0| \leq |\mathcal{S}_{max}|$
 - 2: Choose $\eta \in [0, 1/4)$
 - 3: $k \leftarrow 0$
 - 4: **repeat**
 - 5: Evaluate $f(\mathbf{x}_k), \nabla f(\mathbf{x}_k)$ and $\mathbf{A}_{k,i}, \mathbf{b}_{k,i}$ at \mathcal{S}_k
 - 6: Compute $(\mathbf{p}_k, \mathbf{v}_k, \mathbf{q}_k)$ from (6.43)
 - 7: **if** $\|\mathbf{v}_{k,1}\|_\infty > \epsilon$ **then**
 - 8: **if** (6.44) evaluated at $\mathbf{x}_k + \mathbf{p}_k$ does not hold **then**
 - 9: Compute $\mathcal{Q} \subset \mathcal{S}_k$
 - 10: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k, \mathcal{S}_{k+1} \leftarrow \mathcal{Q}$
 - 11: **else**
 - 12: Compute $\mathcal{Q} \subset \mathcal{S}_k$
 - 13: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{p}_k, \mathcal{S}_{k+1} \leftarrow \mathcal{Q}$
 - 14: **end if**
 - 15: **else**
 - 16: Compute ρ_k from (6.42)
 - 17: **if** $\rho_k < 1/4$ **then**
 - 18: Compute $\mathcal{Q} \subset \mathcal{S}_k$
 - 19: **else**
 - 20: **if** $\rho_k > 3/4$ and $\mathbf{x}_k + \mathbf{p}_k \in \partial \mathcal{S}_k$ **then**
 - 21: Compute $\mathcal{Q} \supset \mathcal{S}_k$ such that $|\mathcal{Q}| \leq |\mathcal{S}_{max}|$
 - 22: **else**
 - 23: $\mathcal{Q} \leftarrow \mathcal{S}_k$
 - 24: **end if**
 - 25: **end if**
 - 26: **if** $\rho_k > \eta$ **then**
 - 27: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{p}_k, \mathcal{S}_{k+1} \leftarrow \mathcal{Q} + \mathbf{p}_k$
 - 28: **else**
 - 29: $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k, \mathcal{S}_{k+1} \leftarrow \mathcal{Q}$
 - 30: **end if**
 - 31: **end if**
 - 32: $k \leftarrow k + 1$
 - 33: **until** KKT conditions are satisfied with ϵ -accuracy
-

6.6 Conclusion

We introduced SQP as a method to solve nonlinear problems and we did an overview of the key numerical aspects to consider in the basic line-search algorithm. We motivated the need of having always-feasible iterates in robotics and exposed pertinent known results in the domain of FSQP. We presented our approach to nonlinearities in the constraints: we design safe linear constraints with respect to such nonlinear constraints. We formulated a modified SQP approach (Safe SQP) that solves nonlinear problems using safe linear constraints.

Chapter 7

Nonlinear Model Predictive Control for biped walking

7.1 Introduction

In this chapter we characterize in detail the MPC scheme we use for biped walking in a crowd.

We start by describing our model of motion of the CoM and how we choose to adapt online the position, orientation and duration of the steps of the robot in Sections 7.2 and 7.3. We then introduce the nonlinear constraints of the robot for kinematics, dynamics, collision avoidance and stopping in Section 7.4. At the same time we present our approach to defining safe linear constraints with respect to those nonlinear constraints. We finish with a description of objectives of the walk in Section 7.5 and the optimization problem we solve in order to compute the motion of the robot in Section 7.6. Part of the work presented here has been published by Bohórquez *et al.* [Bohórquez 2017, Bohórquez 2018].

7.2 Motion of the CoM

Since we are coupling the whole-body and the CoM models through the acceleration of the CoM (2.26), we want this acceleration to be continuous, *i.e.* $\mathbf{c} \in \mathcal{C}^2$. We achieve this with a piecewise constant jerk $\ddot{\mathbf{c}} \in \mathbb{R}^2$.

We choose as state vector $\mathbf{x} \in \mathbb{R}^6$ the position, velocity and acceleration of the CoM in the xy plane

$$\mathbf{x} = (c^x, \dot{c}^x, \ddot{c}^x, c^y, \dot{c}^y, \ddot{c}^y). \quad (7.1)$$

We write the following continuous system with the CoP $\mathbf{p} \in \mathbb{R}^2$ as output

$$\dot{\mathbf{x}}(t) = \hat{\mathbf{A}}\mathbf{x}(t) + \hat{\mathbf{B}}\ddot{\mathbf{c}}(t), \quad (7.2a)$$

$$\mathbf{p}(t) = \mathbf{C}\mathbf{x}(t), \quad (7.2b)$$

where

$$\hat{\mathbf{A}} = \text{diag}_2 \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right), \quad \hat{\mathbf{B}} = \text{diag}_2 \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right), \quad \mathbf{C} = \text{diag}_2 \left([1 \quad 0 \quad -\frac{1}{\omega^2}] \right). \quad (7.3)$$

We discretize this system with a sampling period $\tau > 0$, during which the jerk is kept at a constant value, to obtain:

$$\mathbf{x}_{(i|k)} = \mathbf{A}\mathbf{x}_{(i-1|k)} + \mathbf{B}\ddot{\mathbf{c}}_{(i-1|k)}, \quad (7.4a)$$

$$\mathbf{p}_{(i-1|k)} = \mathbf{C}\mathbf{x}_{(i-1|k)}, \quad (7.4b)$$

where

$$\mathbf{A} = e^{\hat{\mathbf{A}}\tau} = \text{diag}_2 \left(\begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \right), \quad \mathbf{B} = \left(\int_0^\tau e^{\hat{\mathbf{A}}t} dt \right) \hat{\mathbf{B}} = \text{diag}_2 \left(\begin{bmatrix} \frac{\tau^3}{6} \\ \frac{\tau^2}{2} \\ \tau \end{bmatrix} \right). \quad (7.5)$$

7.2.1 Preview horizon

Our preview horizon has N samples separated by periods of time τ . Its duration H can be written as

$$H = N\tau. \quad (7.6)$$

We need to satisfy constraints not only at samples but also between samples. Lengagne *et al.* achieve this with techniques from interval analysis [Lengagne 2007, Lengagne 2009]. Van Loock *et al.* uses the properties of B-splines to guarantee the satisfaction of constraints throughout the preview horizon [Loock 2015]. In this thesis, we do not enforce the satisfaction of constraints between samples. However, this could be done by adding proper extra margins that bound the cubic polynomials of \mathbf{c} and \mathbf{p} between two consecutive samples.

7.3 Position, orientation and duration of the steps

We want the robot to change online the positions of the feet on the ground in the preview horizon [Herdt 2010a, Herdt 2010b]

$$\tilde{\mathbf{s}}_k = (\mathbf{s}_{(1|k)}, \dots, \mathbf{s}_{(J|k)}), \quad (7.7)$$

in order to avoid collisions with people. We also want to change the orientations of the feet on the ground

$$\tilde{\boldsymbol{\theta}}_k = (\theta_{(1|k)}, \dots, \theta_{(J|k)}), \quad (7.8)$$

such that the robot faces the direction of the walk. This is a desired behavior because the robot walks faster forwards than sideways [Arechavaleta 2008, Mombaur 2009]. Figure 7.1 shows the correspondence between the sample i and the step j in the preview horizon. We sample each step N_s times which is equal to eight in the example figure.

Having a fixed step duration and maximal step length limits the maximal speed the robot can achieve (to avoid collisions with people). We need, therefore, to adapt the duration of the steps of the robot. We could choose to do this by directly optimizing for the duration of each step in the horizon in a continuous [Kryczka 2015, Hu 2018, Zhao 2017] or discrete manner [Ibanez 2014]. Alternatively, others have shown that the problem can be made completely linear by either: 1) limiting the number of steps in the preview horizon to one [Khadij 2016], 2) considering the relationship between the divergent component of the dynamics in (2.16) and the position of the future footsteps [Griffin 2017].

We propose to vary the duration of the steps by changing τ [Aftab 2012, Diedam 2009]. Note that since the jerk is constant over intervals τ , that we define to be $1/N_s^{\text{th}}$ of the duration of one step of the robot, when we vary τ we vary the duration of the steps. By doing this, we obtain the following system:

$$\mathbf{x}_{(i|k)}(\tau) = \mathbf{A}(\tau)\mathbf{x}_{(i-1|k)} + \mathbf{B}(\tau)\ddot{\mathbf{c}}_{(i-1|k)}, \quad (7.9)$$

where the output vector $\mathbf{p}_{(i|k)}(\tau)$ (which is a linear combination of $\mathbf{x}_{(i|k)}(\tau)$) and the positions of people $\mathbf{m}_{(i|k)}(\tau)$ vary with τ . We can constrain τ to a range where the steps are not made too fast in order to avoid excessive torque demands on the joints.

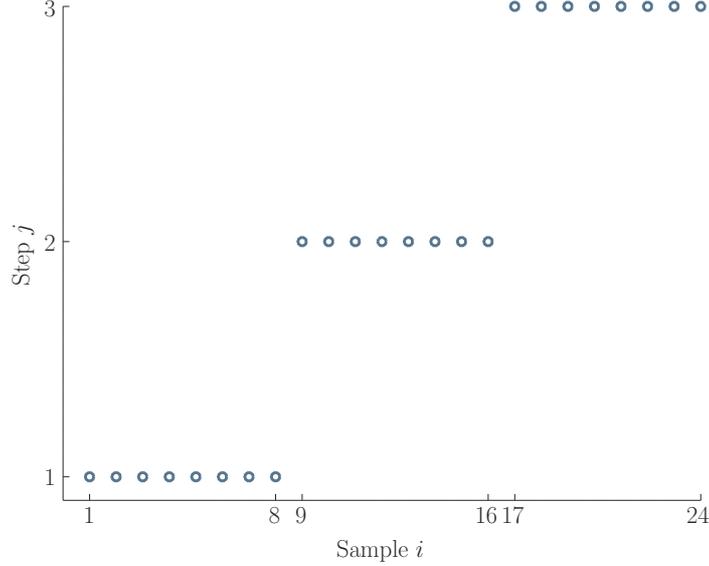


Figure 7.1: Correspondence between samples and steps in the preview horizon with $N_s = 8$.

Remark 4. *When we change τ from one iteration to the next one we are effectively switching the system. Further considerations on stability [Minh 2013] and on strong recursive feasibility [Ciocca 2017] are, therefore, needed. These issues are, nonetheless, beyond the scope of this thesis.*

7.4 Constraints of walking

7.4.1 Kinematics and dynamics

We enforce kinematic constraints on the length of the legs of the robot (through the distance between $\mathbf{c}_{(i|k)}(\tau)$ and \mathbf{s}_j) and on the alternation of the legs as the robot walks (through the sequence $\mathbf{s}_{(j|k)}, \mathbf{s}_{(j+1|k)}, \dots$). Furthermore, we keep the CoP inside the support polygon of the robot (2.12):

$$\begin{cases} \mathbf{p}_{(i|k)}(\tau) \in \mathcal{P}(\mathbf{s}_{(j|k)}, \theta_{(j|k)}), & \text{Dynamical constraint} \\ \mathbf{c}_{(i|k)}(\tau) \in \mathcal{C}(\mathbf{s}_{(j|k)}, \theta_{(j|k)}), & \text{Kinematic constraints} \\ \mathbf{s}_{(j+1|k)} \in \mathcal{S}(\mathbf{s}_{(j|k)}, \theta_{(j|k)}), & \end{cases} \quad (7.10)$$

Figure 7.2 depicts these constraints. The set-valued functions $\mathcal{P}(\mathbf{s}, \theta), \mathcal{C}(\mathbf{s}, \theta)$ specify rectangular regions of width and length $W \times L$ and $w \times l$, respectively. Both rotate around \mathbf{s} with an angle θ . The set-valued function $\mathcal{S}(\mathbf{s}, \theta)$ specifies a halfspace that rotates around \mathbf{s} with an angle θ and is separated from it a distance $w + d_0$. More specifically, we consider

$$\mathcal{P}(\mathbf{s}, \theta) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 : \begin{bmatrix} -l/2 \\ -w/2 \end{bmatrix} \leq \mathbf{R}(\theta)^T(\mathbf{r} - \mathbf{s}) \leq \begin{bmatrix} l/2 \\ w/2 \end{bmatrix} \right\}, \quad (7.11a)$$

$$\mathcal{C}(\mathbf{s}, \theta) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 : \begin{bmatrix} -L/2 \\ -0.5(1 + (-1)^j)W \end{bmatrix} \leq \mathbf{R}(\theta)^T(\mathbf{r} - \mathbf{s}) \leq \begin{bmatrix} L/2 \\ 0.5(1 - (-1)^j)W \end{bmatrix} \right\}, \quad (7.11b)$$

$$\mathcal{S}(\mathbf{s}, \theta) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 : (-1)^j(w + d_0) \leq \mathbf{I}_y \mathbf{R}(\theta)^T(\mathbf{r} - \mathbf{s}) \right\}, \quad (7.11c)$$

where $\mathbf{I}_y = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $\mathbf{R}(\theta)$ is a rotation matrix in $SO(2)$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (7.12)$$

The terms $0.5(1-(-1)^j)W$ and $-0.5(1+(-1)^j)W$ in (7.11b) take the values $W, 0$, respectively, when j is odd and $0, -W$, respectively, when j is even. This allow us to alternate the steps of the robot as it walks.

Turning while walking has been addressed in many different ways [Yagi 2000, Aoi 2004, Behnke 2006, Chestnutt 2005, Chestnutt 2007a, Perrin 2012a, Deits 2014] but never guaranteeing that all kinematic and dynamic constraints are properly satisfied. With MPC we can make sure that the walking motion satisfies constraints at all times. However, the adaptation of the rotations of the steps makes this a nonlinear problem [Naveau 2017] unless, for example, we decide θ in advance [Herdt 2010a, Simone 2017].

Nonlinear dependency on θ

We compute an inner approximation of $\mathcal{P}(\mathbf{s}, \theta)$, $\mathcal{C}(\mathbf{s}, \theta)$, $\mathcal{S}(\mathbf{s}, \theta)$, for all $\theta \in [\underline{\theta}, \bar{\theta}]$, such that the resulting sets do not depend on θ and are linear with respect to \mathbf{s} . That is, we define new functions

$$\bar{\mathcal{P}}(\mathbf{s}) \triangleq \mathcal{W}_{\mathcal{P}} + \mathbf{s}, \quad \bar{\mathcal{C}}(\mathbf{s}) \triangleq \mathcal{W}_{\mathcal{C}} + \mathbf{s}, \quad \bar{\mathcal{S}}(\mathbf{s}) \triangleq \mathcal{W}_{\mathcal{S}} + \mathbf{s}, \quad (7.13)$$

that satisfy

$$\bar{\mathcal{P}}(\mathbf{s}) \subseteq \mathcal{P}(\mathbf{s}, \theta), \quad \bar{\mathcal{C}}(\mathbf{s}) \subseteq \mathcal{C}(\mathbf{s}, \theta), \quad \bar{\mathcal{S}}(\mathbf{s}) \subseteq \mathcal{S}(\mathbf{s}, \theta), \quad (7.14)$$

for all $\theta \in [\underline{\theta}, \bar{\theta}]$.

Let us consider the intersections

$$\bigcap_{\theta} \mathcal{P}(\mathbf{s}, \theta), \quad \bigcap_{\theta} \mathcal{C}(\mathbf{s}, \theta), \quad \bigcap_{\theta} \mathcal{S}(\mathbf{s}, \theta), \quad (7.15)$$

shown in red in Figure 7.3. They are not empty because the function $(\theta, \mathbf{r}) \mapsto \mathbf{R}(\theta)^T \mathbf{r}$ is continuous and we define it over a convex set $[\underline{\theta}, \bar{\theta}]$ of appropriate length. These intersections are, moreover, convex since rectangles are convex sets. We want $\mathcal{W}_{\mathcal{C}}, \mathcal{W}_{\mathcal{P}}$ and $\mathcal{W}_{\mathcal{S}}$ to always be inside these intersections. We define them as the intersection of reduced versions of \mathcal{C}, \mathcal{P} and \mathcal{S} evaluated at the boundaries $\underline{\theta}$ and $\bar{\theta}$, shown in blue in Figure 7.3. That is,

$$\mathcal{W}_{\mathcal{P}} \triangleq \tilde{\mathcal{P}}(\mathbf{s}, \underline{\theta}) \cap \tilde{\mathcal{P}}(\mathbf{s}, \bar{\theta}) \subseteq \bigcap_{\theta} \mathcal{P}(\mathbf{s}, \theta), \quad (7.16a)$$

$$\mathcal{W}_{\mathcal{C}} \triangleq \tilde{\mathcal{C}}(\mathbf{s}, \underline{\theta}) \cap \tilde{\mathcal{C}}(\mathbf{s}, \bar{\theta}) \subseteq \bigcap_{\theta} \mathcal{C}(\mathbf{s}, \theta), \quad (7.16b)$$

$$\mathcal{W}_{\mathcal{S}} \triangleq \tilde{\mathcal{S}}(\mathbf{s}, \underline{\theta}) \cap \tilde{\mathcal{S}}(\mathbf{s}, \bar{\theta}) \subseteq \bigcap_{\theta} \mathcal{S}(\mathbf{s}, \theta), \quad (7.16c)$$

where:

- The rectangle $\tilde{\mathcal{P}}(\mathbf{s}, \theta)$ has length $l - \Delta l$ and width $w - \Delta w$ where $\Delta l = l(1 - \cos(\Delta\theta/2))$ and $\Delta w = w(1 - \cos(\Delta\theta/2))$.
- The rectangle $\tilde{\mathcal{C}}(\mathbf{s}, \theta)$ has length L and width $W - \Delta W$ where $\Delta W = W(1 - \cos(\Delta\theta/2))$.
- The halfspace $\tilde{\mathcal{S}}(\mathbf{s}, \theta)$ is separated from \mathbf{s} a distance $d_0 + \Delta d$ where $\Delta d = 1/2(l - w \tan(\Delta\theta/2)) \sin(\Delta\theta)$.

Constraints (7.10) can then be safely approximated, for all $\theta \in [\underline{\theta}, \bar{\theta}]$, as:

$$\mathbf{s}_{(j+1|k)} \in \bar{\mathcal{S}}(\mathbf{s}_{(j|k)}), \quad \mathbf{c}_{(i|k)}(\tau) \in \bar{\mathcal{C}}(\mathbf{s}_{(j|k)}), \quad \mathbf{p}_{(i|k)}(\tau) \in \bar{\mathcal{P}}(\mathbf{s}_{(j|k)}). \quad (7.17)$$

A similar approach is proposed by Aboudania *et al.* to define the set $\mathcal{W}_{\mathcal{P}}$ [Aboudonia 2017]. However, they define (what would be equivalent to) the sets $\mathcal{W}_{\mathcal{C}}$ and $\mathcal{W}_{\mathcal{S}}$ linear with respect to θ . In this case, a quantitative comparison is not straight-forward to do.

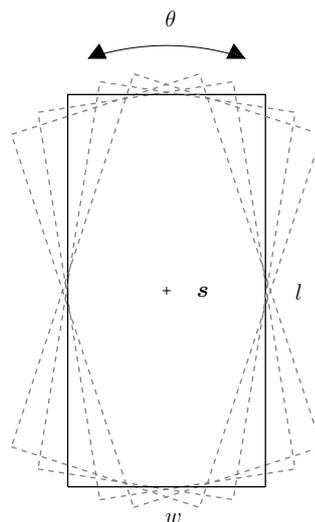
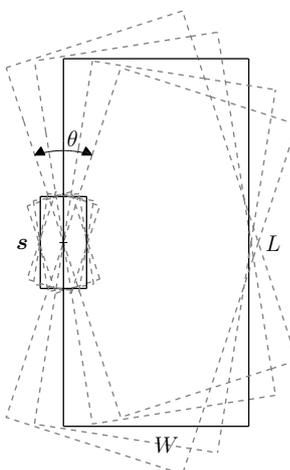
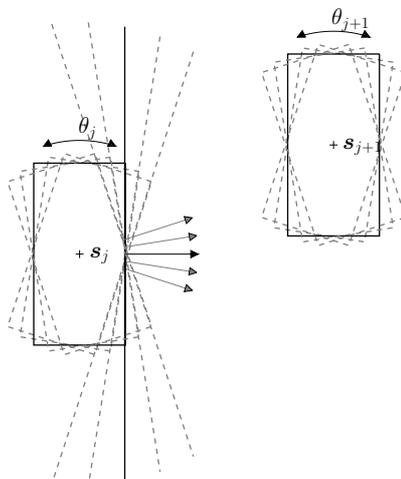
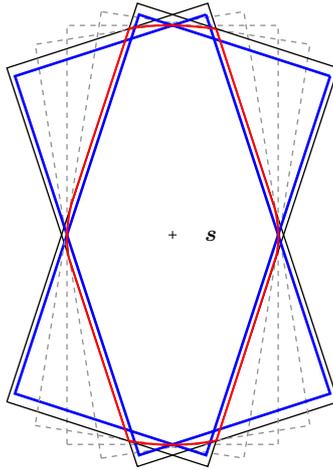
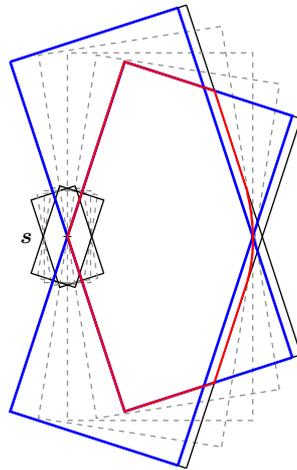
(a) Constraint on the CoP, $\mathcal{P}(\mathbf{s}, \theta)$.(b) Constraint on the CoM, $\mathcal{C}(\mathbf{s}, \theta)$.(c) Constraint on the steps, $\mathcal{S}(\mathbf{s}, \theta)$.

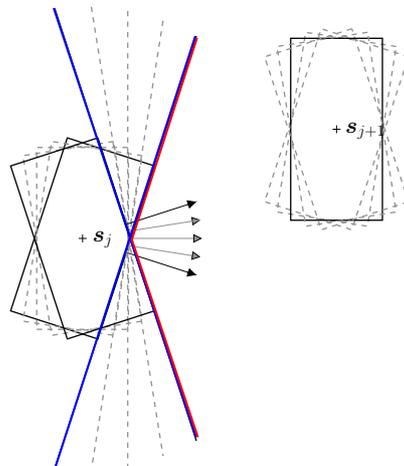
Figure 7.2: Nonlinear kinematic and dynamic constraints of the robot.



(a) Constraint on the CoP, $\bar{\mathcal{P}}(\mathbf{s})$.



(b) Constraint on the CoM, $\bar{\mathcal{C}}(\mathbf{s})$.



(c) Constraint on the steps, $\bar{\mathcal{S}}(\mathbf{s})$.

Figure 7.3: Safe polytopic approximations of the dynamic and kinematic constraints that depend on θ .

Nonlinear dependency on τ

We compute an outer approximation of the range of the functions $\mathbf{p}_{(i|k)}(\tau)$ and $\mathbf{c}_{(i|k)}(\tau)$, for all $\tau \in [\underline{\tau}, \bar{\tau}]$, such that the resulting sets do not depend on τ . These vectors are linear combinations of the state $\mathbf{x}_{(i|k)}(\tau)$ which is, in turn, a cubic polynomial of τ , as can be seen by reordering (7.4):

$$\begin{aligned} \mathbf{x}_{(i|k)}(\tau) &= \mathbf{x}_{(i-1|k)} \\ &+ \left(\text{diag}_2 \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{x}_{(i-1|k)} + \text{diag}_2 \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i-1|k)} \right) \tau \\ &+ \left(\text{diag}_2 \left(\begin{bmatrix} 0 & 0 & 1/2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{x}_{(i-1|k)} + \text{diag}_2 \left(\begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i-1|k)} \right) \tau^2 \\ &+ \left(\text{diag}_2 \left(\begin{bmatrix} 1/6 \\ 0 \\ 0 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i-1|k)} \right) \tau^3. \end{aligned} \quad (7.18)$$

Since the complexity of the bounds we define on $\mathbf{x}_{(i|k)}(\tau)$ depends on the degree of this polynomial, we reduce it by rewriting the last equation of (7.4) as

$$\ddot{\mathbf{c}}_{(i-1|k)} = \frac{\ddot{\mathbf{c}}_{(i|k)} - \ddot{\mathbf{c}}_{(i-1|k)}}{\tau}, \quad (7.19)$$

and substituting it back in the triple integrator to obtain the new system

$$\begin{aligned} \mathbf{x}_{(i|k)}(\tau) &= \left(\text{diag}_2 \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{x}_{(i-1|k)} + \text{diag}_2 \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i|k)} \right) \\ &+ \left(\text{diag}_2 \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{x}_{(i-1|k)} + \text{diag}_2 \left(\begin{bmatrix} 0 \\ 1/2 \\ 0 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i|k)} \right) \tau \\ &+ \left(\text{diag}_2 \left(\begin{bmatrix} 0 & 0 & 1/3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \mathbf{x}_{(i-1|k)} + \text{diag}_2 \left(\begin{bmatrix} 1/6 \\ 0 \\ 0 \end{bmatrix} \right) \ddot{\mathbf{c}}_{(i|k)} \right) \tau^2, \\ \mathbf{x}_{(i|k)}(\tau) &= \boldsymbol{\alpha}_{i0} + \boldsymbol{\alpha}_{i1}\tau + \boldsymbol{\alpha}_{i2}\tau^2, \end{aligned} \quad (7.20)$$

where now the input parameter is $\ddot{\mathbf{c}}_{(i|k)}$. In this way, the state $\mathbf{x}_{(i|k)}(\tau)$ is a quadratic polynomial of τ that preserves the \mathcal{C}^2 continuity of the motion of the CoM.

We define the bounds on $\mathbf{x}_{(i|k)}(\tau)$ by changing its representation from the monomial basis $\{1, \tau, \tau^2\}$ to the Bernstein basis [Lorentz 1953] (traditionally used in Bézier curves):

$$\{b_0(\tau), b_1(\tau), b_2(\tau)\}, \quad (7.21)$$

where

$$b_0(\tau) = (1 - \tau')^2, \quad b_1(\tau) = 2\tau'(1 - \tau'), \quad b_2(\tau) = \tau'^2, \quad (7.22)$$

and

$$\tau' = \frac{\tau - \underline{\tau}}{\bar{\tau} - \underline{\tau}}, \quad (7.23)$$

for a given choice of $\underline{\tau}, \bar{\tau}$. These polynomials have the desirable property that for all $\tau \in [\underline{\tau}, \bar{\tau}]$, $b_n(\tau) \in [0, 1]$, $\sum b_n(\tau) = 1$ and therefore

$$\mathbf{x}_{(i|k)}(\tau) = \boldsymbol{\beta}_{i0}b_0(\tau) + \boldsymbol{\beta}_{i1}b_1(\tau) + \boldsymbol{\beta}_{i2}b_2(\tau) \in \text{conv}\{\boldsymbol{\beta}_{i0}, \boldsymbol{\beta}_{i1}, \boldsymbol{\beta}_{i2}\}. \quad (7.24)$$

So, the polynomial $\mathbf{x}_{(i|k)}(\tau)$ is always contained in the convex hull of the points $\beta_{i0}, \beta_{i1}, \beta_{i2}$ whenever $\tau \in [\underline{\tau}, \bar{\tau}]$. They can be interpreted geometrically as control points of $\mathbf{x}_{(i|k)}(\tau)$ where $\beta_{i0} = \mathbf{x}_{(i|k)}(\underline{\tau})$, $\beta_{i2} = \mathbf{x}_{(i|k)}(\bar{\tau})$ and β_{i1} is the intersection of the tangents to the points $\mathbf{x}_{(i|k)}(\underline{\tau})$ and $\mathbf{x}_{(i|k)}(\bar{\tau})$, as can be seen in Figure 7.4. Details on the transformation from monomial to Bernstein coordinates can be found in Appendix A.

The use of Bernstein polynomials is not uncommon in robotics due to its desirable convexity properties. Fernbach *et al.* used Bézier curves to compute kinematically and dynamically feasible transitions between two configurations of a legged robot in a multi-contact scenario [Fernbach 2018]. Pekarovskiy *et al.* consider online motion generation for reactive robotic tasks with prescribed behavior by employing spline deformation techniques [Pekarovskiy 2018].

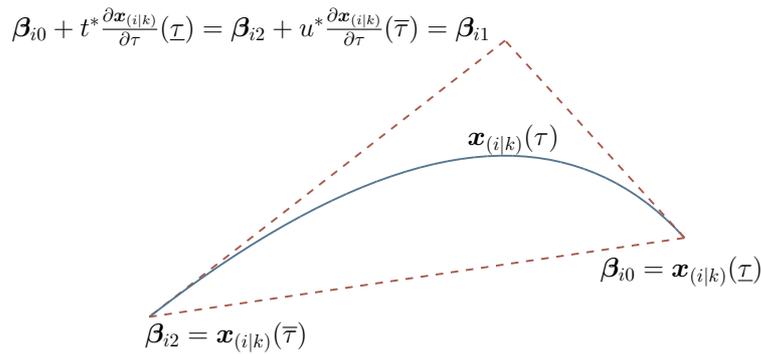


Figure 7.4: Boundaries of $\mathbf{x}_{(i|k)}(\tau)$ defined as $\text{conv}\{\beta_{i0}, \beta_{i1}, \beta_{i2}\}$.

We can define the set-valued function

$$\mathcal{F}_\tau(\mathbf{x}_{(i|k)}) \triangleq \{\beta_{i0}, \beta_{i1}, \beta_{i2}\}, \quad (7.25)$$

such that the convex hull of $\mathcal{F}_\tau(\mathbf{x}_{(i|k)})$ is an outer approximation of $\mathbf{x}_{(i|k)}(\tau)$ in the interval $[\underline{\tau}, \bar{\tau}]$:

$$\forall \tau \in [\underline{\tau}, \bar{\tau}], \quad \mathbf{x}_{(i|k)}(\tau) \in \text{conv}\{\mathcal{F}_\tau(\mathbf{x}_{(i|k)})\}, \quad (7.26)$$

as in (6.32). We can define similar set-valued functions for the vectors $\mathbf{p}_{(i|k)}(\tau), \mathbf{c}_{(i|k)}(\tau)$:

$$\mathcal{F}_\tau(\mathbf{p}_{(i|k)}) \triangleq \{\mathbf{C}\beta_{i0}, \mathbf{C}\beta_{i1}, \mathbf{C}\beta_{i2}\}, \quad (7.27a)$$

$$\mathcal{F}_\tau(\mathbf{c}_{(i|k)}) \triangleq \{\mathbf{I}_c\beta_{i0}, \mathbf{I}_c\beta_{i1}, \mathbf{I}_c\beta_{i2}\}, \quad (7.27b)$$

where \mathbf{C} is as in (7.3) and \mathbf{I}_c is a selection matrix for the position of the CoM in the state vector. In this way, constraints (7.17) can be substituted with the safe linear constraints:

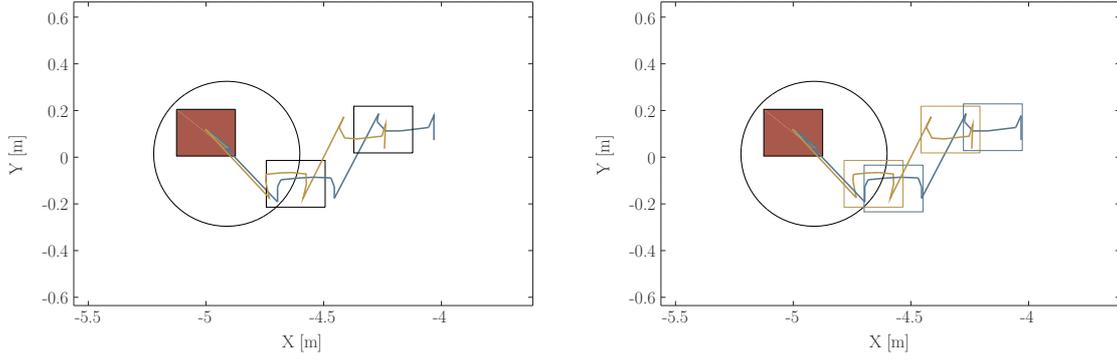
$$\mathbf{s}_{(j+1|k)} \in \bar{\mathcal{S}}(\mathbf{s}_{(j|k)}), \quad \mathcal{F}_\tau(\mathbf{c}_{(i|k)}) \subseteq \bar{\mathcal{C}}(\mathbf{s}_{(j|k)}), \quad \mathcal{F}_\tau(\mathbf{p}_{(i|k)}) \subseteq \bar{\mathcal{P}}(\mathbf{s}_{(j|k)}). \quad (7.28)$$

Relating the position and the duration of the steps

However, it is not always possible to satisfy constraints (7.28) with a single choice of $\mathbf{s}_{(j|k)}$ for all values of τ . In particular, the constraint on the CoP is the most restrictive as can be easily seen in Figure 7.5: different trajectories of the CoP for different τ do not fit all inside a single sequence of steps (the trajectory of the CoP for $\tau = \underline{\tau}$ is shown in yellow and for $\tau = \bar{\tau}$ is shown in blue).

We propose to vary the steps as a function of τ :

$$\tilde{\mathbf{s}}_k(\tau) = (\mathbf{s}_{(1|k)}(\tau), \dots, \mathbf{s}_{(J|k)}(\tau)). \quad (7.29)$$

Figure 7.5: Future trajectories of the CoP for different τ .

Each step $\mathbf{s}_{(j|k)}(\tau)$ is now the sum of a term $\mathbf{s}_{(j|k)}^0$, that is constant during an interval of duration τ , and a term $\mathbf{s}_{(j|k)}^1(\tau)$ that depends on τ

$$\mathbf{s}_{(j|k)}(\tau) = \mathbf{s}_{(j|k)}^0 + \mathbf{s}_{(j|k)}^1(\tau). \quad (7.30)$$

Let the set $\mathcal{H}(j)$ contains the indices of the samples in the preview horizon that correspond to the j^{th} previewed step. We choose $\mathbf{s}_{(j|k)}^1(\tau)$ to be the centroid of the trajectory of the CoP inside the j^{th} step in the preview horizon:

$$\mathbf{s}_{(j|k)}^1(\tau) = \frac{1}{|\mathcal{H}(j)|} \sum_{i \in \mathcal{H}(j)} \mathbf{p}_{(i|k)}(\tau). \quad (7.31)$$

Since $\mathbf{p}_{(i|k)}(\tau) = \mathbf{C}\mathbf{x}_{(i|k)}(\tau) = \mathbf{C}(\boldsymbol{\alpha}_{i0} + \boldsymbol{\alpha}_{i1}\tau + \boldsymbol{\alpha}_{i2}\tau^2)$, we have that:

$$\mathbf{s}_{(j|k)}^1(\tau) = \frac{\mathbf{C}}{|\mathcal{H}(j)|} \left(\sum_{i \in \mathcal{H}(j)} \boldsymbol{\alpha}_{i0} + \sum_{i \in \mathcal{H}(j)} \boldsymbol{\alpha}_{i1}\tau + \sum_{i \in \mathcal{H}(j)} \boldsymbol{\alpha}_{i2}\tau^2 \right), \quad (7.32a)$$

$$= \frac{\mathbf{C}}{|\mathcal{H}(j)|} (\hat{\boldsymbol{\zeta}}_{j0} + \hat{\boldsymbol{\zeta}}_{j1}\tau + \hat{\boldsymbol{\zeta}}_{j2}\tau^2), \quad (7.32b)$$

$$= \boldsymbol{\zeta}_{j0} + \boldsymbol{\zeta}_{j1}\tau + \boldsymbol{\zeta}_{j2}\tau^2. \quad (7.32c)$$

We can then write the polynomials as (we omit the subscripts $(\cdot)_{(i|k)}$, $(\cdot)_{(j|k)}$ to avoid cluttering the notation):

$$\mathbf{s}_{j+1}(\tau) - \mathbf{s}_j(\tau) = (\mathbf{s}_{j+1}^0 + \boldsymbol{\zeta}_{(j+1)0}) - (\mathbf{s}_j^0 + \boldsymbol{\zeta}_{j0}) + (\boldsymbol{\zeta}_{(j+1)1} - \boldsymbol{\zeta}_{j1})\tau + (\boldsymbol{\zeta}_{(j+1)2} - \boldsymbol{\zeta}_{j2})\tau^2, \quad (7.33a)$$

$$\mathbf{p}_i(\tau) - \mathbf{s}_j(\tau) = (\mathbf{C}\boldsymbol{\alpha}_{i0} - (\mathbf{s}_j^0 + \boldsymbol{\zeta}_{j0})) + (\mathbf{C}\boldsymbol{\alpha}_{i1} - \boldsymbol{\zeta}_{j1})\tau + (\mathbf{C}\boldsymbol{\alpha}_{i2} - \boldsymbol{\zeta}_{j2})\tau^2, \quad (7.33b)$$

$$\mathbf{c}_i(\tau) - \mathbf{s}_j(\tau) = (\mathbf{I}_c\boldsymbol{\alpha}_{i0} - (\mathbf{s}_j^0 + \boldsymbol{\zeta}_{j0})) + (\mathbf{I}_c\boldsymbol{\alpha}_{i1} - \boldsymbol{\zeta}_{j1})\tau + (\mathbf{I}_c\boldsymbol{\alpha}_{i2} - \boldsymbol{\zeta}_{j2})\tau^2, \quad (7.33c)$$

along with the set-valued functions:

$$\mathcal{F}_\tau(\mathbf{s}_{j+1} - \mathbf{s}_j) \triangleq \{\boldsymbol{\phi}_{(j+1)0}, \boldsymbol{\phi}_{(j+1)1}, \boldsymbol{\phi}_{(j+1)2}\}, \quad (7.34a)$$

$$\mathcal{F}_\tau(\mathbf{p}_i - \mathbf{s}_j) \triangleq \{\boldsymbol{\lambda}_{i0}, \boldsymbol{\lambda}_{i1}, \boldsymbol{\lambda}_{i2}\}, \quad (7.34b)$$

$$\mathcal{F}_\tau(\mathbf{c}_i - \mathbf{s}_j) \triangleq \{\boldsymbol{\pi}_{i0}, \boldsymbol{\pi}_{i1}, \boldsymbol{\pi}_{i2}\}. \quad (7.34c)$$

The original nonlinear constraints on the dynamics and the kinematics of the robot can then be substituted with the safe linear constraints:

$$\mathcal{F}_\tau(\mathbf{s}_{j+1} - \mathbf{s}_j) \subseteq \overline{\mathcal{S}}, \quad \mathcal{F}_\tau(\mathbf{p}_i - \mathbf{s}_j) \subseteq \overline{\mathcal{P}}, \quad \mathcal{F}_\tau(\mathbf{c}_i - \mathbf{s}_j) \subseteq \overline{\mathcal{C}}. \quad (7.35)$$

7.4.2 Collision avoidance with the crowd

In order to avoid collisions, the CoM $\mathbf{c}_{(i|k)}(\tau)$ has to stay away from the region occupied by the person $\mathbf{m}_{(i|k)}(\tau)$

$$\mathbf{c}_{(i|k)}(\tau) - \mathbf{m}_{(i|k)}(\tau) \notin \text{ball}(\mathbf{0}, D(i\tau)). \quad (7.36)$$

With the model of motion of the crowd described in Section 4.2.2, uncertainties in position D_p and velocity D_v make this radius increase with time (for safety) from the real radius D_{person} of the person

$$D(t) = D_{\text{robot}} + (D_{\text{person}} + D_p + D_v t). \quad (7.37)$$

We compute an outer approximation to the LHS of the inclusion (7.36), for all $\tau \in [\underline{\tau}, \bar{\tau}]$. We can also write $\mathbf{m}_{(i|k)}(\tau)$ in Bernstein basis as $\boldsymbol{\mu}_0 b_0(\tau) + \boldsymbol{\mu}_1 b_1(\tau) + \boldsymbol{\mu}_2 b_2(\tau)$ and, therefore,

$$\mathbf{c}_{(i|k)}(\tau) - \mathbf{m}_{(i|k)}(\tau) = \boldsymbol{\sigma}_{i0} b_0(\tau) + \boldsymbol{\sigma}_{i1} b_1(\tau) + \boldsymbol{\sigma}_{i2} b_2(\tau) \in \text{conv}\{\boldsymbol{\sigma}_{i0}, \boldsymbol{\sigma}_{i1}, \boldsymbol{\sigma}_{i2}\}, \quad (7.38)$$

where $\boldsymbol{\sigma}_{in} = \mathbf{I}_c \boldsymbol{\beta}_{in} - \boldsymbol{\mu}_{in}$. We can then define the set-valued function

$$\mathcal{F}_\tau(\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)}) \triangleq \{\boldsymbol{\sigma}_{i0}, \boldsymbol{\sigma}_{i1}, \boldsymbol{\sigma}_{i2}\}, \quad (7.39)$$

and redefine the constraint on collision avoidance as:

$$\mathcal{F}_\tau(\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)}) \not\subseteq \text{ball}(\mathbf{0}, D(i\tau)). \quad (7.40)$$

We can also prevent collisions between $\text{conv}\{\boldsymbol{\sigma}_{in}\}$ and $\text{ball}(\mathbf{0}, D(i\tau))$ by properly defining separating planes between them [Boyd 2004, Chapter 2]. Mercy *et al.* ensure collision avoidance with a combination of separating hyperplanes and B-spline parametrization of the motion of the CoM [Mercy 2016]. A similar approach is taken by Brossette *et al.* [Brossette 2017] for continuous collision avoidance. In comparison (with Mercy *et al.*), we use similar techniques but we solve different problems: we ensure collision avoidance at a given sample in the preview horizon for all durations of the trajectory, while they ensure it throughout the whole preview horizon for a given duration of the trajectory.

7.4.3 Terminal constraint for stopping

We look for a second mode controller, as in (3.25), for MPC of the form

$$\mathbf{u}_{(i|k)} = \mathbf{K} \mathbf{x}_{(i|k)}, \quad \forall i \in \{N, N+1, \dots\} \quad (7.41)$$

that prevents changes in the CoP; which implies that all its derivatives must be equal to zero. In our triple integrator, the jerk of the CoM is related to the velocity of the CoP as follows:

$$\ddot{\mathbf{c}} = \omega^2 (\dot{\mathbf{c}} - \dot{\mathbf{p}}). \quad (7.42)$$

The feedback matrix \mathbf{K} for (7.41) should be, therefore

$$\mathbf{K} = \text{diag} \left(\begin{bmatrix} 0 & \omega^2 & 0 \end{bmatrix} \right), \quad (7.43)$$

yielding the autonomous system

$$\dot{\mathbf{x}} = (\hat{\mathbf{A}} + \hat{\mathbf{B}}\mathbf{K})\mathbf{x}. \quad (7.44)$$

With the CoP stopped we can now analyze the stability of (7.44) with the eigen-decomposition

$$(\hat{\mathbf{A}} + \hat{\mathbf{B}}\mathbf{K}) = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^{-1}, \quad (7.45)$$

where

$$\mathbf{\Lambda} = \text{diag}_2 \left(\begin{bmatrix} -\omega & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & 0 \end{bmatrix} \right), \quad (7.46)$$

is the matrix of eigenvalues and

$$\mathbf{V}^{-1} = \text{diag}_2 \left(\begin{bmatrix} 0 & -\frac{1}{2\omega} & \frac{1}{2\omega^2} \\ 0 & \frac{1}{2\omega} & \frac{1}{2\omega^2} \\ 1 & 0 & -\frac{1}{\omega^2} \end{bmatrix} \right), \quad (7.47)$$

is the matrix of eigenvectors. We want to make the unstable mode of $\mathbf{V}^{-1}\mathbf{x}$, corresponding to the positive eigenvalue ω , equal to zero. This translates into the following condition:

$$\dot{\mathbf{c}} + \frac{1}{\omega}\ddot{\mathbf{c}} = \mathbf{0}. \quad (7.48)$$

We can substitute it back into (2.16) to obtain an alternative formulation:

$$\mathbf{p} = \mathbf{c} + \frac{1}{\omega}\dot{\mathbf{c}}. \quad (7.49)$$

The RHS is the Capture Point (CP)

$$\boldsymbol{\xi} = \text{diag}_2 \left(\begin{bmatrix} 1 & \frac{1}{\omega} & 0 \end{bmatrix} \right) \mathbf{x} = \mathbf{E}\mathbf{x}, \quad (7.50)$$

which was independently described in [Pratt 2006, Hof 2008, Takenaka 2009].

We want the robot to be *zero-step capturable* [Koolen 2012] at the end of the preview horizon, *i.e.* we want the robot to be able to stop the motion of the CoM without having to make any extra step at the end of the preview horizon. Our terminal constraint is, therefore, defined as:

$$\mathbf{p}_{(N|k)}(\tau) = \boldsymbol{\xi}_{(N|k)}(\tau). \quad (7.51)$$

Remark 5. We do not describe the approximation we do when going from the continuous (7.49) to the discrete constraint (7.51).

Since $\boldsymbol{\xi}_{(i|k)}$ is a linear combination of $\mathbf{x}_{(i|k)}$, as shown in (7.50), then the convex hull of the set-valued function

$$\mathcal{F}_\tau(\boldsymbol{\xi}_{(i|k)}) \triangleq \{\mathbf{E}\boldsymbol{\beta}_{i0}, \mathbf{E}\boldsymbol{\beta}_{i1}, \mathbf{E}\boldsymbol{\beta}_{i2}\}, \quad (7.52)$$

is an outer approximation of $\boldsymbol{\xi}_{(i|k)}(\tau)$, for all $\tau \in [\underline{\tau}, \bar{\tau}]$. The constraint (7.51) can be safely approximated as:

$$\mathcal{F}_\tau(\mathbf{p}_{(N|k)}) = \mathcal{F}_\tau(\boldsymbol{\xi}_{(N|k)}). \quad (7.53)$$

However, this is a constraint that is difficult to satisfy. In practice we relax it by requiring, instead, that

$$\mathcal{F}_\tau(\boldsymbol{\xi}_{(N|k)} - \mathbf{s}_{(J|k)}) \subseteq \bar{\mathcal{P}}, \quad (7.54)$$

and we set (7.51) as an objective of the robot.

7.5 Objectives of the walk

The desired behavior of the robot is to:

1. Keep each person in the crowd at a comfortable distance, *i.e.* no less than d_{ref} from the CoM.

2. Move the CoM at a reference velocity $\dot{\mathbf{c}}_{\text{ref}}$.
3. Keep the CoP at the center of the foot for improved balance.
4. Minimize the jerk of the CoM for smoothness of its motion and regularization.
5. Penalize step durations different from a given reference value $\delta\tau_{\text{ref}}$ where energy expenditure is optimal.
6. Align the feet with the direction where the CoM is moving.
7. Make the robot zero-step capturable at the end of the preview horizon.

In summary, we consider the following (instantaneous) objectives at the i^{th} sample of the preview horizon:

$$\begin{bmatrix} d_{\text{ref}} \\ \dot{\mathbf{c}}_{\text{ref}} \\ \mathbf{s}_{(j|k)} \\ \mathbf{0} \\ \tau_{\text{ref}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \|\mathbf{c}_{(i|k)}(\tau) - \mathbf{m}_{(i|k)}(\tau)\| \\ \dot{\mathbf{c}}_{(i|k)}(\tau) \\ \mathbf{p}_{(i|k)}(\tau) \\ (\ddot{\mathbf{c}}_{(i|k)} - \ddot{\mathbf{c}}_{(i-1|k)})/\tau \\ \tau \\ (\cos(\theta_{(j|k)}), \sin(\theta_{(j|k)})) \times \dot{\mathbf{c}}_{(i|k)}(\tau) \\ \mathbf{p}_{(N|k)} - \boldsymbol{\xi}_{(N|k)} \end{bmatrix} \leq \begin{bmatrix} +\infty \\ \dot{\mathbf{c}}_{\text{ref}} \\ \mathbf{s}_{(j|k)} \\ \mathbf{0} \\ \tau_{\text{ref}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (7.55)$$

which we compile as:

$$\underline{\mathbf{f}}_i \leq \mathbf{f}_i(\ddot{\mathbf{c}}_i, \mathbf{s}_j, \tau, \theta_j) \leq \overline{\mathbf{f}}_i. \quad (7.56)$$

7.6 Optimal Control Problem

7.6.1 NLP formulation

We produce a feasible motion of the CoM and the CoP as well as a feasible position, orientation and duration of the steps at the k^{th} MPC problem by solving the following NLP:

$$\underset{\tilde{\mathbf{c}}_k, \tilde{\mathbf{s}}_k, \tilde{\boldsymbol{\theta}}_k, \tau}{\text{minimize}} \quad \sum_{i=0}^{N-1} \|\mathbf{v}_i\|_{\mathbf{Q}}^2 \quad (7.57a)$$

$$\text{subject to} \quad \underline{\mathbf{f}}_i \leq \mathbf{f}_i(\ddot{\mathbf{c}}_i, \mathbf{s}_j, \tau, \theta_j) - \mathbf{v}_i \leq \overline{\mathbf{f}}_i, \quad (7.57b)$$

$$\mathbf{s}_{j+1} \in \overline{\mathcal{S}}(\mathbf{s}_j, \theta_j), \quad (7.57c)$$

$$\mathbf{c}_i(\tau) \in \overline{\mathcal{C}}(\mathbf{s}_j, \theta_j) \cap \text{ball}(\mathbf{m}_i(\tau), D(i\tau)), \quad (7.57d)$$

$$\mathbf{p}_i(\tau) \in \overline{\mathcal{P}}(\mathbf{s}_j, \theta_j), \quad (7.57e)$$

$$\boldsymbol{\xi}_N(\tau) = \mathbf{s}_J(\tau), \quad (7.57f)$$

$$\tau \in [\underline{\tau}, \overline{\tau}], \quad (7.57g)$$

$$\theta_j \in [\underline{\theta}_j, \overline{\theta}_j], \quad (7.57h)$$

with decision variables $(\tilde{\mathbf{c}}_k, \tilde{\mathbf{s}}_k, \tilde{\boldsymbol{\theta}}_k, \tau)$. \mathbf{Q} is a diagonal matrix of weights. We omit the subscripts $(\cdot)_{(i|k)}, (\cdot)_{(j|k)}$ to avoid cluttering the notation.

Remark 6. *In this formulation we do not consider the safe navigation strategies discussed in Chapter 4.*

7.6.2 Safe SQP approach

We solve each [NLP \(7.57\)](#) using a [Safe SQP](#) method: we solve a sequence of lexicographic least squares subproblems with carefully crafted safe linear constraints. Specifically, we seek to:

$$\text{lex minimize}_{\tilde{\mathbf{c}}_k, \tilde{\mathbf{s}}_k, \tilde{\boldsymbol{\theta}}_k, \tau} \left(\text{viol. of constraints, } \sum_{i=0}^{N-1} \|\mathbf{v}_i\|_Q^2 \right) \quad (7.58a)$$

$$\text{subject to}_{\forall i, \forall j} \underline{\mathbf{f}}_i \leq \mathbf{f}_i(\tilde{\mathbf{c}}_i, \mathbf{s}_j, \tau, \theta_j) + \nabla \mathbf{f}^T(\tilde{\mathbf{c}}_i, \mathbf{s}_j, \tau, \theta_j) - \mathbf{v}_i \leq \bar{\mathbf{f}}_i, \quad (7.58b)$$

$$\mathcal{F}_\tau(\mathbf{s}_{j+1} - \mathbf{s}_j) \subseteq \bar{\mathcal{S}}, \quad (7.58c)$$

$$\mathcal{F}_\tau(\mathbf{c}_i - \mathbf{s}_j) \subseteq \bar{\mathcal{C}}, \quad (7.58d)$$

$$\mathcal{F}_\tau(\mathbf{c}_i - \mathbf{m}_i) \not\subseteq \text{ball}(\mathbf{0}, D(i\tau)), \quad (7.58e)$$

$$\mathcal{F}_\tau(\mathbf{p}_i - \mathbf{s}_j) \subseteq \bar{\mathcal{P}}, \quad (7.58f)$$

$$\mathcal{F}_\tau(\boldsymbol{\xi}_N - \mathbf{s}_j) \subseteq \bar{\mathcal{P}}, \quad (7.58g)$$

$$\tilde{\mathbf{c}}_i \in [\underline{\tilde{\mathbf{c}}}, \bar{\tilde{\mathbf{c}}}], \quad (7.58h)$$

$$\tau \in [\underline{\tau}, \bar{\tau}], \quad (7.58i)$$

$$\theta_j \in [\underline{\theta}_j, \bar{\theta}_j]. \quad (7.58j)$$

7.7 Conclusion

We described the formulation of the [MPC](#) we use for biped walking in great detail. We specified the objectives of the walk and we made special emphasis on how we defined safe linear constraints with respect to the original nonlinear constraints. We formulated each [MPC](#) problem as a [NLP](#) and we solved each of them with a [Safe SQP](#) method.

Chapter 8

Simulations

8.1 Introduction

We present, in this chapter, simulation results and numerical evidence of the performance of the methods and controllers described in the previous chapters.

In Section 8.2 we test the safe navigation strategies we introduced in Chapter 4 on different crowds. In Section 8.3 we show the typical behaviors we obtain when we allow the duration and orientation of the steps of the robot to be adapted online in order to avoid collisions with the crowd. We evaluate the performance of the proposed **Safe SQP** method in comparison with a standard line-search **SQP** method under two different scenarios.

Table 8.1 shows the kinematic parameters of our simulated robot, which is based on the HRP-2 [Kaneko 2004].

8.2 Safe navigation strategies

We evaluate numerically the performance of the proposed safe navigation strategies by making the robot traverse a moving crowd under different environmental conditions and parameters of the controller.

Tables 8.3 and 8.4 show the parameters of the two scenarios of our numerical evaluations whereas Table 8.2 shows parameters that are common for both scenarios. For each combination of size of the crowd and speed along the x -axis of people (M, \dot{m}^x) , we randomly generate and store 100 different crowds that only differ in the initial positions $\{\mathbf{m}_0\}$ and speeds along the y -axis $\{\dot{m}^y\}$ of the people. The initial positions vary *uniformly* over an area of $10 \times 8[\text{m}^2]$ while speeds along the y -axis follow a *normal* distribution $\mathcal{N}(0, 0.2)$. We consider fixed step rotations and step durations in this section.

Table 8.1: Parameters of biped robot and the persons in the crowd

Parameter	Symbol	Value	Unit
Height of CoM (2.17)	c^z	0.80	m
Feet dimensions (7.11)	(w, l)	(0.24, 0.14)	(m, m)
Leg stride (7.11)	(W, L)	(0.30, 0.30)	m
Feet separation (7.11)	d_0	0.01	m
Body radius (3.7)	D_{robot}	0.50	m
Body radius (3.5)	D_{person}	0.50	m

Table 8.2: Common parameters

Parameter	Symbol	Value	Unit
Reference velocity (7.55)	\dot{c}_{ref}^x	0.5	ms^{-1}
	\dot{c}_{ref}^y	0	ms^{-1}
Sampling time (7.4)	τ	0.1	s
Samples per step	N_s	8	samples
Step duration	$N_s\tau$	0.8	s
Interval of step duration	$N_s\Delta\tau$	0	s
Step rotation	θ	0	deg
Interval of rotations	$\Delta\theta$	0	deg

Table 8.3: Parameters of simulations with no uncertainty

Parameter	Symbol	Value	Unit
Horizon duration (7.6)	H	{1.0, 1.8, 2.6}	s
Radius of FoV (4.3)	D_{fov}	{0.5, 1, 2, 3, 4}	m
Size of crowd (3.6)	M	{8, 16, 32}	-
Velocity of the crowd	\dot{m}^x	{0.25, 0.5, 1, 1.5, 2}	ms^{-1}
	\dot{m}^y	$\mathcal{N}(0, 0.2)$	ms^{-1}
Uncertainty (4.10)	D_p	0	m
	D_v	0	ms^{-1}

During simulations, we test each configuration of the controller ($H, D_{\text{fov}}, D_p, D_v$) against each of the 100 crowds generated for each combination (M, \dot{m}^x). Simulations last 20[s] or until a failure occurs, *i.e.* when the kinematic or dynamical constraints are violated or when the robot collides with a person.

8.2.1 No uncertainty in the motion of the crowd

Figure 8.1 synthesizes the results obtained from all simulations with no uncertainty in the motion of the crowd. Plots on the left show the failure rate as a function of the radius of the FoV and for different walking speeds of the crowd. Plots on the right report the corresponding median anticipation time to failure, which is the elapsed time between the moment the earliest alarm was triggered and the moment the failure happened. It is worth mentioning that, in these simulations with no uncertainty, all failures are anticipated as early as permitted by the limits of the FoV and the prediction horizon, and no false alarms are raised.

We can see two different regimes represented with either dashed lines or solid lines. The threshold is attained when the radius D_{fov} of the FoV reaches

$$H(\dot{m}^x + \dot{c}_{\text{ref}}^x), \quad (8.1)$$

which is the minimum value for detecting all people that might collide with the robot over a horizon of duration H , with walking speeds \dot{m}^x for the people and \dot{c}_{ref}^x for the robot.

If the FoV is smaller than this minimum value, the anticipation time for collisions grows linearly with the radius of the FoV and the failure rate decreases accordingly. It does not,

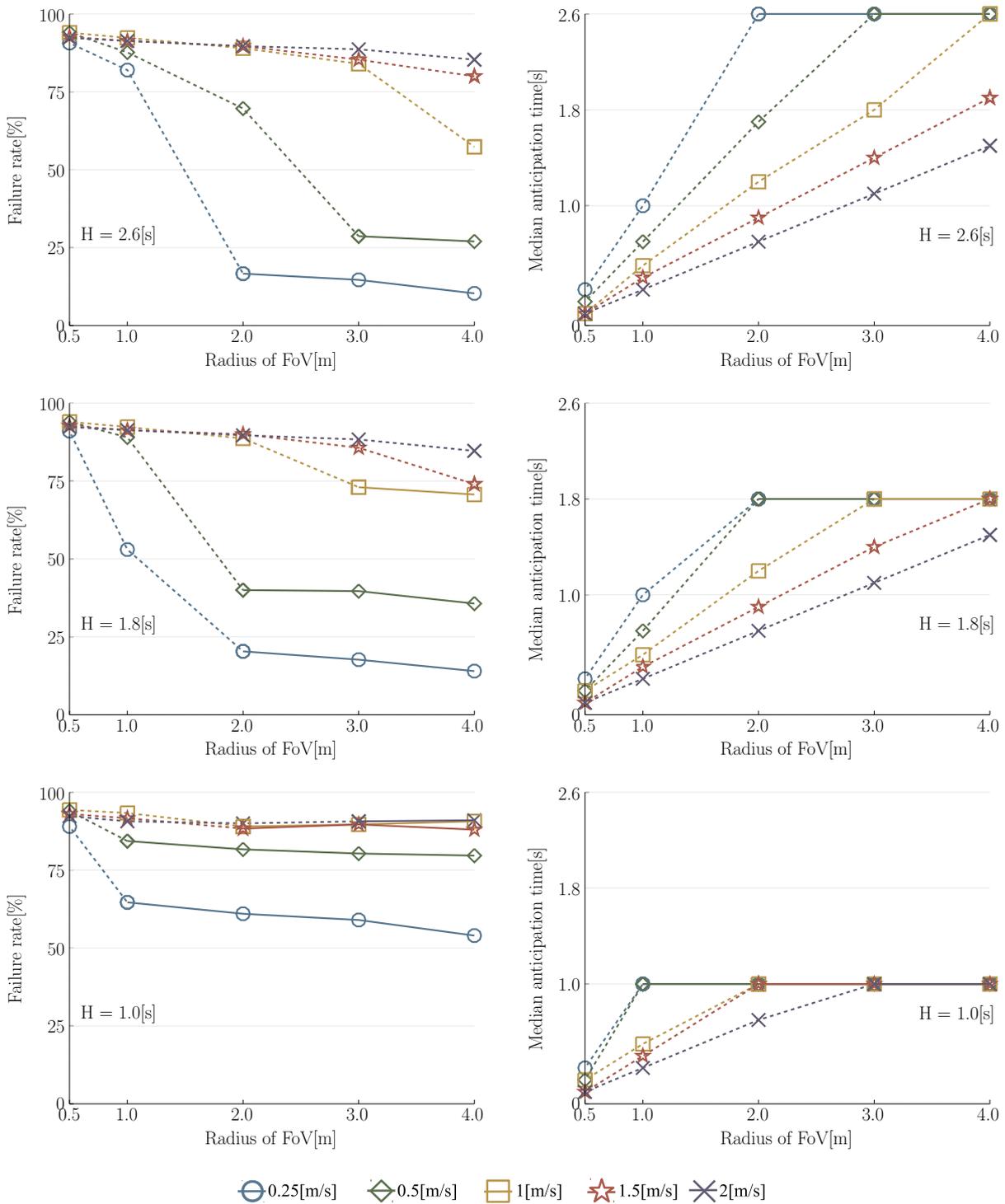


Figure 8.1: Failure rate and median anticipation time for different speeds of the crowd and different durations of the horizon.

Table 8.4: Parameters of simulations with uncertainty

Parameter	Symbol	Value	Unit
Horizon duration (7.6)	H	1.8	s
Radius of FoV (4.3)	D_{fov}	4	m
Size of crowd (3.6)	M	16	-
Velocity of the crowd	\dot{m}^x	0.5	ms^{-1}
	\dot{m}^y	$\mathcal{N}(0, 0.2)$	ms^{-1}
Uncertainty (4.10)	D_p	$\{0, 0.15, 0.30\}$	m
	D_v	$\{0, 0.05, 0.10\}$	ms^{-1}

however, decrease with longer prediction horizons: it appears that longer prediction horizons can degrade collision avoidance if the radius of the FoV is not large enough, as the robot is making decisions without proper knowledge of the surrounding crowd.

If the FoV is larger than this minimum value, the anticipation time for collisions reaches exactly the duration H of the prediction horizon, and the failure rate reaches a minimum, which does not vary further with the radius of the FoV.

When the robot walks in an inattentive crowd that moves at a standard human walking speed (between $1[\text{ms}^{-1}]$ and $1.5[\text{ms}^{-1}]$ [Ralston 1958]), the probability of collisions is always higher than 50%. The robot does not have enough kinematic and dynamic capabilities to avoid collisions with people walking that fast. We can, however, anticipate these collisions as early as needed, with appropriate FoV and prediction horizon following the simple suggestions of (8.1), and initiate an emergency stop.

8.2.2 Uncertainty in the motion of the crowd

Tables 8.5 and 8.6 show, for passive safety and collision mitigation strategies respectively, the number of failures that take place when there is uncertainty in the perception of people. In each table, the first two columns are the values of uncertainty. The next two columns present the median velocities of both the robot and the person at the time of collision along the direction of the contact. Cases with no collision are indicated with a dash “-”. The last column shows the proportion of simulations that ended with a failure.

The failure rate of the emergency stop strategy varies little with the uncertainty in position but increases with the uncertainty in velocity. Uncertainty causes this strategy to make unnecessary and undesirable stops that lead to collisions. In the deferrable emergency stop strategy, the failure rate seems unaffected by the uncertainty in position as well but decreases with the uncertainty in velocity.

It is important to note that, in our model of the crowd, the motion of people does not become more *erratic* with increased uncertainty. What increases is the size of the areas where people might be in the future. With only uncertainty in position, these areas have equal size throughout the horizon. People appear to be larger and there is less space to evade them. With only uncertainty in velocity, these areas increase in size along the preview horizon. People appear to be larger at the distance but reduce their size as the robot approaches them.

The velocity of the robot at the moment of collision is either zero or close to zero but negative, as expected. This suggests that the robot is trying to walk away from people before the collision happens.

The controller that implements Hierarchy 4 walks away from people as much as possible,

Table 8.5: Collisions of passive safety strategies

Uncertainty		Collision velocity[ms ⁻¹]		Failure rate[%]
Position[m]	Velocity[ms ⁻¹]	Robot	Person	
0	0	-0.04	0.48	27
0	0.05	0.00	0.52	34
0	0.10	0.00	0.41	36
0.15	0	-0.01	0.44	28
0.15	0.05	0.00	0.40	38
0.15	0.10	0.00	0.41	38
0.30	0	0.00	0.46	34
0.30	0.05	0.00	0.50	40
0.30	0.10	0.00	0.41	39

(a) Hierarchy 1: emergency stop

Uncertainty		Collision velocity[ms ⁻¹]		Failure rate[%]
Position[m]	Velocity[ms ⁻¹]	Robot	Person	
0	0	-0.05	0.36	32
0	0.05	0.00	0.55	15
0	0.10	0.00	0.55	9
0.15	0	-0.01	0.42	31
0.15	0.05	0.00	0.52	15
0.15	0.10	0.00	0.55	9
0.30	0	0.00	0.45	35
0.30	0.05	0.00	0.55	15
0.30	0.10	0.00	0.53	9

(b) Hierarchy 1: deferrable emergency stop

Table 8.6: Collisions of collision mitigation strategies

Uncertainty		Collision velocity[ms ⁻¹]		Failure rate[%]
Position[m]	Velocity[ms ⁻¹]	Robot	Person	
0	0	-0.33	0.39	19
0	0.05	-0.71	0.60	4
0	0.10	-0.53	0.59	2
0.15	0	-0.19	0.49	15
0.15	0.05	-0.74	0.58	5
0.15	0.10	-0.64	0.58	2
0.30	0	-0.14	0.56	15
0.30	0.05	-0.48	0.58	5
0.30	0.10	-0.28	0.57	2

(a) Hierarchy 3: relaxation of collision avoidance

Uncertainty		Collision velocity[ms ⁻¹]		Failure rate[%]
Position[m]	Velocity[ms ⁻¹]	Robot	Person	
0	0	-0.60	0.39	21
0	0.05	-	-	11
0	0.10	-	-	13
0.15	0	-	-	22
0.15	0.05	-	-	10
0.15	0.10	-	-	13
0.30	0	-	-	27
0.30	0.05	-	-	12
0.30	0.10	-	-	13

(b) Hierarchy 4: relaxation of collision avoidance and feasibility

Table 8.7: Parameters of simulation with adaptation of the step duration

Parameter	Symbol	Value	Unit
Size of crowd (3.6)	M	16	persons
Vel. of the crowd	\dot{m}_k^x	-1.5	m.s ⁻¹
	\dot{m}_k^y	$\mathcal{N}(0, 0.4)$	m.s ⁻¹
Interval of step duration	$N_s \Delta \tau$	40	ms
Interval of step rotations	$\Delta \theta$	0	deg

even by following trajectories that lead to infeasible future states. This results in fewer collisions than any other strategy only because the robot is deciding to fall on the ground instead. On the other hand, with the controller that implements Hierarchy 3, the order of the priorities requires the motion of the robot to be dynamically and kinematically feasible during the whole prediction. The resulting number of collisions is higher but the total number of failures is lower.

The controller that implements Hierarchy 3 *delays* collisions as much as possible. The corresponding optimal control action is, in most cases, to make the robot walk away from people while in some other specific cases (*e.g.* when people are coming towards the robot from all directions at the same speed) it is to make it remain still. In any case, the difference of velocities at the moment of collision is guaranteed to be no bigger than it would be if the robot was at rest. Collisions are, therefore, less severe.

8.3 Safe Sequential Quadratic Programming

8.3.1 Adapting the duration of the steps

We make the robot walk in a crowd that has a higher average walking speed than the nominal speed of the robot. We want to see the action of the controller adapting the duration of the steps in order to walk faster.

Figure 8.2 shows the results obtained from one illustrative simulation where the robot adapts the duration of the steps in order to walk faster. The parameters of the controller and the crowd are specified in Table 8.7. The top subfigure shows in solid blue the x component of the steps ($s_j^x - c^x$) made by the robot during the simulation. Values above and below the red line indicate forward and backward motion, respectively. The vertical dashed lines mark the instants at which steps were made. The magenta dashed lines indicate the maximal step the robot can make. The bottom subfigure shows the x component of the velocity of the robot (\dot{c}^x) during the simulation. The magenta dashed lines indicate the maximal speed the robot can reach at a nominal step duration of 0.8[s]. The green dash-dotted line indicates the objective speed. During the first 2[s] there are no persons present in the Field of View (FoV). The robot walks freely and tries to reach its reference velocity \dot{c}_{ref} making steps of nominal duration (0.8[s]).

Between 2[s] and 16[s] the robot encounters people walking in opposite direction and moves backwards to avoid them. At first (between 2[s] and 3[s]) the robot only makes larger steps while maintaining its nominal step duration. However, its maximal speed under such conditions is only about -0.75m.s^{-1} whereas people in the crowd walk at -1.5m.s^{-1} . Consequently, it is forced (between 3[s] and 11[s]) to also make faster steps to be able to avoid

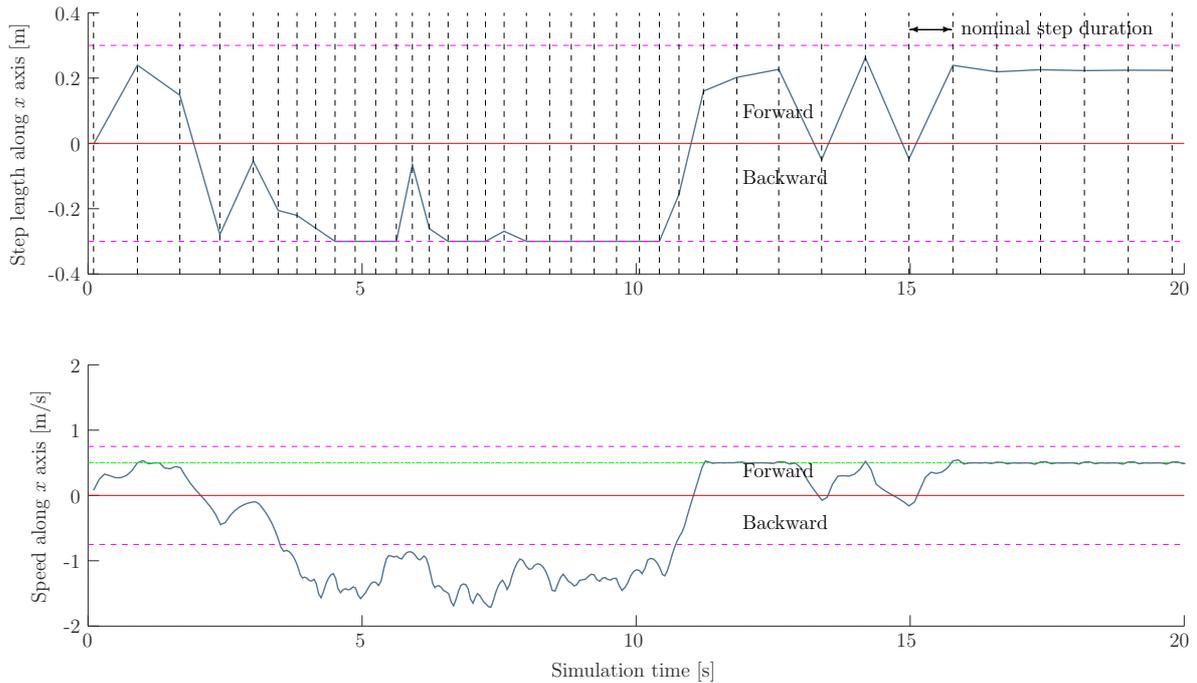


Figure 8.2: Adapting the step duration while walking in a crowd.

collisions.

At 11[s] the robot resumes its forward walking motion once it has overcome most people in its way. It has to stop occasionally a couple of times to walk around the last members of the crowd. From 16[s] until 20[s] the robot continues walking uninterrupted at its objective velocity.

The adaptation of $[\underline{\tau}, \bar{\tau}]$ allows the robot to avoid infeasibilities due to difficult collision-avoidance situations. The key here is the objective on the reference minimal distance from people: it makes the cost function take into account potential collisions in the future so that the optimal solution moves the system away from infeasibilities.

8.3.2 Adapting the rotation of the steps

Figure 8.3 shows the results obtained in a typical simulation where the robot adapts the rotation of the steps in order to face its walking direction. The parameters of the controller and the crowd for this scenario are specified in Table 8.8. The robot walks to the right and the crowd walks in the opposite direction. The top subfigure shows the trajectory of the feet over the course of the simulation. The rest of the subfigures are snapshots of the simulation at different instants. In these snapshots, the black circle represents the body of the robot, the filled rectangles are the steps currently on the ground, the unfilled rectangles are the previewed future steps and the red circles with numbers represent people. The plot at the top shows all the steps made by the robot during the simulation. The robot changes its orientation several times to avoid the people in the crowd. This is illustrated in the different snapshots shown in the same figure. Once the robot overcomes the crowd, at around 18[s], it changes its orientation back to 0 degrees to more efficiently track its reference velocity \dot{c}_{ref} .

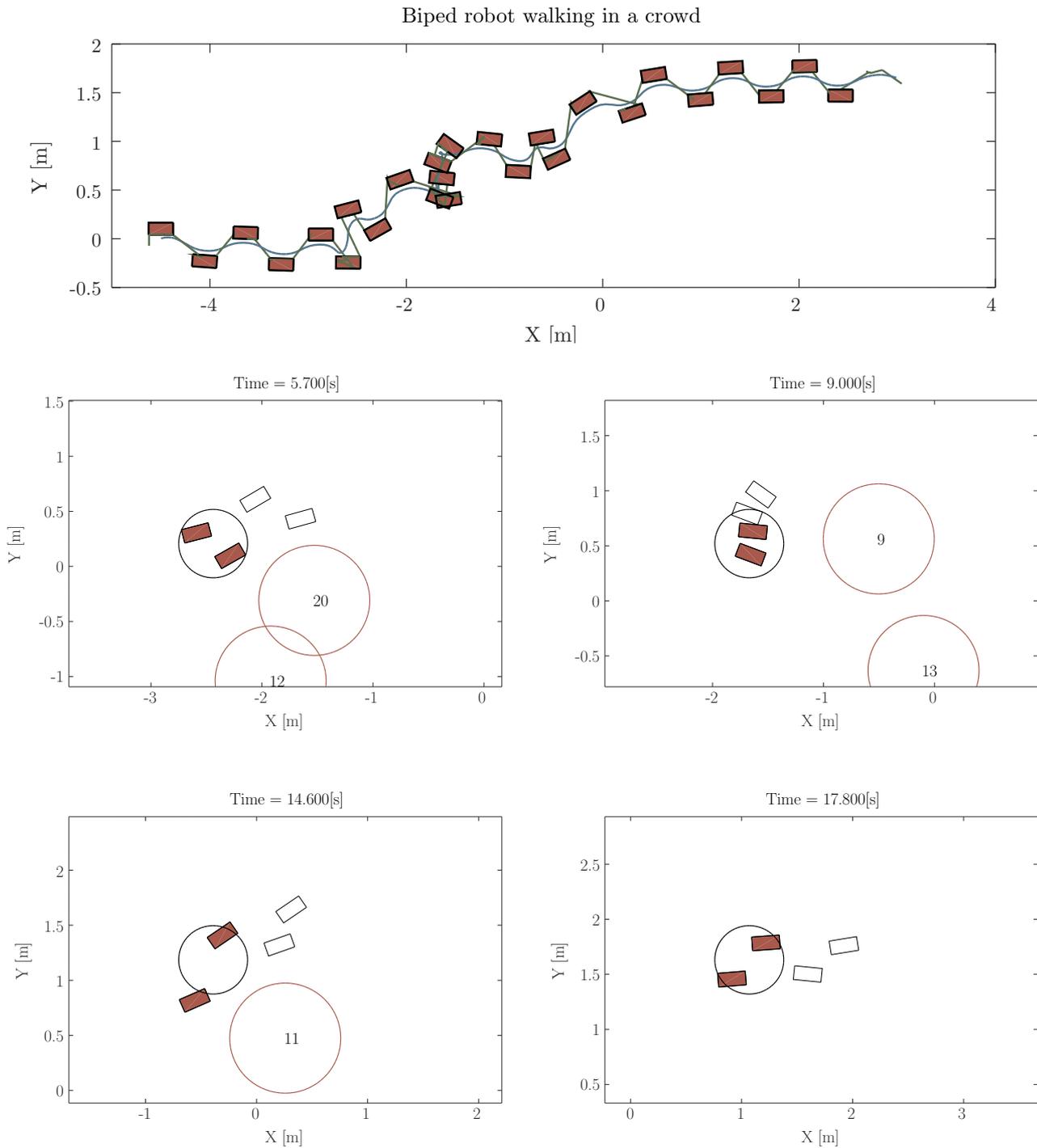


Figure 8.3: Adapting the rotation of the feet of the robot while walking in a crowd.

Table 8.8: Parameters of simulation with adaptation of step rotation

Parameter	Symbol	Value	Unit
Size of crowd (3.6)	M	20	persons
Vel. of the crowd	\dot{m}_k^x	-0.5	m.s ⁻¹
	\dot{m}_k^y	$\mathcal{N}(0, 0.5)$	m.s ⁻¹
Interval of step duration	$N_s \Delta \tau$	0	ms
Interval of step rotations	$\Delta \theta$	30	deg

Table 8.9: Parameters of simulation for comparison with SQP

Parameter	Symbol	Value	Unit
Interval of step duration	$N_s \Delta \tau$	80	ms
Interval of step rotation	$\Delta \theta$	30	deg

8.3.3 Comparison with SQP

We test how fast the [Safe SQP](#) scheme arrives to a feasible solution compared to the line-search [SQP](#) we described in Section 6.2. The [SQP](#) scheme solves the problem (7.58) by doing continuous linearizations of the constraints whereas the [Safe SQP](#) solves it using safe polytopic approximations (7.57). In both cases, our termination criteria is constraint satisfaction with arbitrary ϵ -precision, in this case, $\epsilon = 10^{-6}$. We solve the related [QP](#) and lexicographic problems subproblems using [LexLS](#). Table 8.9 shows the parameters of the robot we used in these simulations.

In active set methods, the computation time needed to solve the i^{th} optimization problem is related to the number of factorizations (activations and deactivations of constraints in the working set) needed to arrive to the optimal active set \mathcal{A}_i^* . Accurately guessing this set leads to a lower number of factorizations and faster computation times. We warm-start the i^{th} problem using the following heuristics:

- We take advantage of the periodicity of the motion of the robot by taking the corresponding optimal active set from two physical steps ago as a guess

$$\mathcal{A}_i^{\text{guess}} = \mathcal{A}_{i-2N_s}^*, \quad (8.2)$$

where N_s is the number of times we sample each physical step of the robot.

- Similarly, we use the solution of the $(i - 2N_s)^{th}$ problem as solution guess

$$\mathbf{x}_i^{\text{guess}} = \mathbf{x}_{i-2N_s}^*. \quad (8.3)$$

Walking with no crowd

We test the robot walking at a reference speed of $0.5[m.s^{-1}]$ along the x axis with no people in the environment during 20[s].

Figure 8.4 shows the results obtained with the **Safe SQP** scheme. The plot at the top-left reports the proportion of problems that were solved within a given amount of time. The lower and upper¹ bounds represent the **CPU** time spent on the instruction at line 6 in Algorithm 3 and on the whole algorithm, respectively. The plot at the top-right details the proportion of problems that were solved with a given number of factorizations of the active set. Each addition/subtraction of a constraint to/from the working set constitutes a *factorization*. The plot at the bottom-left reports the proportion of problems that were solved within a given number of Newton iterations of the **Safe SQP** algorithm. The plot at the bottom-right shows the evolution over time of the factorizations of the active set needed to terminate each optimization problem. The theoretical lower bound is the minimal number of factorizations needed to arrive to the optimal active set from the active set guess.

Our guessing strategy starts acting at the $2N^{\text{th}}$ problem. At this time, the motion of the robot is still not periodic enough (presumably due to transitory effects and initial conditions) for our active set guess to be accurate. During this phase, the theoretical lower bound of the number of factorizations stays between 2 and 50. After around 50 problems, the motion stabilizes into a cycle and the lower bound drops to zero for the rest of the simulation. As a result, 65% of the problems are solved with just one active set factorization or less and almost 80% of the problems are solved in 4[ms] or less. A feasible solution is found for 100% of the problems after a single Newton iteration.

Figure 8.5 shows the results obtained with the **SQP** scheme. In this case, the lower and upper bounds in the histogram of computation time represent the **CPU** time spent on the instruction at line 6 in Algorithm 2 and on the whole algorithm, respectively. The theoretical lower bound of active set factorizations does not converge to zero over time (presumably because solutions are not optimal given in our termination criteria) implying that our guessing strategy is not accurate in this scheme. Overall, this results in a slower numerical scheme with almost 60% of the problems solved in 10[ms] or less. 100% of the problems had a feasible solution after five Newton iterations.

Walking in a crowd

We test the robot walking at a reference speed of $0.5[\text{ms}^{-1}]$ along the x axis in a crowd of 10 persons during 20[s]. We warm-start the **MPC** problems with the guessing strategies (8.2) and (8.3).

Figure 8.6 shows the results obtained with the **Safe SQP** scheme. The walking motion of the robot starts to converge to a cycle at around problem 50, as before. However, as soon as people approach the robot and evasion actions are needed, the motion stops being periodic. We solve 60% of the problems in 65[ms] or less. Our guessing strategy is, therefore, rather unsuitable for this problem. However, we solve 92% of the problems with a single Newton iteration and 100% of the problems with two Newton iterations. The difference, 8% of the problems, comes from the relaxation of the infeasible safe linear constraints (6.43).

Figure 8.7 shows the results obtained with the **SQP** scheme. Similarly, it seems clear that our guessing strategy is not optimal for aperiodic motions as we require up to 85[ms] to solve 60% of the problems. Regarding feasibility: we solve around 92% of the problems with five iterations or less and 98% of the problems within 10 iterations; the remaining 2% of the problems are infeasible.

The “shape” of the histograms of computation time and active set factorizations for both the **Safe SQP** and the **SQP** schemes are similar for this scenario. Being able to obtain a feasible solution in a single Newton iteration is an advantage only if not many active set factorizations are needed, *i.e.* if the active set guess is close to the optimal.

¹Our implementation is written in **Octave** and is not optimized for performance. We estimate that the upper bound can be lowered significantly.

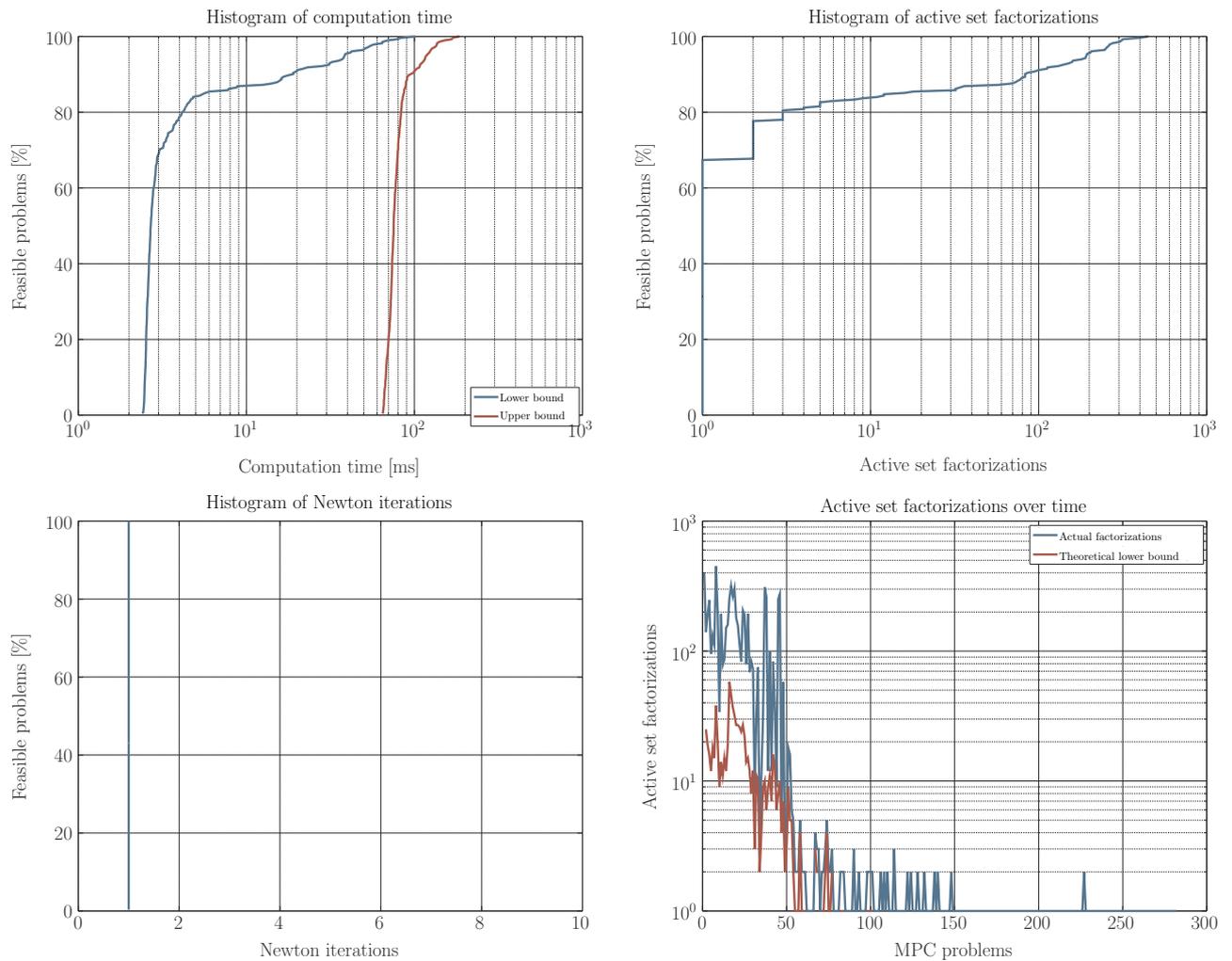


Figure 8.4: The robot walking with no crowd under a [Safe SQP](#) scheme.

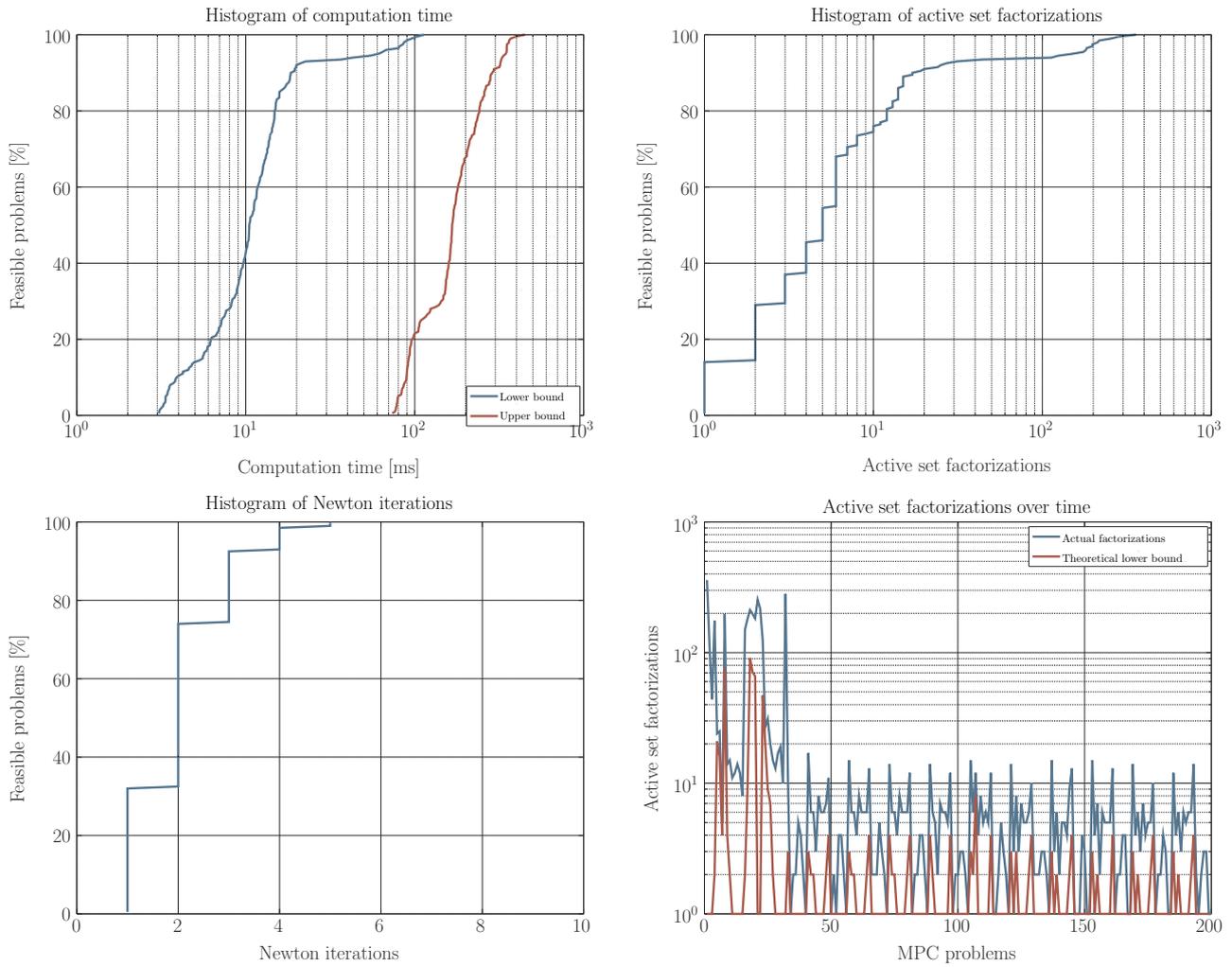


Figure 8.5: The robot walking with no crowd under a SQP scheme.

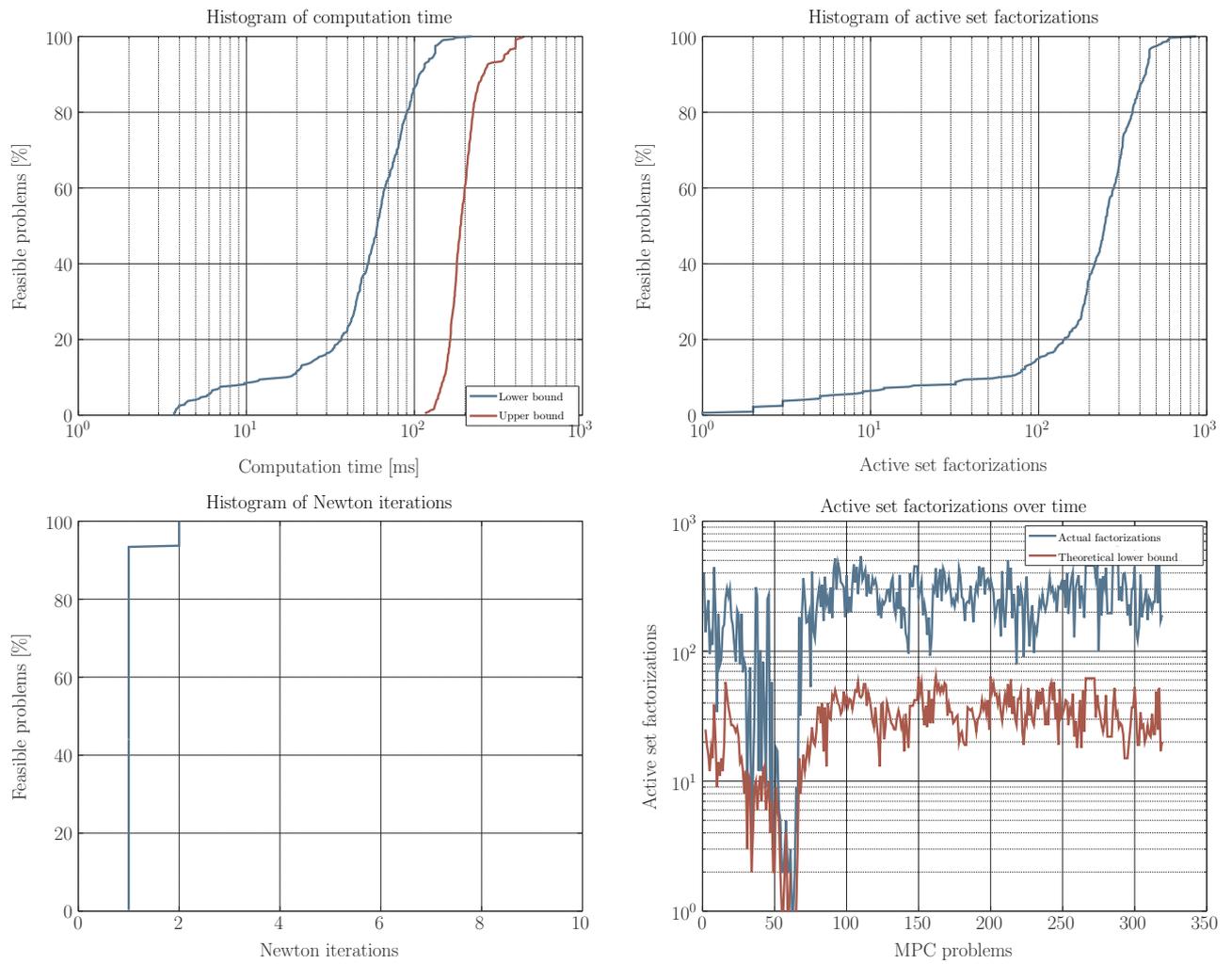


Figure 8.6: The robot walking in a crowd under a [Safe SQP](#) scheme.

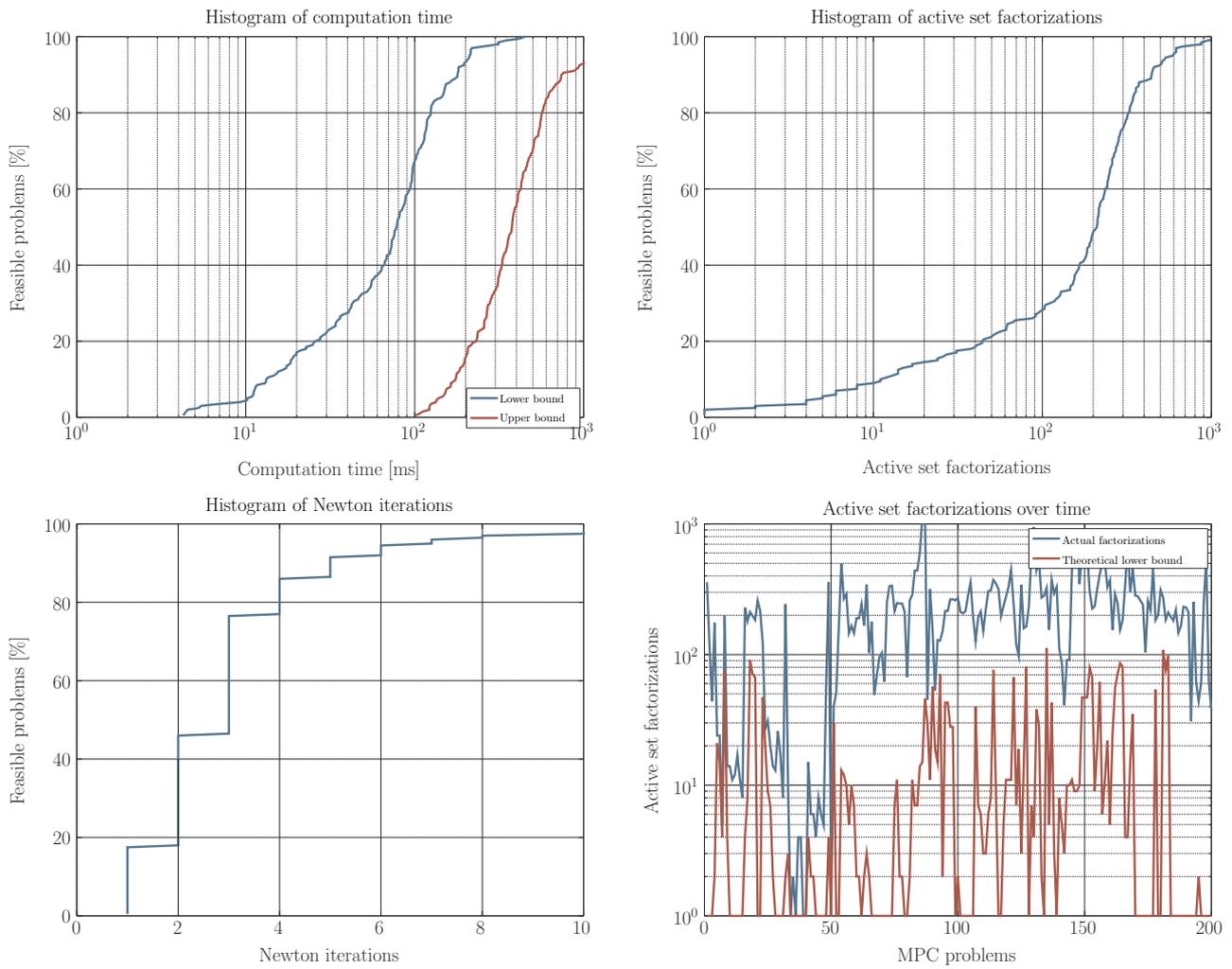


Figure 8.7: The robot walking in a crowd under a SQP scheme.

8.4 Conclusion

We simulated an HRP-2 robot walking in different crowds.

We tested the collision mitigation strategies introduced in Chapter 4 and compared them to standard passive safety controllers. Given maximal relative walking speed of the crowd, we determined the minimal radius of the FoV needed in order to have a desired anticipation time to collisions. We showed that, when there is uncertainty in the position and velocity of the crowd, collision mitigation strategies reduce the number of collisions and, moreover, they are less severe when they occur.

We made the robot adapt the rotation and duration of its steps in order to avoid collisions with the crowd using safe linear constraints. We also tested the performance of the [Safe SQP](#) method, proposed in Chapter 6, compared to a standard [SQP](#) when trying to solve the MPC introduced in Chapter 7. We showed that the proposed method could arrive to a feasible solution in just one Newton iteration, as desired. We realized that correctly warm-starting the NLP problem is crucial to have short computation times and that good active set and solution guesses are not trivial to propose.

Chapter 9

Conclusion

9.1 Summary

This thesis aimed at improving the safety of the navigation of a biped robot in a crowd and the safety of the computations of such motions. We started our dissertation by introducing the dynamical model of motion of the CoM in Chapter 2. Following the requirements of the motion of the robot and the assumption we made on the walking scenario we concluded, in Chapter 3, that the problem of avoiding failures and collisions could be addressed with optimal control techniques and, in particular, with an MPC scheme. In Chapter 4 we explained in detail the setting of the crowd navigation problem and defined the properties of the proposed *collision mitigation* strategy. We formulated controllers, written as lexicographic problems, that synthesize motions that satisfy such properties. At this point we stepped back in order to reflect on the implications of having strict priorities among the objectives of a robot and how this can constitute a plausible approach to dealing with the complexities of the social interactions and expectations between robots and people. We presented some numerical aspects of lexicographic programming in Chapter 5. We then addressed the problem of computing feasible Newton iterates in Chapter 6. We proposed to create safe linear constraints with respect to the nonlinear constraints, solving the nonlinear problem with a series of convex problems in Algorithm 3 (trust-region *Safe SQP*). We presented in abundant detail the MPC scheme we use to generate walking motions of the robot in a crowd in Chapter 7. We made emphasis in how we dealt with the nonlinearities introduced in the constraints when we allowed the rotations and the duration of the steps to vary. We evaluated the controllers we proposed in a series of simulations described in Chapter 8. Given appropriate perception capabilities of the robot, we showed that the collision mitigation controllers reduced the number and severity of collisions in comparison with passive safety specially when there is uncertainty in the motion of the crowd. We showed that our approach to nonlinear constraints allows the robot to produce a higher variety of motions with reduced computation costs under some scenarios. In more complex scenarios, such as a moving crowd, we acknowledge how sensible the computation costs can be with respect to the warm-start strategy employed.

9.2 Perspectives

The results presented in this work can lead to a number of possible future research directions.

9.2.1 Safe navigation strategies

Major research directions

- Develop a general safe navigation model. A *backup plan* takes the robot to a safe state in case something goes wrong. Safety is guaranteed if such safe state can be reached within some arbitrary time at any moment. In the specific case of walking, safety and progress “point” in the same direction: we can produce a single trajectory that is safe and that allows us to make progress towards the objectives of the robot. We argue that this is not true in the general case and we have to calculate two different trajectories: one for progress and another for safety. The robot would try to pursue its objectives as long as that does not impede it from reaching (if necessary) the safe state on time. The length of the two trajectories are independent. We vary the level of safety of the robot by changing the length of the safe trajectory.
- Develop a strategy that minimizes damages that occur after an inevitable collision. Example case: the robot is working on the airframe of a new aircraft and it notices an inattentive coworker walking directly towards its position. The collision is by now unavoidable and the fall will damage the airframe. However, if it moves a couple of steps backwards the collision will still take place but it will fall far from the airframe.

Minor research directions

- Prove that the lexicographic controller described in Hierarchy 3 produces motions that satisfy Definitions 4 and 5.
- Account for occlusions in the FoV. Until now it is not the case in Definition 2.

9.2.2 Lexicographic programming

Investigate the numerical methods to solve nonlinear lexicographic least-squares programming problems, first discussed by Wieber *et al.* [Wieber 2017].

9.2.3 Safe SQP

Prove the conjectures of Section 6.5 regarding the termination and convergence to first-order and second-order critical points of Algorithm 3.

9.2.4 Nonlinear MPC for biped walking

Major research directions

- Study the stability and strong recursive feasibility of the switched system (7.9).
- Integrate to the proposed MPC scheme the adaptation of the height of the CoM [Brasseur 2015] and the application of external forces [Serra 2016].

Minor research directions

- Guarantee the satisfaction of constraints between samples in the preview horizon.
- Define sets \mathcal{W}_P , \mathcal{W}_C and \mathcal{W}_S (7.16) that vary linearly with respect to the orientation of the foot θ .
- Define a more refined objective function for the orientation of the steps.

9.2.5 Simulations

Major research directions

Determine a way to guess the optimal active set when the robot is walking in a crowd while adapting the rotation and duration of its steps.

Minor research directions

- Address the uncertainty of the motion of people in a more standard way, *i.e.* where the robot can be at any position inside the uncertainty circle and not necessarily at the center.
- Compare [Safe SQP](#) with [FSQP](#) methods in the scenario of the robot walking in a crowd.

Appendix A

Bernstein polynomials

Given a quadratic polynomial $x(\tau)$ for $\tau \in [\underline{\tau}, \bar{\tau}]$ with coordinates $\alpha = (\alpha_0, \alpha_1, \alpha_2)$ in the monomial basis $\{1, \tau, \tau^2\}$, we first find the equivalent coordinates in the normalized monomial basis $\{1, \tau', \tau'^2\}$, for $\tau' \in [0, 1]$, by defining $\tau = \underline{\tau} + (\bar{\tau} - \underline{\tau})\tau'$ and solving for all τ'

$$\alpha_0 + \alpha_1\tau + \alpha_2\tau^2 = \alpha'_0 + \alpha'_1\tau' + \alpha'_2\tau'^2,$$

to obtain

$$\begin{bmatrix} \alpha'_0 \\ \alpha'_1 \\ \alpha'_2 \end{bmatrix} = \begin{bmatrix} 1 & \underline{\tau} & \underline{\tau}^2 \\ 0 & \bar{\tau} - \underline{\tau} & 2(\bar{\tau} - \underline{\tau})\underline{\tau} \\ 0 & 0 & (\bar{\tau} - \underline{\tau})^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

Then we find its coordinates $\beta = (\beta_0, \beta_1, \beta_2)$ in the Bernstein basis $\{(1 - \tau')^2, 2\tau'(1 - \tau'), \tau'^2\}$ by solving, once again, for all τ'

$$\alpha'_0 + \alpha'_1\tau' + \alpha'_2\tau'^2 = \beta_0(1 - \tau')^2 + \beta_1 2\tau'(1 - \tau') + \beta_2\tau'^2,$$

to obtain:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1/2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha'_0 \\ \alpha'_1 \\ \alpha'_2 \end{bmatrix}.$$

Appendix B

Concepts of convex analysis

B.1 Sets and polyhedra

A set $\mathcal{S} \subset \mathbb{R}^n$ is *convex* if the line that connects any two points in \mathcal{S} is also in \mathcal{S} , *i.e.* for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}$ and for any $\gamma \in [0, 1]$ we have that

$$\gamma \mathbf{x}_1 + (1 - \gamma) \mathbf{x}_2 \in \mathcal{S}. \quad (\text{B.1})$$

Let the subsets $\mathcal{M}_1 \subset \mathcal{M}$ and $\mathcal{M}_2 \subset \mathcal{M}$ be non-empty. The *Hausdorff distance* between them in the metric space (\mathcal{M}, d) , denoted $d_H(\mathcal{M}_1, \mathcal{M}_2)$, is defined as

$$d_H(\mathcal{M}_1, \mathcal{M}_2) = \max\left\{ \sup_{\mathbf{x} \in \mathcal{M}_1} \inf_{\mathbf{y} \in \mathcal{M}_2} d(\mathbf{x}, \mathbf{y}), \sup_{\mathbf{y} \in \mathcal{M}_2} \inf_{\mathbf{x} \in \mathcal{M}_1} d(\mathbf{x}, \mathbf{y}) \right\}. \quad (\text{B.2})$$

The distance $d_H(\mathcal{M}_1, \mathcal{M}_2)$ is equal to zero if and only if \mathcal{M}_1 and \mathcal{M}_2 have the same closure.

A *convex hull* of a finite set of points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, denoted $\text{conv}\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, is the set of all convex combinations of those points

$$\text{conv}\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \triangleq \left\{ \gamma_1 \mathbf{x}_1 + \dots + \gamma_n \mathbf{x}_n : \sum_i \gamma_i = 1, \forall i \gamma_i \geq 0 \right\}. \quad (\text{B.3})$$

A *conic hull* of a finite set of points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, denoted $\text{cone}\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, is the set of all conic combinations of those points

$$\text{cone}\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \triangleq \left\{ \gamma_1 \mathbf{x}_1 + \dots + \gamma_n \mathbf{x}_n : \forall i \gamma_i \geq 0 \right\}. \quad (\text{B.4})$$

A *polyhedron* $\mathcal{P} \subset \mathbb{R}^n$ is the solution of a finite number of linear equalities and inequalities:

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^n : \underline{\mathbf{b}} \leq \mathbf{A}\mathbf{x} \leq \bar{\mathbf{b}} \right\}, \quad (\text{B.5})$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}, \underline{\mathbf{b}} = \begin{bmatrix} \underline{b}_1 \\ \vdots \\ \underline{b}_m \end{bmatrix}, \bar{\mathbf{b}} = \begin{bmatrix} \bar{b}_1 \\ \vdots \\ \bar{b}_m \end{bmatrix}. \quad (\text{B.6})$$

Equalities can be considered in this formulation with $\underline{b}_i = \bar{b}_i$. All polyhedra are convex sets because they are the result of the intersection (an operation that preserves convexity) of hyperplanes and halfspaces (both of which are convex sets). A convex hull is a polyhedron and, conversely, any polyhedron can also be written as the convex hull of its vertices [Boyd 2004, Chapter 2]. A bounded polyhedron is a *polytope*.

B.2 Functions

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *convex* if the line that connects two points on the graph of f is above or on the graph, *i.e.* if for any two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for any $\gamma \in [0, 1]$ we have that

$$\gamma f(\mathbf{x}_1) + (1 - \gamma)f(\mathbf{x}_2) \geq f(\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2). \quad (\text{B.7})$$

Let \mathcal{K} be a *proper convex cone* in \mathbb{R}^m . A function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is \mathcal{K} -*convex* if for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for any $\gamma \in [0, 1]$ we have that

$$\gamma g(\mathbf{x}_1) + (1 - \gamma)g(\mathbf{x}_2) \in g(\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2) + \mathcal{K}. \quad (\text{B.8})$$

A set-valued function $\mathcal{F} : \mathcal{X} \rightsquigarrow \mathcal{Y}$ maps each $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ to a set $\mathcal{F}(\mathbf{x}) \subset \mathcal{Y} \subseteq \mathbb{R}^m$. \mathcal{F} is also said to be a map from \mathcal{X} to the power set of \mathcal{Y} , that is, $\mathcal{F} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$. This set-valued function \mathcal{F} is said to be *convex-valued* if, for any $\mathbf{x} \in \mathcal{X}$, the set $\mathcal{F}(\mathbf{x})$ is convex. \mathcal{F} is said to be \mathcal{K} -*convex* if, for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ and for any $\gamma \in [0, 1]$ we have that

$$\gamma\mathcal{F}(\mathbf{x}_1) + (1 - \gamma)\mathcal{F}(\mathbf{x}_2) \subset \mathcal{F}(\gamma\mathbf{x}_1 + (1 - \gamma)\mathbf{x}_2) + \mathcal{K}. \quad (\text{B.9})$$

Let $\mathcal{F}_1 : \mathcal{X} \rightsquigarrow \mathcal{Y}$ and $\mathcal{F}_2 : \mathcal{Y} \rightsquigarrow \mathcal{Z}$ be two set-valued functions. The composition map of \mathcal{F}_1 and \mathcal{F}_2 is the map $\mathcal{F}_2 \circ \mathcal{F}_1 : \mathcal{X} \rightsquigarrow \mathcal{Z}$ such that

$$(\mathcal{F}_2 \circ \mathcal{F}_1)(\mathbf{x}) = \bigcup_{\mathbf{y} \in \mathcal{F}_1(\mathbf{x})} \mathcal{F}_2(\mathbf{y}). \quad (\text{B.10})$$

for $\mathbf{x} \in \mathcal{X}$.

Appendix C

Warm start of MPC

C.1 MPC as a sequence of neighboring SQPs

We use some of the techniques discussed in [Diehl 2009].

C.1.1 Linearization of the collision avoidance constraint

In order to avoid clutter in the notation, we do not include the dependency on τ of the vectors \mathbf{c} and \mathbf{m} and the scalar d in the following. Let D_i be the minimal distance the CoM has to maintain for people in the crowd at the sample time i in the preview horizon, then it follows that constraint (7.36) can be written as

$$\frac{(\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)})^T}{\|\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)}\|} (\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)}) \geq D_i, \quad (\text{C.1})$$

which reveals that the CoM is restricted to the moving halfspace $\mathcal{H}(\mathbf{m}_{(i|k)}, D_i)$, that is

$$\mathbf{c}_{(i|k)} \in \mathcal{H}(\mathbf{m}_{(i|k)}, D_i), \quad (\text{C.2})$$

where

$$\mathcal{H}(\mathbf{o}, R) \triangleq \{\mathbf{r} : \hat{\mathbf{n}}^T(\mathbf{r} - \mathbf{o}) \geq R, \quad \hat{\mathbf{n}} = \text{normal}(\mathbf{r} - \mathbf{o})\}. \quad (\text{C.3})$$

The constraint (C.2) is clearly nonlinear because of the normal vector

$$\hat{\mathbf{n}} = \text{normal}(\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)}). \quad (\text{C.4})$$

Let us make explicit this dependency with a change in the notation of \mathcal{H}

$$\mathcal{H}(\hat{\mathbf{n}}, \mathbf{o}, R) \triangleq \{\mathbf{r} : \hat{\mathbf{n}}^T(\mathbf{r} - \mathbf{o}) \geq R\}. \quad (\text{C.5})$$

However, if τ is small enough then we can assume that $\hat{\mathbf{n}}_{(i|k)} = \text{normal}(\mathbf{c}_{(i|k)} - \mathbf{m}_{(i|k)})$ from the k^{th} computation of the MPC does not differ too much from $\hat{\mathbf{n}}_{(i|k-1)} = \text{normal}(\mathbf{c}_{(i|k-1)} - \mathbf{m}_{(i|k-1)})$ obtained in the previous computation $k-1$

$$\lim_{\tau \rightarrow 0} \|\hat{\mathbf{n}}_{(i|k)} - \hat{\mathbf{n}}_{(i|k-1)}\| = 0, \quad (\text{C.6})$$

and we can use it to approximate the original constraint and obtain a completely linear formulation

$$\mathbf{c}_{(i|k)} \in \mathcal{H}(\hat{\mathbf{n}}_{(i|k-1)}, \mathbf{m}_{(i|k)}, D_i). \quad (\text{C.7})$$

Bibliography

- [Aboudonia 2017] A. Aboudonia, N. Scianca, D. D. Simone, L. Lanari and G. Oriolo. *Humanoid gait generation for walk-to locomotion using single-stage MPC*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids). IEEE, nov 2017. (Cited on page 50.)
- [Aftab 2012] Z. Aftab, T. Robert and P.-B. Wieber. *Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme*. In 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012). IEEE, nov 2012. (Cited on page 48.)
- [Ahn 2018] J. Ahn, O. Campbell, D. Kim and L. Sentis. *Fast Kinodynamic Bipedal Locomotion Planning with Moving Obstacles*. 2018. (Cited on page 13.)
- [Al Homsy 2016] S. Al Homsy, A. Sherikov, D. Dimitrov and P.-B. Wieber. *A hierarchical approach to minimum-time control of industrial robots*. In IEEE International Conference on Robotics and Automation (ICRA), pages 2368–2374, 2016. (Cited on page 26.)
- [Aoi 2004] S. Aoi, K. Tsuchiya and K. Tsujita. *Turning control of a biped locomotion robot using nonlinear oscillators*. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA 2004. IEEE, 2004. (Cited on page 50.)
- [Arechavaleta 2008] G. Arechavaleta, J.-P. Laumond, H. Hicheur and A. Berthoz. *An Optimality Principle Governing Human Walking*. IEEE Transactions on Robotics, vol. 24, no. 1, pages 5–14, feb 2008. (Cited on page 48.)
- [Asama 2004] H. Asama and T. Fraichard. *Inevitable collision states - a step towards safer robots*. Advanced Robotics, vol. 18, pages 1001–1024, 2004. (Cited on page 14.)
- [Aubin 1991] J.-P. Aubin. Viability theory. Birkhäuser, 1991. (Cited on page 14.)
- [Ayaz 2009] Y. Ayaz, A. Konno, K. Munawar, T. Tsujita and M. Uchiyama. *Planning footsteps in obstacle cluttered environments*. 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Jul 2009. (Cited on page 13.)
- [Behnke 2006] S. Behnke. *Online trajectory generation for omnidirectional biped walking*. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006. (Cited on page 50.)
- [Bohórquez 2016] N. Bohórquez, A. Sherikov, D. Dimitrov and P.-B. Wieber. *Safe navigation strategies for a biped robot walking in a crowd*. In IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 379–386, Cancun, Mexico, November 2016. IEEE. (Cited on page 20.)

- [Bohórquez 2017] N. Bohórquez and P.-B. Wieber. *Adaptive step duration in biped walking: a robust approach to nonlinear constraints*. In IEEE RAS International Conference on Humanoid Robots 2017, Birmingham, United Kingdom, November 2017. (Cited on page 47.)
- [Bohórquez 2018] N. Bohórquez and P.-B. Wieber. *Adaptive step rotation in biped walking*. In IEEE/RSJ International Conference on Intelligent Robots and Systems 2018, Madrid, Spain, October 2018. (Cited on page 47.)
- [Bouraine 2012] S. Bouraine, T. Fraichard and H. Salhi. *Provably Safe Navigation for Mobile Robots with Limited Field-of-Views in Dynamic Environments*. *Autonomous Robots*, vol. 32, no. 3, pages 267–283, April 2012. (Cited on page 15.)
- [Boyd 2004] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. (Cited on pages 30, 40, 56, and 80.)
- [Brasseur 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 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids). Institute of Electrical & Electronics Engineers (IEEE), nov 2015. (Cited on pages 9 and 77.)
- [Bronshtein 1975] E. M. Bronshtein and L. D. Ivanov. *Approximation of convex sets by polyhedrons*. *Sib. Mat. Zh.*, no. 16, page 1110–1112, 1975. (Cited on page 42.)
- [Brossette 2017] S. Brossette and P.-B. Wieber. *Collision avoidance based on separating planes for feet trajectory generation*. In *Humanoids 2017 - IEEE RAS International Conference on Humanoid Robots*, pages 509–514, Birmingham, United Kingdom, November 2017. IEEE. (Cited on page 56.)
- [Cannon 2018] M. Cannon. *Lecture Notes: C21 Model Predictive Control*. Available: https://markcannon.github.io/assets/downloads/teaching/C21_Model_Predictive_Control/mp July 2018. (Cited on page 17.)
- [Chestnutt 2003] J. Chestnutt, J. Kuffner, K. Nishiwaki and S. Kagami. *Planning Biped Navigation Strategies in Complex Environments*. In *IEEE Int. Conf. Humanoid Robots*, 2003. (Cited on page 13.)
- [Chestnutt 2004] J. Chestnutt and J. Kuffner. *A tiered planning strategy for biped navigation*. 4th IEEE/RAS International Conference on Humanoid Robots, 2004., 2004. (Cited on page 13.)
- [Chestnutt 2005] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins and T. Kanade. *Footstep Planning for the Honda ASIMO Humanoid*. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005. (Cited on pages 13 and 50.)
- [Chestnutt 2007a] J. Chestnutt, P. Michel, J. Kuffner and T. Kanade. *Locomotion among dynamic obstacles for the honda ASIMO*. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, oct 2007. (Cited on pages 13 and 50.)
- [Chestnutt 2007b] J. Chestnutt, K. Nishiwaki, J. Kuffner and S. Kagami. *An adaptive action model for legged navigation planning*. 2007 7th IEEE-RAS International Conference on Humanoid Robots, Nov 2007. (Cited on page 13.)

- [Chevallereau 2009] C. Chevallereau, J. Grizzle and C.-L. Shih. *Asymptotically Stable Walking of a Five-Link Underactuated 3-D Bipedal Robot*. IEEE Transactions on Robotics, vol. 25, no. 1, pages 37–50, feb 2009. (Cited on page 7.)
- [Ciocca 2017] M. Ciocca, P.-B. Wieber and T. Fraichard. *Strong Recursive Feasibility in Model Predictive Control of Biped Walking*. In HUMANOIDS 2017 - IEEE-RAS International Conference on Humanoid Robots, Birmingham, United Kingdom, November 2017. (Cited on pages 18 and 49.)
- [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, feb 2005. (Cited on page 7.)
- [Conn 2000] A. Conn, N. Gould and P. Toint. Trust region methods. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000. (Cited on page 44.)
- [Da Silva Filho 2017] J. G. Da Silva Filho and T. Fraichard. *Human Robot Motion: A Shared Effort Approach*. In European Conference on Mobile Robotics, Paris, France, September 2017. (Cited on page 21.)
- [Deits 2014] R. Deits and R. Tedrake. *Footstep planning on uneven terrain with mixed-integer convex optimization*. 2014 IEEE-RAS International Conference on Humanoid Robots, Nov 2014. (Cited on pages 13 and 50.)
- [Deits 2015] R. Deits and R. Tedrake. Computing large convex regions of obstacle-free space through semidefinite programming, pages 109–124. Springer International Publishing, Cham, 2015. (Cited on page 42.)
- [Diedam 2009] H. Diedam. Fast nmpc algorithms for humanoid robot walking. Master’s thesis, University of Heidelberg, 2009. (Cited on page 48.)
- [Diehl 2009] M. Diehl, H. J. Ferreau and N. Haverbeke. *Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation*. In Nonlinear Model Predictive Control, pages 391–417. Springer Science, 2009. (Cited on page 82.)
- [Diehl 2017] M. Diehl and S. Gros. *Numerical Optimal Control (preliminary and incomplete draft)*, May 2017. (Cited on page 37.)
- [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 31.)
- [Dimitrov 2015] D. Dimitrov, A. Sherikov and P.-B. Wieber. Efficient resolution of potentially conflicting linear constraints in robotics. Submitted to IEEE TRO (05/August/2015), August 2015. (Cited on page 32.)
- [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 26 and 32.)
- [Faiz 1999] T. N. Faiz. *Real-time and optimal trajectory generation for nonlinear systems*. PhD thesis, University of Delaware, 1999. (Cited on page 41.)

- [Fernbach 2017] P. Fernbach, S. Tonneau, A. Del Prete and M. Taïx. *A Kinodynamic steering-method for legged multi-contact locomotion*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, page 7p., Vancouver, Canada, September 2017. (Cited on page 13.)
- [Fernbach 2018] P. Fernbach, S. Tonneau and M. Taïx. CROC: Convex Resolution Of Centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem. working paper or preprint, May 2018. (Cited on page 54.)
- [Fiorini 1998] P. Fiorini and Z. Shillert. *Motion Planning in Dynamic Environments using Velocity Obstacles*. International Journal of Robotics Research, vol. 17, pages 760–772, 1998. (Cited on page 14.)
- [Fox 1997] D. Fox, W. Burgard and S. Thrun. *The dynamic window approach to collision avoidance*. IEEE Robotics & Automation Magazine, vol. 4, pages 23–33, 1997. (Cited on page 13.)
- [Fraichard 2007] T. Fraichard. *A Short Paper about Motion Safety*. Proceedings 2007 IEEE International Conference on Robotics and Automation, Apr 2007. (Cited on page 22.)
- [Fraichard 2012] T. Fraichard and T. M. Howard. Handbook of intelligent vehicles, chapitre Iterative Motion Planning and Safety Issue, pages 1433–1458. Springer London, London, 2012. (Cited on page 13.)
- [Fraichard 2014] T. Fraichard, R. Paulin and P. Reignier. *Human-Robot Motion: An Attention-Based Navigation Approach*. In IEEE Int. Symp. on Robot and Human Interactive Communication (ROMAN), Edinburgh, United Kingdom, August 2014. (Cited on page 28.)
- [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 9.)
- [Griffin 2017] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa and J. Pratt. *Walking stabilization using step timing and location adjustment on the humanoid robot, Atlas*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, sep 2017. (Cited on page 48.)
- [Grizzle 2014] J. W. Grizzle, C. Chevallereau, R. W. Sinnet and A. D. Ames. *Models, feedback control, and open problems of 3D bipedal robotic walking*. Automatica, vol. 50, no. 8, pages 1955–1988, aug 2014. (Cited on page 7.)
- [Harada 2009] K. Harada, M. Morisawa, S. Nakaoka, K. Kaneko and S. Kajita. *Kinodynamic Planning for Humanoid Robots Walking on Uneven Terrain*. JRM, vol. 21, pages 311–316, 2009. (Cited on page 13.)
- [Hauser 1994] J. Hauser and C. C. Chung. *Converse Lyapunov functions for exponentially stable periodic orbits*. Systems & Control Letters, vol. 23, no. 1, pages 27–34, jul 1994. (Cited on page 15.)
- [Herdt 2010a] A. Herdt, N. Perrin and P.-B. Wieber. *Walking without thinking about it*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, oct 2010. (Cited on pages 48 and 50.)

- [Herdt 2010b] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur and M. Diehl. *Online Walking Motion Generation with Automatic Foot Step Placement*. *Advanced Robotics -Utrecht-*, vol. 24, no. 5-6, pages 719–737, 2010. (Cited on page 48.)
- [Hof 2008] A. L. Hof. *The ‘extrapolated center of mass’ concept suggests a simple control of balance in walking*. *Human Movement Science*, vol. 27, no. 1, pages 112–125, feb 2008. (Cited on page 57.)
- [Homsy 2016] S. A. Homsy. *Online generation of time- optimal trajectories for industrial robots in dynamic environments*. Theses, Université Grenoble Alpes, March 2016. (Cited on page 24.)
- [Hornung 2012] A. Hornung and M. Bennewitz. *Adaptive level-of-detail planning for efficient humanoid navigation*. 2012 IEEE International Conference on Robotics and Automation, May 2012. (Cited on page 13.)
- [Horst 1984] R. Horst. *On the convexification of nonlinear programming problems: An applications-oriented survey*. *European Journal of Operational Research*, vol. 15, no. 3, pages 382–392, mar 1984. (Cited on page 41.)
- [Hu 2018] W. Hu, I. Chatzinikolaïdis, K. Yuan and Z. Li. *Comparison Study of Nonlinear Optimization of Step Durations and Foot Placement for Dynamic Walking*. *CoRR*, vol. abs/1805.02155, 2018. (Cited on page 48.)
- [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, oct 2004. (Cited on page 5.)
- [Ibanez 2014] A. Ibanez, P. Bidaud and V. Padois. *Emergence of humanoid walking behaviors from mixed-integer model predictive control*. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, sep 2014. (Cited on page 48.)
- [Isermann 1982] H. Isermann. *Linear lexicographic optimization*. *Operations-Research-Spektrum*, vol. 4, no. 4, pages 223–228, Dec 1982. (Cited on page 24.)
- [ISO 2011] ISO. *Robots and robotic devices – safety requirements for industrial robots – part 1: Robots*. ISO 10218-1:2011. ISO, Geneva, Switzerland, 2011. (Cited on page 15.)
- [ISO 2014] ISO. *Robots and robotic devices – safety requirements for personal care robots*. ISO 13482:2014. ISO, Geneva, Switzerland, 2014. (Cited on page 15.)
- [Johnson 1985] K. L. Johnson. *Contact mechanics*. Cambridge University Press, 1985. (Cited on page 25.)
- [Jones 2010] C. N. Jones and M. Morari. *Polytopic Approximation of Explicit Model Predictive Controllers*. *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pages 2542–2553, nov 2010. (Cited on page 42.)
- [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 page 11.)

- [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 page 60.)
- [Kanoun 2011] O. Kanoun, F. Lamiroux 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 31 and 32.)
- [Kerrigan 2001] E. C. Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2001. (Cited on page 18.)
- [Khadiv 2016] M. Khadiv, A. Herzog, S. Moosavian and L. Righetti. *Step Timing Adjustment: A Step Toward Generating Robust Gaits*. In Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots, 2016. (Cited on pages 41 and 48.)
- [Khatib 1986] O. Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*. The International Journal of Robotics Research, vol. 5, no. 1, pages 90–98, 1986. (Cited on page 13.)
- [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 page 57.)
- [Kryczka 2015] P. Kryczka, P. Kormushev, N. G. Tsagarakis and D. G. Caldwell. *Online re-generation of bipedal walking gait pattern optimizing footstep placement and timing*. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, sep 2015. (Cited on page 48.)
- [Kuindersma 2015] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion and R. Tedrake. *Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot*. Autonomous Robots, vol. 40, no. 3, pages 429–455, jul 2015. (Cited on page 13.)
- [Large 2005] F. Large, C. Laugier and Z. Shiller. *Navigation Among Moving Obstacles Using the NLVO: Principles and Applications to Intelligent Vehicles*. Autonomous Robots, vol. 19, no. 2, pages 159–171, 2005. (Cited on page 14.)
- [Latombe 1991] J.-C. Latombe. Robot motion planning. Kluwer, Boston, MA, 1991. (Cited on page 13.)
- [LaValle 2006] S. M. LaValle. Planning algorithms. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>. (Cited on pages 13 and 22.)
- [Lawrence 1996] C. T. Lawrence and A. L. Tits. *Nonlinear equality constraints in feasible sequential quadratic programming**. Optimization Methods and Software, vol. 6, no. 4, pages 265–282, jan 1996. (Cited on page 40.)
- [Lawrence 2001] C. T. Lawrence and A. L. Tits. *A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm*. SIAM Journal on Optimization, vol. 11, no. 4, pages 1092–1118, jan 2001. (Cited on page 40.)

- [Lebbah 2005] Y. Lebbah, C. Michel, M. Rueher, D. Daney and J.-P. Merlet. *Efficient and Safe Global Constraints for Handling Numerical Constraint Systems*. SIAM Journal on Numerical Analysis, vol. 42, no. 5, pages 2076–2097, jan 2005. (Cited on page 41.)
- [Lebbah 2007] Y. Lebbah, C. Michel and M. Rueher. *An efficient and safe framework for solving optimization problems*. Journal of Computational and Applied Mathematics, vol. 199, pages 372–377,, February 2007. (Cited on page 41.)
- [Lengagne 2007] S. Lengagne, N. Ramdani and P. Fraisse. *Guaranteed Computation of Constraints for Safe path Planning*. In Humanoids, Pittsburgh, PA, United States, November 2007. (Cited on page 48.)
- [Lengagne 2009] S. Lengagne, P. Fraisse and N. Ramdani. *Planning and fast re-planning of safe motions for humanoid robots: Application to a kicking motion*. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, oct 2009. (Cited on page 48.)
- [Liu 2012] H. Liu, Q. Sun and T. Zhang. *Hierarchical RRT for humanoid robot footstep planning with multiple constraints in complex environments*. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2012. (Cited on page 13.)
- [Liu 2017] S. B. Liu, H. Roehm, C. Heinzemann, I. Lutkebohle, J. Oehlerking and M. Althoff. *Provably safe motion of mobile robots in human environments*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, sep 2017. (Cited on page 22.)
- [Loock 2015] W. V. Loock, G. Pipeleers and J. Swevers. *B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction*. Mechanical Sciences, vol. 6, no. 2, pages 163–171, sep 2015. (Cited on page 48.)
- [Lorentz 1953] G. Lorentz. Bernstein polynomials. Mathematical expositions. University of Toronto Press, 1953. (Cited on page 53.)
- [Louembet 2007] C. Louembet. *Generation de trajectoires optimales pour systemes differentiellement plats application aux manoeuvres d’attitude sur orbite*. PhD thesis, Université Bordeaux 1, 2007. (Cited on page 41.)
- [Macek 2009] K. Macek, D. A. Vasquez Govea, T. Fraichard and R. Y. Siegwart. *Towards Safe Vehicle Navigation in Dynamic Urban Scenarios*. Automatika, November 2009. (Cited on pages 13 and 15.)
- [McGeer 1990a] T. McGeer. *Passive walking with knees*. In Proceedings., IEEE International Conference on Robotics and Automation. IEEE Comput. Soc. Press, 1990. (Cited on page 7.)
- [McGeer 1990b] T. McGeer. *Passive Dynamic Walking*. The International Journal of Robotics Research, vol. 9, no. 2, pages 62–82, apr 1990. (Cited on page 7.)
- [Mercy 2016] T. Mercy, W. V. Loock and G. Pipeleers. *Real-time motion planning in the presence of moving obstacles*. In 2016 European Control Conference (ECC). IEEE, jun 2016. (Cited on page 56.)
- [Miettinen 1998] K. Miettinen. Nonlinear multiobjective optimization. Springer US, 1998. (Cited on page 30.)

- [Minh 2013] V. T. Minh. *Stability for switched dynamic hybrid systems*. Mathematical and Computer Modelling, vol. 57, no. 1-2, pages 78–83, jan 2013. (Cited on page 49.)
- [Mombaur 2009] K. Mombaur, A. Truong and J.-P. Laumond. *From human to humanoid locomotion—an inverse optimal control approach*. Autonomous Robots, vol. 28, no. 3, pages 369–383, dec 2009. (Cited on page 48.)
- [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 page 8.)
- [Nadim Faiz 2001] S. A. Nadim Faiz and R. Murray. *Differentially Flat Systems with Inequality Constraints: An Approach to Real-Time Feasible Trajectory Generation*. Journal of Guidance, Control, and Dynamics, vol. 24, no. 2, pages 219–227, 2001. (Cited on page 41.)
- [Naveau 2017] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Soueres. *A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control*. IEEE Robotics and Automation Letters, vol. 2, no. 1, pages 10–17, jan 2017. (Cited on page 50.)
- [Nocedal 2006] J. Nocedal and S. J. Wright. Numerical optimization. Springer, New York, NY, USA, second édition, 2006. (Cited on pages 35, 36, 37, 38, 39, and 44.)
- [Panier 1993] E. R. Panier and A. L. Tits. *On combining feasibility, descent and superlinear convergence in inequality constrained optimization*. Mathematical Programming, vol. 59, no. 1, pages 261–276, Mar 1993. (Cited on page 40.)
- [Pekarovskiy 2018] A. Pekarovskiy, T. Nierhoff, S. Hirche and M. Buss. *Dynamically Consistent Online Adaptation of Fast Motions for Robotic Manipulators*. IEEE Transactions on Robotics, vol. 34, no. 1, pages 166–182, feb 2018. (Cited on page 54.)
- [Perrin 2012a] N. Perrin, O. Stasse, L. Baudouin, F. Lamiroux and E. Yoshida. *Fast humanoid robot collision-free footstep planning using swept volume approximations*, Robotics. IEEE Transactions on, page 439, 2012. (Cited on pages 13 and 50.)
- [Perrin 2012b] N. Perrin, O. Stasse, F. Lamiroux, Y. J. Kim and D. Manocha. *Real-time footstep planning for humanoid robots among 3D obstacles using a hybrid bounding box*. 2012 IEEE International Conference on Robotics and Automation, May 2012. (Cited on page 13.)
- [Pham 2014] Q. Pham, S. Caron, P. Lertkultanon and Y. Nakamura. *Planning Truly Dynamic Motions: Path-Velocity Decomposition Revisited*. CoRR, vol. abs/1411.4045, 2014. (Cited on page 13.)
- [Pratt 2006] J. Pratt, J. Carff, S. Drakunov and A. Goswami. *Capture Point: A Step toward Humanoid Push Recovery*. In Humanoid Robots, 2006 6th IEEE-RAS International Conference on, pages 200–207, dec. 2006. (Cited on page 57.)
- [Quinlan 1993] S. Quinlan and O. Khatib. *Elastic Bands: Connecting Path Planning and Control*. In ICRA (2), pages 802–807. IEEE Computer Society Press, 1993. (Cited on page 13.)

- [Ralston 1958] H. J. Ralston. *Energy-speed relation and optimal speed during level walking*. Internationale Zeitschrift für angewandte Physiologie einschließlich Arbeitsphysiologie, vol. 17, no. 4, pages 277–283, 1958. (Cited on page 63.)
- [Rawlings 2016] J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Publishing, 2016. (Cited on page 15.)
- [Scianca 2016] N. Scianca, M. Cagnetti, D. D. Simone, L. Lanari and G. Oriolo. *Intrinsically stable MPC for humanoid gait generation*. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). IEEE, nov 2016. (Cited on page 15.)
- [Seder 2007] M. Seder and I. Petrovic. *Dynamic window based approach to mobile robot motion control in the presence of moving obstacles*. Proceedings 2007 IEEE International Conference on Robotics and Automation, Apr 2007. (Cited on page 14.)
- [Serra 2016] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov and P.-B. Wieber. *A Newton method with always feasible iterates for Nonlinear Model Predictive Control of walking in a multi-contact situation*. In IEEE-RAS 2016 - International Conference on Humanoid Robots (Humanoids), pages 932–937, Cancun, Mexico, November 2016. IEEE. (Cited on pages 9 and 77.)
- [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 page 9.)
- [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 page 26.)
- [Sherikov 2016] A. Sherikov. *Balance preservation and task prioritization in whole body motion control of humanoid robots*. PhD thesis, Communauté Université Grenoble Alpes, 2016. (Cited on pages ix, 5, 7, 16, and 17.)
- [Siciliano 1991] B. Siciliano and J.-J. Slotine. *A general framework for managing multiple tasks in highly redundant robotic systems*. In Fifth International Conference on Advanced Robotics Robots in Unstructured Environments. IEEE, 1991. (Cited on page 32.)
- [Simone 2017] D. D. Simone, N. Scianca, P. Ferrari, L. Lanari and G. Oriolo. *MPC-based humanoid pursuit-evasion in the presence of obstacles*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, sep 2017. (Cited on page 50.)
- [Sreenath 2010] K. Sreenath, H.-W. Park, I. Poulakakis and J. W. Grizzle. *A Compliant Hybrid Zero Dynamics Controller for Stable, Efficient and Fast Bipedal Walking on MABEL*. The International Journal of Robotics Research, vol. 30, no. 9, pages 1170–1193, sep 2010. (Cited on page 7.)
- [Sun 2015] Z. Sun, Y. Tian, H. Li and J. Wang. *A superlinear convergence feasible sequential quadratic programming algorithm for bipedal dynamic walking robot via discrete mechanics and optimal control*. Optimal Control Applications and Methods, vol. 37, no. 6, pages 1139–1161, dec 2015. (Cited on page 40.)

- [Tajima 2009] R. Tajima, D. Honda and K. Suga. *Fast running experiments involving a humanoid robot*. In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pages 1571–1576, May 2009. (Cited on page 8.)
- [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 8, 15, and 57.)
- [Tamiz 1995] M. Tamiz, D. F. Jones and E. El-Darzi. *A review of Goal Programming and its applications*. Annals of Operations Research, vol. 58, no. 1, pages 39–53, Jan 1995. (Cited on page 31.)
- [Van den Broeck 2011] L. Van den Broeck, M. Diehl and J. Swevers. *A model predictive control approach for time optimal point-to-point motion control*. Mechatronics, vol. 21, no. 7, pages 1203–1212, oct 2011. (Cited on page 23.)
- [Vasquez 2009] D. Vasquez, T. Fraichard and C. Laugier. *Growing Hidden Markov Models: An Incremental Tool for Learning and Predicting Human and Vehicle Motion*. The International Journal of Robotics Research, vol. 28, no. 11-12, pages 1486–1506, 2009. (Cited on page 28.)
- [Villa 2017] N. A. Villa and P.-B. Wieber. *Model predictive control of biped walking with bounded uncertainties*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids). IEEE, nov 2017. (Cited on page 17.)
- [Vukobratović 2004] M. Vukobratović and B. Borovac. *ZERO-MOMENT POINT — THIRTY FIVE YEARS OF ITS LIFE*. International Journal of Humanoid Robotics, vol. 01, no. 01, pages 157–173, mar 2004. (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 12 and 14.)
- [Wieber 2006] 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 page 8.)
- [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 page 14.)
- [Wieber 2015] P.-B. Wieber, R. Tedrake and S. Kuindersma. Springer handbook of robotics, chapitre Modeling and Control of Legged Robots. Springer London, London, 2015. (Cited on pages 5 and 8.)
- [Wieber 2017] P.-B. Wieber, A. Escande, D. Dimitrov and A. Sherikov. Geometric and numerical aspects of redundancy, pages 67–85. Springer International Publishing, Cham, 2017. (Cited on pages 26, 33, and 77.)
- [Wright 2004] S. J. Wright and M. J. Tenny. *A Feasible Trust-Region Sequential Quadratic Programming Algorithm*. SIAM Journal on Optimization, vol. 14, no. 4, pages 1074–1105, jan 2004. (Cited on page 40.)

- [Yagi 2000] M. Yagi and V. Lumelsky. *Synthesis of turning pattern trajectories for a biped robot in a scene with obstacles*. In Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113). IEEE, 2000. (Cited on page 50.)
- [Zaytsev 2015] 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 page 17.)
- [Zhao 2017] M. Zhao and S. Yu. *Push recovery by stepping with step timing adjustment*. In 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, dec 2017. (Cited on page 48.)
- [Zhu 2010] Z. Zhu, W. Zhang and Z. Geng. *A feasible SQP method for nonlinear programming*. Applied Mathematics and Computation, vol. 215, no. 11, pages 3956 – 3969, 2010. (Cited on page 40.)