



HAL
open science

Balance Preservation and Collision Mitigation for Biped Robots

Matteo Ciocca

► **To cite this version:**

Matteo Ciocca. Balance Preservation and Collision Mitigation for Biped Robots. Artificial Intelligence [cs.AI]. Université Grenoble Alpes [2020-..], 2020. English. NNT : 2020GRALM042 . tel-03065088

HAL Id: tel-03065088

<https://theses.hal.science/tel-03065088>

Submitted on 14 Dec 2020

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 L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Matteo Ciocca

Thèse dirigée par **Thierry Fraichard**
et co-encadrée par **Pierre-Brice Wieber**

préparée au sein du **Laboratoire d'Informatique de Grenoble et Inria Grenoble Rhône-Alpes**
dans l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Préservation de l'équilibre et atténuation des collisions pour les robots bipèdes

Balance Preservation and Collision Mitigation for Biped Robots

Thèse soutenue publiquement le **27 Mai 2020**,
devant le jury composé de :

Gérard Bailly

Directeur de Recherche CNRS, Président

Florent Lamiroux

Directeur de Recherche CNRS, Rapporteur

Vincent Padois

Maître de Conférences à Sorbonne Université, Rapporteur

Julien Pettré

Directeur de Recherche INRIA, Examineur

Thierry Fraichard

Chargé de Recherche INRIA, Directeur de thèse

Pierre-Brice Wieber

Chargé de Recherche INRIA, Co-encadrant de thèse



Abstract

This work focuses on two challenging tasks for humanoid robots: bipedal balance and collision avoidance in a dense crowd. We solve these tasks on a limited time horizon in which we can anticipate the consequences of robot actions.

We can guarantee that the robot is able to stop in a few steps and avoid falling. When the robot is not planning to stop but to continue walking, we show the guarantee to avoid falling is not lost but it depends on the length of the time horizon. It is impossible to know beforehand what people will do next, so we cannot guarantee that no collision will ever occur. Over a limited time horizon we can guarantee *Passive Safety*: the robot is able to stop before a collision occurs. This safety guarantee is combined with fall avoidance in a *Model Predictive Control* scheme. The capacity for the robot to react and avoid collisions is constrained once a step is planted on the ground, until the next step is initiated. With the control scheme outlined above the robot reacts not only at each step initiation but also in between. We show that reacting only once per step (thus saving computational power) does not degrade collision avoidance capability.

The time left for people to react and avoid collisions once the robot has stopped (to guarantee Passive Safety) might not be enough. We propose a new control scheme called *Collision Mitigation* that guarantees fall avoidance while aiming to leave as much time as possible for the people to react. As a result, the robot collides less often and later than when it guarantees Passive Safety. This scheme can be adapted to take different priorities into account. For example, when the members of a crowd are divided in robots and people, the robot should leave as much time as possible for the people to react and then, if possible, for the other robots. Or when the robot must reach a target location at the utmost important and people might obstruct the motion of the robot, the robot can jostle people if necessary to reach the location.

Résumé

Ce travail se concentre sur deux tâches difficiles pour les robots humanoïdes: l'équilibre bipède et la prévention des collisions dans une foule dense. Nous résolvons ces tâches sur un horizon temporel limité dans lequel nous pouvons anticiper les conséquences des actions des robots.

Nous pouvons garantir que le robot est capable de s'arrêter en quelques pas et d'éviter de tomber. Lorsque le robot ne prévoit pas de s'arrêter mais de continuer à marcher, nous montrons que la garantie d'éviter la chute n'est pas perdue mais qu'elle dépend de la longueur de l'horizon temporel. Il est impossible de savoir à l'avance ce que les gens feront ensuite, et nous ne pouvons donc pas garantir qu'aucune collision ne se produira jamais. Sur un horizon temporel limité, nous pouvons garantir la sécurité passive : le robot est capable de s'arrêter avant qu'une collision ne se produise. Cette garantie de sécurité est combinée avec l'évitement de chutes dans un schéma de contrôle prédictif. La capacité du robot à réagir et à éviter les collisions est limitée une fois qu'un pied est posé au sol, et ce jusqu'au prochain pas. Avec le schéma de commande décrit ci-dessus, le robot réagit non seulement à chaque pas, mais aussi entretemps. Nous montrons que le fait de ne réagir qu'une seule fois par pas (ce qui permet d'économiser de la puissance de calcul) ne dégrade pas la capacité d'éviter les collisions.

Le temps laissé aux personnes pour réagir et éviter les collisions une fois que le robot s'est arrêté (pour garantir la sécurité passive) pourrait ne pas être suffisant. Nous proposons un nouveau système de contrôle appelé Collision Mitigation qui garantit l'évitement de chutes tout en visant à laisser aux personnes le plus de temps possible pour réagir. Ainsi, le robot entre en collision moins souvent et plus tard que lorsqu'il garantit la sécurité passive. Ce système peut être adapté pour prendre en compte différentes priorités. Par exemple, lorsque les membres d'une foule sont divisés en robots et en personnes, le robot doit laisser autant de temps que possible aux personnes pour réagir et ensuite, si possible, aux autres robots. Ou encore, lorsque le robot doit atteindre une cible de la plus haute importance et que des personnes pourraient entraver le mouvement du robot, ce dernier peut, si nécessaire, bousculer les personnes pour atteindre le lieu en question.

Contents

1	Introduction	1
1.1	Context and motivation	1
1.2	Contributions and Outline	4
1.2.1	List of publications	5
1.2.2	Outline	5
1.3	Notation	6
2	Balance	7
2.1	Introduction	7
2.2	Biped robot	8
2.2.1	Mechanics	8
2.2.2	Contact forces and CoM relationship when walking on flat ground	9
2.2.3	Walking cycle	13
2.2.4	Whole-body restrictions on CoM dynamics	15
2.3	Balance Preservation	16
2.3.1	Capturability	17
2.4	Model Predictive Control of biped walking	18
2.4.1	MPC Approach	18
2.4.2	Walking motion generation with automatic footstep placement	19
2.4.3	Strong Recursive Feasibility	20
2.4.4	Capturability terminal constraint	20
2.5	Conclusion	21
3	Strong Recursive Feasibility in MPC of Biped Walking	23
3.1	Introduction	23
3.2	Problem formulation	23
3.3	Parameters	24
3.3.1	Robot	24
3.3.2	MPC	24
3.4	Numerical approach	25
3.5	SRF with the capturability terminal constraint	26
3.5.1	Numerical evidence from randomly selected vertices	29
3.5.2	Vertices verification	29
3.6	SRF without the capturability terminal constraint	30
3.7	Discussion	31
3.7.1	Providing a numerical proof	31
3.7.2	Ongoing research with the capturability terminal constraint	32
3.8	Conclusion	33

4	Motion Safety	34
4.1	Introduction	34
4.2	Crowd	35
4.2.1	Behavior	35
4.2.2	Pedestrian model	35
4.2.3	Model of the future	36
4.3	Physical attributes and interactions	38
4.3.1	Biped robot and people	38
4.3.2	Collision	39
4.4	Robot's perception	39
4.4.1	Field of View	39
4.4.2	Perception uncertainties	40
4.5	MPC-based safe biped navigation in a crowd	40
4.5.1	Collision avoidance with the crowd	40
4.5.2	Conservativeness in time	41
4.5.3	Optimization	43
4.5.4	Safety guarantee	45
4.6	Conclusion	46
5	Re-planning Effect on MPC-based Safe Biped Navigation in a Crowd	47
5.1	Introduction	47
5.2	Planning period and initial planning phase in MPC	47
5.3	Simulation parameters	48
5.3.1	Crowd	48
5.3.2	Robot	48
5.3.3	MPC	49
5.4	Results	52
5.4.1	Collisions and uncertainty	55
5.5	Discussion	55
5.5.1	Constraints on the swing phase	55
5.5.2	Varying the dynamicity of the crowd	56
5.5.3	Varying the sampling period	56
5.6	Conclusion	56
6	Collision Mitigation	57
6.1	Introduction	57
6.2	The Time To React (TTR)	57
6.3	Motion planning solving a lexicographic optimization problem	61
6.4	A lexicographic programming approach to maximize TTR	62
6.5	Simulation parameters and settings	63
6.6	Mitigating the risk of collision in the crowd scenario	63
6.6.1	Collision risk	63
6.6.2	Collision velocity	65
6.7	Conclusions	67
7	Prioritizing People and Jostling	68
7.1	Introduction	68
7.2	Prioritizing people in a mixed crowd	69
7.3	Jostling	69
7.4	Simulation parameters and settings	69
7.5	The effect of prioritizing people and jostling on the crowd scenario	70

7.5.1	Results for prioritizing people	70
7.5.2	Results for jostling	71
7.6	Conclusions	74
8	Conclusions	75
8.1	Summary	75
8.2	Future Work	76
A	Subtleties on the implementation of MPC for biped robots	78
A.1	Introduction	78
A.2	Derivation of the Discrete Time CoM Model	78
A.3	Center of Pressure position during the time interval	81
A.4	How the step selection matrices cycle	83
A.5	Kinematic constraints during the DS phase	84
	Bibliography	86

List of Figures

1.1	Research and test of humanoid robots.	1
1.2	Humanoid robots are controlled with a cascade of controllers [Dafarra 2018, Kajita 2010]. The MPC, the controller under analysis in this work, is just one of the many controllers that makes the robot moves. We assume that the output of MPC is perfectly tracked by next controllers.	3
2.1	The figure (taken from [HRP 2020]) shows HRP-2 humanoid robot (stature: 1.54m with $N = 30$) [Kaneko 2004].	7
2.2	The figure (taken from [Kajita 2014, pag.106]) shows the CoM at a constant height c^z above the ground, and the CoP \mathbf{p} inside the contact surface.	11
2.3	The figure is a top view of footprints (with rectangular shapes) onto the ground of a robot walking to the right (or to the left). We note the step and stride lengths.	13
2.4	The figure shows how we sample DS phases and the sampled gait and step cycles.	14
2.5	Representation of the walking cycle model (2.33) when starting from the end of the horizon N is assumed to be time-invariant, which implies, in particular, that the number and positions of the contacts do not change.	22
3.1	The figures represent when the robot is walking and is planning to stop (Figure 3.1a-3.1b), and when the robot considers actually making a new step (Figure 3.1c).	24
3.2	Representations of the closed convex polytopes $\mathcal{F}_i^N(\mathbf{T})$ and $\mathcal{K}_i^{1 N}(\mathbf{T})$, and SRF definition with time-varying constraints.	26
3.3	Representation of the 3 possible walking moments at time t_k and t_{k+1}	27
4.1	Representation of a biped robot (HRP-2) in a dense crowd.	34
4.2	Modeling the future in 1D space. Deterministic model (gray lines): future motion is known and available beforehand. Conservative model (represented by the gray area): we consider all the possible future motions. Probabilistic model (represented by a gray gradient): bound a probability distribution on pedestrian future occupancy.	37
4.3	Physical 2D modeling of the biped robot (and its perception) and the person.	39
4.4	Representation of the unsafe zone and its conservativeness in time.	42
4.5	Passive Safety has already been developed and used with several robotics platforms.	46
5.1	How often and when to re-plan the walking motion.	51
5.2	The inattentive crowd scenario.	52
5.3	Results of re-planning effect on success rate.	54

6.1	We simulate 1D scenario where a robot is walking away from a person. The robot aiming at Goal 6.2.2 (in green) without additionally guaranteeing to be at rest before entering the unsafe zone, achieved the greatest <i>TTR</i> since it is less constrained.	61
6.2	The figure shows the cumulative plot of the collision risk in percentage along the time of all simulations for PS and CM.	64
6.3	The figure shows a simulation for PS where the speed of the robot is not 0 at collision time, but it is converging to 0.	65
6.4	The figure shows the cumulative plot of the collision risk in percentage along the relative collision velocity of all simulations for PS and CM.	66
7.1	Two variants of the inattentive crowd scenario, with people prioritization (left), and jostling (right).	70
7.2	The figure shows the cumulative plot of collision risk in percentage along the time of all simulations for original CM (Hierarchy 1), transparent lines, and Hierarchy 2, in solid lines.	71
7.3	The figure shows the cumulative plot of collision risk in percentage along the time of all simulations of <i>small</i> crowd sizes for a robot that does not care about avoiding colliding with people (Hierarchy 4) and a robot that enables jostling (Hierarchy 3).	72
7.4	The figure shows the cumulative plot of collision risk in percentage along the time of all simulations of <i>big</i> crowd sizes for a robot that does not care about avoiding colliding with people (Hierarchy 4) and a robot that enables jostling (Hierarchy 3).	73
A.1	The figures represent the top view of the same biped robot walking for few footsteps. When we impose the DS kinematic constraints (A.41) along the walking motion of the robot (Figure A.1a), when we remove those constraints (Figure A.1b-A.1c) but we limit the distance between footprints (Figure A.1c).	85

List of Acronyms

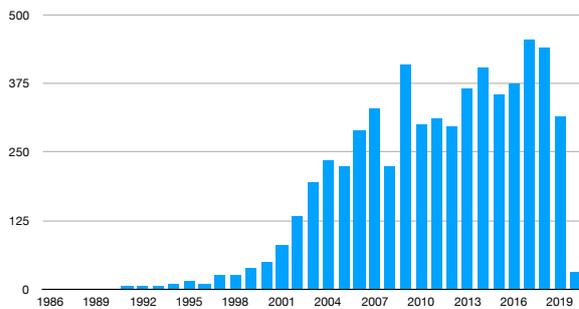
CM	Collision Mitigation
CoM	Center of Mass
CoP	Center of Pressure
CP	Capture Point
DS	Double Support
FoV	Field of View
ICS	Inevitable Collision State
LP	Linear Programming
MPC	Model Predictive Control
PFS	Passive Friendly Safety
PS	Passive Safety
QP	Quadratic Program
SFI	Strong Forward Invariance
SRF	Strong Recursive Feasibility
SS	Single Support

Chapter 1

Introduction

Figure 1.1: Research and test of humanoid robots.

(a) Number of publication per year under the search of “Humanoid Robotics” for Conference/Journals/Books in [IE 2020] (dated 12/03/2020).



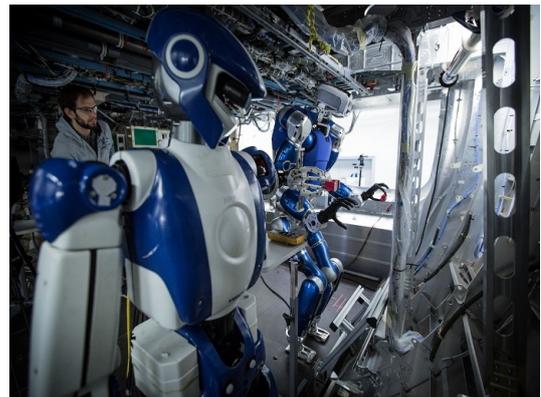
(b) Fukushima Daiichi nuclear disaster in 2011.



(c) DARPA Robotics Challenge (DRC) in 2015.



(d) The European project *Comanoid* in 2019.



1.1 Context and motivation

Humanoid robotics is one of the most fascinating research area in the field of robotics. Humanoid robots are comparable to humans at least in their motion capabilities and, given their versatility to perform a variety of activities (such as sensing, manipulation or locomotion), they are expected to have a tremendous impact on social and industrial settings. These

machines for example are meant to autonomously perform tasks in environment shared with humans, or to eliminate human involvement in dangerous and/or repetitive tasks and allow them to focus on high-value operations.

Humanoid robotics became a big research area since the early 2000s (Figure 1.1a). Nevertheless, existing humanoid robots back in 2011 were not ready to be deployed for a disaster response as the Fukushima (Japan) nuclear power plant accident [FUK 2011] (Figure 1.1b). In fact, some of the damage could have been minimized if it had been possible to access the manual valves inside the reactor building.

In order to accelerate this robotics research, the U.S. *DARPA* agency organized a “Robotics Challenge” [DRC 2015] (Figure 1.1c) where some remarkable progress has been made in terms of emergency response. And since 2014 the EU Research and Innovation programme called “*Horizon 2020*”^[1] invested part of its funding in projects that involve humanoid robots in environments shared with humans such as: dense crowds [CBP 2018], shopping mall [MuM 2016], or industrial settings [SHS 2015]. The European project *Comanoid* for example demonstrated successfully the deployment of biped robots to achieve human-robot collaboration in well-identified Airbus airliner assembly lines, which are inaccessible to wheeled platforms [Kheddar 2019] (Figure 1.1d). With the steady decline in prices owing to reduced sensor and actuator costs, humanoid robots are also gaining traction across all industry verticals (it is expected the global shipments to reach 1 million units by 2024 [GMI 2016]), and robotic companies are starting the first partnerships with big industries. Four-legged robots for example are used to evaluate robotic inspection and maintenance [AN2 2018, BDA 2020], but also *Ford* announced its plan to develop its biped robot “Digit” in cooperation with *Agility Robotics*, which will be used for delivery purposes [AR2 2019].

In this thesis, we focus only on two tasks of humanoid robots: balance (for bipedal locomotion), and motion safety capability (the capability to avoid collisions) in environments shared with humans.

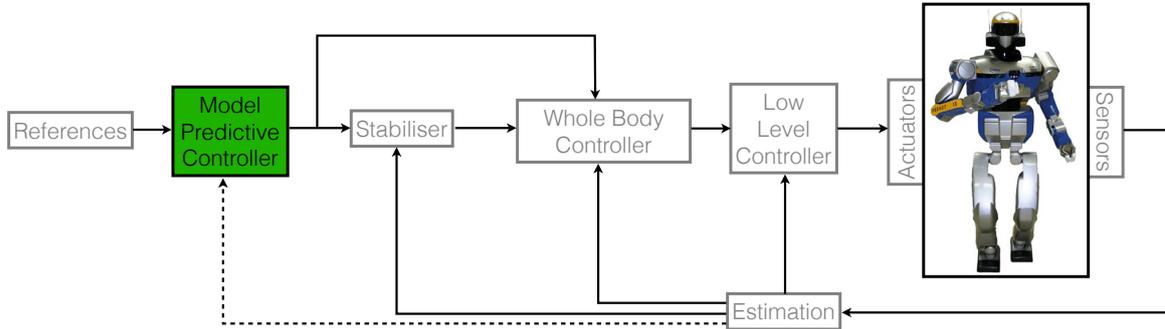
The concept of balance for humanoid robots can be formalized with the help of the viability theory [Wieber 2002]. The application of this theory makes it possible to formally separate situations where a fall is inevitable and which are therefore to be avoided, from situations where it is avoidable and which are therefore to be preferred. The same approach was proposed to formalize collision avoidance in dynamic environments, formally separating situations where a collision is inevitable and which are therefore to be avoided from situations where it is avoidable and which are therefore to be preferred. Such a formalization allows to demonstrate for example that existing approaches did not answer completely to the security requirements of robots [Fraichard 2006]. But to transform these formal advances into practical ones, the following aspect had to be taken into account. A robot can only anticipate the consequences of its actions on a limited time horizon (or planning horizon), if only because the future is uncertain.

To that end, it was proposed a *Partial Motion Planning* approach for collision avoidance [Petti 2005] and a similar *Model Predictive Control* (MPC) approach for the balance of humanoid robots [Wieber 2006a]: an iterative control process which computes explicitly the actions of the robot and their consequences over a limited time horizon. This obviously requires formulating objectives that can be achieved over a limited time horizon.

For the balance of humanoid robots when walking, the existing research suggests that the

¹<https://ec.europa.eu/programmes/horizon2020/>

Figure 1.2: Humanoid robots are controlled with a cascade of controllers [Dafarra 2018, Kajita 2010]. The MPC, the controller under analysis in this work, is just one of the many controllers that makes the robot moves. We assume that the output of MPC is perfectly tracked by next controllers.



length of the time horizon should cover 2 or 3 steps [Zaytsev 2015, Kajita 2003, Koolen 2012, Carver 2009]. Following the concept of Capturability [Koolen 2012], we can guarantee that the robot is able to stop in a few steps without falling (for whatever horizon length - covering whatever number of steps) and this is enough to guarantee that it is able to simply avoid falling. In MPC this is achieved by imposing a capturability condition at the end of the horizon [Sherikov 2016]. However, when the robot is not really planning to stop but actually consider making a new step at the end of the horizon to continue walking, the guarantee to avoid falling is lost. The *first* contribution of this work is to provide a numerical evidence that, despite the sudden plan change, the guarantee to avoid falling is not lost. But the guarantee depends on the length of the time horizon. We observe that many time horizons covering 2 steps lost this guarantee.

For the motion safety capability, it is impossible to guarantee *absolute motion safety* which is that no collision will ever occur, if only in the presence of non-cooperative or even hostile behaviour. What can actually be guaranteed over a limited time horizon, is *Passive Safety (PS)*: the robot is able to stop before a collision occurs. If a collision is inevitable, at least the robot will be at rest when that happens. No collision would happen if everybody behaved that way, so in a sense the robot will have done its share [Bouraine 2014]. Capturability has previously been used to successfully control the movement of a humanoid robot in a crowd and to guarantee both fall avoidance and PS in a single MPC scheme [Bohorquez 2016].

For biped robots, once a step is planted on the ground, it usually stays there at a constant position until the next step is initiated. This naturally constrains the capacity for the robot to react and adapt its motion in between steps. As a result, the walking strategies in [Chestnutt 2005, Karkowski 2016, Garimort 2011] propose to re-plan the walking motion to adapt to changing environments once per step, only when a new step is initiated. In contrast, the MPC scheme outlined above re-plans the walking motion not only at each step initiation but also in between (8 times per step). Obviously re-planning more often than once per step comes at the expense of computational power. The *second* contribution of this work is to show that, when controlling the movement of a humanoid robot in a crowd, we can favor to re-plan the walking motion to adapt to changing environments once per step, instead of 2, 4 or 8 times per step, since motion safety capabilities of the robot are not degraded. We thus save computational power without deteriorating the robot’s motion safety capability. But re-planning should only happen just before initiating

the next step, or during the time that the robot has only one step in contact with the ground.

PS has been criticized because while the robot makes sure to always have time to stop before a collision happens, this time might not be enough for people around to actually react and avoid the collision once the robot has stopped [Macek 2008]. Based on the criticism of **PS**, we conveyed that people around the robot could potentially attempt to react and avoid collisions if they have enough time to do so. For this reason, we claim the following: more time to react for the surrounding environment reduces the number of collisions. In the literature, a variant of **PS** was proposed under the name of *Passive Friendly Safety* (**PFS**) [Macek 2008, Mitsch 2013]: the robot will be at rest before collisions happen, leaving enough time for the surrounding environment to react and avoid collisions. Both **PS** and **PFS** however limit the time to react the robot can leave for people because the robot is constrained to stop before collisions happen. We show that if the robot is not constrained to stop before collisions happen, it can further improve this time. For this reason, we propose to control the movement of a humanoid robot in a crowd that guarantee fall avoidance while aiming to leave as much time to react as possible for the people (instead of guaranteeing **PS**) in a single **MPC** scheme, that we call *Collision Mitigation* (**CM**). The *third* contribution of this work is to show that, when controlling the movement of a humanoid robot in a crowd, we can favor **CM** instead of **PS** to reduce the number of collisions and postpones these collisions later in time.

We consider the case where a humanoid robot moves in a crowd and the members of this crowd are equally divided in robots and people. The *fourth* contribution of this work is to control the movement of a humanoid robot in this particular crowd scenario with a new single **MPC** scheme. This scheme guarantees fall avoidance while aiming to leave as much time to react as possible for the people and then, if possible, for the other robots. When this revised version of **CM** takes priorities into account, *e.g.* prioritizing people, the simulation results reflect these priorities. The biped robot collides always later with people and reduces the number of collisions with them. For the other robots instead, collisions happen earlier and more often.

Last, we propose another situation for the robot moving in a crowd (populated with only people). We suppose the robot must reach a target location at the utmost important, and this time people might obstruct the motion of the robot. The *fifth* (and last) contribution is to control the movement of a humanoid robot in this situation with a new single **MPC** scheme. This scheme guarantees fall avoidance while enabling the robot to jostle people of the crowd if necessary to reach the target location. We compare this scheme in the situation outlined above with a robot that does not enable jostling but only aims to reach the target location as soon as possible. Our results show that the robot collides less and collisions are postponed in time when we enable jostling. However, If we increase the density of the crowd, both robots converges to equally perform in terms of collisions. This is because the robot might not have enough space to jostle among people.

1.2 Contributions and Outline

The contributions of this thesis are:

- We bring the *Strong Recursive Feasibility* (**SRF**) concept from **MPC** theory in the field of humanoid robotics. The relationship between **SRF** and viability is the following: **SRF** guarantees that all feasible states are viable at all time. In case of balance preservation for biped robots, no feasible state can lead to a fall of the robot.

- We introduce two new **MPC** variables: *planning period* and *initial planning phase*, that in our case decide how often and when to re-plan the walking motion. Bigger planning period means to re-compute less often the walking motion and save computational power.
- We propose a single **MPC** scheme that guarantees fall avoidance while aiming to leave as much time to react as possible for the people to avoid colliding with the robot. This scheme is a revised version of *Collision Mitigation* proposed in [Bohorquez 2018a] where fall avoidance was not guaranteed because the robot would *sacrifice* its capturability for aiming to leave as much time to react as possible for the people to avoid colliding with the robot.
- We propose an **MPC** scheme for a robot moving in a crowd used to *prioritize* collision avoidance with specific members of the crowd.
- We propose an **MPC** scheme for a robot moving in a crowd, that enables the robot push its way through the crowd, *i.e.* to *jostle* people, if necessary to reach the target location.

1.2.1 List of publications

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

- M. Ciocca, P.-B. Wieber and T. Fraichard. *Strong recursive feasibility in model predictive control of biped walking*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pages 730–735, Nov 2017
- M. Ciocca, P.-B. Wieber and T. Fraichard. *Effect of Planning Period on MPC-based Navigation for a Biped Robot in a Crowd*. In IROS 2019 - IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1–8, Macau, China, November 2019. IEEE

And presented at the workshop:

- *From freezing to jostling robots: Current challenges and new paradigms for safe robot navigation in dense crowds* in the International Conference on Intelligent Robots (IROS) 2018, Madrid, Spain.

1.2.2 Outline

The thesis starts with the introduction of the dynamical model for the robot, the problem of balance and **MPC** scheme for biped walking in Chapter 2. The contribution to the guarantee of avoiding falling even when the robot is not really planning to stop but consider to make a new step to continue walking is explained and explored in Chapter 3. The problem of robot’s motion safety capability in a crowd is presented in Chapter 4. In this chapter, we present the choice of the crowd behavior and the **MPC** scheme that guarantee both fall avoidance and passive safety. The contribution to the re-planning frequency on the **MPC**-based safe navigation of a humanoid robot in a crowd is explained and explored in Chapter 5. The **MPC** scheme that guarantees fall avoidance while aiming to leave as much time to react as possible for the people to avoid colliding with the robot is presented in Chapter 6. Last, the two new **MPC** schemes: one is used to *prioritize* collision avoidance with specific members of the crowd, and the second to enable *jostling*, are presented in Chapter 7.

1.3 Notation

Software names

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

General scalars, vectors, matrices

- Vectors and matrices are denoted by letters in a bold font: \mathbf{v} , \mathbf{M} .
- Scalars are denoted using the standard italic font: N, n .
- Special functions are written in plain text: `diag`(\cdot).
- $(\cdot)^\top$ – transpose of a matrix or a vector.
- Block diagonal matrices:

$$\text{diag}_2(\mathbf{M}) = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \quad (1.1)$$

- Inequalities between vectors $\mathbf{v} \geq \mathbf{r}$ are interpreted component-wise.
- $\|\cdot\|_p$ 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 (or Euclidean) norm.

Sets

- Sets are written in calligraphic \mathcal{S}, \mathcal{P} or Capital Greek letters Ω, \mathbb{T} .
- Special sets are written in blackboard bold $\mathbb{R}, \mathbb{N}, \mathbb{Z}$.
- \mathbb{R} is the set of real numbers.
- \mathbb{R}^n is the set of real-valued vectors.
- $\mathbb{R}^{n \times m}$ is the set of real-valued matrices.

Other

- Function names in mathematical expressions are written in the regular font: `func`(\mathbf{x}, \mathbf{y}).

Chapter 2

Balance

Figure 2.1: The figure (taken from [HRP 2020]) shows HRP-2 humanoid robot (stature: 1.54m with $N = 30$) [Kaneko 2004].



2.1 Introduction

In this chapter we introduce the robot dynamic model and the control scheme, that employs this model, to generate a stable walking motion. We start describing the general mechanical model of the biped robot (Section 2.2.1). We explain how the motion of the center of mass is linearly related to the contact forces applied on the ground (Section 2.2.2), and how we model a walking cycle for the robot (Section 2.2.3). When walking on a flat ground, the robot needs to satisfy a set of constraints to comply with the whole-body kinematic and dynamical structure (Section 2.2.4). In this work, if the robot satisfies those constraints not only in the present but also into the future, it preserves its balance (Section 2.3). A sufficient condition to preserve balance is the robot's ability to stop in a given number of steps, called *capturability* (Section 2.3.1). The control scheme that employs these model and constraints is called *Model Predictive Control (MPC)* and it is explained in Section 2.4. In our implementation we point out an issue that will be the core of the next chapter.

2.2 Biped robot

2.2.1 Mechanics

A biped robot (such as HRP-2 in Figure 2.1) is composed by $R+1$ rigid bodies, called links, connected by R rotational joints with 1 degree of freedom each. This multi-body system is *free floating*, *i.e.* none of the links have a priori constant pose with respect to a general frame of reference, so the robot can move freely in 3D environment. The configuration space \mathcal{Q} of the robot can then be characterized by: the joint configurations and, the *position* and *orientation* of a frame attached to a robot's link called the *base* frame^[1]. This space is $\mathcal{Q} = \mathbb{R}^n \times \mathbb{R}^3 \times SO(3)$, and its elements are triplets composed by the *generalized coordinates*:

$$\mathbf{q} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{bmatrix} \in \mathcal{Q}, \quad (2.1)$$

where $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_n\} \in \mathbb{R}^n$ are joint angles and $(\boldsymbol{\rho}, \boldsymbol{\theta}) \in \mathbb{R}^3 \times SO(3)$ denotes the origin and orientation of the base frame.

The biped robot is equipped with rotary actuators at each joint α_i to change the robot's posture, but we do not include additional actuators, *e.g.* thrusters, that can directly act on $(\boldsymbol{\rho}, \boldsymbol{\theta})$ and would enable the robot to fly [Nava 2018]. The multi-body system is then *under-actuated*. Moreover, we consider the biped robot in contact with the environment, so it is affected by forces of interactions. The corresponding Lagrangian dynamics of the multi-body system has the following structure [Wieber 2016]:

$$\mathbf{M}(\mathbf{q}) \left(\ddot{\mathbf{q}} + \begin{bmatrix} 0 \\ \mathbf{g} \\ 0 \end{bmatrix} \right) + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \boldsymbol{\eta} \\ 0 \\ 0 \end{bmatrix} + \sum_i \mathbf{J}_i^\top(\mathbf{q}) \boldsymbol{\lambda}_i, \quad (2.2)$$

where $\mathbf{M} \in \mathbb{R}^{(N+6) \times (N+6)}$ is the generalized inertia matrix, $\mathbf{g} \in \mathbb{R}^3$ is the constant gravity acceleration vector, $\mathbf{N} \in \mathbb{R}^{(N+6)}$ is the vector of Coriolis and centrifugal effects, $\boldsymbol{\eta} \in \mathbb{R}^N$ is the vector of joint torques (provided by the actuators), $\boldsymbol{\lambda}_i \in \mathbb{R}^3$ is a force exerted by the environment on the robot and $\mathbf{J}_i \in \mathbb{R}^{3 \times (N+6)}$ is the associated Jacobian matrix [Wieber 2006b].

Any robot's posture must be restricted to the ones without interpenetration of robot body parts, *i.e.* self-collisions, and that takes into account hardware restrictions, *e.g.* actuator limits. These limitations can be represented by *kinematic* and task-related constraints [Sherikov 2016]:

$$\phi \leq \phi(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\eta}) \leq \bar{\phi}. \quad (2.3)$$

From the Lagrangian dynamics, it appears the fact that joint torques $\boldsymbol{\eta}$ cannot act directly on $(\boldsymbol{\rho}, \boldsymbol{\theta})$: joint torques cannot make the robot move around. Thus robot locomotion requires external contact forces. We consider the general case where these contact forces $\boldsymbol{\lambda}_i$ are the only one between the feet and the ground: ground reaction forces. However other parts of the robot can produce helpful support [Kudruss 2015], or produce other types of locomotion, *e.g.* crawling [Kuehn 2016].

When the robot takes a step, *i.e.* establishes a contact with the ground, an impact occurs. Impacts are assumed to be collisions with rigid surface and they are normally modeled as a *non-smooth* transition between the pre- and post-impact velocities of the system [Hurmuzlu 2004]:

$$(\mathbf{q}^+, \dot{\mathbf{q}}^+) = \Delta(\mathbf{q}^-, \dot{\mathbf{q}}^-). \quad (2.4)$$

¹The base frame is typical central to the body (pelvis or trunk) or in an extremity (foot or hand).

In robot locomotion, the continuous dynamics (2.2) is suddenly interrupted every time the robot takes a step (2.4), for this reason the dynamics becomes *hybrid*.

Forces, required for locomotion, can only be applied to contacts already made. Unless the biped robot can fasten its feet to the ground at will, *e.g.* with some sort of clamping device (such as vacuum suction cup) on the sole of the feet, it can only push and not pull on the ground. This can be modeled as a *complementarity* condition between contacts and forces [Hurmuzlu 2004]. Denote with the superscripts $(\cdot)^{\parallel}$ and $(\cdot)^{\perp}$ tangential and normal components of vectors with respect to the contact surfaces, and let be \mathbf{r}_i the position vector of the i^{th} contact, we have

$$r_i^{\perp} \lambda_i^{\perp} = 0, \quad \text{with} \quad r_i^{\perp} \geq 0, \quad \lambda_i^{\perp} \geq 0. \quad (2.5)$$

The unilateral nature of this contact interaction (the feet can only push on the ground) imposes the robot to exploit ground friction to walk. However, it must avoid foot slipping and this is achieved by satisfying the following constraint [Sherikov 2016]:

$$\|\boldsymbol{\lambda}_i^{\parallel}\| \leq \mu_i \lambda_i^{\perp}, \quad (2.6)$$

where μ_i is the friction coefficient of the i^{th} contact interaction.

Actuation of the joints can produce a sequence of postures that must always satisfy kinematic constraints (2.3) and account (indirectly) for ground reaction forces, *e.g.* with (2.4)-(2.5)-(2.6), in order to realize a desired balancing or walking motion [Sherikov 2016].

2.2.2 Contact forces and CoM relationship when walking on flat ground

When ground reaction forces act on the robot, they directly affect the base frame position $\boldsymbol{\rho}$. If we choose the Center of Mass (CoM) position \mathbf{c} as base frame position: $\mathbf{c} = \boldsymbol{\rho}$, the under-actuated part of the Lagrangian dynamics boils down to the Newton and Euler equations of motion of the robot taken as a whole [Wieber 2006b]. They correspond to:

$$m(\ddot{\mathbf{c}} + \mathbf{g}) = \sum_i \boldsymbol{\lambda}_i \quad (2.7)$$

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

where m is the total mass of the robot, and \mathbf{L} is the centroidal angular momentum of the entire robot. We sum the $\mathbf{c} \times$ of Newton equation (2.7) to the Euler equation (2.8) and we obtain:

$$m \mathbf{c} \times (\ddot{\mathbf{c}} + \mathbf{g}) + \dot{\mathbf{L}} = \sum_i (\mathbf{r}_i \times \boldsymbol{\lambda}_i) \quad (2.9)$$

Denote with $(\cdot)^x$, $(\cdot)^y$, $(\cdot)^z$ the Cartesian coordinates of a frame with origin on the contact surface of one foot on the ground and the axis z normal to it. Now we divide the result (2.9) by the z coordinate of the Newton equation (2.7):

$$\frac{m \mathbf{c} \times (\ddot{\mathbf{c}} + \mathbf{g}) + \dot{\mathbf{L}}}{m(\ddot{c}^z + g^z)} = \frac{\sum_i (\mathbf{r}_i \times \boldsymbol{\lambda}_i)}{\sum_i \lambda_i^z}. \quad (2.10)$$

Assume all the contact points lie in a horizontal plane and for simplicity the plane is located at zero height, therefore $\forall i, r_i^z = 0$. This allows us to simplify the cross products in (2.10) in the following way:

$$\begin{bmatrix} c^y \\ -c^x \end{bmatrix} - \frac{c^z}{(\ddot{c}^z + g^z)} \begin{bmatrix} (\ddot{c}^y + g^y) \\ -(\ddot{c}^x + g^x) \end{bmatrix} + \frac{1}{m(\ddot{c}^z + g^z)} \begin{bmatrix} \dot{L}^x \\ \dot{L}^y \end{bmatrix} = \frac{1}{\sum_i \lambda_i^z} \sum_i \begin{bmatrix} r_i^y \\ -r_i^x \end{bmatrix} \lambda_i^z. \quad (2.11)$$

We multiply the equation with a rotation matrix $R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and we obtain:

$$\mathbf{c}^{x,y} - \frac{c^z}{(\ddot{c}^z + g^z)}(\ddot{\mathbf{c}}^{x,y} + \mathbf{g}^{x,y}) + \frac{1}{m(\ddot{c}^z + g^z)}R\dot{\mathbf{L}}^{x,y} = \frac{\sum_i \mathbf{r}_i^{x,y} \lambda_i^z}{\sum_i \lambda_i^z}. \quad (2.12)$$

There is a point on the contact surface where the horizontal momenta produced by the contact forces, λ_i , with respect to this point are equal to zero:

$$\sum_i (\mathbf{p}^{x,y} - \mathbf{r}_i^{x,y}) \lambda_i^z = 0, \quad (2.13)$$

this point is called Center of Pressure (**CoP**), $\mathbf{p}^{x,y}$ [Sardain 2004]. And it is exactly what appears on the RHS of equation (2.12)

$$\mathbf{p}^{x,y} = \frac{\sum_i \mathbf{r}_i^{x,y} \lambda_i^z}{\sum_i \lambda_i^z}. \quad (2.14)$$

We consider the case of walking on a *horizontal ground*, that implies $\mathbf{g}^{x,y} = 0$. Moreover, we consider that the **CoM** moves strictly *horizontally* above the ground: c^z is constant and $\ddot{c}_i^z = 0$ (see Figure 2.2). And equation (2.12) becomes:

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

Last, we consider the variations of the angular momentum equal to zero, $\dot{\mathbf{L}}^{x,y} = 0$. This is not true in reality: execution of limb motions implies certain values of rate of angular momentum, but it has proven to be sufficient to realize robot locomotion [Feng 2016]. Variations of the angular momentum $\mathbf{L}^{x,y}$ however can be estimated using multi-mass models [Shimmyo 2013] and/or can be bounded: to generate vertical motion [Serra 2016] or because accounted as source of uncertainty [Villa 2019a]. We end up with a linear differential equation that relates the dynamics of the **CoM** with contact forces:

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

where

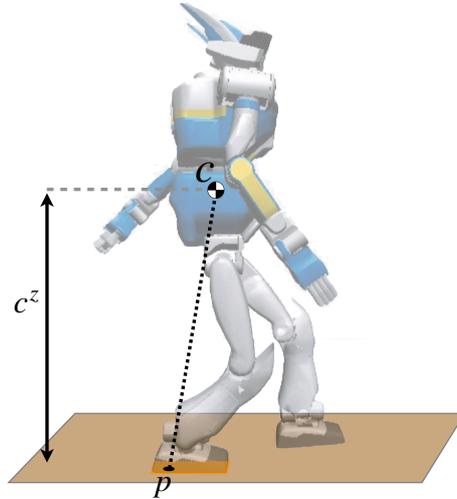
$$\omega = \sqrt{\frac{g^z}{c^z}}. \quad (2.17)$$

This model has been experimentally validated for the balance of human beings while standing [Winter 1998], and it has been widely used for control of legged robots [Englsberger 2011, Kajita 2010, Romualdi 2018]. From now on we remove the x and y superscripts.

For control purpose, the dynamics (2.16) has been originally proposed as a triple integrator by [Kajita 2003] where **CoM** jerk, $\ddot{\mathbf{c}}$, is the control input. This was later employed and extended in [Diedam 2008, Herdt 2010, Agravante 2016] and many other works. The triple integrator implies a smooth variation of **CoM** acceleration. This results in a smooth change of **CoP** position (2.16), which helps the realization of robot locomotion. This model however leads to some subtle difficulties in terms of realization of robot locomotion but they can be fixed by appropriate control design choices [Wieber 2006a]. Discussion on this issue can be found in Appendix A.3.

A second or third order model can use the **CoP** position as state [Scianca 2016, Kajita 2010].

Figure 2.2: The figure (taken from [Kajita 2014, pag.106]) shows the **CoM** at a constant height c^z above the ground, and the **CoP** p inside the contact surface.



This is suggested since modern robots can be equipped with force sensors in the feet to estimate the position of the **CoP** [Englsberger 2014, Kaneko 2004, Kaneko 2009]. Another way is to use the **CoP** position as control input [Villa 2019a]. In this case, the dynamics can be transformed in a modal form, exposing its stable and unstable parts [Englsberger 2011, Takenaka 2009, Krause 2012].

In this work, we consider a third order model that use **CoP** velocity $\dot{p} \in \mathbb{R}^2$ as control input [Sherikov 2016]. This choice is not subjected at subtleties of the triple integrator with **CoM** acceleration as control input, and yet implies a smooth change of **CoP** position. The state of this model is the following:

$$\mathbf{x} = [c^x \quad \dot{c}^x \quad \ddot{c}^x \quad c^y \quad \dot{c}^y \quad \ddot{c}^y]^\top. \quad (2.18)$$

By differentiating the dynamics (2.16) we have the following linear time invariant system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{G}\mathbf{x} + \mathbf{H}\dot{p} \\ \mathbf{p} = \mathbf{D}\mathbf{x}, \end{cases} \quad (2.19)$$

$$(2.20)$$

where

$$\mathbf{G} = \text{diag}_2 \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix} \right), \quad \mathbf{H} = \text{diag}_2 \left(\begin{bmatrix} 0 \\ 0 \\ -\omega^2 \end{bmatrix} \right), \quad \mathbf{D} = \text{diag}_2 \left([1 \quad 0 \quad -\frac{1}{\omega^2}] \right). \quad (2.21)$$

We consider that there are *no uncertainties* in measurements of the state and/or in parameters of the robot as in [Sherikov 2016]. We refer to [Villa 2019b] for the study of uncertainties in modeling and control of legged robots.

Discrete-time CoM model

Standard approaches to control legged robots rely on discrete-time models. We discretize time in a sequence of time instants t_i , $i \in \mathbb{N}$. We represent the value of any variable at time t_i as $(\cdot)_i$. Each time interval $[t_i, t_{i+1}]$ has a fixed duration T , usually called *sampling period*, in seconds. From now on we refer to this type of discretization in case of discrete-time models.

The control input is generally kept constant for a fixed time duration, this approach is called zero-order hold for controls [Ogata 1995, Chapter 5].

$$\dot{\mathbf{p}}_j = \dot{\mathbf{p}}_i, \quad \text{for } t_i \leq t_j < t_{i+1}. \quad (2.22)$$

By doing so, step-by-step in Appendix A.2, the model (2.19) corresponds to

$$\begin{cases} \mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\dot{\mathbf{p}}_i \\ \mathbf{p}_i = \mathbf{D}\mathbf{x}_i, \end{cases} \quad (2.23)$$

$$(2.24)$$

where^[2]

$$\mathbf{A} = \text{diag}_2(e^{\mathbf{G}T}) = \text{diag}_2 \left(\begin{bmatrix} 1 & \sinh(T\omega)/\omega & \cosh(T\omega)/\omega^2 - 1/\omega^2 \\ 0 & \cosh(T\omega) & \sinh(T\omega)/\omega \\ 0 & \sinh(T\omega)\omega & \cosh(T\omega) \end{bmatrix} \right), \quad (2.25)$$

$$\mathbf{B} = \text{diag}_2 \left(\left(\int_0^T e^{\mathbf{G}t} dt \right) \mathbf{H} \right) = \text{diag}_2 \left(\begin{bmatrix} T - \sinh(T\omega)/\omega \\ 1 - \cosh(T\omega) \\ -\omega \sinh(T\omega) \end{bmatrix} \right). \quad (2.26)$$

It is then possible to describe a CoM trajectory by evolving for example the state \mathbf{x}_i N -times following the dynamics (2.23):

$$\bar{\mathbf{c}}_i = \text{diag}_N(\mathbf{I}_c) (\mathbf{U}_1 \mathbf{x}_i + \mathbf{U}_2 \bar{\mathbf{p}}_i), \quad (2.27)$$

which relates the sequence $\bar{\mathbf{c}}_i \in \mathbb{R}^{2N}$ of N CoM (2D) positions^[3] (from time t_{i+1} to t_{i+N}),

$$\bar{\mathbf{c}}_i = \{\mathbf{c}_{(i+1|i)}, \dots, \mathbf{c}_{(i+N|i)}\} \quad (2.28)$$

with a sequence $\bar{\mathbf{p}}_i \in \mathbb{R}^{2N}$ of CoP velocities^[4] (from time t_i to t_{i+N-1}),

$$\bar{\mathbf{p}}_i = \{\dot{\mathbf{p}}_{(i|i)}, \dots, \dot{\mathbf{p}}_{(i+N-1|i)}\} \quad (2.29)$$

with

$$\mathbf{U}_1 = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}\mathbf{B} & \mathbf{B} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{B} \end{bmatrix}, \quad (2.30)$$

and $\mathbf{I}_c \in \mathbb{R}^{2 \times 6}$ a selection matrix to extract the CoM position from the state. The same state evolution can be used to describe a CoP trajectory, from (2.23) and (2.24):

$$\bar{\mathbf{p}}_i = \mathbf{U}_3 \mathbf{x}_i + \mathbf{U}_4 \bar{\mathbf{p}}_i, \quad (2.31)$$

where $\bar{\mathbf{p}}_i \in \mathbb{R}^{2N}$ describes the evolution of the CoP position,

$$\bar{\mathbf{p}}_i = \{\mathbf{p}_{(i+1|i)}, \dots, \mathbf{p}_{(i+N|i)}\} \quad (2.32)$$

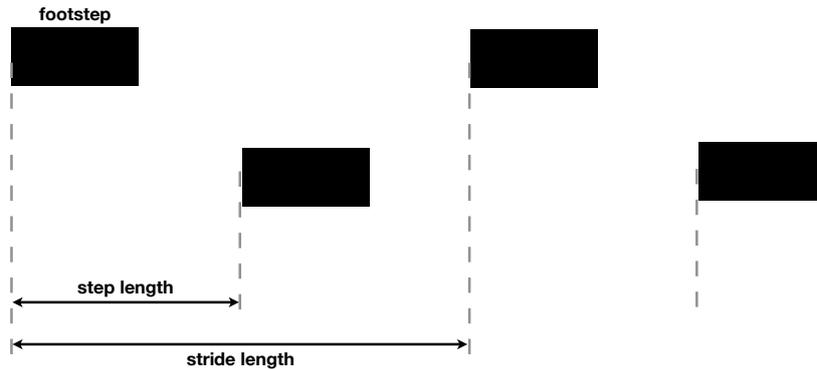
with $(\mathbf{U}_3, \mathbf{U}_4)$ similar to $(\mathbf{U}_1, \mathbf{U}_2)$.

² Euler's number: e .

³The state $\mathbf{x}_{(\beta|\alpha)}$ of the evolution is described at time t_α and is reached at time t_β .

⁴The control input $\mathbf{u}_{(\beta|\alpha)}$ of the sequence is described at time t_α to be executed at time t_β and kept constant for a duration T .

Figure 2.3: The figure is a top view of footprints (with rectangular shapes) onto the ground of a robot walking to the right (or to the left). We note the step and stride lengths.



2.2.3 Walking cycle

Modeling of biped robot walking is inspired by human walking [Rose 2005]. Walking is a sequence of regularly recurring events. At each event there is, at least, one foot on the ground. A gait (or stride) cycle is defined as the period between a foot contact on the ground to the next contact of the same foot on the ground again. The stride length is defined as the distance between successive ground contacts of the same foot. The *step cycle* (s_d) is defined as the period between a foot contact on the ground to the next contact of the opposite foot, *i.e.* half gait cycle. The step length is the distance between the point of initial contact of one foot and the point of initial contact of the opposite foot. We consider the case where the step cycle is divided into two events: Single Support (SS) and Double support (DS) phases. Lengths are represented in Figure 2.3 and cycles in Figure 2.4b.

A biped robot starts in DS, that is, with both feet in contact with the ground. Then it lifts one 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 SS. The moment the swinging leg hits the ground produces an impact and the walker returns to the DS again. In our case, the heel rise is absent during the SS phase and the (flat) foot is lifted all together. This can limit the walking speed. Rotation about the front edge of the stance foot before heel strike or rotation through the use of toe joint [Kaneko 2011] can benefit the walking speed but it would increase the difficulty of controlling the robot.

We model changes of the foot positions on the ground in order to facilitate their *automatic adjustment*, which helps for example in: disturbance compensation [Sherikov 2016], physical interaction [Agravante 2016] and tracking of desired walking speed [Herdt 2010]. This automatic adjustment might be complex. When the robot walks on a flat ground, it may have two feet contacts with the ground at the same time, *e.g.* when the robot is in DS phase of a walking cycle. Multiple contacts with the same surface may lead to a contact surface of a complex shape. This area depends on the position and orientation of the feet onto the ground: these variables can be accounted in the automatic adjustment [Naveau 2017, Bohórquez 2018b]. We can also adjust the step cycle duration along the walking [Bohórquez 2017]. In this work, we build a discrete-time model for the changes of the foot positions on the ground and we made the following simplifications to reduce the complexity of the automatic adjustment as is done in [Herdt 2010]:

- we consider that foot rotations are fixed in advance: we disable the possibility to reorient the feet along the walking.
- we consider a fixed time duration for both DS and SS phases. As a consequence, we

have a fixed duration s_d .

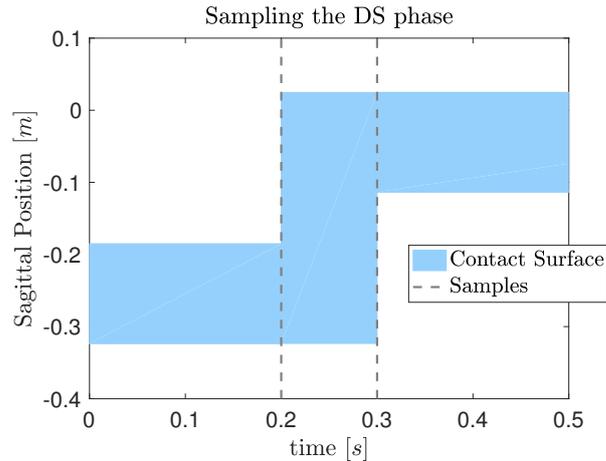
- at each sample, we consider that the robot puts its weight on only *one* foot in contact with the ground. Let s^{pc} be the next foot that will hit the ground, starting the **DS** phase and let s^c be the current foot that will be lifted, ending the **DS** phase. We sample as follows:
 - we sample exactly when s^{pc} hits the ground, when the **DS** phase starts. In this case s^c is considered the contact with the ground supporting the weight of the robot. At this sample, s^{pc} is already fixed.
 - we sample exactly when s^c is lifted from the ground, when the **DS** phase ends. In this case s^{pc} is considered the contact with the ground supporting the weight of the robot. At this sample, we update s^c with s^{pc} .

So the sampling period T is considered the duration of the **DS** phases.

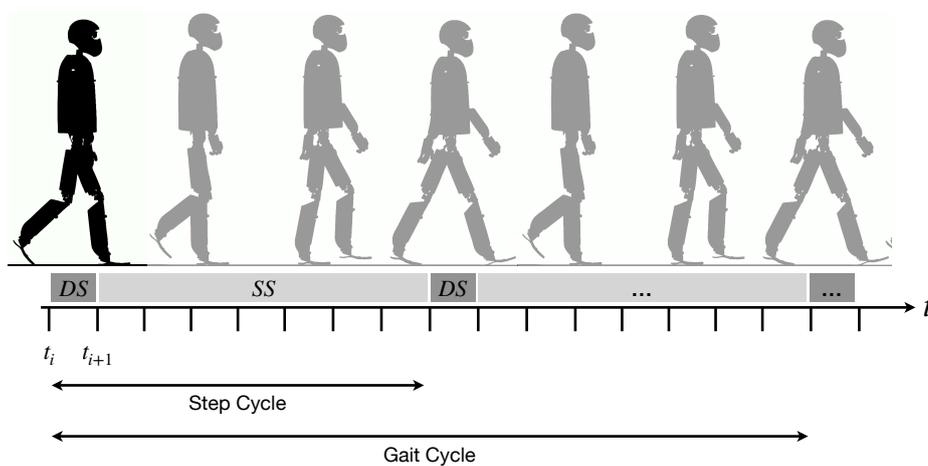
We represent some of these aspect in Figure 2.4.

Figure 2.4: The figure shows how we sample **DS** phases and the sampled gait and step cycles.

- (a) The robot starts in **SS** phase on position s^c . At 0.2[s] the next foot s^{pc} hits the ground: the **DS** phase starts. And s^c is lifted from the ground at 0.3[s]: the **DS** phase ends. The contact surface in blue depends on the foot or feet in contact with the ground. We sample when the **DS** phase starts and ends (samples in dashed gray lines).



- (b) In the figure (taken and adjusted from [Pratt 2012]) is shown when **DS** and **SS** phases happens, and the duration of a gait and step cycle. The sampling period T is the duration of the **DS** phases.



With these simplifications, we can model the evolution of a walking cycle for N time instants as

$$\bar{\mathbf{s}}_i = \mathbf{V}_i^{pc} \mathbf{s}_i^{pc} + \mathbf{V}_i^f \mathbf{s}_i^f, \quad (2.33)$$

where $\bar{\mathbf{s}}_i \in \mathbb{R}^{2N}$ is the sequence of footsteps ahead (from time t_{i+1} to t_{i+N}),

$$\bar{\mathbf{s}}_i = \{\mathbf{s}_{(i+1|i)}, \dots, \mathbf{s}_{(i+N|i)}\} \quad (2.34)$$

In case of walking, once a step is planted on the ground, it usually stays there at a constant position for quite sometime until the next step is initiated. As a consequence, we represent as $\mathbf{s}_{[j|i]}$, the j -th footstep planted for several time instants, and we can group the sequence $\bar{\mathbf{s}}_i$ in the number of footsteps only as

$$\bar{\mathbf{s}}_i = \{\mathbf{s}_{[1|i]}, \dots, \mathbf{s}_{[J|i]}\} \quad (2.35)$$

And $\mathbf{s}_i^{pc} \in \mathbb{R}^2$ can be the next foot that will hit the ground or identical to the current step position onto the ground $\mathbf{s}^c \in \mathbb{R}^2$. In both cases the variable \mathbf{s}_i^{pc} is always fixed. And $\mathbf{s}_i^f \in \mathbb{R}^{2m}$ is a sequence of m adjustable foot positions. The matrices $(\mathbf{V}_i^{pc}, \mathbf{V}_i^f)$ vary cyclically and determine which foot is onto the ground at what time

$$\mathbf{V}_i^{pc} = \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2N \times 2}, \quad \mathbf{V}_i^f = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{I} & \dots \end{bmatrix} \in \mathbb{R}^{2N \times 2m}. \quad (2.36)$$

We explain in details these matrices and how they cycle in Appendix A.4. In this work we do not account for the motion of the feet in the air. It could be modeled however with a 3-rd order polynomial as in [Sherikov 2016, Section 4.4.3].

2.2.4 Whole-body restrictions on CoM dynamics

The joint torques $\boldsymbol{\eta}$ that will actuate the entire mechanical structure are chosen with an *inverse dynamics* (ID) problem [Sherikov 2016]. This problem takes as input generalized coordinates \mathbf{q} and contact forces $\boldsymbol{\lambda}_i$ and as output find the joint torques that satisfy the Lagrangian dynamics (2.2)

$$\boldsymbol{\eta} = \text{ID}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\lambda}_i). \quad (2.37)$$

From the evolution of CoM dynamics (2.19)-(2.20), we can obtain a sequence of contact forces (thanks to the CoP). And, there exists a relationship between CoM and the whole-body kinematics (thus generalized coordinates)

$$\dot{\mathbf{c}} = \mathbf{J}_{\text{CoM}} \dot{\mathbf{q}}, \quad (2.38)$$

where \mathbf{J}_{CoM} is a Jacobian matrix [Sugihara 2002]. It is then possible to generate a desired walking motion via CoM dynamics and use it to move the whole-body of the robot.

As we explained in Section 2.2.1, generalized coordinates and contact forces need to comply with mechanical constraints, *e.g.* joint limits and friction constraints. If we ensure

the realization of the **CoM** motion for the whole-body hardware restrictions and with the limitations of each physical interaction, it is then possible to use the inverse dynamics problem tracking the **CoM** motion and contact forces in order to realize a walking motion. This motivates the introduction of a set of constraints on the **CoM** dynamics and contact forces that reflects whole-body restrictions.

Consider only the case of one contact with the ground, hereby called the j^{th} footstep $\mathbf{s}_j \in \mathbb{R}^2$. Since contact forces with the ground are unilateral, the **CoP** is always constrained within a convex hull \mathcal{P} , called the *support polygon*

$$\mathbf{p} \in \mathcal{P}(\mathbf{s}_j) \quad (2.39)$$

that varies depending on the position and orientation of \mathbf{s}_j . In our case foot rotation is fixed in advance, so the support polygon depends only on the position of \mathbf{s}_j . In our work the feet of the robot have a rectangular shape, and the set-valued function $\mathcal{P}(\mathbf{s})$ specifies a rectangular region of width and length $w \times l$ around the position \mathbf{s} :

$$\mathcal{P}(\mathbf{s}) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 \mid \begin{bmatrix} -l/2 \\ -w/2 \end{bmatrix} \leq (\mathbf{r} - \mathbf{s}) \leq \begin{bmatrix} l/2 \\ w/2 \end{bmatrix} \right\}. \quad (2.40)$$

We consider that friction forces constraints to avoid slipping (2.6) are always satisfied, but additional constraints can be imposed [Caron 2015]. We consider that the position of the $(j+1)^{\text{th}}$ footstep with respect to the position of the j^{th} footstep is restricted to a region \mathcal{S} where the legs do not cross:

$$\mathbf{s}_{j+1} \in \mathcal{S}(\mathbf{s}_j). \quad (2.41)$$

The set-valued function $\mathcal{S}(\mathbf{s})$ specifies a halfspace around \mathbf{s} and is separated from it by a distance $w + f_s$

$$\mathcal{S}(\mathbf{s}) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 \mid (-1)^j (w + f_s) \leq \mathbf{I}_y(\mathbf{r} - \mathbf{s}) \right\}, \quad (2.42)$$

where $\mathbf{I}_y = [0 \ 1]$. Last, the **CoM** position is constrained to a closed convex region \mathcal{C} due to the maximal leg length of the robot [Brasseur 2015]:

$$\mathbf{c} \in \mathcal{C}(\mathbf{s}_j). \quad (2.43)$$

The set-valued functions $\mathcal{C}(\mathbf{s})$ specifies a rectangular region of width and length $W \times L$ around the position \mathbf{s} :

$$\mathcal{C}(\mathbf{s}) \triangleq \left\{ \mathbf{r} \in \mathbb{R}^2 \mid \begin{bmatrix} -L/2 \\ -W/2 \end{bmatrix} \leq (\mathbf{r} - \mathbf{s}) \leq \begin{bmatrix} L/2 \\ W/2 \end{bmatrix} \right\}. \quad (2.44)$$

If we can satisfy these kinematic (2.41)-(2.43) and dynamic (2.39) constraints, we comply with the whole-body kinematic and dynamical structure of the robot when walking on a flat ground.

2.3 Balance Preservation

We assume that violation of the set of constraint (2.41)-(2.43)-(2.39) indicates a fall. And the preservation of balance means avoiding falls at all future moments. So it is necessary to not only satisfy that set of constraints in the present time but also into the future. In fact, there can be a set of states where the robot has not fallen yet (all the constraints are satisfied at the present time), but is bound to fall inexorably [Wieber 2016]. Anticipation is crucial! In order to avoid falling, the robot must be in a state where it *can* avoid falling. Such states are so-called *viable*, and the set of viable states is called *viability kernel* (concepts from Viability

Theory [Aubin 2009]). The same approach can be used to formalize collision avoidance in dynamic environments, what is called *Inevitable Collision State (ICS)* [Fraichard 2003], which separate situations where a collision is inevitable (to be avoided) from situations where it is avoidable (to be preferred). In general it is difficult to check if the state of the robot is inside the viability kernel because the problem to compute this set of states appears to be intractable in the case of humanoid robots [Wieber 2002]. It is however possible to isolate a subset of these viable states, which are demonstrated to be balanced.

2.3.1 Capturability

There are 2 standard set of states: (i) the set of states for which the robot reaches a cyclic motion or (ii) the set of states for which the robot stops after a given number of steps. As a consequence, 2 standard approaches to control biped robots are (i) to consider that the robot keeps repeating indefinitely the same cyclic motion [Takenaka 2009, Tajima 2009, Nagasaka 2004, Scianca 2016] or (ii) that it stops after a given number of steps [Takanishi 1989, Hun-ok Lim 2002, Bohorquez 2016], what corresponds to *capturability* [Koolen 2012].

In this work, we favor capturability, since we almost always want the robot to eventually stop, or at least be able to stop. And, on the collision front, these states can be combined effortlessly with the ones that guarantee *passive safety* [Bouraine 2014]: if a collision is inevitable, at least the robot will be at rest when that happens. Passive safety is discussed further in Section 4.5.4.

The set of capturable states is identified analytically, for the linear system derived above (2.19), with the help of

$$\boldsymbol{\xi} = \mathbf{c} + \frac{\dot{\mathbf{c}}}{\omega}. \quad (2.45)$$

This variable is called the *Capture Point (CP)* $\boldsymbol{\xi}$. When considering the robot walking on a flat ground, the CP has the same constant CoM height and when projected on the floor it is the point where a robot has to step to come to a complete rest. Among the capturable states, we identify the ones that make the robot *0-step capturable*, *i.e.* for which the robot can stop without having to make any further step [Koolen 2012]. From now on we consider the projected position (in 2D coordinates) onto the floor of $\boldsymbol{\xi}$.

The CoM dynamics (2.16) can be transformed in a modal form, exposing its stable and unstable parts. The unstable part yields exactly the same definition as in (2.45). Reformulating (2.16) and (2.45), we have that the CP diverges from the CoP \mathbf{p} , but the CoM converges to the CP:

$$\begin{bmatrix} \dot{\mathbf{c}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} \mathbf{p}. \quad (2.46)$$

If $\boldsymbol{\xi}$ is inside the support polygon,

$$\boldsymbol{\xi} \in \mathcal{P}(\mathbf{s}_j), \quad (2.47)$$

we can have $\boldsymbol{\xi} = \mathbf{p}$ so the CP and the CoP do not move

$$\dot{\mathbf{p}} = \dot{\boldsymbol{\xi}} = \mathbf{0}, \quad (2.48)$$

and the robot comes to a stop without having to make any further step. The set of states that satisfy condition (2.47) are *0-step capturable* (hence viable).

So realizing a stable walking motion requires to take decisions that satisfy a set of constraints, and to work with predictions of the future outcome of decisions made in the

present. One of few suitable methods to handle this problem is Model Predictive Control (MPC) [Wieber 2008]. It has been used therefore extensively for the control of biped robots [Wieber 2018].

2.4 Model Predictive Control of biped walking

2.4.1 MPC Approach

Standard approaches to MPC rely on discrete-time models. In this work, we employ a linear discrete-time time-invariant model of the form

$$\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i, \quad (2.49)$$

where $\mathbf{x}_i \in \mathbb{R}^a$, $\mathbf{u}_i \in \mathbb{R}^b$ and \mathbf{x}_{i+1} are respectively the state, control and successor state vectors, and (\mathbf{A}, \mathbf{B}) are matrices of proper dimensions. An MPC scheme is an iterative planning process: it solves iteratively a finite-horizon optimization problem, until some objectives (or goals) are reached.

At time t_i , given the state \mathbf{x}_i , in this work assumed to be known and exact, MPC aims to find a motion, *i.e.* a sequence of N control inputs or N -actions

$$\boldsymbol{\pi}_i^N = \{\mathbf{u}_{(i|i)}, \mathbf{u}_{(i+1|i)}, \dots, \mathbf{u}_{(i+N-1|i)}\}. \quad (2.50)$$

The action $\mathbf{u}_{(\beta|\alpha)}$ is computed at time t_α and is planned to be executed at time t_β and kept constant for a duration T . The motion is valid for N time instants (from t_i to t_{i+N}) and it has a duration NT , where N is called *planning horizon*^[5]. The motion is called *feasible* when the pair $(\mathbf{x}_i, \boldsymbol{\pi}_i^N)$ satisfies a time-varying set of constraints, in this work *linear*, applied for $[t_i, t_{i+N}]$:

$$\mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \leq \mathbf{d}_i, \quad (2.51)$$

where \mathbf{E}_i and \mathbf{d}_i are respectively a time-varying matrix and vector of proper dimensions. An MPC scheme classically chooses a sequence $\boldsymbol{\pi}_i^N$ that minimizes the deviation from a given objective $(\mathbf{u}_{\text{ref}}, \mathbf{x}_{\text{ref}})$:

$$\underset{\boldsymbol{\pi}_i^N}{\text{minimize}} \quad f(\boldsymbol{\pi}_i^N). \quad (2.52)$$

Typically, the reference deviation is expressed as a least-squares objective, for example,

$$f(\boldsymbol{\pi}_i^N) = \sum_{j=1}^N (\|\boldsymbol{\Gamma}_u(\mathbf{u}_{(i+j-1|i)} - \mathbf{u}_{\text{ref}})\|_2^2 + \|\boldsymbol{\Gamma}_x(\mathbf{x}_{(i+j|i)} - \mathbf{x}_{\text{ref}})\|_2^2), \quad (2.53)$$

where $\boldsymbol{\Gamma}_u$ and $\boldsymbol{\Gamma}_x$ are weighting matrices. Constraints (2.51) and reference deviation (2.52) are satisfied and minimized altogether, solving the following Quadratic Program (QP) [Nocedal 2006, Boyd 2004]:

$$\underset{\boldsymbol{\pi}_i^N}{\text{minimize}} \quad f(\boldsymbol{\pi}_i^N) \quad (2.54a)$$

$$\text{s.t.} \quad \mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \leq \mathbf{d}_i, \quad (2.54b)$$

⁵ N is typically called *prediction horizon* in the Model Predictive Control literature. In general however, $\boldsymbol{\pi}$ and $\boldsymbol{\varphi}$ can have different horizon lengths: control and prediction horizon lengths [Kerrigan 2001].

which can be solved with off-the-shelf software, *e.g.* qpOASES [Ferreau 2017]. When we account for bounded disturbances also the state \mathbf{x}_i can become a decision variable when solving (2.54) [Mayne 2005]. The first action $\boldsymbol{\kappa}_1 = \mathbf{u}_{(i|i)}$ is executed, *i.e.* applied to the system:

$$\mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\boldsymbol{\kappa}_1. \quad (2.55)$$

Then at time t_{i+1} the whole planning process is repeated, until the objective is reached.

Remark 1. *Given an horizon N , we do not account the time for computing a plan solving the QP (2.54), and we assume no time delay to execute the control action.*

2.4.2 Walking motion generation with automatic footstep placement

We use an MPC scheme that generates a walking motion online with automatic footstep placement [Herdt 2010]. A motion $\boldsymbol{\pi}_i^N$ is composed by a sequence of CoP input velocities and a sequence of adjustable footsteps.

$$\boldsymbol{\pi}_i^N = \begin{bmatrix} \bar{\mathbf{p}}_i \\ \mathbf{s}_i^f \end{bmatrix} \in \mathbb{R}^{2N+2m} \quad (2.56)$$

The CoM, CoP trajectories (2.27)-(2.31) and the model the evolution of a walking cycle (2.33) are used to formulate the following constraints that we want the motion to satisfy:

$$\forall k \in \{1, \dots, N\}, \begin{cases} \mathbf{p}_{(i+k|i)} \in \mathcal{P}(\mathbf{s}_{[j|i]}), & \text{Dynamical constraint} \\ \mathbf{c}_{(i+k|i)} \in \mathcal{C}(\mathbf{s}_{[j|i]}), & \text{Kinematic constraints.} \\ \mathbf{s}_{[j+1|i]} \in \mathcal{S}(\mathbf{s}_{[j|i]}) \end{cases} \quad (2.57)$$

Furthermore, we make sure the maximum length of the legs is respected between two samples: during the DS with

$$\begin{cases} \mathbf{c}_{(i+q|i)} \in \mathcal{C}(\mathbf{s}_{[j+1|i]}), \\ \mathbf{c}_{(i+q+1|i)} \in \mathcal{C}(\mathbf{s}_{[j|i]}) \end{cases} \quad \text{DS Kinematic constraints,} \quad (2.58)$$

where t_{i+q} is the instant the robot plans to start the DS phase, and t_{i+q+1} is the instant the robot plans to end the DS phase. The constraints (2.57)-(2.58):

- are *linear* with respect to the CoP velocity $\dot{\mathbf{p}}$ and the adjustable steps \mathbf{s}^f [Sherikov 2016], thanks to the simplification made for the step automatic adjustment in Section 2.2.3. And they can be formulated in the linear form (2.51).
- define all together a *closed* convex set of solutions since they impose closed convex regions: for the CoP position with (2.40), limiting its velocity, for the CoM position with (2.44)-(2.58), limiting its velocity and acceleration, and for the foot position since the halfspace region (2.42) is combined with the CoP and CoM regions.
- vary *cyclically* because of the time-varying model of the walking cycle (2.33), with a fixed period: the fixed step cycle duration.

The action $\boldsymbol{\kappa}_1$ of the motion (2.56) is the first constant CoP velocity input that updates the system (2.23). And, cyclically, the first future footstep position $\mathbf{s}^f(1)$ updates \mathbf{s}^{pc} as explained in Appendix A.4.

In MPC, the capacity to satisfy constraints, which is usually called *feasibility*, is classically guaranteed *recursively* [Mayne 2000]. An important aspect of our application is that the MPC scheme has to adapt continuously to the dynamic environment of the robot: collision avoidance [Bohorquez 2016], physical interaction with humans [Agravante 2016], or visual feedback [Dune 2011]. We aim therefore at guaranteeing recursive feasibility for all possible scenarios, which is called strong recursive feasibility.

2.4.3 Strong Recursive Feasibility

Let the set of solutions of (2.51) be

$$\mathcal{F}_i^N \triangleq \{(\mathbf{x}_i, \boldsymbol{\pi}_i^N) \mid \mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \leq \mathbf{d}_i\}. \quad (2.59)$$

Two useful projections of \mathcal{F}_i^N are

$$\mathcal{K}_i^{1|N} \triangleq \{(\mathbf{x}_i, \boldsymbol{\kappa}_1) \mid \exists \boldsymbol{\pi}_i^N, (\mathbf{x}_i, \boldsymbol{\pi}_i^N) \in \mathcal{F}_i^N, \boldsymbol{\kappa}_1 = \mathbf{u}_{(i|i)}\}, \quad (2.60)$$

the set of state-action pairs of a feasible motion, and

$$\mathcal{X}_i^N \triangleq \{\mathbf{x}_i \mid \exists \boldsymbol{\pi}_i^N, (\mathbf{x}_i, \boldsymbol{\pi}_i^N) \in \mathcal{F}_i^N\}, \quad (2.61)$$

generally called the *feasible set*, the set of states for which there exists at least one feasible motion.

Strong Recursive Feasibility (SRF) [Kerrigan 2001] guarantees that applying the first action of *any* pairs in $\mathcal{K}_i^{1|N}$ to the system (2.49), a state in \mathcal{X}_{i+1}^N is reached, *independently* from the minimization of *any* objective function (2.52). We define SRF in the following way:

Definition 2.4.1 (Strong Recursive Feasibility). *An MPC scheme is strongly recursive feasible if and only if*

$$\begin{aligned} \forall i, \forall (\mathbf{x}_i, \boldsymbol{\kappa}_1) \in \mathcal{K}_i^{1|N}, \\ \mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\boldsymbol{\kappa}_1 \in \mathcal{X}_{i+1}^N. \end{aligned} \quad (2.62)$$

Remark 2. SRF is “stronger” than the standard Recursive Feasibility since the latter depends on a optimal solution of a specific cost function (2.52) [Kerrigan 2001, Section 5.3]. Changing the cost function at any t_i and/or not reaching an optimal solution do not affect SRF property.

Remark 3. The relationship between SRF and viability is the following: SRF guarantees that $\forall i$, at time t_i all feasible states are viable. In case of balance preservation for biped robots, no feasible state can lead to a fall of the robot.

Strong recursive feasibility is classically obtained by introducing a *terminal constraint* at the end of the planning horizon [Kerrigan 2001, Section 5.7.2] to make sure the system remains feasible indefinitely after the end of the horizon.

2.4.4 Capturability terminal constraint

We favored the capability of the robot to stop, being capturable, as a sufficient condition to maintain balance without violating the constraints (2.57)-(2.58). Only an additional constraint on the final state $\mathbf{x}_{(i+N|i)}$ formalizes conditions under which the robot can be stopped. This condition on the final state acts as a terminal constraint of the MPC scheme. Once reached the state $\mathbf{x}_{(i+N|i)}$, a simple controller can stabilize the system after the horizon N [Mayne 2014].

Since we want the robot to stop at the end of the planning horizon, we construct a controller in such a way that the CoP will be maintained at a constant position: $\mathbf{p}_{(i+N|i)} = \mathbf{p}_{(i+N+j|i)}$, $\forall j \in \mathbb{N}^+$. Position of the CoP for the dynamics (2.23) can always be maintained with a trivial linear controller: $\dot{\mathbf{p}}_i = \mathbf{0}$. It is however necessary to analyze the stability of the system. We know that the system has an unstable mode [Englsberger 2011].

We will nullify this mode rendering the system stable [Muske 1993]. We identify the eigenvalues of the matrix \mathbf{A} in (2.25) via the matrix factorization called eigendecomposition:

$$(1, e^{T\omega}, e^{-T\omega}). \quad (2.63)$$

The matrix \mathbf{A} has one unstable eigenvalue: $e^{T\omega}$, which is outside of the unit circle since $T\omega > 0$. Its corresponding mode is $\mathbf{m}_u = \dot{\mathbf{c}}_i + \ddot{\mathbf{c}}_i \omega^{-1}$. Nullifying \mathbf{m}_u means to impose the following equality constraint

$$\dot{\mathbf{c}}_i + \ddot{\mathbf{c}}_i \omega^{-1} = \mathbf{0}. \quad (2.64)$$

If this equation holds, the state $[\mathbf{c}_{i+j}, \dot{\mathbf{c}}_{i+j}, \ddot{\mathbf{c}}_{i+j}]$ of the *autonomous* system (because $\dot{\mathbf{p}}_i = 0$) converges to $[\mathbf{p}_i, \mathbf{0}, \mathbf{0}]$ as $j \rightarrow \infty$ while the **CoP** does not move. Notice that the unstable mode corresponds to the derivative of the **CP** ξ , see Section 2.3.1, so we are actually imposing

$$\dot{\xi}_i = \mathbf{0}. \quad (2.65)$$

When the **CP** is not moving (2.65), the projected position onto the floor of **CP** coincide with the position of the **CoP**, see the second line of the system (2.46). This it implies the **CP** is above the support polygon, defined with (2.57), and the robot can stop without making further steps: it is *0-step capturable*. From this analysis, we impose

$$\dot{\xi}_{(i+N|i)} \in \mathbb{T}, \quad (2.66)$$

where

$$\mathbb{T} \triangleq \{\mathbf{r} \in \mathbb{R}^2 \mid \mathbf{r} = \mathbf{0}\}, \quad (2.67)$$

as terminal constraint, making sure that after the end of the horizon, a simple controller can stabilize the system (2.49). From now if we include a terminal constraint in (2.51) that defines a terminal set \mathbb{T} , the solution, state-action pairs and feasible sets are represented by: $\mathcal{F}_i^N(\mathbb{T})$, $\mathcal{K}_i^{1|N}(\mathbb{T})$ and $\mathcal{X}_i^N(\mathbb{T})$.

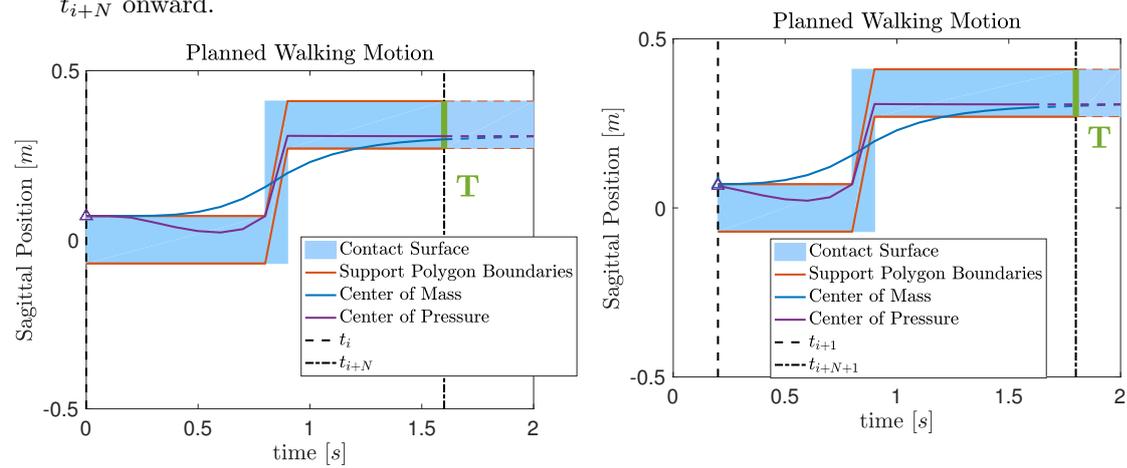
At time t_i , suppose the model of the walking cycle (2.33), starting from the end of the horizon N is assumed to be time-invariant, which implies, in particular, that the number and positions of the contacts do not change, represented in Figure 2.5. In this case, after t_{i+N} , the state can remain feasible in \mathbb{T} since it is a subset of all possible feasible states (defined in the footstep where the robot will stop). At time t_{i+1} , **SRF** is guarantee by *construction* and the **MPC** is strongly feasible. The problem is when at time t_{i+1} the robot is not really planning to stop, and considers actually making a new step at the end of the planning horizon. With such a sudden change, **SRF** is not guaranteed anymore. We address this issue in Chapter 3.

2.5 Conclusion

We presented a linear model that relates the **CoM** of the robot to the contact forces applied on the ground (2.16). We model changes of the foot positions on the ground in order to facilitate their automatic adjustment (2.33). With those models we build a set of constraints that the robot needs to satisfy in order to comply with the whole-body kinematic and dynamical structure. We additionally imposed a capturability constraint that allows the robot to stop in a given number of steps since we almost always want the robot to eventually stop. Models and constraints were employed in a single **MPC** scheme.

Figure 2.5: Representation of the walking cycle model (2.33) when starting from the end of the horizon N is assumed to be time-invariant, which implies, in particular, that the number and positions of the contacts do not change.

- (a) At time t_i it is assumed the number and positions of the contacts do not change from time t_{i+N} onward. (b) The assumption made at time t_i is valid.



Chapter 3

Strong Recursive Feasibility in MPC of Biped Walking

3.1 Introduction

Chapter 2 presented the *Model Predictive Control* (MPC) of biped walking that generates walking motion with automatic footstep placement. We favored the capability of the robot to stop, being capturable, as a sufficient condition to maintain balance. This was formalized in a capturability terminal constraint (imposed at the end of the planning horizon). This constraint works also as a sufficient condition to guarantee strong recursive feasibility (Definition 2.4.1) for the control scheme. However, it is explained in Section 3.2 how strong recursive feasibility guarantee is lost when the robot is not really planning to stop but considers actually making a new step. In this Chapter we demonstrate numerically (Section 3.5) that strong recursive feasibility is actually guaranteed, even when a new step is added in the planning horizon. But the guarantee depends on the length of the horizon (Section 3.5). On the other hand, if we remove the capturability terminal constraint we lose the robot's ability to stop and we lose completely strong recursive feasibility for several horizon lengths (Section 3.6). Finally, we discuss the ongoing research on the choice of the capturability constraint and recursive feasibility in the literature of control for biped walking (Section 3.7).

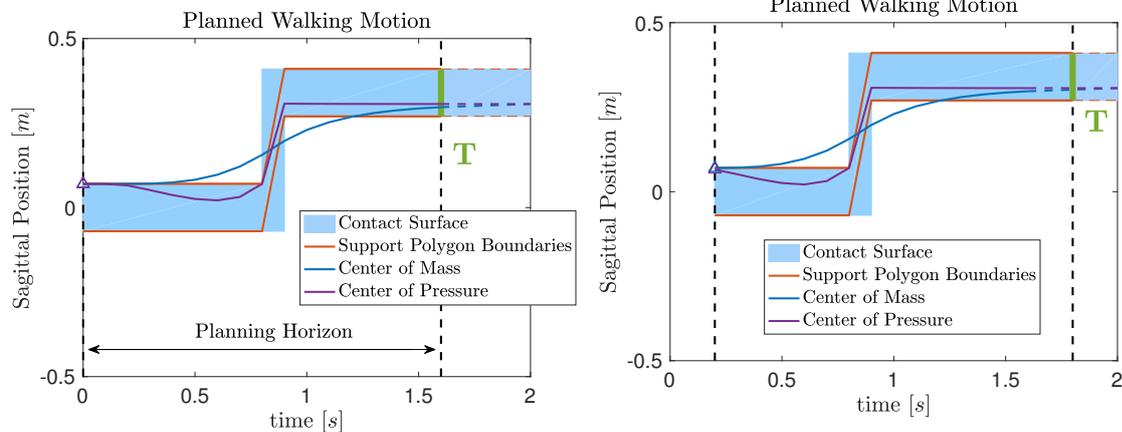
3.2 Problem formulation

We can see in Fig. 3.1a how a capturability terminal constraint makes sure that the system remains feasible indefinitely. This way, when the planning horizon advances as in Fig. 3.1b, we are sure that the MPC scheme remains feasible: SRF is guaranteed by construction. In this case the robot can keep planning to stop until it really stops, and then re-start walking. In such a way, the robot would walk for few steps and come to a stop periodically. Since in our application the robot needs to adapt continuously to the dynamic environment, *e.g.* avoid collision or physical interaction, we want the robot to stop only in case of necessity and continue walking otherwise. When the robot is walking and is not really planning to stop but considers actually making a new step, with such a sudden plan change, SRF is not guaranteed anymore, as shown in Fig. 3.1c. Note that this issue occurs *independently* of the planning horizon length N .

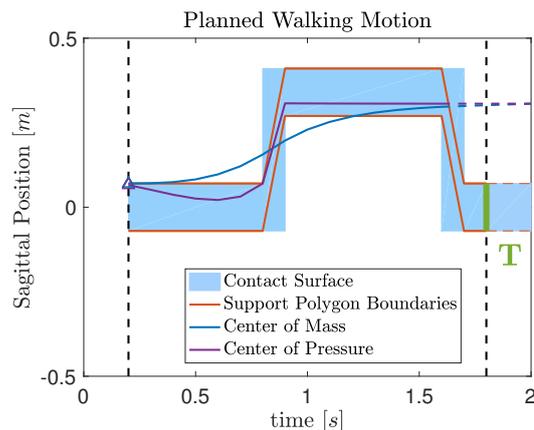
We investigate this issue and we provide a numerical evidence that SRF is actually guaranteed, even when a new step is added in the planning horizon. We found that in this case, the SRF guarantee depends on the horizon length N . Since this issue is introduced by the capturability terminal constraint, we remove it and we investigate again SRF. In this case, the MPC scheme loses SRF guarantee for all horizon lengths N tested.

Figure 3.1: The figures represent when the robot is walking and is planning to stop (Figure 3.1a-3.1b), and when the robot considers actually making a new step (Figure 3.1c).

- (a) A capturability terminal constraint (that defines a terminal constraint set T) makes sure that the motion of the biped robot remains feasible indefinitely. (b) When the planning horizon advances, thanks to the fixed terminal constraint set T , the MPC scheme remains feasible.



- (c) When the robot considers making a new step, with such a sudden change, recursive feasibility is not guaranteed anymore.



3.3 Parameters

3.3.1 Robot

The parameters of the biped robot were selected according to the kinematics of the robot HRP-2 [Kaneko 2004], see Table 3.1.

3.3.2 MPC

In MPC-based walking strategies such as [Bohorquez 2016, Kajita 2003, Herdt 2010, Scianca 2016, Herdt 2010] or [Naveau 2017], the sampling period T is usually small. In our framework, the sampling period is set to 0.1[s], and it corresponds to the duration of the DS phase. The duration of the SS phase is set to 0.7[s], resulting in a step cycle of 0.8[s/step]. These choices proved to realize stable walking motions [Dune 2011].

Table 3.1: Robot parameters.

Parameter	Symbol	Value	Unit
Height of CoM (2.17)	c^z	0.8	m
Min. Feet Separation (2.42)	f_s	0.07	m
Feet Dimensions (2.40)	(l, w)	(0.24, 0.14)	(m,m)
Step Length (2.44)	(L, W)	(0.24, 0.30)	m

The number of planned steps ahead is chosen accordingly to $\lceil (NT/s_d) \rceil$, that depends on the planning horizon, sampling period and step cycle. When $N \geq 1$, the number of adjustable future steps is $m = \lceil ((N-1)T/s_d) \rceil$: details in Appendix A.4. In our case, when $N = 8$ the robot is always planning a full step cycle, and when $N = 16$ the robot is always planning a full gait cycle. The parameters are summarized in Table 3.2.

Table 3.2: MPC scheme parameters.

Parameter	Symbol	Value	Unit
Planning Period	N	{8, 9, ..., 17, 18}	-
Sampling Period	T	0.1	s
Step Cycle	s_d	0.8	s/step
Planned Steps Ahead	-	$\lceil (NT/s_d) \rceil$	step

3.4 Numerical approach

The sets $\mathcal{F}_i^N(\mathbb{T})$, $\mathcal{K}_i^{1|N}(\mathbb{T})$ (showed in Figure 3.2a) and $\mathcal{X}_i^N(\mathbb{T})$ are convex polytopes [Boyd 2004] following definitions in Section 2.4.3. And in our case they also happen to be closed, as explained in Section 2.4.2. Thanks to these properties, it is *sufficient* to check SRF (2.62) only on the vertices of $\mathcal{K}_i^{1|N}(\mathbb{T})$, as can be seen on Fig. 3.2b. The number of vertices of these polytopes is finite, but enumerating all of them from definition (2.59) and (2.60) is actually an *NP-hard* problem [Borwein 1987] although various algorithms exist for this enumeration, *e.g.* [Löfberg 2012, Fukuda 1996, Jones 2004]. In this work we are going to use a randomized shooting approach.

Numerical evidence. The method we propose is based on the observation that the simplex method for Linear Programming (LP) always terminates on a so-called basic solution, which is actually a *vertex* [Bertsimas 1997]. We propose therefore to solve LPs of the form

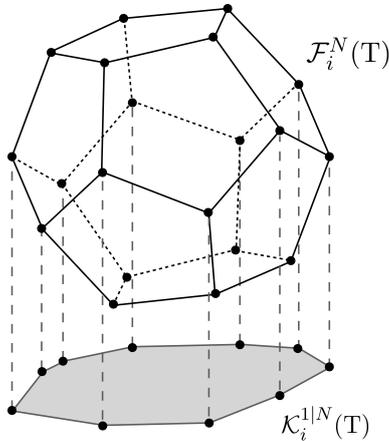
$$\underset{\mathbf{x}_i, \boldsymbol{\pi}_i^N}{\text{minimize}} \quad \boldsymbol{\gamma}^T \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \quad (3.1a)$$

$$\text{s.t.} \quad \mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \leq \mathbf{d}_i, \quad (3.1b)$$

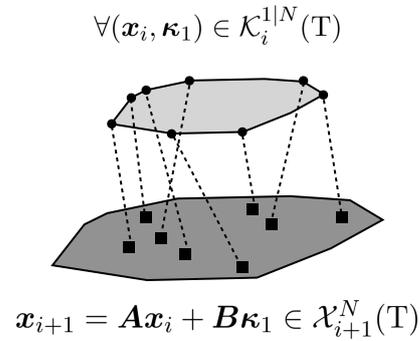
for randomly chosen directions $\boldsymbol{\gamma} \in \mathbb{R}^{6+2N+2m}$, will provide a random selection of vertices of $\mathcal{F}_i^N(\mathbb{T})$. Instead of projecting each vertex $\mathcal{F}_i^N(\mathbb{T})$ onto $\mathcal{K}_i^{1|N}(\mathbb{T})$, we decided to choose

Figure 3.2: Representations of the closed convex polytopes $\mathcal{F}_i^N(\mathbf{T})$ and $\mathcal{K}_i^{1|N}(\mathbf{T})$, and **SRF** definition with time-varying constraints.

(a) The polytope $\mathcal{F}_i^N(\mathbf{T})$, defined in (2.59), and its projection $\mathcal{K}_i^{1|N}(\mathbf{T})$ defined in (2.60).



(b) **SRF**, Definition 2.4.1 with time-varying constraints.



randomly the entries of γ that multiply $(\mathbf{x}_i, \boldsymbol{\kappa}_1)$ and set to *zero* the others. And **SRF** is checked simply verifying, with each vertex found, that the following **LP**

$$\underset{\boldsymbol{\pi}_{i+1}^N}{\text{minimize}} \quad \boldsymbol{\psi}^T \boldsymbol{\pi}_{i+1}^N \quad (3.2a)$$

$$\text{s.t.} \quad \mathbf{E}_{i+1} \begin{bmatrix} \mathbf{A}\mathbf{x}_i + \mathbf{B}\boldsymbol{\kappa}_1 \\ \boldsymbol{\pi}_{i+1}^N \end{bmatrix} \leq \mathbf{d}_{i+1}, \quad (3.2b)$$

has a solution for any direction $\boldsymbol{\psi} \in \mathbb{R}^{2N+2m}$, chosen arbitrarily. This is a randomized approach (random shooting) which cannot guarantee to enumerate all the vertices, so it can only provide a *numerical evidence*.

3.5 SRF with the capturability terminal constraint

In our **MPC** framework, when the robot wants to continue walking, a new step appears at the end of the planning horizon *cyclically*. It is sufficient to check **SRF** not for all time, $\forall i$, but only for that moment. We indicate the time before and after the appearance of a new step at the end of the planning horizon, t_k and t_{k+1} , respectively in Fig. 3.1a and in Fig. 3.1c. We investigate the **SRF** guarantee for several planning horizons lengths, N . Depending on N , there are 3 possible walking moments from time t_k to t_{k+1} :

1. At time t_k the robot starts the **DS** phase, and t_{k+1} the robot ends the **DS** phase.
2. At time t_k the robot is in **SS** phase, and at time t_{k+1} the robot starts the **DS** phase.
3. For both time t_k and t_{k+1} , the robot is in **SS** phase.

These cases are represented in Figure 3.3. In the MPC scheme, at time t_k , the 3 different projections $\mathcal{K}_k^{1|N}(\mathbb{T})$ are:

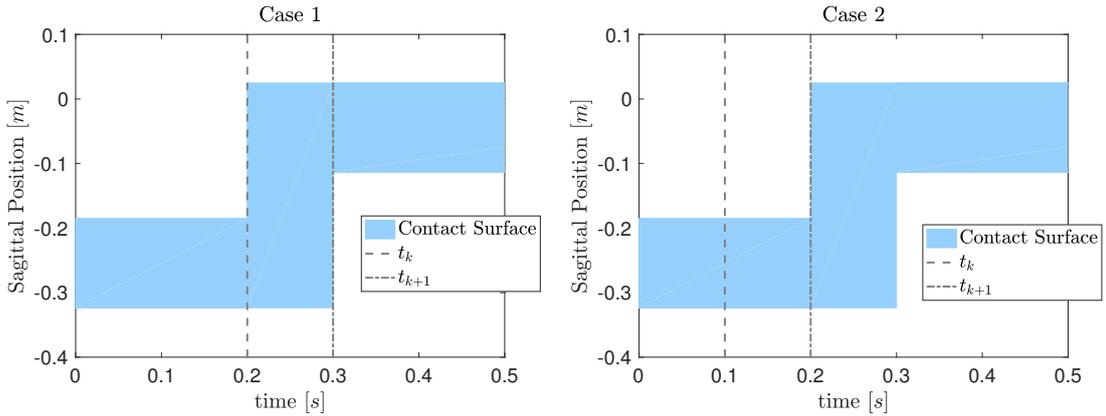
1. $t_k = t_{sDS}$, where $\mathcal{K}_{sDS}^{1|N} \triangleq \{(\mathbf{x}, \boldsymbol{\kappa}_1)\} \in \mathbb{R}^{10}$, $\boldsymbol{\kappa}_1 = [\dot{\mathbf{p}}_{(k|k)}, \mathbf{s}_k^{pc\top}]^\top$.
The foot position \mathbf{s}_k^c is fixed but we explore all the possible DS phases the robot can start, using \mathbf{s}_k^{pc} as decision variable.
2. $t_k = t_{eSS}$, where $\mathcal{K}_{eSS}^{1|N} \triangleq \{(\mathbf{x}, \boldsymbol{\kappa}_1)\} \in \mathbb{R}^{10}$, $\boldsymbol{\kappa}_1 = [\dot{\mathbf{p}}_{(k|k)}, \mathbf{s}_k^f(1)]^\top$.
We have $\mathbf{s}_k^c = \mathbf{s}_k^{pc}$ and the first future adjustable step $\mathbf{s}_k^f(1)$ defines all the possible DS phases the robot can start at time t_{k+1} .
3. $t_k = t_{SS}$, where $\mathcal{K}_{SS}^{1|N} \triangleq \{(\mathbf{x}, \boldsymbol{\kappa}_1)\} \in \mathbb{R}^8$, $\boldsymbol{\kappa}_1 = \dot{\mathbf{p}}_{(k|k)}$.
We have $\mathbf{s}_k^c = \mathbf{s}_k^{pc}$.

At time t_{k+1} :

1. \mathbf{s}^c is updated with \mathbf{s}^{pc} : $\mathbf{s}_{k+1}^c = \mathbf{s}_k^{pc}$
2. \mathbf{s}^{pc} is updated with $\mathbf{s}^f(1)$: $\mathbf{s}_{k+1}^{pc} = \mathbf{s}_k^f(1)$
3. $\mathbf{s}_{k+1}^{pc} = \mathbf{s}_k^{pc}$ and $\mathbf{s}_{k+1}^c = \mathbf{s}_k^c$.

Figure 3.3: Representation of the 3 possible walking moments at time t_k and t_{k+1} .

- (a) At time t_k the robot starts the DS phase, (b) At time t_k the robot is in SS phase, and at t_{k+1} the robot ends the DS phase. (c) For both time t_k and t_{k+1} , the robot is in SS phase.



- (c) For both time t_k and t_{k+1} , the robot is in SS phase.

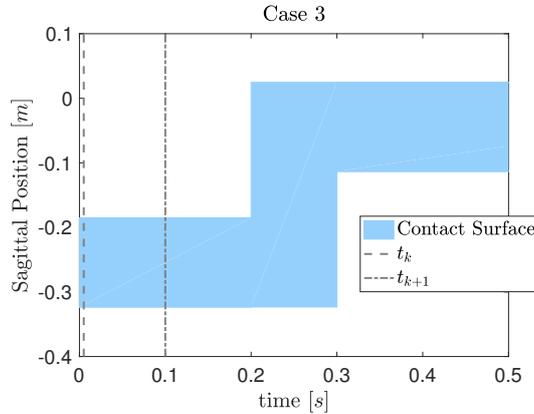


Table 3.3: Numerical results from randomly selected vertices of SRF in MPC of biped walking.

(a) Numerical results with the capturability terminal constraint.

N	Steps ahead	Set $\mathcal{F}_k^N(\mathbb{T})$		Projection $\mathcal{K}_k^{1 N}(\mathbb{T})$			Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$		SRF
		cstr	dim $\mathbb{R}^{(\cdot)}$	$\mathcal{K}_{sDS}^{1 N}$	$\mathcal{K}_{SS}^{1 N}$	$\mathcal{K}_{eSS}^{1 N}$	$\exists \boldsymbol{\pi}_{k+1}^N$	$\nexists \boldsymbol{\pi}_{k+1}^N$	
8	1	90	26	x			523	522	No
9	2	94	26			x	1074	753	No
10		110	30		x		470	205	No
11		118	32		x		398	195	No
12		126	34		x		338	149	No
13		134	36		x		358	113	No
14		142	38		x		374	0	Yes
15		150	40		x		390	0	Yes
16		166	44	x			1246	0	Yes
17	3	170	44			x	3465	0	Yes
18		186	48		x		2123	0	Yes

(b) Vertices verification with the capturability terminal constraint.

N	Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\exists \boldsymbol{\pi}_{k+1}^N$		Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\nexists \boldsymbol{\pi}_{k+1}^N$	
	Original	Verified	Original	Verified
8	523	363	522	362
9	1074	1074	753	753
10	470	316	205	134
11	398	380	195	186
12	338	196	149	80
13	358	329	113	66
14	374	165	0	0
15	390	285	0	0
16	1246	495	0	0
17	3465	3461	0	0
18	2123	638	0	0

(c) Numerical results without the capturability terminal constraint.

N	Steps ahead	Set \mathcal{F}_k^N		Projection $\mathcal{K}_k^{1 N}$			Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$		SRF
		cstr.	dim $\mathbb{R}^{(\cdot)}$	$\mathcal{K}_{sDS}^{1 N}$	$\mathcal{K}_{SS}^{1 N}$	$\mathcal{K}_{eSS}^{1 N}$	$\exists \boldsymbol{\pi}_{k+1}^N$	$\nexists \boldsymbol{\pi}_{k+1}^N$	
14	2	140	38		x		36	220	No
15		148	40		x		50	219	No
16	3	164	44	x			5	59	No
17		168	44			x	0	826	No
18		184	48		x		5	199	No

3.5.1 Numerical evidence from randomly selected vertices

At time t_k , three million vertices were found with (3.1) by choosing three million random directions γ . Within this set of vertices, many of them were duplicates, thus removed. There is no strong reason behind the choice of three million since it is a randomized shooting approach and the search of all vertices can never be said <concluded>. So we chose a reasonably high number. Successively, the LP (3.2) was used to check a feasible solution for each vertex at time t_{k+1} . This process was repeated for several horizon lengths $N \in \{8, 9, \dots, 17, 18\}$. MATLAB R2016a was used to run the linear programming problems (3.1) and (3.2) with the `linprog` function (simplex method). The results of this numerical approach are summarized in Table 3.3a.

The system can become infeasible when the robot is not planning to stop and consider to make a new step in the near future, *e.g.* when planning a full step cycle ($N = 8$). As the horizon length increases, the robot can plan 2 steps ahead: $N \in \{9, \dots, 12, 13\}$, but we still do not guarantee the feasibility of the system when the robot considers to make a new step. We obtained a numerical evidence that the system remains feasible when the robot plans with: $N \in \{14, 15\}$, when the robot plans a full gait cycle $N = 16$ and when the robot starts planning 3 steps ahead: $N \in \{17, 18\}$.

$\mathcal{F}_k(\mathbb{T})$ dimension As a side note, when passing from 1 to 2 steps ahead (or from 2 to 3 steps ahead), the dimension of $\mathcal{F}_k(\mathbb{T})$ does not increase. When $N = 8$ (1 step ahead), the decision variables are: the state of the robot \mathbf{x} , the control sequence $\boldsymbol{\pi}^N$ and \mathbf{s}^{pc} . When passing to $N = 9$ (2 steps ahead), the planning horizon is increased, but the number of adjustable steps m does not, and we lose the decision variable \mathbf{s}^{pc} . So the dimension of $\mathcal{F}_k(\mathbb{T})$ remains unchanged.

3.5.2 Vertices verification

In theory, the simplex method for LP always terminates on a vertex. In practice, we want to verify that the LP always terminated on a vertex. We want to understand how many solutions are actually vertices.

Each vertex is *verified* with the following procedure. Suppose you have a point \mathbf{v} that belongs to a polytope of dimension n , with inequality and equality constraints. An inequality constraint that becomes an equality constraint for that point \mathbf{v} is said to be *active* (equality constraints are always active). If there are at least n *linearly independent* constraints that are active for \mathbf{v} , \mathbf{v} is a vertex.

Take the set of vertices \mathcal{V} found with the numerical evidence, Section 3.5.1. Each vertex $\mathbf{v} \in \mathcal{V}$ has dimension D : dimension of $\mathcal{F}_k^N(\mathbb{T})$. From the set of constraints (3.1b), we do

$$\mathbf{w} = \mathbf{E}_k \mathbf{v} - \mathbf{d}_k \quad (3.3)$$

and we look for the number of zero (absolute value is smaller than 10^{-8}) elements of \mathbf{w} . These elements identify a set of linear active constraints from (3.3):

$$\mathbf{A}_{\text{cstr}} \mathbf{v} = \mathbf{b}_{\text{cstr}}. \quad (3.4)$$

If there are at least D linearly independent active constraints:

$$\text{rank}(\mathbf{A}_{\text{cstr}}) = D, \quad (3.5)$$

where `rank` performs a singular value decomposition and returns the number of singular values of \mathbf{A}_{cstr} that are larger than 10^{-11} , then we confirm \mathbf{v} as a vertex.

We repeated this process for each N and we summarized the results in Table 3.3b. For each N we found at least a vertex that did not pass the test. When $N = 18$ for example, up to 70% of the vertices did not pass the test. However, after this verification, we have the same binary answers: **SRF** is guaranteed for the planning horizon lengths $N \in \{14, \dots, 18\}$.

3.6 SRF without the capturability terminal constraint

Without the capturability terminal constraint, the system is no longer feasible indefinitely by construction. And, when the planning horizon advances, even when a new step is *not* added in the planning horizon (Fig. 3.1b), we do not know if the system remains feasible. With sufficiently long planning horizon N , we have implicitly the equivalent of a terminal constraint [Boccia 2014, Grüne 2012], but we can't know how long is "sufficiently long" without testing numerically.

We used the same numerical test as in Section 3.5.1 (with the vertex verification in Section 3.5.2). This time we only chose 10.000 random directions for γ . We obtained a numerical proof that **SRF** is not guarantee for the set of horizon lengths $N \in \{14, \dots, 17, 18\}$ without the capturability terminal constraint, see the results in Table 3.3c. The horizon length $N = 18$ is yet not "sufficiently long" to guarantee **SRF**. Hereby, we list several important aspects of this result:

- it is sufficient to check **SRF** not for all time, $\forall i$, but only for the step cycle (even when a new step is *not* added in the planning horizon). Since we found at least 1 vertex that was not a feasible solution at time t_{k+1} (when a new step is added in the planning horizon), we obtained a numerical proof that **SRF** is not guaranteed. Thus, we do not need to run the test for all other times of the step cycle.
- we did not need to run the test for the planning horizon lengths $N \in \{8, 9, \dots, 12, 13\}$. Because the polytope \mathcal{F}_k^N without the terminal constraint contains the polytope with the terminal constraint

$$\mathcal{F}_k^N(\mathbb{T}) \subseteq \mathcal{F}_k^N, \quad (3.6)$$

and with both our approaches, we have already a numerical proof that **SRF** is not guaranteed for those horizon lengths N , see Tables 3.3a.

3.7 Discussion

Hereby, we discuss on few aspects of this contribution, and possible future directions.

3.7.1 Providing a numerical proof

The Multi-Parametric Toolbox (MPT) [Herceg 2013] uses algorithms found in the literature to enumerate vertices. The vertex enumeration can be delegated to an external CDD solver, wrapped in the toolbox that use the double description method [Fukuda 1996]. We tested this toolbox on our polytopes, but they have a very large dimension and vertex enumeration became complicated. In Table 3.4a we report the results from the toolbox. We verified the results as it was done in Section 3.5.2 (Table 3.4b). And in Table 3.4c we compared the results found with our approach and the toolbox. Sometimes the total number of vertices found by the toolbox were less than the vertices found by our numerical approach, and for some horizon length the toolbox gave us completely different results. For example when $N = 10$, the toolbox did not find any vertices for which $\exists \pi_{k+1}^N$. The toolbox could not compute a projection for $N > 15$. From this toolbox however we used `.isBounded` function to verify that all the polytopes were actually closed: for all N tested (with and without terminal constraint), we obtained a positive answer.

Preliminary results with MPT We represent the polytope \mathcal{F}_i^N by half-plane description^[1]. We project it onto $\mathcal{K}_i^{1|N}$ and we enumerate the vertices. The projection is done using the Fourier-Motzkin Elimination: it is numerically robust for polytopes of large dimension, and for enumerating vertices we use the double description method [Fukuda 1996]. Each vertex is used to check SRF by simply verifying that the LP (3.2) has a solution (at time t_{i+1}). MPT was used to build polytopes, project them and enumerate the vertices of the projection, see Algorithm 1. The vertex enumeration is done by the external CDD solver, wrapped in the toolbox. MPT tries to enumerate all vertices, providing *numerical proof*.

Algorithm 1 Vertices enumeration with MPT

- 1: $\mathcal{F}_i^N = \text{polyhedron}(\mathbf{E}_i, \mathbf{d}_i)$
 - 2: $\mathcal{K}_i^{1|N} = (\mathcal{F}_i^N).\text{projection}(\text{dim}, \text{'ifourier'})$;
 - 3: $(\mathbf{x}_i, \boldsymbol{\kappa}_1) = (\mathcal{K}_i^{1|N}).\text{computeVRep}()$;
-

¹A polytope \mathcal{P} is represented in half-plane description as: $\mathcal{P} = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{A}_e\mathbf{x} = \mathbf{b}_e\}$.

Table 3.4: Preliminary results with MPT-3 of **SRF** and comparison with our numerical approach.

(a) Numerical results from MPT-3 toolbox of **SRF**.

N	Set \mathcal{F}_k^N		Projection $\mathcal{K}_k^{1 N}$		Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$		SRF
	cstr	dim $\mathbb{R}^{(\cdot)}$	cstr	dim $\mathbb{R}^{(\cdot)}$	$\exists \boldsymbol{\pi}_{k+1}^N$	$\nexists \boldsymbol{\pi}_{k+1}^N$	
8	90	26	22	10	768	768	No
9	94	26	25	10	1536	1024	No
10	110	30	17	8	0	224	No
11	118	32	24	8	512	256	No
12	126	34	22	8	384	192	No
13	134	36	21	8	384	96	No
14	142	38	21	8	480	0	Yes
15	150	40	21	8	320	0	Yes

(b) Vertices verification from MPT-3.

N	Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\exists \boldsymbol{\pi}_{k+1}^N$		Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\nexists \boldsymbol{\pi}_{k+1}^N$	
	Original	Verified	Original	Verified
8	768	768	768	768
9	1536	1536	1024	1024
10	0	0	224	0
11	512	512	256	256
12	384	384	192	192
13	384	384	96	96
14	480	480	0	0
15	320	320	0	0

(c) The verified vertices \mathbf{v} from numerical evidence (NE) and MPT are compared. A vertex is identical (or ID) if the absolute value of each element in $\mathbf{v}_{\text{NE}} - \mathbf{v}_{\text{MPT-3}}$ is smaller than 10^{-8} .

N	Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\exists \boldsymbol{\pi}_{k+1}^N$			Vertices $(\mathbf{x}_k, \boldsymbol{\kappa}_1)$ for $\nexists \boldsymbol{\pi}_{k+1}^N$		
	EVID	PR	ID	EVID	PR	ID
8	363	768	61	362	768	64
9	1074	1536	270	753	1024	183
10	316	0	0	134	0	0
11	380	512	330	186	256	112
12	196	384	178	80	192	61
13	329	384	312	66	96	53
14	165	480	165	0	0	0
15	285	320	285	0	0	0

3.7.2 Ongoing research with the capturability terminal constraint

In the literature of **MPC** for biped robots, many applications are successful without capturability constraint [Dune 2011, Kajita 2003, Agravante 2016]. Successful applications explored only feasible states that were viable. So what kind of improvements the capturability constraint bring to the **MPC** scheme? First, a capturability constraint guarantees capturability, *i.e.* the ability to stop. Second, the capturability constraint implies viability. When

the robot plans to stop in a fixed number of steps, the constraint guarantees **SRF** for all planning horizon lengths. In case of sudden plan change: when the robot considers to make a new step at the end of the horizon, **SRF** is yet guaranteed but not for all horizon lengths. An **MPC** scheme with capturability constraint is more reliable in balance preservation and we provided several planning horizon lengths (relatively small) to achieve **SRF** in case of sudden plan change.

The capturability constraint has been recently extended for walking up stairs [Pajon 2019] or for compensating potential perturbations [Villa 2019b]. However, both cases lack an evaluation of the issue highlighted in this Chapter, so we do not know for which horizon lengths **SRF** is actually guaranteed when the robot considers to make a new step at the end of the horizon. An important contribution to recursive feasibility using a capturability terminal constraint is proposed in [Scianca 2020]. It differs from our approach in the followings:

- they generate walking motions with 2 **MPC** schemes: the *first* one generates candidate step positions, and the *second* generates the **CoM/CoP** trajectories (based on the candidate step positions).
- they do not use the kinematic constraints (2.43) applied to the **CoM**.
- they impose a fixed **CoP** position after the end of the horizon, in the middle of the support polygon.

And they propose an analytical solution that provides a *lower bound* on the horizon length N for which recursive feasibility is guaranteed for the *second* **MPC**. The lower bound depends on kinematic capabilities of the robot, *e.g.* size of the feet and height of the **CoM**, and postulate an upper bound on the **CoP** velocity. We would like to work in this direction since an analytical solution of this kind is quickly adaptable for the **MPC** of any biped robots. Currently, if we change any parameters of the robot, we need to re-evaluate **SRF** for all horizon lengths. Finally, we want to study the **SRF** guarantee also when adjusting footstep orientation and step cycle duration all together in a single **MPC** [Bohorquez 2018a].

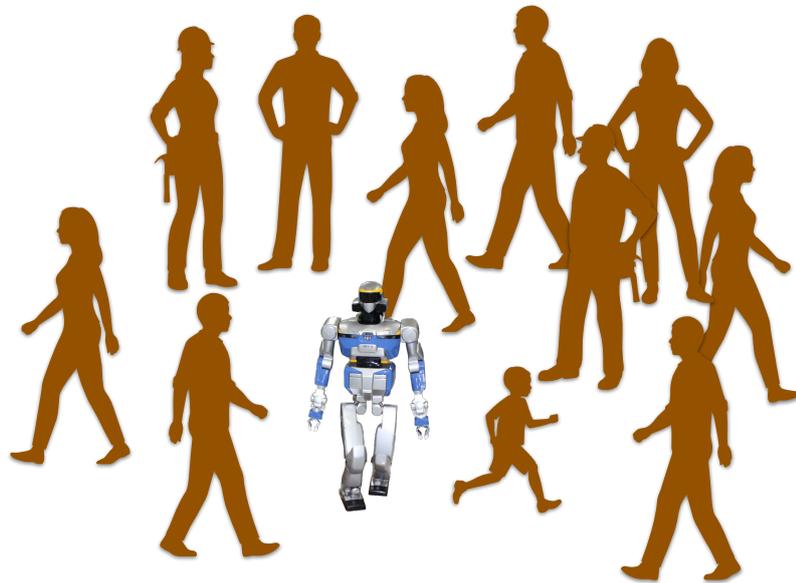
3.8 Conclusion

We investigated the **SRF** guarantee in **MPC** for biped walking. We showed that with the capturability terminal constraint this guarantee is achieved for horizon lengths $N \in \{14, \dots, 18\}$. Without this constraint, the guarantee is lost for all tested lengths. It is then profitable to include the capturability terminal constraint in the **MPC** scheme to guarantee capturability: the robot is then able to stop in finite time, and guarantee **SRF**.

Chapter 4

Motion Safety

Figure 4.1: Representation of a biped robot (HRP-2) in a dense crowd.



4.1 Introduction

We want biped robots to navigate safely in close proximity of people and in this work we focus on a biped robot walking in a dense crowd, represented in Figure 4.1. Navigate among people is challenging since their future behavior can be difficult to predict. Therefore environments shared with humans can be categorized as *dynamic and uncertain environments*.

Before placing biped robots in a real crowd, we need to evaluate the performance of their physical capabilities in a simulated scenario. This should potentially feature both fixed and moving people whose future behavior is unknown and we should limit the sensors of the simulated robot to provide a partial knowledge of its surroundings. In this work we are interested in the robot's capability to avoid collisions, *i.e.* its motion safety. A collision is avoided if the robot respects some kind of thresholds or bounds applied to the motion that will execute, *e.g.* minimum distance from people. The robot therefore needs to anticipate people behaviors, *i.e.* modeling their future behavior, and to render explicit these bounds.

In this Chapter, we describe how we model a simulated dense crowd as well as how to model their future behavior (Section 4.2). Then we explain the sensors of the robot and

how it perceives people, with relative perception errors (Section 4.4). We define collisions with a physical description of the simulated robot and people in Section 4.3. We extend the model predictive control scheme of biped walking for safe navigation purposes among people (Section 4.5): we include collision avoidance constraints and navigation objectives. For a robot moving in a crowd, there are situations where collisions are inevitable: limited physical capabilities and limited knowledge of the future do not help in this regard (Section 4.5.4). Our model predictive control scheme addresses this issue by guaranteeing the robot is at rest before a collision occurs, this is called passive motion safety (Section 4.5.4).

4.2 Crowd

4.2.1 Behavior

One possible crowd behavior could be that every person is very careful in trying to avoid collisions with the robot, *i.e.* cooperative (or joint) collision avoidance [Trautman 2010, Althoff 2012, Silva 2018]. This scenario relates to human aware navigation [Kruse 2013] that focuses on socially acceptable and legible robot behaviors, rather than explicitly solving the collision avoidance problem. In fact, this can lead to a strong bias on the evaluation of the robot’s motion safety with potentially no collisions ever happening what and would not be very meaningful [Fraichard 2006].

Another extreme could be a crowd specifically aiming at collisions with the robot. This scenario represents a hostile or adversarial environment to the robot, that was actually observed in many case studies where children persistently obstructed the robot’s activity in a shopping mall in Japan [Brscić 2015]. This scenario also focuses on socially acceptable behaviors and it would corrupt the evaluation of the robot’s motion safety, *e.g.* only collisions, and would not be very meaningful.

One way to evaluate the robot motion’s safety capabilities and the robot’s motion safety capabilities alone, could be to consider that the robot is actually the only one in charge to avoid collisions [Bohorquez 2016, Wu 2012, Bouraine 2014]. Colloquially the crowd can be said *inattentive* or moving *blindly*, in that it does not account for the presence of the robot. This scenario can also be adopted for benchmarking purposes. In [Liu 2017, Cao 2019] a robot was simulated in a human environment where human-trajectories were played from public datasets of real pedestrian crowds, *e.g.* from CVL lab [Pellegrini 2009] (Zurich) or from UCY lab [Lerner 2007] (Cyprius), so people appeared to walk inattentive to the presence of the robot. The set of human-trajectories however can be limited to the number of crowd recordings.

In this thesis, we decided to work with this inattentive crowd scenario. To fully consider the crowd inattentive, people will (also) not try to avoid each other, *i.e.* collisions among people are disregarded. Since people are not aware of their surroundings and do not actively avoid collisions with the robot or with each other, we assume that they walk at a *constant* velocity.

4.2.2 Pedestrian model

Since the future behavior of people can be difficult to predict, modeling pedestrians is challenging. Motion models are usually built from analyzing human movement. Some of the sophisticated models analyze human intention [Grasso 1998], which are usually highly complex, and walking gait [Laumond 2017, Bissacco 2009]. Motion capture technology has been used to record several human trajectories and they were found to obey a simple nonholonomic system, *i.e.* the unicycle model [Arechavaleta 2008], which has been used for motion prediction [Batkovic 2018, Schneider 2013]. In the scenario where pedestrians do not change

their walking orientation, *e.g.* no turning maneuvers, it is then possible to describe their motion with a basic motion model: a single or double integrator [Kooij 2014, Cao 2019].

Since the crowd is assumed to walk blindly, with respect to the robot and each other, their motion can be chosen arbitrarily. We consider the case where each person follows a *straight* trajectory (different from person to person), *e.g.* people crossing a road on a zebra cross. With this choice, we model each pedestrian with a double integrator model. Let the state of the model for the j -th person in a 2D space be:

$$\boldsymbol{\eta}^j = [\mathbf{z}^j \quad \dot{\mathbf{z}}^j]^\top, \quad (4.1)$$

where $\mathbf{z}^j \in \mathbb{R}^2$ is the person position and his velocity $\dot{\mathbf{z}}^j \in \mathbb{R}^2$. We have then the following LTI system:

$$\dot{\boldsymbol{\eta}}^j = \boldsymbol{\Sigma}_1 \boldsymbol{\eta}^j + \boldsymbol{\Sigma}_2 \ddot{\mathbf{z}}^j, \quad (4.2)$$

where

$$\boldsymbol{\Sigma}_1 = \text{diag}_2 \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right), \quad \boldsymbol{\Sigma}_2 = \text{diag}_2 \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right). \quad (4.3)$$

In discrete time, using the discretization explained in Section 2.2.2, we have

$$\boldsymbol{\eta}_{i+1}^j = \boldsymbol{\Gamma}_1 \boldsymbol{\eta}_i^j + \boldsymbol{\Gamma}_2 \ddot{\mathbf{z}}_i^j, \quad (4.4)$$

where

$$\boldsymbol{\Gamma}_1 = \text{diag}_2 (e^{\boldsymbol{\Sigma}_1 T}) = \text{diag}_2 \left(\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right), \quad (4.5)$$

$$\boldsymbol{\Gamma}_2 = \text{diag}_2 \left(\left(\int_0^T e^{\boldsymbol{\Sigma}_1 t} dt \right) \boldsymbol{\Sigma}_2 \right) = \text{diag}_2 \left(\begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \right). \quad (4.6)$$

4.2.3 Model of the future

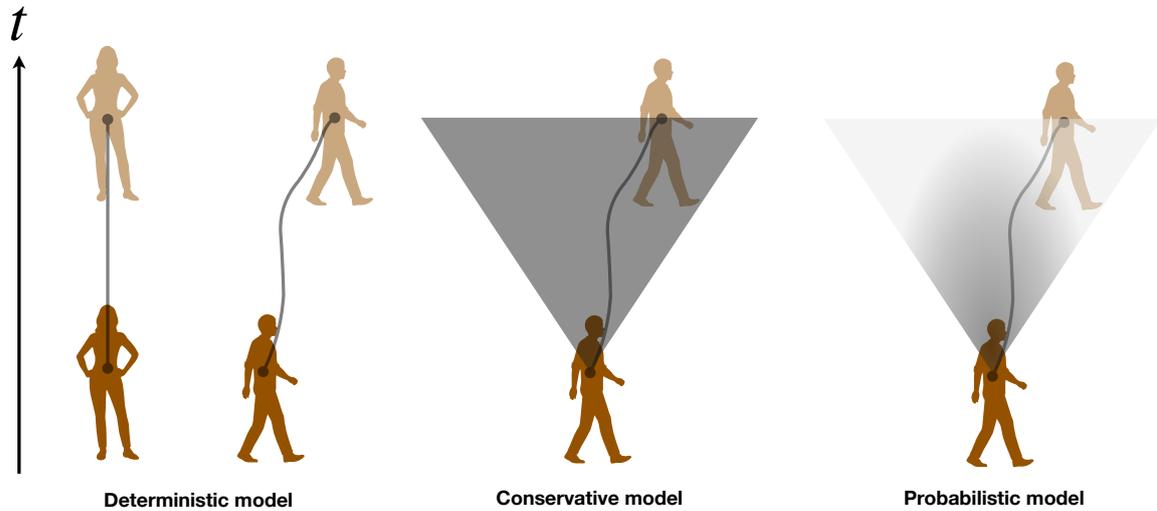
Modeling the environment's future behavior, *e.g.* the pedestrian's future behavior, is one of the safety criteria presented in [Fraichard 2006], crucial for robot motion safety. The robot therefore needs a model that anticipates the motion of people to avoid collisions in the present and into the future. Models of the future broadly fall into three classes: *deterministic*, *conservative* and *probabilistic* (Figure 4.2).

Deterministic models are used when future motions are known beforehand. From a motion safety point of view, deterministic models are useful as long as their prediction of the future evolution of the environment is reliable. Unfortunately, for a crowd of people it is impossible to know beforehand what people will do next. Probabilistic and conservative models address this issue.

There are two popular methods to build probabilistic models. One method consists in two stages. A first stage of "learning": observe the environment (through sensors) in order to identify and model possible motion patterns or plans. And then a second stage of "prediction": learned patterns are used in order to predict future motions. These models aim to improve the accuracy of the prediction (to be as close as possible to the real motion) [Ellis 2009, Chen 2016, Keller 2014, Vasquez Govea 2009] for a long time horizon ($\approx 5 - 7$ s) [Ellis 2009, Chen 2016], but the general drawback is the high computational complexity that comes from the "learning" stage [Keller 2014].

A second method is to use stochastic transition function to describe motion patterns. A probabilistic model is obtained for example through an Extended Kalman Filter [Kushleyev 2009].

Figure 4.2: Modeling the future in 1D space. Deterministic model (gray lines): future motion is known and available beforehand. Conservative model (represented by the gray area): we consider all the possible future motions. Probabilistic model (represented by a gray gradient): bound a probability distribution on pedestrian future occupancy.



The model takes the current measurement of pedestrian position and/or velocity, and it assumes the pedestrian will follow a certain behavior, *e.g.* maintains constant motion, at all future times. A probability of pedestrian (physical) occupation is assigned along the predicted model. A Gaussian noise can represent the probability distribution, with an increasing variance value across time representing the uncertainty of the prediction. The probability distribution is bounded up to two standard deviations from the mean (95% of the normal distribution). The future occupancy within the bound is considered reliable for motion safety purposes [Bautin 2010, Althoff 2010]. The uncertainty will grow with time and, at some point, the distribution will be so wide, that will occupy the entire space. While the uncertainty grows, the probability of pedestrian occupancy diffuses, *i.e.* flattens, and eventually cancels out: the pedestrian “disappears” from the whole space. So pedestrian’s future trajectories are predicted until the probability of pedestrian occupancy flattens [Kushleyev 2009].

Conservative models simply consider all the possible pedestrian’s future trajectories. Accordingly, each pedestrian is assigned a reachable set, *i.e.* the set of positions it can potentially occupy in the future, to represent its future motion [Bouraine 2014, Liu 2017]. In general, the computational complexity is relatively low since these models do not pass from any “learning” phase and use few observations about the pedestrian, *e.g.* their estimated current position. *All* the pedestrian (physical) occupation along the predicted model is assumed to be reliable for motion safety purposes and it is based on some sort of *bounded* behavior assumption. For example: each person’s motion is limited by a maximum walking speed in any direction. Once decided this bounded behavior, there can be a time in which the monotonous growth of the region occupied by the pedestrian is such that, eventually, the whole space will be that region. Accordingly, after this time any motion would not be safe, so pedestrian’s future trajectories are predicted no further than that time.

Probabilistic or conservative models? It is a crucial question. The common factor is that both deal with parameters that are constrained within certain bounds, *i.e.* on the

uncertainty distribution or on the assumed behavior. Since any form of motion safety will always be guaranteed with respect to certain bounds, we could choose independently one of these two models. In this work, we choose a conservative model to anticipate a pedestrian's future behavior. In our crowd scenario: (i) people walk at a constant velocity and they do not try to avoid the robot and (ii) collisions among people are disregarded (see Section 4.2.1). The robot uses these information to anticipate the motion of people.

The *assumption* that people walk at a constant velocity, *i.e.* accelerations equal to zero, determines the anticipative motion of the j -th person computed at time t_i for N time instants:

$$\forall k \in \{1, \dots, N\}, \ddot{\mathbf{z}}_{(i+k-1|i)}^j = 0. \quad (4.7)$$

If not for our crowd scenario only but in general, this is a strong assumption on the anticipative motion. We claim however that this is valid when the future behavior is anticipated for about one or two seconds ($\approx 1 - 2$ s).

It is then possible to describe a pedestrian's future position by evolving for example the state $\boldsymbol{\eta}_i$ for N -times following the dynamics (4.4) with assumption (4.7):

$$\bar{\mathbf{z}}_i^j = \underset{N}{\text{diag}}(\mathbf{I}_z) (\mathbf{U}_5 \boldsymbol{\eta}_i^j), \quad (4.8)$$

where $\bar{\mathbf{z}}_i \in \mathbb{R}^{2N}$

$$\bar{\mathbf{z}}_i = \{\mathbf{z}_{(i+1|i)}^j, \dots, \mathbf{z}_{(i+N|i)}^j\}, \quad (4.9)$$

and

$$\mathbf{U}_5 = [\boldsymbol{\Gamma}_1 \quad \boldsymbol{\Gamma}_1^2 \quad \dots \quad \boldsymbol{\Gamma}_1^N]^\top, \quad (4.10)$$

with $\mathbf{I}_z \in \mathbb{R}^{2 \times 4}$ a selection matrix to extract the person position from the state.

4.3 Physical attributes and interactions

4.3.1 Biped robot and people

The biped robot occupies an area defined by a circle of ray D_{robot} . Let \mathcal{A} be the area occupied by the robot's body when the CoM is at position \mathbf{c} (area in gray, Figure 4.3a)

$$\mathcal{A} \triangleq \text{ball}(\mathbf{c}, D_{\text{robot}}), \quad (4.11)$$

where

$$\text{ball}(\mathbf{x}_c, R) \triangleq \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_c\| \leq R\}. \quad (4.12)$$

As for humans, each person occupies an area defined by a circle [Liu 2017, Bohorquez 2016, Bouraine 2014]. Consider a crowd composed of Z pedestrians. Let \mathcal{B}^j be the area occupied by the j^{th} person as a circle of radius D_{person} (area in blue, Figure 4.3a). Given the position of the person, \mathbf{z}^j , this space is:

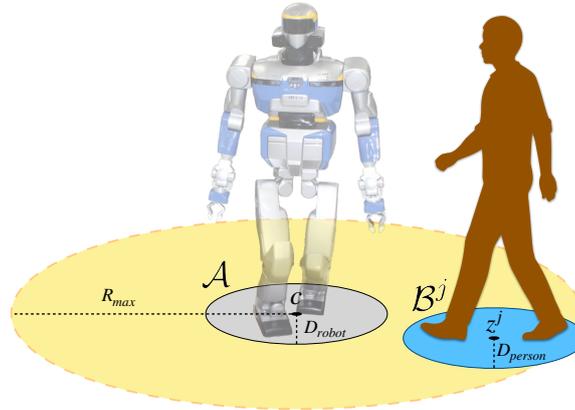
$$\mathcal{B}^j \triangleq \text{ball}(\mathbf{z}^j, D_{\text{person}}). \quad (4.13)$$

Let \mathcal{B} be the space occupied by all Z pedestrians in a crowd:

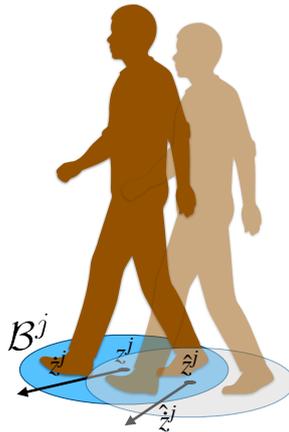
$$\mathcal{B} = \bigcup_{j \in \{1, \dots, Z\}} \mathcal{B}^j. \quad (4.14)$$

Figure 4.3: Physical 2D modeling of the biped robot (and its perception) and the person.

- (a) The figure shows the areas occupied by the biped robot and a person: these areas are circles. The position \mathbf{c} is the center of the robot circle with ray D_{robot} . And \mathbf{z}^j position is the center of the person circle with ray D_{person} . We consider “no contact” between these two areas as long as $\|\mathbf{c} - \mathbf{z}^j\| \geq D_{\text{robot}} + D_{\text{person}}$. Finally, the FoV of the robot is a circle centered in \mathbf{c} with R_{max} as ray: the maximal distance around the robot at which it is capable of perceiving people.



- (b) The figure shows the uncertainties in position and velocity estimations of a person ($\hat{\mathbf{z}}^j, \hat{\dot{\mathbf{z}}}^j$) with respect to the real position \mathbf{z}^j and velocity $\dot{\mathbf{z}}^j$.



4.3.2 Collision

In the crowd scenario, we are interested in identifying a physical contact event between a human and the robot, *i.e.* a collision. We say that a biped robot is in *collision* if

$$\mathcal{A} \cap \mathcal{B} \neq \emptyset. \quad (4.15)$$

The minimum safe distance (no contact) is

$$\sigma_0 = D_{\text{robot}} + D_{\text{person}}. \quad (4.16)$$

Collisions between pedestrian body and robot feet are disregarded.

4.4 Robot's perception

4.4.1 Field of View

We consider a biped robot equipped with a range sensor, *e.g.* laser telemeter or range camera, and it can only perceive a subset of agents that surrounds it. This subset is the

robot's **FoV**. Its shape is arbitrary and it depends on the current surrounding of the robot and the maximum range of its sensor. Occlusions in perception are disregarded, making the robot aware of the current position of everybody within the **FoV**.

In our case, the **FoV** has a *circular* shape centered in \mathbf{c} and R_{\max} is the maximal distance around the robot at which it is capable of perceiving people composing the crowd (area in yellow, Figure 4.3a). The j^{th} person in the crowd is inside the **FoV** if

$$\mathcal{B}^j \cap \text{ball}(\mathbf{c}, R_{\max}) \neq \emptyset. \quad (4.17)$$

4.4.2 Perception uncertainties

We consider uncertainties in position and velocity estimations of the crowd with respect to the real position \mathbf{z}^j and velocity $\dot{\mathbf{z}}^j$ (showed in Figure 4.3b):

$$\tilde{\mathbf{z}}^j = \mathbf{z}^j - \hat{\mathbf{z}}^j, \quad \tilde{\dot{\mathbf{z}}}^j = \dot{\mathbf{z}}^j - \hat{\dot{\mathbf{z}}}^j. \quad (4.18)$$

4.5 MPC-based safe biped navigation in a crowd

The **MPC** scheme in Section 2.4.2 generates a walking motion online with automatic footstep placement. We included a capturability terminal constraint (2.66) to ensure the robot's ability to stop in a finite time and maintain its balance. In this section we expand this scheme to navigate safely in a crowd.

4.5.1 Collision avoidance with the crowd

In this work, the robot avoids collisions by not entering the circle occupied by each person. For the j -th person,

$$\mathbf{c}_{(i+k|i)} - \mathbf{z}_{(i+k|i)}^j \notin \text{ball}(\mathbf{0}, \sigma_{(i+k|i)}), \quad (4.19)$$

this is a nonlinear, non convex constraint. Let $\sigma_{(i+k|i)}$ be the minimal distance between the center of the j -th person and the **CoM** of the robot planned for time t_{i+k} . Then it follows that (4.19) can be represented by the Euclidean distance as:

$$\mathbf{n}_{(i+k|i)}^j (\mathbf{c}_{(i+k|i)} - \mathbf{z}_{(i+k|i)}^j) \geq \sigma_{(i+k|i)}, \quad (4.20)$$

where $\mathbf{n}_{(i+k|i)}^j \in \mathbb{R}^2$ is a unit normal vector that points from the center of the j -th person to the **CoM** of the robot:

$$\mathbf{n}_{(i+k|i)}^j = \frac{(\mathbf{c}_{(i+k|i)} - \mathbf{z}_{(i+k|i)}^j)^\top}{\|\mathbf{c}_{(i+k|i)} - \mathbf{z}_{(i+k|i)}^j\|}, \quad (4.21)$$

and we want to impose this constraint along the planning horizon

$$\bar{\mathbf{n}}_i^j = \{\mathbf{n}_{(i+1|i)}^j, \dots, \mathbf{n}_{(i+N|i)}^j\}. \quad (4.22)$$

The separation distance σ includes perception uncertainties (see Section 4.4.2) as "additional safety distance":

$$\sigma_{(i+k|i)} = \sigma_0 + \|\tilde{\mathbf{z}}^j\| + \|\tilde{\dot{\mathbf{z}}}^j\| (Tk). \quad (4.23)$$

Given the magnitude of the uncertainty in position $\|\tilde{\mathbf{z}}^j\|$, we consider that the real position \mathbf{z}^j lies in a circle of radius $\|\tilde{\mathbf{z}}^j\|$ centered at the estimated position $\hat{\mathbf{z}}^j$. Furthermore, given the magnitude of the uncertainty in velocity $\|\tilde{\dot{\mathbf{z}}}^j\|$, the radius of this circle increases at a rate $\|\tilde{\dot{\mathbf{z}}}^j\| (Tk)$ as it moves with the estimated velocity $\hat{\dot{\mathbf{z}}}^j$. We discuss important aspects about the area defined by the magnitude of the uncertainty in Section 4.5.2.

The **CoM** of the robot is restricted to a moving *half-space* \mathcal{H} [Boyd 2004, Section 2.2.1]:

$$\mathbf{c}_{(i+k|i)} \in \mathcal{H}(\mathbf{n}_{(i+k|i)}^j, \mathbf{z}_{(i+k|i)}^j, \sigma_{(i+k|i)}), \quad (4.24)$$

where

$$\mathcal{H}(\mathbf{n}, \mathbf{o}, R) \triangleq \{\mathbf{r} : \mathbf{n}^T(\mathbf{r} - \mathbf{o}) \geq R\}. \quad (4.25)$$

Note that $\mathbf{n}_{(i+k|i)}^j$ is a nonlinear expression for the variables $\mathbf{c}_{(i+k|i)}$ and $\mathbf{z}_{(i+k|i)}^j$. Since $\mathbf{c}_{(i+k|i)}$ depends on the decision variables, (4.24) is a nonlinear constraint. When T is small enough, \mathbf{n} from the i^{th} computation of the **MPC** does not differ too much from \mathbf{n} obtained in the previous $(i-1)^{\text{th}}$ computation

$$\lim_{T \rightarrow 0} \|\mathbf{n}_{(i+k|i)}^j - \mathbf{n}_{(i+k-1|i-1)}^j\| = 0, \quad (4.26)$$

and we can use it to approximate the original constraint and obtain a completely linear formulation as

$$\mathbf{n}_{(i+k-1|i-1)}^j (\mathbf{c}_{(i+k|i)} - \mathbf{z}_{(i+k|i)}^j) \geq \sigma_{(i+k|i)}, \quad (4.27)$$

and impose it along the planning horizon:

$$\forall k \in \{1, \dots, N\}, \mathbf{c}_{(i+k|i)} \in \mathcal{H}(\mathbf{n}_{(i+k-1|i-1)}^j, \mathbf{z}_{(i+k|i)}^j, \sigma_{(i+k|i)}). \quad (4.28)$$

This approximation is *safe* with respect to the nonlinear problem because it is an *outer* approximation to the left-hand side of the inclusion (4.19). We impose the constraint (4.28) along the evolution of the robot position \mathbf{c}_i (2.27), using the evolution of each person position \mathbf{z}_i^j (4.8) appearing inside the **FoV**.

This half-space method comes at a very low computational cost and it has been successfully applied to industrial robots [Al Homsy 2016]. The constraint (4.28) allows to ensure, at each time instant t_i , a *separation* between robot and people, *i.e.* no collision. When generating a collision-free motion, ensuring that collisions are avoided at each t_i does not guarantee *continuous* collision avoidance: that there is no collision during each time interval (t_i, t_{i+1}) , *i.e.* between samples. In this work, it is assumed that avoiding collisions at time t_i and t_{i+1} implies to avoid collisions between samples. It is however possible to enforce an *additional* set of collision avoidance constraints (4.28) in between samples, similar to (2.58) [Brossette 2017, Zheng 2020]. Continuous collision avoidance (between samples) is also guaranteed in [Mercy 2016] with (4.24) by B -spline parametrization of the **CoM** motion. The parametrization however restricts the set of possible motions.

4.5.2 Conservativeness in time

We consider that the real position \mathbf{z}^j lies in a circle of radius $\|\tilde{\mathbf{z}}^j\|$ centered at the estimated position $\hat{\mathbf{z}}^j$. For this reason, the area \mathcal{B}^j occupied by the j^{th} person (4.13), is somewhere inside an area called $\mathcal{W}^j(t)$

$$\mathcal{B}^j \subseteq \mathcal{W}^j(t). \quad (4.29)$$

The estimated position of the j^{th} person $\hat{\mathbf{z}}^j$ walking with the estimated speed $\hat{\dot{\mathbf{z}}}^j$ at time t occupies the area $\mathcal{W}^j(t)$. This area is modeled as a circle with a radius that accounts for D_{person} and robot's perception errors, $(\|\tilde{\mathbf{z}}^j\|, \|\hat{\dot{\mathbf{z}}}^j\|)$,

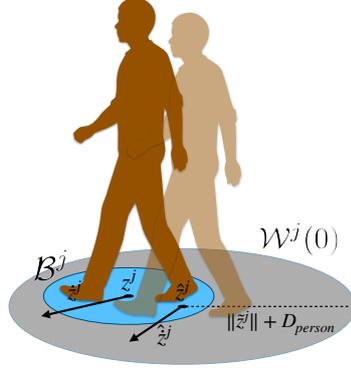
$$\mathcal{W}^j(t) \triangleq \text{ball}(\hat{\mathbf{z}}^j + \hat{\dot{\mathbf{z}}}^j t, D_{\text{person}} + \|\tilde{\mathbf{z}}^j\| + \|\hat{\dot{\mathbf{z}}}^j\| t). \quad (4.30)$$

We call this area *unsafe zone* because there is a person inside but the robot does not know exactly where. Let \mathcal{W} be the unsafe zone occupied by the estimation of all Z pedestrians in a crowd:

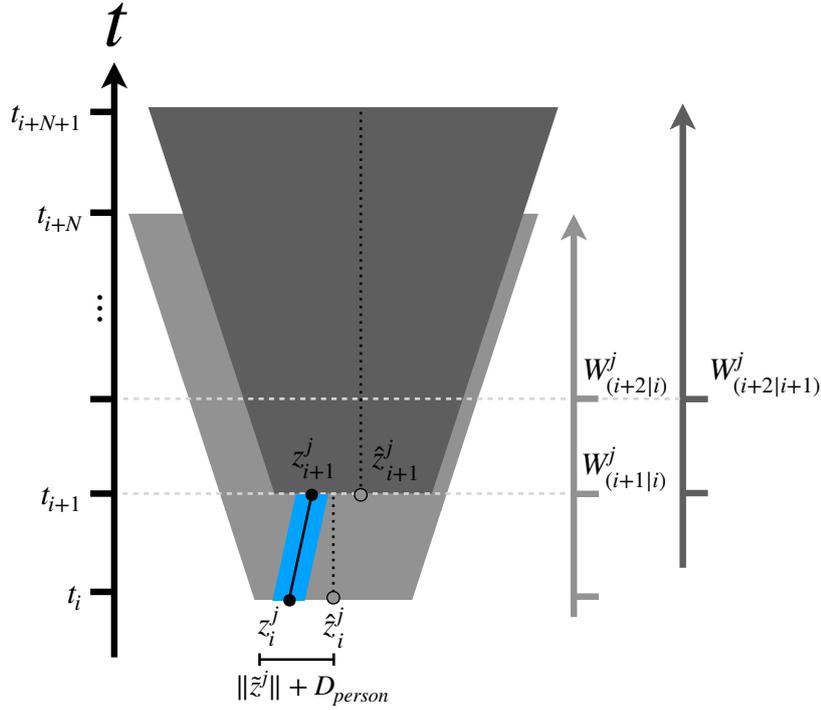
$$\mathcal{W}(t) = \bigcup_{j \in \{1, \dots, Z\}} \mathcal{W}^j(t). \quad (4.31)$$

Figure 4.4: Representation of the unsafe zone and its conservativeness in time.

- (a) The unsafe zone $\mathcal{W}^j(t)$ is an area where there is a person (\mathcal{B}^j) inside it but the robot does not know exactly where the person is. The unsafe zone is modeled as a circle with a radius that accounts for D_{person} and robot's perception errors, $(\|\tilde{z}^j\|, \|\hat{z}^j\|)$. The figure shows $\mathcal{W}^j(0)$.



- (b) The figure shows in 1D space how the monotonous growth of $\mathcal{W}_{(i+k|i)}^j$ is updated (or reset) at each computation time when we gather a new estimate of the person's position (in this case the estimate in velocity \hat{z}^j is 0). As a consequence, each predicted unsafe zone computed at time t_i fits always inside the one re-computed at time t_{i+1} . In blue, the area \mathcal{B}^j occupied by the j^{th} person is somewhere inside \mathcal{W}^j .



so we have:

$$\mathcal{B} \subseteq \mathcal{W}(t). \quad (4.32)$$

In our discrete-time model implementation, let $\mathcal{W}_{(i+k|i)}^j$ be the unsafe zone computed at time t_i and predicted for time t_{i+k} as

$$\mathcal{W}_{(i+k|i)}^j \triangleq \text{ball}(\hat{z}_i^j + \hat{z}_i^j(Tk), D_{\text{person}} + \|\tilde{z}^j\| + \|\tilde{z}^j\|(Tk)). \quad (4.33)$$

The radius of this circle increases at a rate $\|\dot{\tilde{\mathbf{z}}^j}\|$ (Tk) as it moves with the estimated velocity $\hat{\mathbf{z}}^j$. The monotonous growth of \mathcal{W}^j however is updated (or reset) at each computation time when we gather a new estimates of the person’s position and velocity. As a consequence, each predicted area computed at time t_i *fits always inside* to the one re-computed at time t_{i+1} . In general we have:

$$\mathcal{W}_{(i+k|i+k)}^j \subseteq \mathcal{W}_{(i+k|i)}^j. \quad (4.34)$$

In Figure 4.4 we represent these 2 properties (4.29) and (4.34) of \mathcal{W}^j . The prediction of the unsafe zone is always reduced at each computation time: it cannot be bigger than before.

4.5.3 Optimization

We would like a feasible walking motion to minimize the deviation from a set of objectives.

Objectives

On the balance front, we want to keep the **CoP** as close as possible to the center of the foot on the ground along the planning, to improve the robustness of the robot against perturbations [Wieber 2006a], such as possible collisions. We could furthermore minimize the **CoM** jerk component to improve fall avoidance of the biped robot [Kajita 2003], but we addressed this issue (maintain balance) with the capturability terminal constraint (2.66).

On the collision front, when the robot is traversing the crowd, the objectives can fall on a wide spectrum. We can focus on sociability (adherence to explicit high-level socio-cultural conventions), on comfort (absence of annoyance and stress for humans), and/or naturalness (robot tries to mimic human behaviour patterns). Most of these objectives take into account some common sense rules and comply with social conventions. We could for example use a repulsive potential function [Khatib 1985], where the repulsive force applied to the robot is based on *social forces*: a measure for the internal motivation of the individuals to perform certain movements [Helbing 1995]. The robot could additionally try to walk at a *comfortable distance* from each person: same concept of collision avoidance constraints, but as a reference function [Bohorquez 2018a]. Last, but not least, a penalty function can be used to respect a desired *social norm* [Chen 2017], *e.g.* “passing a person on the right”, breaking the collision avoidance symmetry problem^[1].

In this work however, we are only interested in the robot’s motion safety and not in socially acceptable behaviors. For this reason, we can either ask the robot to reach a desired location or to follow a desired walking speed $\dot{\mathbf{c}}_{\text{ref}}$. Whatever choice of these objectives, they should make the robot traverse a moving crowd (to evaluate its motion safety). We choose the robot should walk at a certain walking speed.

The objective on the balance front (**CoP** as close as possible to the center of the foot) and the objective to make the robot traverse the crowd (follow desired walking speed) are expressed as a convex quadratic function of the form (2.53) (see [Herdt 2010] for details):

$$f_w(\boldsymbol{\pi}_i^N) = \sum_{k=1}^N \left(\|\dot{\mathbf{c}}_{\text{ref}} - \dot{\mathbf{c}}_{(i+k|i)}\|^2 + \|\mathbf{p}_{(i+k|i)} - \mathbf{s}_{(i+k|i)}\|^2 \right). \quad (4.35)$$

¹When there exists at least two equally efficient ways to avoid a collision, such as overtaking a person on the right or on the left.

The MPC scheme computes a motion by solving the following QP:

$$\underset{\pi_i^N}{\text{minimize}} \quad f_w(\pi_i^N) \quad (4.36a)$$

$$\text{s.t.} \quad \begin{cases} \text{physical constraints (2.57)-(2.58),} \\ \text{terminal constraint (2.66),} \\ \text{collision avoidance constraints (4.28).} \end{cases} \quad (4.36b)$$

Newton step

While the linear approximation in (4.28) is safe with respect to the nonlinear problem (4.19), it might over-constrain the robot's behavior. To overcome this problem, to treat collision avoidance as a nonlinear problem, we apply a Newton method [Nocedal 2006] each time we solve the QP (4.36). At the i^{th} computation, the QP is solved iteratively to reduce the collision avoidance approximation. Let $\bar{\mathbf{n}}_{i,k}^j$ the sequence of normal vectors of the j -th person at iteration k . This is updated with the optimal solution of the previous iteration $k-1$, $\pi_{i,k-1}^*$, and the iteration process stops when:

$$\|\pi_{i,k}^* - \pi_{i,k-1}^*\| \leq \epsilon. \quad (4.37)$$

The maximum number of iterations is set to `max_Iter` and ϵ is chosen arbitrarily small: order of one thousandth. We summarize this procedure in Algorithm 2. Thanks to the safe linear approximation of (4.19), feasible iterates are always generated. If we do not satisfy (4.37) after `max_Iter` iterations, we can however use the solution $\pi_{i,\text{max_Iter}}^*$ since it is feasible.

Algorithm 2 Network Step for the j -th person

Require: $\bar{\mathbf{n}}_{i-1}^j, \mathbf{z}_i$

- 1: Compute a motion π_i^* from (4.36) with $\bar{\mathbf{n}}_{i-1}^j$
 - 2: Compute a CoM sequence $\bar{\mathbf{c}}_i$ from π_i^*
 - 3: Compute $\bar{\mathbf{n}}_i^j$ from (4.21) with $\bar{\mathbf{c}}_i$ and $\bar{\mathbf{z}}_i$
 - 4: Set $\bar{\mathbf{n}}_{i,0}^j \leftarrow \bar{\mathbf{n}}_i^j$, $\bar{\mathbf{c}}_{i,0} \leftarrow \bar{\mathbf{c}}_i$ and $\pi_{i,0}^* \leftarrow \pi_i^*$
 - 5: $k \leftarrow 0$
 - 6: **repeat**
 - 7: Compute a motion $\pi_{i,k+1}^*$ from (4.36) with $\bar{\mathbf{n}}_{i,k}^j$
 - 8: **if** $\|\pi_{i,k+1}^* - \pi_{i,k}^*\| \leq \epsilon$ **then**
 - 9: $k \leftarrow \text{max_Iter}$
 - 10: **else**
 - 11: Compute a CoM sequence $\bar{\mathbf{c}}_{i,k+1}$ from $\pi_{i,k+1}^*$
 - 12: Compute $\bar{\mathbf{n}}_{i,k+1}^j$ from (4.21) with $\bar{\mathbf{c}}_{i,k+1}$ and $\bar{\mathbf{z}}_i^j$
 - 13: $k \leftarrow k + 1$
 - 14: **end if**
 - 15: **until** the maximum number of iteration `max_Iter` is reached
-

Safe optimal behavior

Classical approaches for collision avoidance such as velocity damping [Faverjon 1987], control barrier functions [Chen 2018], repulsive potential field [Khatib 1985], invariance control [Kimmel 2017] and momentum limitations [Tsai 2014], come with a major drawback. They optimize the motion with arbitrary restrictions (or constraints), *i.e.* based on arbitrary

parameters, related to the robot-obstacles distance and sometimes loosely to the dynamics of the robot. For this reason, they generate *suboptimal* behaviors such as: unnecessary detours for a biped robot walking among obstacles [Agrawal 2017, Fig. 7-8] when using control barrier functions or decelerating/stopping in proximity of obstacles when using velocity damping [Kanehiro 2008].

Some of those classical approaches furthermore assume that avoiding collisions is always possible. This is impossible to guarantee in a partially unknown dynamic environment such as a crowd. We discuss this safety issue in the next section.

4.5.4 Safety guarantee

For a biped robot moving in a crowd, two things that should be avoided are to fall and to collide with people.

On the balance front, with the capturability terminal constraint (2.66), we can guarantee that the robot will always be able to stop in a few footsteps and maintain its balance forever. On the collision front, only with the complete knowledge of the surrounding environment and its future evolution it is possible to determine *whether* the robot, given its current configuration and limited physical capabilities, achieves “no collision ever”, dubbed *absolute motion safety*, thanks to the concept of ICS [Fraichard 2003]. This corresponds to satisfy at all time a set of collision avoidance constraints such as (4.24). And it is the assumption of classical approaches on collision avoidance, see Section 4.5.3. In practice, such knowledge is possible to obtain only in specific environments that we are not interested in: *static*, *freezing* or *periodic* [Bouguerra 2019]. In dynamic and uncertain environments such as a crowd, “no collision ever” is impossible to guarantee [Fraichard 2006].

It is nonetheless possible to guarantee that the robot will be able to stop before a collision takes place, should this collision be inevitable. No collision would happen if everybody behaved that way, and in this sense the robot will have done its share. This property is called passive motion safety [Bouraine 2014], and it has already been used effectively with several robotics platforms, see Figure 4.5.

Remark 4. *In our case, absolute motion safety corresponds to guarantee SRF for the MPC scheme (4.36) because the set of collision avoidance constraints would be satisfied at all time.*

Fall avoidance and Passive Safety guarantees

Passive motion safety or called also *Passive Safety (PS)* guarantees that the robot is able to stop before a collision occurs: if a collision is inevitable, at least the robot will be at rest when that happens. Capturability has previously been used in MPC scheme for biped robots to guarantee both balance and PS. This scheme combines the conservative model that anticipates the motion of people (4.8) and the capturability terminal constraint (2.66) to ensure that the robot can stop (keeping its balance indefinitely) before any collision happens [Bohorquez 2016].

The main idea is that at time instant t_i the robot tries to compute a motion for N time instants solving (4.36) in which: it will walk without colliding for few footsteps and additionally maintain balance in the last footstep forever, *i.e.* stop. If it cannot compute such motion, it can always execute all the remaining $N - 1$ actions of the last computed motion at time instant t_{i-1} as *fallback*. Thanks to conservativeness in time explained in Section 4.5.2, the last motion that was safe, it is still safe since the unsafe zone cannot grow in time. As a consequence, the robot walks without colliding and it will stop balancing on its last footstep on the ground (balance guaranteed) before any collision occurs (PS guaranteed).

The robot, before executing the remaining actions of the fallback motion, can re-evaluate the situation to check if, based on new measurements of the surrounding crowd, it is possible

to postpone as long as possible the moment when it will stop. We can go, after executing each action, through a simple loop to find a motion valid for the largest number of time instants: $\{1, 2, \dots, N\}$, that satisfies all constraints.

Figure 4.5: Passive Safety has already been developed and used with several robotics platforms.

(a) humanoid robots [Morisawa 2005].



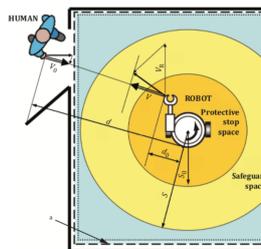
(b) self-driving cars [Macek 2008].



(c) autonomous helicopters [Choudhury 2014].



(d) It corresponds to the *emergency stop* procedures required for industrial and personal care robots [ISO 2014].



4.6 Conclusion

Pedestrian's future behavior is crucial for robot motion safety. We chose to consider all the possible pedestrian's future trajectories, using a conservative model of the future. We use this model to design collision avoidance constraints for our biped robot, accounting for perception errors on the estimation of people's velocity/position around it.

We then use those constraints in a MPC-based biped navigation scheme to generate safe walking motions in proximity of people. Last, in case a collision will be inevitable, this scheme guarantees the robot will stop balancing on its last footstep on the ground (balance guaranteed) before any collision occurs (PS guaranteed). This is possible thanks to the capturability constraint and conservative model of the future.

Chapter 5

Re-planning Effect on MPC-based Safe Biped Navigation in a Crowd

5.1 Introduction

We control a biped robot moving in a crowd using the MPC scheme presented in Chapter 4, which generates stable walking motions with automatic step placement and guarantees both fall avoidance and passive motion safety. For biped robots, once a step is planted on the ground, it usually stays there at a constant position until the next step is initiated. This naturally constrains the capacity for the robot to react and adapt its motion in between steps. As a result, the walking strategies in [Chestnutt 2005, Karkowski 2016, Garimort 2011] propose to re-plan the walking motion to adapt to changing environments once per step, only when a new step is initiated. In contrast, our MPC scheme re-plans the walking motion not only at each step initiation but also in between. Obviously re-planning more often than once per step comes at the expense of computational power.

In this Chapter we measure if re-planning the walking motion more often than once per step can lead to an improvement in collision avoidance when navigating in a crowd. To compare our approach with the walking strategies in the literature, we introduce in Section 5.2 two new MPC variables:

1. planning period.
2. initial planning phase.

In our case they are used to decide:

1. how often to re-plan the walking motion per step.
2. when to re-plan the walking motion along the step.

The challenging scenario of a biped robot walking against an inattentive crowd, used for our evaluation purposes, is described in Section 5.3.1. We used the robot parameters as in Chapter 3 and Section 5.3.3 presents the MPC parameters. The results of collision avoidance performances obtained using simulated crowd scenarios are finally presented and discussed in Section 5.4.

5.2 Planning period and initial planning phase in MPC

The standard MPC approach, as explained in Section 2.4.1, is an iterative planning process that operates as follows: at time t_i , a motion π_i^N made up of a sequence of N control actions is computed. The motion satisfies a set of constraints imposed from t_i to t_{i+N} , N is the planning horizon. The first control action of the motion is executed. At

time t_{i+1} the planning process is repeated, until an objective is reached. The sampling period T denotes the fixed duration (in seconds) between two consecutive time instants $[t_i, t_{i+1}]$.

The planning process in the standard MPC approach is repeated at every time t_i . In certain situations though, it might be difficult to compute a new motion at each time, it might be interesting to compute instead a new motion every M times, *e.g.* to save computational power. In such a case, it would correspond to execute M actions of the motion π_i^N before repeating the planning process at time t_{i+M} . Henceforth, M denotes what we will call the *planning period* (equal to 1 in standard MPC)

Definition 5.2.1 (Planning Period). *The planning period M ,*

$$M \in \{1, 2, \dots, N\}, \quad (5.1)$$

is the number of time instants before the planning process is repeated.

When $M > 1$ the planning process will not be repeated at every time t_i . In case we deal with some sort of *cycle* (with a period greater than T), it is however possible to choose the set of instants at which the planning process will be repeated along the cycle by introducing

Definition 5.2.2 (Initial Planning Phase). *The initial planning phase ϕ ,*

$$\phi \in \{0, \dots, M - 1\}, \quad (5.2)$$

is the time instant t_ϕ considered the initial time at which the planning process starts.

The planning process will be repeated at every time $t_{iM+\phi}$.

5.3 Simulation parameters

5.3.1 Crowd

We summarize the crowd behavior chosen in Section 4.2:

- people do not account for the presence of the robot.
- people do not try to avoid each other, *i.e.* collisions among people are disregarded.
- each person walk at a constant velocity.
- each person follows a straight trajectory.

In this Chapter, we consider the worst case scenario where the robot is walking in the opposite direction of the crowd. We consider 100 different random crowds, composed of Z people that differ in the initial positions \mathbf{z}_0 and speeds along y , $\{z^y\}$. The initial positions of the people vary uniformly over an area of $10 \times 8[\text{m}^2]$. For all people the velocity along x is chosen constant $0.5[\text{m/s}]$, while for each person speed along y is picked randomly in an interval $[-0.2, 0.2][\text{m/s}]$. Parameters are summarized in Table 5.1a.

5.3.2 Robot

The parameters of the biped robot were selected according to the kinematics of the robot HRP-2 as it was done in Chapter 3, but we recall these parameters in Table 5.1b.

5.3.3 MPC

The sampling period T and the step cycle s_d are chosen as in Section 3.3.2. The robot plans always a full gait cycle, $N = 16$, since it is a standard choice for our MPC-based biped locomotion [Dune 2011, Sherikov 2016, Herdt 2010]. We control the robot with the MPC scheme in Section 4.5.4 that guarantees fall avoidance and PS.

We chose arbitrarily the parameter for the Newton method: the maximum number of iterations `max.Iter` is 5, and the convergence factor ϵ is 10^{-4} . We made no comparison with any other parameter choice. When the robot walks against a crowd, the robot perceives people around it within a FoV based on a specific sensor, *e.g.* laser scanner. The choice of FoV is not treated fully as in [Bouraine 2014], but its radius R_{\max} is chosen greater than a certain lower bound in order to guarantee PS [Bohorquez 2016]. This lower bound depends on the plan duration NT and the robot’s and people’s walking velocities:

$$R_{\max} \geq NT(\dot{z}^x + \dot{c}_r^x) \quad (5.3)$$

We account for different magnitudes of uncertainties ($\|\tilde{z}^j\|, \|\tilde{\dot{z}}^j\|$) when the robot estimates people’s position and velocity. All these MPC and perception parameters are summarized in Table 5.1c.

In Chapter 4, the robot tries to plan a new walking motion at each time instant, this corresponds to $M = 1$. Up to now, we can only find $M = 1$ in the literature of MPC for biped robots: [Dafarra 2018, Tsagarakis 2017, Naveau 2017, Caron 2019, Gouaillier 2010]. In our framework, when $M = 1$, the robot re-plans 8 times per step, every 0.1[s]. Since most walking strategies proposed to re-plan the walking motion once per step (to adapt to changing environments), we can compare this choice with our MPC framework by changing the planning period to $M = 8$. Doing so, the robot re-plans less often, every 0.8[s]. We additionally consider intermediate cases. When the robot re-plans 4 times per step, every 0.2[s], and when it re-plans 2 times per step, every 0.4[s]. We represent these re-planning choices in Figure 5.1a, and we show in Table 5.1d the relationship between the re-planning per step frequency and planning period M .

In our implementation, we choose t_0 as the time the robot starts a DS phase. When the robot re-plans once per step in Figure 5.1a, this corresponds to set the initial planning phase as $\phi = 0$. In Figure 5.1b we show how the robot could choose instead to start planning at different phases, *e.g.* with $\phi = 1$ when the robot ends DS phases. If the robot can only re-plan less often than at each time instant, when is the best moment (w.r.t. its motion safety capabilities) to re-plan along the step? How often to re-plan and when to re-plan are investigated in Section 5.4.

The additional parameters M and ϕ have *no* impact on the safety guarantees of the MPC scheme in Section 4.5.4. For example, if the robot cannot compute a new motion at time t_i , it can always execute all the remaining $N - M$ actions of the fallback motion computed at time t_{i-M} , $\forall M$. It will walk without colliding and stop balancing on its last footstep on the ground (balance guaranteed) before any collision occurs (PS guaranteed).

Table 5.1: Simulation parameters.**(a)** Crowd Parameters.

Parameter	Symbol	Value	Unit
Crowd Size	Z	$\{8, 16, 24, 32\}$	ppl
Number of randomly generated crowds (for size Z)	-	100	-
Velocity of the j -th person	\dot{z}^x	0.5	m/s
	\dot{z}^y	$[-0.2, 0.2]$	m/s
Body radius (4.11)-(4.13)	D_{person}	0.5	m
	D_{robot}	0.5	m
min. separation distance (4.16)	σ_0	1	m

(b) Robot Parameters.

Parameter	Symbol	Value	Unit
Height of CoM (2.17)	c^z	0.8	m
Min. Feet Separation (2.42)	f_s	0.07	m
Feet Dimensions (2.40)	(l, w)	(0.24, 0.14)	(m,m)
Step Length (2.44)	(L, W)	(0.24, 0.30)	m

(c) MPC, Newton method and perception parameters.

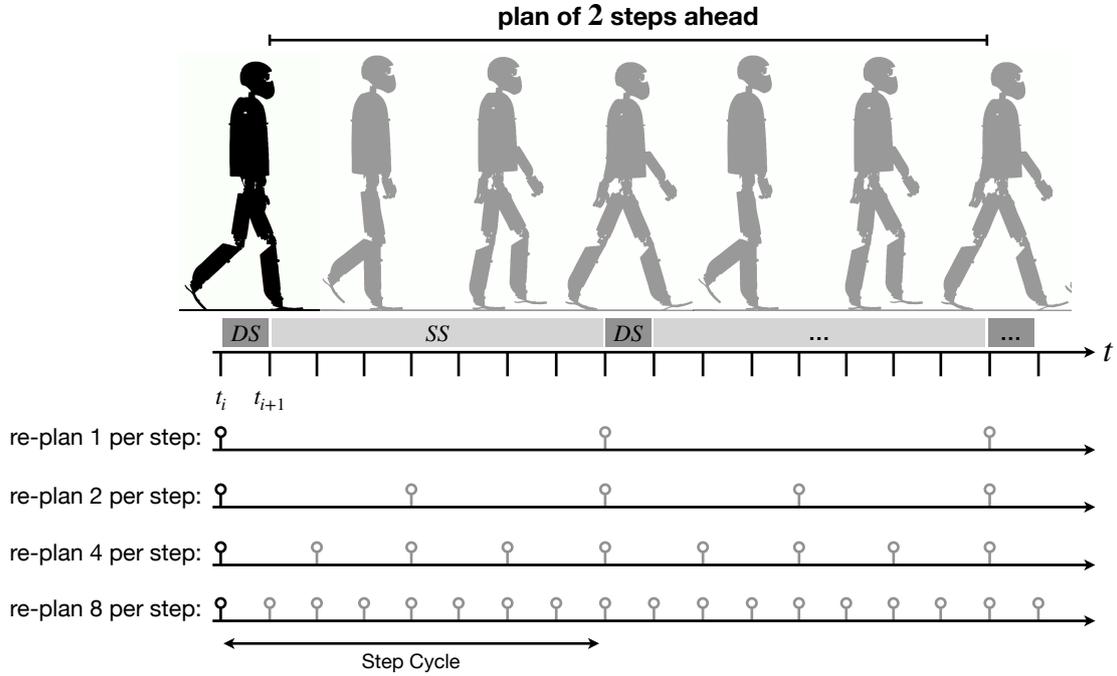
Parameter	Symbol	Value	Unit
Sampling Period	T	0.1	s
Planning Period	M	$\{1, 2, 4, 8\}$	-
Initial Planning Phase	ϕ	$\{0, \dots, M - 1\}$	-
Planning Horizon	N	16	-
Adjustable Steps Ahead	m	2	step
Step Cycle	s_d	0.8	s/step
Reference Speed (4.35)	$(\dot{c}_r^x, \dot{c}_r^y)$	(0.5, 0.0)	m/s
Newton step Iterations	<code>max_iter</code>	5	-
Convergence Parameter	ϵ	10^{-4}	-
FoV distance (4.17)	R_{max}	4	m
Uncertainty	$\ \tilde{z}^j\ $	$\{0, 0.15, 0.30\}$	m
	$\ \tilde{\dot{z}}^j\ $	$\{0, 0.05, 0.10\}$	m/s

(d) Relationship between re-planning per step frequency and planning period M .

	Value			
Re-planning per step frequency	1	2	4	8
Planning Period M	8	4	2	1

Figure 5.1: How often and when to re-plan the walking motion.

- (a) The figure shows when at time t_i we plan a walking motion pattern of 2 steps ahead with a fixed step cycle. Below, we can see when the planning events happen by re-planning 1, 2, 4 or 8 times per step. In our case, re-planning 8 times per step corresponds to re-plan at each time instant.



- (b) We choose t_0 as the time the robot starts a *DS* phase. The figure shows the case of re-planning 1 time per step and for several choices of initial planning phase ϕ . By varying ϕ , the robot re-plans the walking motion at different time along the step cycle.

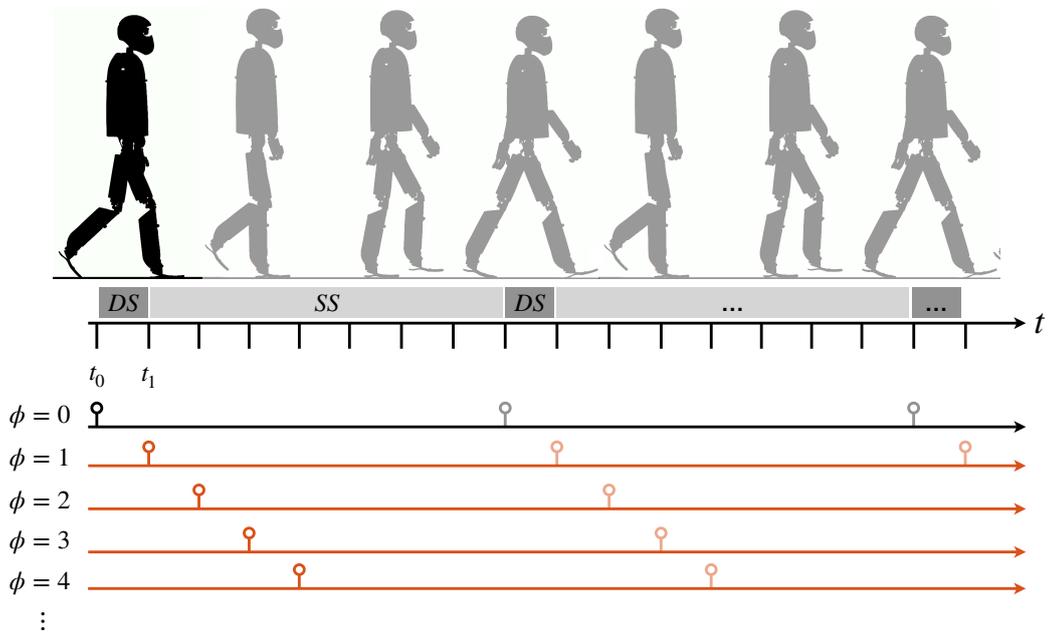
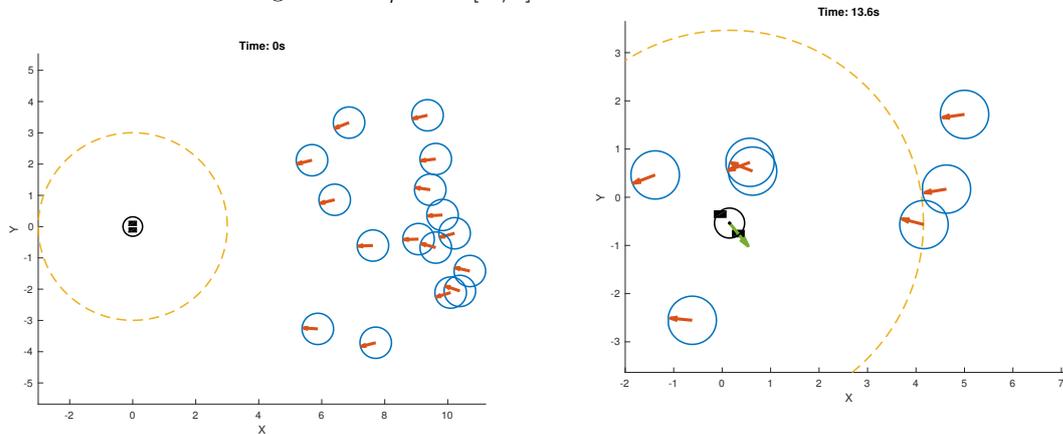


Figure 5.2: The inattentive crowd scenario.

- (a) Time t_0 in a crowd of size $Z = 16$: robot in black, people in blue (collisions among people are disregarded), the FoV is the dashed circle and the instantaneous velocity of each person is represented by the red vector. The robot is asked to walk to the right with $\dot{c}_r^x = 0.5$ [m/s].
- (b) Frame of the biped robot that is traversing an inattentive crowd. The instantaneous velocity of the CoM is represented by the green vector.



5.4 Results

The effect of re-planning per step on success rate for a biped robot traversing a moving crowd is evaluated numerically, as seen in Fig. 5.2. We consider a “success” if the robot walks and does not collide for 20[s]. And we stop the simulation before if people collide with the robot. The initial positions of the people are randomly chosen in front of the robot outside the FoV. For each uncertainty ($\|\tilde{z}^j\|, \|\tilde{\dot{z}}^j\|$), we simulate 100 crowds of Z people.

We start by evaluating all the possible initial planning phases ϕ per re-planning frequency in the crowd scenario. Each point of Figure 5.3a corresponds to 3600 crowd scenarios, varying all the possible combination of uncertainty ($\|\tilde{z}^j\|, \|\tilde{\dot{z}}^j\|$) and crowd size Z for a choice of re-planning frequency. 100% of success rate means that the robot did not collide for 3600 simulations.

Re-planning less often means to save computational power but it comes at the expense of a decay in success. The results indicates that when re-planning less often than 8 times per step, choosing *never* to re-plan when the robot starts DS phases (in our settings $\phi > 0$), leads to a significant improvement in success rate.

Recall from Section 2.2.3 that at the time the robot starts a DS phase, the next step is fixed: this helps to reduce the complexity of the step adjustment. The results here indicate that re-planning less often, but keeping the re-plan when a DS phase starts, worsen the robot motion safety capabilities adaptation to changing environments.

We see in Figure 5.3a that re-planning when the DS phase ends and/or along the SS phase only, the success of the biped improves at least by 20% when re-planning once per step, but only less than 5% when re-planning 2, 4 times per step. Overall, the biped reaches the “best” success rate:

- when re-planning 2, 4 times per step and one of the re-planning occurs when the DS phase ends ($\phi^* = 1$).
- when re-planning once per step and the re-planning occurs in the middle of SS phases only ($\phi^* = 4$).

Note in the Figure when $\phi = 4$, there is less than 5% difference in success rate between re-planning once and 2 times per step.

We now compare the choices of re-planning frequency (for both $\phi = 0$ and ϕ^*) versus the crowd sizes. Each point of the plot in Figure 5.3b corresponds to 900 crowd scenarios, varying all the possible combination of uncertainty ($\|\tilde{z}_k\|, \|\hat{z}_k\|$) for a choice of re-planning frequency and crowd size Z . We report the number of success in perceptual as “success rate”: 100% means that the robot did not collide for 900 simulations.

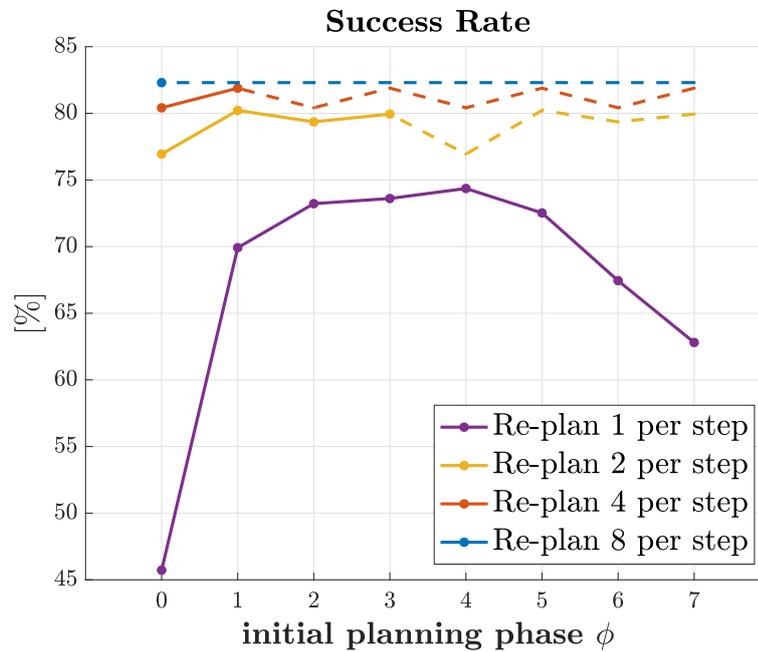
Obviously, the success of the robot decreases when we increase the density of the crowd: represented by the number of people Z , that the robot needs to traverse.

In the plot, the dashed lines are the results with $\phi = 0$ for all choices of re-planning frequency. The solid lines are the results where we set ϕ^* for each re-planning frequency. When re-planning once per step for example, the robot improves its success from 20% ($Z = 8$) to 30% ($Z = \{16, 24, 32\}$) if it re-plans in the middle of **SS** phases (with ϕ^*), instead of re-planning when it starts the **DS** phases ($\phi = 0$).

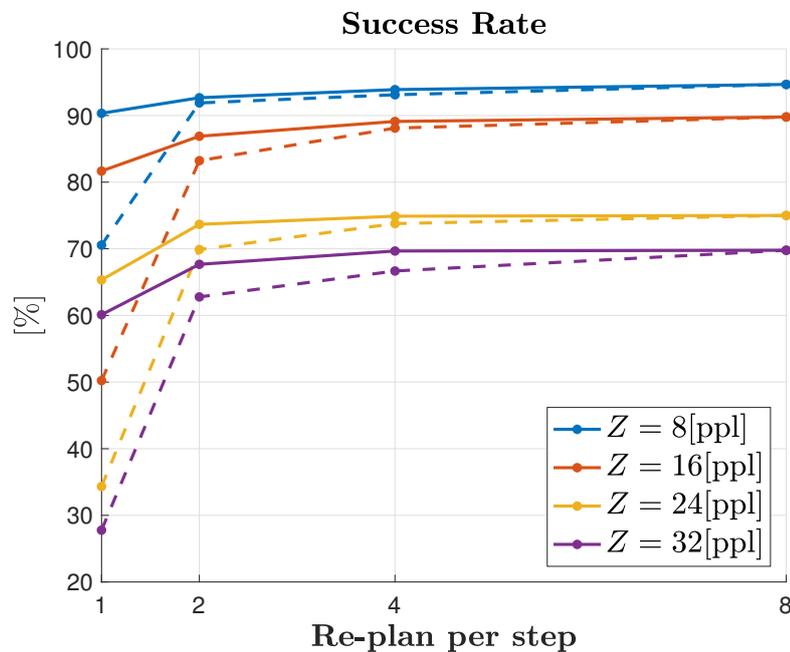
We conclude that, for a robot traversing a moving crowd, we can favor to re-plan only once per step and in the middle of **SS** phases only, instead of 2, 4 or 8 times per step to save computational power, since success improves only by less than 10%.

Figure 5.3: Results of re-planning effect on success rate.

- (a) The success rate for different re-planning frequencies and initial planning phases ϕ is showed in solid line. When $\phi = M + j$ with $j \in \mathbb{N}$ we have duplicated results in dashed lines.



- (b) The success rate for different re-planning frequencies and crowd sizes Z . In dashed line, for all re-planning frequencies, one of the re-planning occurs when the robot starts the **DS** phase ($\phi = 0$). In solid line, when the robot re-plans 2, 4 times per step, one of the re-planning occurs when the **DS** phase ends ($\phi^* = 1$). And when it re-plans once per step, re-planning occurs in the middle of **SS** phases only ($\phi^* = 4$).



5.4.1 Collisions and uncertainty

We now give some insight on the relationship between success rate and uncertainty. We summarize in Table 5.2 the success rate in all the simulations that counts all crowd sizes Z and all initial planning phases ϕ , for each re-planning frequency and various amount of uncertainty ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$). Given a re-planning frequency for any uncertainty pair ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$), the total number of simulations is

$$\text{simulations} = 100 \times (\text{n. of } Z) \times (\text{n. of } \phi). \quad (5.4)$$

When re-planning once per step for example, the success rate is calculated over 3200 simulations.

When there is uncertainty, the robot accounts for it by being more cautious: uncertainty is represented as an additional *area* where people might be into the future and the robot needs to avoid it. As result, the success rate increases in the presence of uncertainty. We can observe that uncertainty on speed has a higher impact on success rate than uncertainty on position. This is marked in the table when re-planning 1 per step.

Table 5.2: Relationship between collisions and uncertainty.

Uncertainty		Success Rate [%] for re-planning			
$\ \tilde{z}^j\ $ [m/s]	$\ \tilde{z}^j\ $ [m]	1 per step	2 per step	4 per step	8 per step
0.00	0.00	61.72	80.38	82.25	83.00
	0.15	59.78	76.44	77.50	79.25
	0.30	61.34	74.62	76.38	77.50
0.05	0.00	69.06	82.44	85.12	88.25
	0.15	68.72	79.88	80.00	81.00
	0.30	66.44	76.75	80.25	81.75
0.10	0.00	75.22	83.44	84.88	85.75
	0.15	73.47	80.62	83.50	83.75
	0.30	71.31	77.50	80.50	80.50

5.5 Discussion

Hereby, we discuss on few aspects of this contribution, and possible future directions.

5.5.1 Constraints on the swing phase

We explained in Section 2.2.3 that we do not account for the motion of the feet in the air, but it could be modeled however with a 3-rd order polynomial as in [Sherikov 2016, Section 4.4.3]. Furthermore, we do not account for constraints on the swing foot during the SS as it is done in [Diedam 2008]. Those constraints are simple bounds on the position of the next step depending on the current position of the foot in the air. When we re-plan not at each time instant, but more scarcely along the SS, the robot adjusts less often the position of the next step where the current step in the air will land and this might be infeasible without the additional constraints on the swing phase. Based on this observation, our future work is to include those bounds and re-evaluate the experiments when the robot re-plans less often than at each time instant along the SS phase: $\phi \geq 2$.

5.5.2 Varying the dynamicity of the crowd

When the simulated robot walks in the same inattentive crowd that moves at a standard human walking speed (between $1[\text{ms}^{-1}]$ and $1.5[\text{ms}^{-1}]$ [Ralston 1958]), it was observed that the success rate drops dramatically when the robot re-plan at each time instant [Bohorquez 2016], and the same result is expected for all choice of re-planning. The robot does not have enough kinematic and dynamic capabilities to avoid collisions with people walking that fast. In fact, the robot's maximal walking speed, with a fixed stride length ($\times 2$ step length) and step duration, is approximately $0.6[\text{ms}^{-1}]$.

When the simulated robot walks among people that do not move, *i.e.* $\forall j, \dot{z}^j = 0$, we simulate a static environment. When unexpected static people appear inside the FoV, if the robot can guarantee PS, respecting the lower bound on the radius R_{\max} (5.3), it always has a fallback motion at hand to stop in front of those people and avoid colliding with them: we achieve absolute motion safety. As a consequence the success rate would be 100% for all choices of re-planning.

It could be interesting to see how the success rate for all choices of re-planning varies for different crowd velocities in between the extreme points explained above. The focus of interest would be the case that performed worst: when the robot re-plans once per step at the start of DS phases ($\phi = 0$). And for example finding the smallest crowd velocity the success rate decays drastically for that choice.

5.5.3 Varying the sampling period

In our framework, when the robot tries to plan a new walking motion at each time instant, it re-plans 8 times per step. This is due to the choice of step cycle duration and sampling period T . As a future direction, we could increase the length of the sampling period instead of re-planning less often that at each time instant. It has been shown for example that when considering only the balance of a biped robot, re-planning more often than every $0.2[\text{s}]$ to potential perturbations leads to no practical improvement in the maximal tracking error [Villa 2019a]. In that work the sampling period T was set to $0.2[\text{s}]$. This choice would correspond to re-planning at each time instant and 4 times per step. By increasing the sampling period we could have the same effect of re-planning at each time instant.

5.6 Conclusion

Most existing walking strategies propose to re-plan the walking motion, adapting to changing environments only once at every step. Currently, our MPC scheme for biped robots re-plans the walking motion 8 times per step. We investigated if re-planning the walking motion more often than once per step: 2, 4 or 8, can lead to an improvement in motion safety capabilities when a biped robot is navigating in a crowd.

Our results show that we can favor to re-plan once per step, instead of 2, 4 or 8 times per step, since the motion safety capabilities of the robot improves only by less than 10%. There is, however, an important condition to satisfy: the robot re-plans once per step in the middle of SS phases only. We can conclude that it is possible to save computational power with our MPC for biped robots by re-planning less often the walking motion to adapt to changing environments, without deteriorating the robot's motion safety capabilities.

Chapter 6

Collision Mitigation

6.1 Introduction

Passive Safety (**PS**) has been criticized because while the robot makes sure to always have time to stop before a collision happens, this time might not be enough for people around to actually react and avoid the collision once the robot has stopped [Macek 2008].

We begin this Chapter by introducing the Time To React or *TTR* for the people (Section 6.2): the time left before the robot enters the unsafe zone (where collisions could eventually happen) when following a particular motion. Our definition of *TTR* is different from the one that can be found in the literature of risk indicators for autonomous vehicles [Lefèvre 2014, Sontges 2018, Wagner 2018]: the time available for the driver to act before the collision is inevitable. Based on the criticism of **PS**, we conveyed that people around the robot could potentially attempt to react and avoid collisions if they have enough time to do so. For this reason, we claim the following: more *TTR* for the surrounding environment reduces the number of collisions. For this reason, we would like the robot to leave as much *TTR* as possible for the people.

A variant of **PS** was proposed under the name of *Passive Friendly Safety* (**PFS**): the robot will be at rest before collisions happen, leaving enough time for the surrounding environment to react and avoid collisions. Both **PS** and **PFS** however limit the *TTR* the robot can leave for people because the robot is constrained to stop before collisions happen. We show that if the robot is not constrained to stop before collisions happen, it can further improve *TTR*.

For this reason, we propose a new **MPC**-based biped navigation scheme called *Collision Mitigation* (**CM**) that no longer guarantee **PS** but aims at leaving as much *TTR* as possible for the people (Section 6.4). We then measure if controlling the robot with **CM** lead to an improvement in collision avoidance when navigating in a crowd with respect to the **MPC**-based biped navigation scheme introduced in Chapter 4, that guarantees **PS**. The challenging scenario of a biped robot walking against an inattentive crowd as in Chapter 5 is used for our evaluation purposes. In this scenario, we compare the number of collisions (collision risk), collision time and relative collision velocity of both navigation strategies. These results obtained using simulated crowd scenarios are finally presented and discussed in Section 6.6.1-6.6.2.

6.2 The Time To React (*TTR*)

From the Ethics Commission of the German Federal Ministry of Transport and Infrastructure [Ethics Commission 2017], we cite:

“[...] collision avoidance systems are governed by the same principle as airbags or seat belts. Death caused an airbag inflating improperly remains a wrong, but the manufacturer will not be held liable for it if they have done everything that might be reasonably expected to minimize such risks.”

However, a definition of such “risk” is not provided. Because we are mainly interested in the risk of collision, we propose the following definition:

Definition 6.2.1 (Collision Risk). *The number of collisions.*

Following the ethics recommendation, the robot must aim to achieve a *zero* collision risk, in other words: “no collisions ever happen”. But this is impossible to guarantee [Fraichard 2003] and for this reason the robot must constantly aim at minimizing such risk, achieving:

Goal 6.2.1 (Collision Risk Minimization). *The robot minimizes the collision risk.*

Define $\mathcal{A}^\pi(t)$ the area occupied by the robot following a motion $\pi \in \Pi$ at time t , where Π is the set of all possible motions of a finite duration^[1] starting from the current state of the robot. Let’s introduce the *Time To React* or *TTR* for the surrounding environment as:

Definition 6.2.2 (TTR). *The time to react or TTR is the time left before the robot enters the unsafe zone^[2] $\mathcal{W}(t)$ following a motion π :*

$$TTR(\pi) \triangleq \{t^* \mid 0 \leq t < t^*, \mathcal{A}^\pi(t) \cap \mathcal{W}(t) = \emptyset, \mathcal{A}^\pi(t^*) \cap \mathcal{W}(t^*) \neq \emptyset\}. \quad (6.1)$$

People around the robot could *potentially* attempt to react and avoid collisions if they have enough time to do so. For this reason, we claim the following:

Assumption 6.2.1. *More TTR for the surrounding environment reduces the collision risk.*

Beyond the planning horizon, which is constrained by the sensors horizon (*e.g.* radius of the **FoV** as in (5.3)), anything can happen and the entire environment can be suddenly unsafe: we consider that beyond the planning horizon, everywhere is unsafe.

Define $\mathcal{A}_{(i+k|i)}^\pi$ the area occupied by the robot at time t_{i+k} that is following the motion π computed at time t_i . We can calculate the *TTR* at each computation time when the robot is not inside the unsafe zone:

$$TTR_i(\pi) \triangleq \{t_{i+k^*} \mid \forall k \in [0, \dots, k^* - 1], \mathcal{A}_{(i+k|i)}^\pi \cap \mathcal{W}_{(i+k|i)} = \emptyset, \mathcal{A}_{(i+k^*|i)}^\pi \cap \mathcal{W}_{(i+k^*|i)} \neq \emptyset\} \quad (6.2)$$

As explained in Section 4.5.2, some parts of the unsafe zone can become safe later, when the uncertainty is reduced with new observations of the surrounding environment, but the $TTR(\pi)$ *never* gets worse, as shown in the following Lemma.

Lemma 6.2.1. *Suppose that the robot is outside the unsafe zone at time t_i , then:*

$$TTR_{i+1}(\pi) \geq TTR_i(\pi). \quad (6.3)$$

Proof. Suppose that at time t_i :

$$\mathcal{A}_{(i+k|i)}^\pi \cap \mathcal{W}_{(i+k|i)} = \emptyset. \quad (6.4)$$

¹For example the motion duration used in this work: NT .

²It is the area where there is a person inside but the robot does not know exactly where.

Following the same motion π , the predicted area that the robot will occupy at time t_{i+k} does not change:

$$\mathcal{A}_{(i+k|i+1)}^\pi \equiv \mathcal{A}_{(i+k|i)}^\pi. \quad (6.5)$$

As explained in Section 4.5.2, thanks to the conservativeness in time of our model for the unsafe zone, the unsafe zone predicted at time t_{i+1} always fits inside the one predicted at time t_i :

$$\mathcal{W}_{(i+k|i+1)} \subseteq \mathcal{W}_{(i+k|i)}. \quad (6.6)$$

Therefore, at the $(i+1)^{th}$ computation time we have

$$\mathcal{A}_{(i+k|i+1)}^\pi \cap \mathcal{W}_{(i+k|i+1)} = \mathcal{A}_{(i+k|i)}^\pi \cap \mathcal{W}_{(i+k|i+1)} \subseteq \mathcal{A}_{(i+k|i)}^\pi \cap \mathcal{W}_{(i+k|i)} = \emptyset. \quad (6.7)$$

As a consequence $TTR_{i+1}(\pi) \geq TTR_i(\pi)$. \square

The Lemma 6.2.1 considers the situation when the robot does *not* recompute a new motion at time t_{i+1} . If the robot keeps following the motion π , $TTR_{i+1}(\pi)$ cannot get worse, but searching for a new motion π^n at time t_{i+1} , it might actually improve: $TTR_{i+1}(\pi^n) \geq TTR_{i+1}(\pi)$, especially since the unsafe zone could be reduced with new observations of the surrounding environment.

The impossibility to achieve zero collision risk led to settle with a safety guarantee called **PS**: the robot is at rest when a collision happens. It guarantees that the robot is always able to stop before a collision occurs: if a collision is inevitable, at least the robot will be at rest when that happens. No collision would happen, so the collision risk would be minimal (zero), if everybody behaved that way: in this sense, the robot will have done its share. **PS** is achieved by imposing the robot to stop before entering the unsafe zone and staying at rest for the entire duration inside it:

$$\forall t \geq TTR(\pi), \mathcal{A}^\pi(t) \equiv \mathcal{A}^\pi(TTR(\pi)). \quad (6.8)$$

And the set of motions that the robot must follow to guarantee **PS** before entering the unsafe zone is:

$$\Pi^{PS} \triangleq \{\pi \in \Pi \mid \exists t^* \leq TTR(\pi), \forall t \geq t^*, \mathcal{A}^\pi(t) \equiv \mathcal{A}^\pi(t^*)\}. \quad (6.9)$$

This is of course a subset of the set of all motions:

$$\Pi^{PS} \subset \Pi. \quad (6.10)$$

PS has been criticized because while the robot makes sure to always have time to stop before a collision happens, this time might not be enough for people around to actually react and avoid the collision once the robot has stopped [Macek 2008]. Let's now borrow a basketball example that explain the relationship between **PS** and **TTR**.

Consider a robot as a defender in basketball. One of the main objectives of a defender in basketball is to use his or her body to impede the ball-carrier's advance toward the basket. The defender's only absolute way to achieve this is to stand directly and as quickly as possible in the ball-carrier's path, ideally triggering a charging foul in case a collision occurs while the defender is still³. This corresponds to minimizing **TTR** while maintaining **PS**. This is to show that **PS** is "indifferent" to **TTR**: it is possible to maintain **PS** while looking to maximize, or minimize **TTR**, or do anything else in between.

A variant of **PS** was proposed under the name of **PFS**: the robot will be at rest before a collision happens, leaving T_s time for the surrounding environment to react and avoid

³[https://en.wikipedia.org/wiki/Personal_foul_\(basketball\)#Charging_and_blocking](https://en.wikipedia.org/wiki/Personal_foul_(basketball)#Charging_and_blocking)

collisions. This time can be based on further assumptions about the environment, *e.g.* lower bound braking power and an upper bound reaction time to avoid collisions [Mitsch 2013]. **PFS** is achieved by imposing the robot to remain at rest for enough time T_s , before entering the unsafe zone:

$$\forall t \geq TTR(\boldsymbol{\pi}) - T_s, \mathcal{A}^\pi(t) \equiv \mathcal{A}^\pi(TTR(\boldsymbol{\pi}) - T_s). \quad (6.11)$$

And the set of motions that the robot must follow to guarantee **PFS** before entering the unsafe zone is:

$$\Pi_{T_s}^{PFS} \triangleq \{\boldsymbol{\pi} \mid \exists t^* \leq TTR(\boldsymbol{\pi}) - T_s, \forall t \geq t^*, \mathcal{A}^\pi(t) \equiv \mathcal{A}^\pi(t^*)\}. \quad (6.12)$$

Suppose we choose $T_{s_1} \geq T_{s_2} \geq 0$, we have the following set inclusion^[4]:

$$\Pi_{T_{s_1}}^{PFS} \subseteq \Pi_{T_{s_2}}^{PFS} \subseteq \Pi_0^{PFS} = \Pi^{PS}. \quad (6.13)$$

For example, the robot that achieves **PS** can choose to stop (and remain at rest) from $TTR(\boldsymbol{\pi}) - T_{s_1}$, achieving also **PFS** for T_{s_1} . In general, when in doubt about the environment and how much T_s is necessary, Assumption 6.2.1 would actually relate Goal 6.2.1 to:

Goal 6.2.2 (Maximization of TTR). *The robot leaves as much TTR as possible for the surrounding environment.*

$$\underset{\boldsymbol{\pi}}{\text{maximize}} \quad TTR(\boldsymbol{\pi}) \quad (6.14a)$$

It is possible to combine this goal with the previous safety guarantees. Based on the set relationships

$$\Pi_{T_s}^{PFS} \subseteq \Pi^{PS} \subset \Pi, \quad (6.15)$$

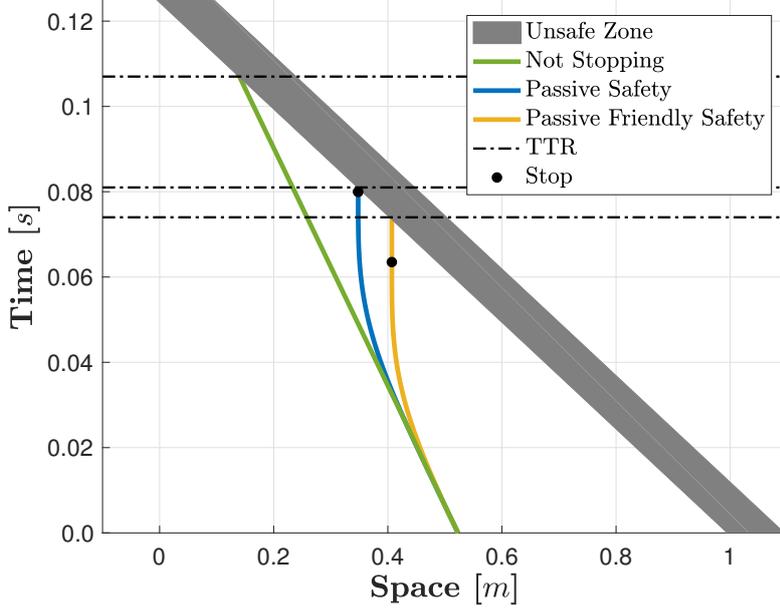
all the possible TTR that **PFS** can achieve are achievable by **PS**, and all the possible TTR that **PS** can achieve are achievable by any general motion. For this reason, a general motion fulfills Goal 6.2.2 as good as, or at least better than any motion that guarantees only **PS**. And **PS** fulfills Goal 6.2.2 as good as, or at least better than any motion that guarantees **PFS**. We have then

$$\underset{\boldsymbol{\pi} \in \Pi}{\text{maximize}} \quad TTR(\boldsymbol{\pi}) \geq \underset{\boldsymbol{\pi} \in \Pi^{PS}}{\text{maximize}} \quad TTR(\boldsymbol{\pi}) \geq \underset{\boldsymbol{\pi} \in \Pi_{T_s}^{PFS}}{\text{maximize}} \quad TTR(\boldsymbol{\pi}). \quad (6.16a)$$

In Figure 6.1 a robot aims at Goal 6.2.2 (in green), additionally guarantees **PS** (in blue) or guarantees **PFS** (in yellow) while a person is walking towards it. **PFS** imposes the robot to be at rest waiting for the surrounding people to avoid collisions. If the person does not react to avoid the coming collision, the robot waiting at rest reduces TTR . Since the robot can do more when it is not at rest, TTR is improved when it guarantees only **PS**. And, the TTR is further improved if the robot is not required to stop just before entering the unsafe zone, but only aiming at Goal 6.2.2.

⁴As a side note, when the robot achieves **PFS** for T_s that corresponds to its maximum value, the entire motion duration, the robot remains at rest at all time.

Figure 6.1: We simulate 1D scenario where a robot is walking away from a person. The robot aiming at Goal 6.2.2 (in green) without additionally guaranteeing to be at rest before entering the unsafe zone, achieved the greatest *TTR* since it is less constrained.



6.3 Motion planning solving a lexicographic optimization problem

For a biped robot moving in a crowd, two things that should be avoided are to fall and to collide with people. Additionally, the robot should minimize the deviation from some given references. These goals or objectives might conflict: avoiding a collision might lead the robot to fall onto the ground, in that case at least one of them must be *relaxed*. This conflict can be approached with Goal Programming [Escande 2014]. This technique has already been used effectively for the control of all kinds of robots, humanoid [Sherikov 2015], industrial [Al Homsy 2016] and self-driving vehicles [Blumentals 2017]. It accounts for multiple potentially incompatible objectives in complete safety, which is essential to deal with complex situations. In our case, the objectives can be expressed as a set of linear inequalities of the form

$$\mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} \leq \mathbf{d}_i \quad (6.17)$$

Relaxation can be done by introducing a violation \mathbf{w} as in

$$\mathbf{E}_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\pi}_i^N \end{bmatrix} - \mathbf{w}_i \leq \mathbf{d}_i \quad (6.18)$$

Note that the objective is satisfied if $\|\mathbf{w}_i\| = 0$. We assign *priority levels* for different objectives, deciding which of them can be relaxed. Objectives with the same importance are defined in the same level. Once assigned the priority levels, we try to satisfy objectives by minimizing their violations accordingly, that is, as a lexicographic least squares problem [Dimitrov 2015]. This ensures objectives with lower priority are optimized as far as they do not interfere with the optimization of objectives with higher priority.

In the lexicographic programming approach, we have a sequence of objectives to be satisfied. These objectives are competing with each other: it is impossible to satisfy them all

at the same time. So we arrange these objectives according to relative importance in a hierarchy. We then try to solve them “sequentially”. First, we solve the most important one. Then, among the set of solutions of the most important one, we solve the second objective in importance. And so on until the least important objective.

By defining P priority levels, each objective is described by a pair $(\mathbf{E}_i, \mathbf{d}_i)$, and we look to

$$\text{lex minimize}_{\pi_i^N, \{\mathbf{w}_i^j\}} \{ \|\mathbf{w}_i^1\|^2, \dots, \|\mathbf{w}_i^P\|^2 \} \quad (6.19a)$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{E}_i^1 \\ \vdots \\ \mathbf{E}_i^P \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \pi_i^N \end{bmatrix} - \begin{bmatrix} \mathbf{w}_i^1 \\ \vdots \\ \mathbf{w}_i^P \end{bmatrix} \leq \begin{bmatrix} \mathbf{d}_i^1 \\ \vdots \\ \mathbf{d}_i^P \end{bmatrix}. \quad (6.19b)$$

Solving problem (6.19) means to plan a motion π_i^N for the robot to perform. We solved it with our in-house solver LexLS and we refer to [Dimitrov 2015] for algorithmic details.

6.4 A lexicographic programming approach to maximize TTR

Collision avoidance could conflict with preserving balance. We consider here that preserving balance has higher priority. As a result, the first objective of the robot is to preserve balance as defined by the set of walking constraints (2.57)-(2.58) with the capturability terminal constraint (2.66). Balance is preserved as long as those constraints are not violated. As explained in Chapter 3, for several horizon lengths N we do not have any constraint violation when the robot is planning to stop and when a new step is added at the end of the horizon to continue walking.

Once balance is preserved, the robot aims at Goal 6.2.2. Suppose only 1 person is present inside the FoV. Collision avoidance constraints are assigned in *temporal order* to the priority levels: avoid a collision at time t_{i+k} corresponds to satisfy the k^{th} collision avoidance constraint as

$$d_{(i+k|i)}^{k+1} \geq \sigma_{(i+k|i)}^{k+1}. \quad (6.20)$$

where $d_{(i+k|i)}^{k+1}$ corresponds to the approximated euclidean distance between the person and the robot (left hand side of (4.27)), which is assigned to the $(k+1)$ priority level, $k \in \{1, \dots, N\}$. If the minimal violation of the level $(k+1)$ is

$$(w_i^{k+1})^* = 0 \quad (6.21)$$

the robot avoids entering the unsafe zone when executing the k^{th} action of π_i^N . The temporal order of these constraints in the priority levels ensures to leave as much TTR as possible for the surrounding people (Goal 6.2.2).

If the robot cannot avoid entering the unsafe zone at time t_{i+k} , we have

$$d_{(i+k|i)}^{k+1} < \sigma_{(i+k|i)}^{k+1}. \quad (6.22)$$

The violation w_i^{k+1} accounts for the *penetration* into the unsafe zone:

$$d_{(i+k|i)}^{k+1} + w_i^{k+1} = \sigma_{(i+k|i)}^{k+1}. \quad (6.23)$$

Among all possible options, π_i^N minimizes such violation in a least-squares manner (6.19a). Let's call δ the displacement between two consecutive minimized violations w as

$$(\delta_i^k)^* = (w_i^{k+1})^* - (w_i^k)^*. \quad (6.24)$$

Dividing this quantity by the time interval duration T , we have the average velocity along that displacement

$$(\bar{\delta}_i^k)^* = \frac{(\delta_i^k)^*}{T}. \quad (6.25)$$

This is the component of the *average velocity* of the robot in the direction of the estimated position of the person inside the unsafe zone.

At the lowest priority ($N + 2$), we minimize the deviation $\|\mathbf{w}_i^{N+1}\|^2$ from the references (4.35): **CoP** as close as possible to the center of the foot and follow a desired walking speed.

We call this scheme *Collision Mitigation (CM)* and it is represented by Hierarchy 1.

Hierarchy 1: Collision Mitigation
1: Capturable walking motion (2.57)-(2.58)-(2.66)
2: Minimize penetration unsafe zone (6.20) at t_{i+1}
⋮
N: Minimize penetration unsafe zone (6.20) at t_{i+N-1}
N+1: Minimize penetration unsafe zone (6.20) at t_{i+N}
N+2: Minimize reference deviation (4.36a)

6.5 Simulation parameters and settings

The parameters

- of the crowd are in Table 5.1a.
- of the robot are in Table 5.1b, and the **FoV** of the robot is 4[m].
- of the **MPC** are in Table 5.1c but we limit the experiments to $M = 1$ ($\phi = 0$).

We use the crowd settings as explained in Section 5.3.1.

6.6 Mitigating the risk of collision in the crowd scenario

The effect of mitigating the risk of collision (Goal 6.2.1) for a biped robot traversing a moving crowd is evaluated numerically. The collision risk (Definition 6.2.1) counts the number of collision events (or failure): when we stop the simulation because people collide with the robot in 20s or less. The initial positions of the people are randomly chosen in front of the robot outside the **FoV**. For each uncertainty ($\|\tilde{\mathbf{z}}^j\|, \|\tilde{\mathbf{z}}^j\|$), we simulate 100 crowds of Z people.

6.6.1 Collision risk

We compare in the crowd scenario a robot controlled with the **MPC** scheme used in Chapter 5, that we call hereby just **PS**, and the same robot controlled with the **MPC** scheme called **CM** (Hierarchy 1). We compute the *cumulative* collision risk along the simulation time, showed in Figure 6.2. The cumulative (frequency) plot shows the percentage [%] out of 3600 crowd scenarios^[5], varying all the possible combination of uncertainty ($\|\tilde{\mathbf{z}}^j\|, \|\tilde{\mathbf{z}}^j\|$)

⁵These scenarios are used for both **PS** and **CM**

for a choice of crowd size Z , of collision risk that are less than or equal to particular time.

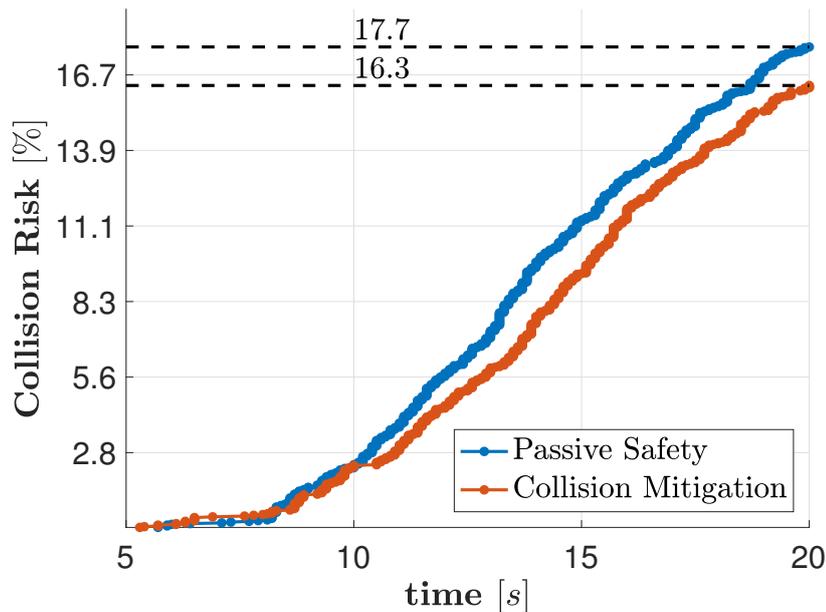
In our simulations, people are inattentive thus they will *never* react to avoid collisions. It is however possible to compute the time when collisions happen. Increasing the TTR means that collisions would happen later in time. In Figure 6.2, for any collision risks, the robot controlled with **CM** collides always later than the one controlled with **PS**.

The figure shows that **CM** outperforms **PS** in terms of collision risk: **CM** better fulfills Goal 6.2.1. Due to the cumulative sum, we have many collision events in the top right of the plot whereas few events in the bottom left. Statistics in the bottom left are considered less reliable because they are based on few events only (around 20 events only, $\approx 0.5\%$).

As long as the robot can avoid entering the unsafe zone for the entire plan duration (NT), the walking motion of the robot with **PS** is *equivalent* to the one of **CM**: the motion satisfies the same set of constraints (including the capturability terminal constraint) and it is optimized with the same cost function. The same robot controlled either with **PS** or **CM** follows the same path. If at any time, we have a motion that guarantees **PS** (of whatever length), next time we can plan either **PS** or **CM** and both will have an opportunity to improve TTR (as explained in Lemma 6.2.1). But, when the robot cannot avoid the unsafe zone for the entire plan duration, **CM** plans a *less* constrained motion before entering the unsafe zone^[6], and it is always at least as good, if not better than **PS** with respect to Goal 6.2.2.

When the robot is inside the unsafe zone, the one controlled with **PS** stays at rest waiting for a new safe motion to continue walking. The one controlled with **CM** is not imposed to remain at rest, but instead is trying to minimize the penetration (6.23) and can keep walking. We observe that searching a motion to minimize the penetration over time inside the unsafe zone helps in postponing collisions later in time and reducing collision risk.

Figure 6.2: The figure shows the cumulative plot of the collision risk in percentage along the time of all simulations for **PS** and **CM**.



⁶capturability is no longer imposed before entering the unsafe zone

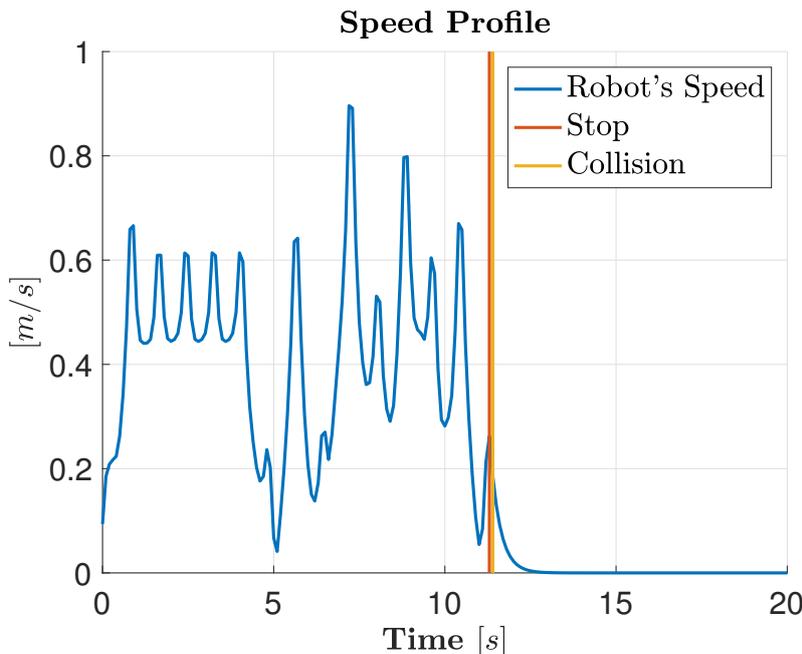
6.6.2 Collision velocity

We compare **PS** and **CM** in terms of impact. The kinetic energy transferred to a person can be used as a metric for injury of people [Shea]. In our case, we consider the relative collision velocity between a robot and person that is proportional to the kinetic energy transferred to the person. This velocity is calculated as the difference between the robot and person velocity components in the direction of the point of collision.

We compute the cumulative collision risk along the magnitude of relative collision velocity, showed in Figure 6.4. The cumulative (frequency) plot shows the percentage [%] out of 3600 crowd scenarios, varying all the possible combination of uncertainty ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$) for a choice of crowd size Z , of collision risk that are less than or equal to particular relative collision velocity.

PS guarantees that the robot is at rest when collisions happen (*i.e.* zero robot's kinetic energy during impacts). This implies that the robot does not add (or remove) kinetic energy to impacts. The robot cannot actively collide with people. The blue curve confirms this theory: the relative collision velocity with **PS** is always lower or equal than the maximum velocity of people. We can however observe that for $\approx 0.5\%$ of collision risk (top left of blue curve) the robot added a small amount of kinetic energy to the collision. Our hypothesis is that the capturability constraint (2.66) to stop the robot does not imply immediately 0 speed of the robot, but converging to 0. So for those cases, the robot was stopping while balancing with a single foot on the ground. One simulation that represents this situation is showed in Figure 6.3. In the case of **CM** up to 11% of collision risk the robot did not add (or remove) kinetic energy with people (as for **PS**) but it is not the case for the remaining 5%. **CM** is *worse* than **PS** in this respect since there are a significant amount of collision events where the robot actively collided with people.

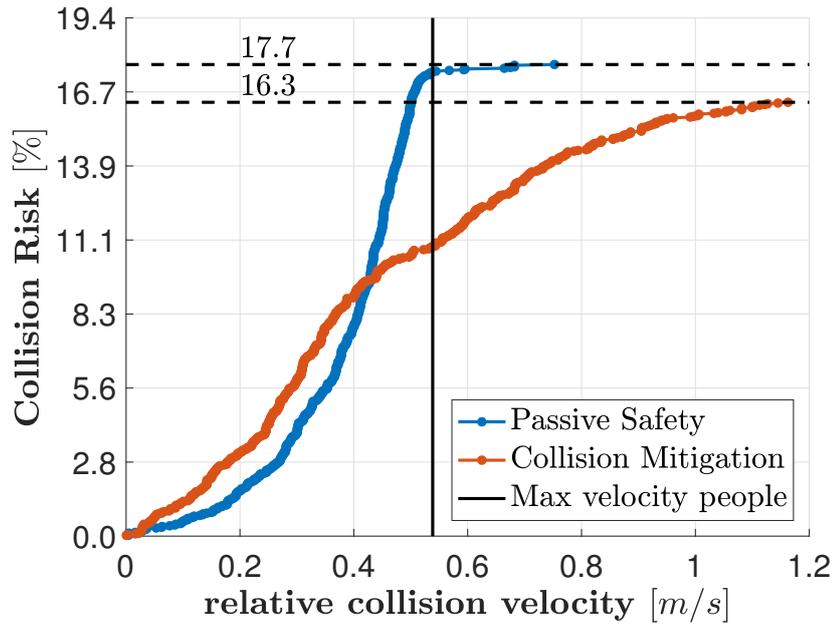
Figure 6.3: The figure shows a simulation for **PS** where the speed of the robot is not 0 at collision time, but it is converging to 0.



The goal of **CM** is to postpone collisions as much as possible. We described in Section 6.4

that there is a relationship between this goal and the minimization of the average velocity of the robot in the direction of the estimated position of the person inside the unsafe zone. This velocity is the closest component to the collision velocity of the robot, and it was expected that this minimization would help the robot to not actively collide with people and even reduce further the relative collision velocity. This minimization however did not have the effect expected. Our hypothesis is that the robot controlled with **CM** prefers to postpone collisions with stronger impact later in time.

Figure 6.4: The figure shows the cumulative plot of the collision risk in percentage along the relative collision velocity of all simulations for **PS** and **CM**.



6.7 Conclusions

PS has been criticized because while the robot makes sure to always have time to stop before a collision happens, this time might not be enough for people around to actually react and avoid the collision once the robot has stopped [Macek 2008]. We introduced a variable called Time To React or *TTR*: the time left before the robot enters the unsafe zone (where collisions could eventually happen) when following a particular motion.

TTR improves in time thanks to the conservativeness in time of our model for the unsafe zone: some part of the zone can become safe later, when the uncertainty is reduced with new observations of the surrounding environment. We showed how the set of all possible motions that achieve **PS** is a subset of the set of all possible motions of a finite duration starting from the current state of the robot. The robot can then achieve a better *TTR* if it is not imposed to be at rest before entering the unsafe zone. For this reason, we propose a new **MPC**-based navigation scheme called **CM** that no longer guarantee **PS** but aims at leaving as much *TTR* as possible for the people. We investigated if controlling the robot with **CM** leads to an improvement in collision avoidance when navigating in a crowd with respect to the **MPC**-based biped navigation scheme introduced in Chapter 4, that guarantees **PS**. Our results show that **CM** outperforms the control scheme that guarantees **PS** in terms of collisions: collisions happen less often and later in time (people are left with more time to react and avoid collisions). But **PS** guarantees that the robot does not actively collide with people. With **CM** instead the robot actively collides with people for a significant amount of collisions. **CM** is worse than **PS** in terms of impact.

Chapter 7

Prioritizing People and Jostling

7.1 Introduction

In this Chapter, we start by proposing a variant of the crowd scenario: we equally divide the members of the crowd in robots and people. In this variant of the crowd scenario, we propose a new MPC-based biped navigation scheme that first aims at leaving as much *TTR* as possible for the people and then, if possible, for the other robots (Section 7.2). When the biped robot distinguishes between robots and people, we measure if controlling the robot with this new scheme leads to an improvement in collision avoidance with respect to CM.

Next, we propose another situation for the robot in the crowd scenario (populated with only people). We suppose the robot must reach a target location at the utmost important, and this time people might obstruct the motion of the robot. We propose another MPC-based biped navigation scheme for this situation that enables the robot to jostle people of the crowd if necessary to reach the target location (Section 7.3). We measure if controlling the robot with this new scheme leads to an improvement in collision avoidance when navigating in a crowd with respect to a robot that does not enable jostling but only aims to reach the target location as soon as possible.

The challenging scenario of a biped robot walking against an inattentive crowd is used for our evaluation purposes. In this scenario, we compare the number of collisions (collision risk) and collision time of both navigation strategies. These results obtained using simulated crowd scenarios are finally presented and discussed in Sections 7.5.1-7.5.2.

7.2 Prioritizing people in a mixed crowd

Suppose the crowd is equally divided in robots and people, and our biped robot (that is traversing the crowd) can distinguish between them in its FoV. We want the robot to aim at Goal 6.2.2 first with respect to people and then, if possible, with respect to other robots. To do so, we first mitigate collision with respect to people, and then with the other robots. This is represented by the Hierarchy 2.

7.3 Jostling

There are situations where the robot must reach a location at the utmost importance, *e.g.* for a fire fighter robot [COL 2017], but people in a crowd can obstruct the motion of the robot. We can deal with these situations with Hierarchy 3. The difference with Hierarchy 1-2 is that minimizing the deviation from a target location the robot should reach at the end of the horizon

$$\|\mathbf{c}_{\text{ref}} - \mathbf{c}_{(i+N|i)}\|^2 \quad (7.1)$$

has higher priority than aiming at Goal 6.2.2. The effect is that the robot will *jostle* people of the crowd if necessary to reach the target location. It will mitigate collisions but only as far as it does not impede reaching the target location.

Hierarchy 2: Prioritize People

- 1: Capturable walking motion (2.57)-(2.58)-(2.66)
- 2: Minimize penetration unsafe zone (6.20) for People at t_{i+1}
- ⋮
- N+1: Minimize penetration unsafe zone (6.20) for People at t_{i+N}
- N+2: Minimize penetration unsafe zone (6.20) for Robot at t_{i+1}
- ⋮
- 2N+1: Minimize penetration unsafe zone (6.20) for Robot at t_{i+N}
- 2N+2: Minimize reference deviation (4.36a)

Hierarchy 3: Jostling

- 1: Capturable walking motion (2.57)-(2.58)-(2.66)
- 2: Minimize reference deviation: reach the target location (7.1)
- 3: Minimize penetration unsafe zone (6.20) at t_{i+1}
- ⋮
- N+2: Minimize penetration unsafe zone (6.20) at t_{i+N}

7.4 Simulation parameters and settings

The parameters

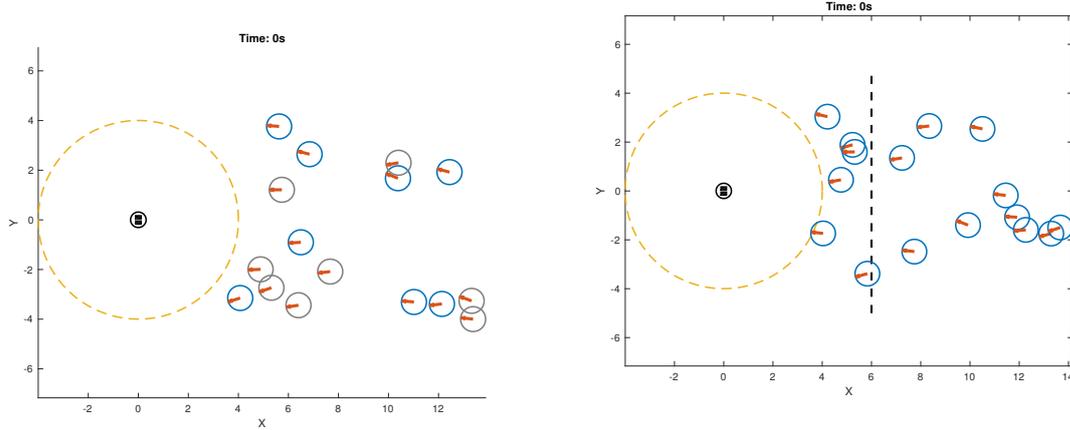
- of the crowd are in Table 5.1a.

- of the robot are in Table 5.1b, and the FoV of the robot is 4[m].
- of the MPC are in Table 5.1c but we limit the experiments to $M = 1$ ($\phi = 0$).

We use the crowd settings as explained in Section 5.3.1.

Figure 7.1: Two variants of the inattentive crowd scenario, with people prioritization (left), and jostling (right).

- (a) Time t_0 in a crowd of size $Z = 16$. The crowd is equally divided in robots (gray circles) and people (blue circles), and our biped robot, in black, can distinguish if someone entering the FoV is a robot or a person. The robot is asked to walk to the right with $\dot{c}_r^x = 0.5$ [m/s].
- (b) Time t_0 in a crowd of size $Z = 16$. The robot must reach a target location (vertical dashed black line) but people in a crowd can obstruct the motion of the robot.



7.5 The effect of prioritizing people and jostling on the crowd scenario

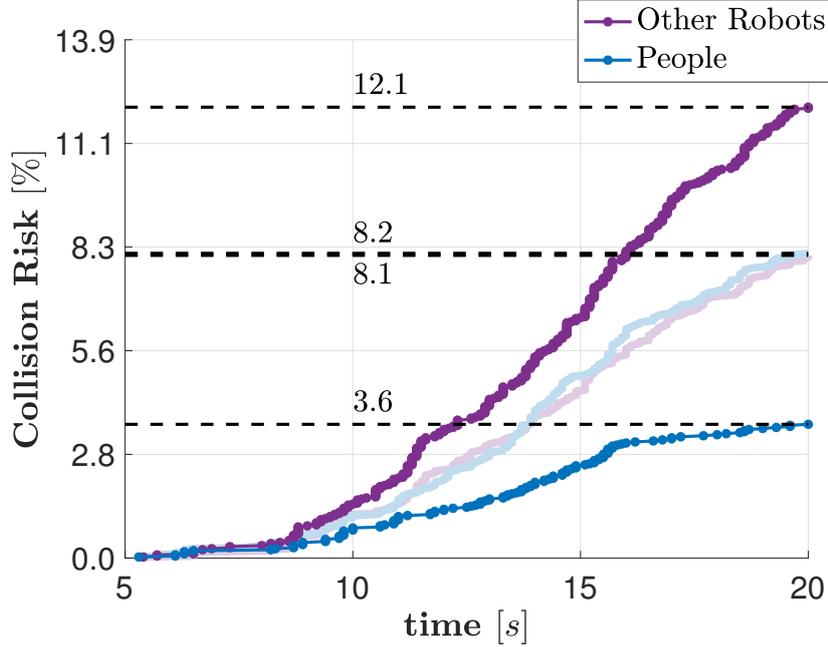
The effect of prioritizing people and jostling for a biped robot traversing a moving crowd is evaluated numerically. To evaluate the prioritization of people, we use a variant of the crowd scenario: the crowd is equally divided in robots and people both inattentive to the presence of our biped robot, as seen in Fig. 7.1a. To evaluate jostling we consider the situation where the robot must reach a target location inside the moving crowd as represented in Fig. 7.1b, and the crowd composed by people only.

The collision risk (Definition 6.2.1) counts the number of collision events (or failure): when we stop the simulation because people or other robots collide with our biped robot in 20s or less. The initial positions of all members of a crowd is randomly chosen in front of our biped robot outside the FoV. For each uncertainty ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$), we simulate 100 crowds of Z people and robots.

7.5.1 Results for prioritizing people

We compare in the crowd scenario a robot controlled with CM and the same robot controlled with Hierarchy 2 in the variant of the inattentive crowd scenario represented in Fig. 7.1a. We compute the cumulative collision risk along the simulation time, showed in Figure 7.2. The cumulative (frequency) plot shows the percentage [%] out of 3600 crowd scenarios, varying all the possible combination of uncertainty ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$) for a choice of crowd size Z ,

Figure 7.2: The figure shows the cumulative plot of collision risk in percentage along the time of all simulations for original **CM** (Hierarchy 1), transparent lines, and Hierarchy 2, in solid lines.



of collision risk that are less than or equal to particular time.

When the robot mitigates the risk of collision with other robots and people equally, the collision risk of other robots and people is similar. When the biped robot distinguishes between robots and people, the biped robot collides always later with people and reduces its collision risk with them (by $\approx 5\%$). For the other robots instead, collisions happen earlier and more often (collision risk is increased by $\approx 4\%$). The overall collision risk in this variant of inattentive crowd scenario was actually reduced with the choice of prioritizing people (from 16.3% to 15.7%). When it is possible to distinguish between robots and people, the robot fulfills better Goal 6.2.1 with respect to people if it first mitigates collision with them and then if possible with other robots. Hierarchy 2 is better than **CM** in this regard.

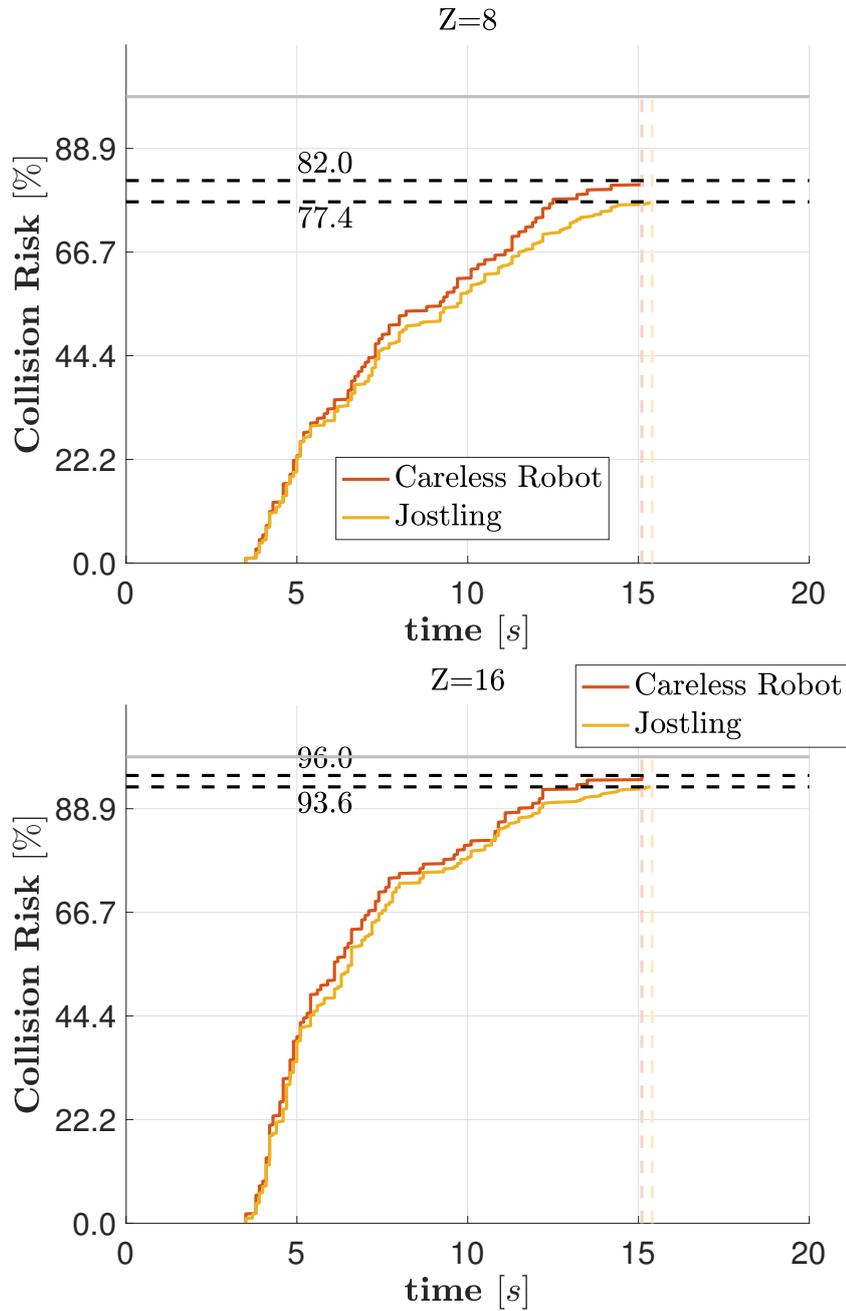
7.5.2 Results for jostling

Hierarchy 4: Careless

- 1: Capturable walking motion (2.57)-(2.58)-(2.66)
- 2: Minimize reference deviation: reach the target location (7.1)

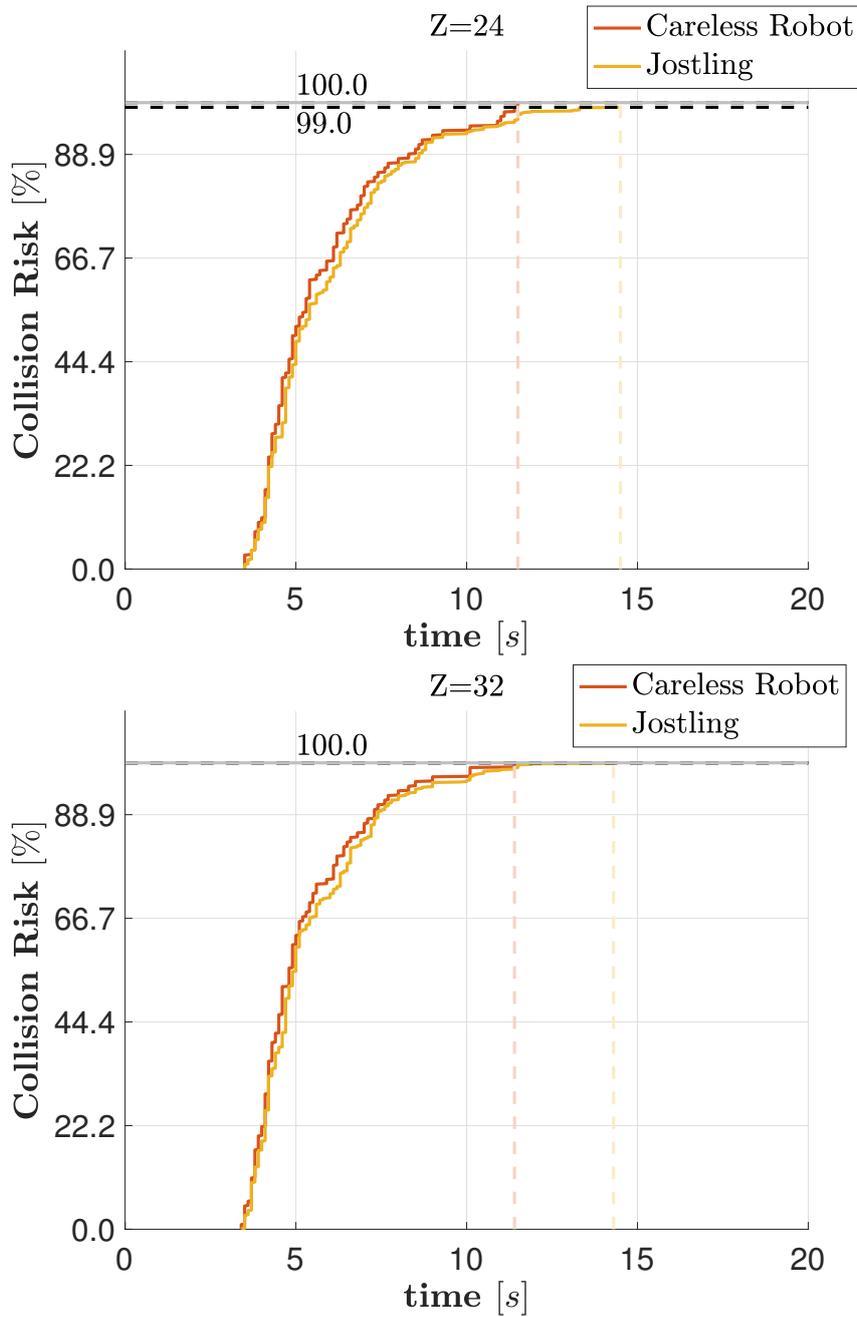
We compare a robot controlled with Hierarchy 3 and the same robot controlled with Hierarchy 4. The robot controlled with the latter hierarchy aims only to reach the target location as fast as possible without trying to avoid collisions with the surrounding people, we therefore call it *careless* robot. We compute the cumulative collision risk along the simulation time, showed in Figure 7.3-7.4. The cumulative (frequency) plot shows the percentage [%] out of 900 crowd scenarios for a crowd size Z , varying all the possible combination of uncertainty ($\|\tilde{z}^j\|, \|\tilde{z}^j\|$), of collision risk that are less than or equal to particular time.

Figure 7.3: The figure shows the cumulative plot of collision risk in percentage along the time of all simulations of *small* crowd sizes for a robot that does not care about avoiding colliding with people (Hierarchy 4) and a robot that enables jostling (Hierarchy 3).



In all cases, whereas walking (without falling) to reach a target location is at utmost importance, we observed that by introducing jostling for the robot, the collision risk decreases and collisions are postponed later in time. In this scenario whatever crowd density, collisions could not be postponed later than ≈ 15 s. The jostling robot is always at least as good, if not better than the careless robot with respect to Goal 6.2.1. This is of course related to the fact that mitigating collision risk has a lower priority than reaching a location in the case of *jostling* robot, or it is not asked at all to the *careless* robot.

Figure 7.4: The figure shows the cumulative plot of collision risk in percentage along the time of all simulations of *big* crowd sizes for a robot that does not care about avoiding colliding with people (Hierarchy 4) and a robot that enables jostling (Hierarchy 3).



Obviously, collision risk increases when we increase the density of the crowd Z , that the robot needs to traverse. Introducing jostling when the robot needs to traverse a crowd of few people $Z = 8$, the collision risk decreased significantly by 5%, and 3% for $Z = 16$. When the density of the crowd increases both strategies basically collide in all the simulations (100% collision risk) but yet the jostling robot postpones collisions later than the careless robot. If we increase further the density of the crowd Z , both jostling and careless robot might equally perform in all crowd scenarios because the robot might not have enough space to jostle among people.

7.6 Conclusions

In this Chapter, we introduced a variant of the crowd scenario: we equally divide the members of the crowd in robots and people. In this variant of the crowd scenario, we proposed a new MPC-based biped navigation scheme that first aims at leaving as much *TTR* as possible for the people and then, if possible, for the other robots. When the biped robot distinguishes between robots and people, we investigated if controlling the robot with this new scheme leads to an improvement in collision avoidance when navigating in a crowd with respect to CM. Our results show that with CM the robot collides with other robots and people equally. On the other hand, when we control the robot with the new scheme the biped robot collides always later with people and reduces its collision risk with them. For the other robots instead, collisions happen earlier and more often. The new scheme takes priorities into account and that the simulation results reflect these priorities. The new scheme outperforms CM in terms of collisions.

We also introduced another situation for the robot in the crowd scenario (that is no longer populated by other robots but only people). We suppose the robot must reach a target location at the utmost important, and in this case people of the crowd can obstruct the motion of the robot. We proposed another MPC-based biped navigation scheme for this situation that enable the robot to jostle people of the crowd if necessary to reach the target location. We investigated if controlling the robot with this new scheme leads to an improvement in collision avoidance when navigating in a crowd with respect to a robot that does not enable jostling but only aims to reach the target location as soon as possible. Our results show that the robot collides less and collisions are postponed in time when we enable jostling. Obviously, when we increase the density of the crowd that the robot needs to traverse, the robot that enables jostling starts to collide more often but yet less and later than the robot that only aims to reach the target location as soon as possible. If we increase further the density of the crowd, both robots might equally perform in terms of collisions. This is because the robot might not have enough space to jostle among people.

Chapter 8

Conclusions

8.1 Summary

This thesis contributes to the problem of balance preservation for biped locomotion and motion safety capability (the capability to avoid collisions) in a crowd.

In Chapter 2, we started the dissertation by presenting a linear dynamical model that relates the **CoM** of the robot to the contact forces applied on the ground. We model changes of the foot positions on the ground in order to facilitate their automatic adjustment. With those models, we build a set of constraints that the robot needs to satisfy to comply with the whole-body kinematic and dynamical structure. We additionally imposed a capturability constraint that guarantees the robot is able to stop in a few steps without falling and this is enough to guarantee that it is able to simply avoid falling. Models and constraints were employed in a single **MPC** scheme: an iterative control process which computes explicitly the actions of the robot and their consequences over a limited time horizon.

When the robot is not really planning to stop but actually consider making a new step at the end of the horizon to continue walking, the guarantee to avoid falling is lost. We investigated this issue in Chapter 3. In the same chapter we provided a numerical evidence that, despite the sudden plan change, the guarantee to avoid falling is not lost. But the guarantee depends on the length of the time horizon.

In Chapter 4 we addressed the problem of motion safety capability in a crowd. Pedestrian's future behavior is crucial for this capability. We chose to consider all the possible pedestrian's future trajectories, using a conservative model of the future. Thanks to this model we can guarantee *Passive Safety* (**PS**): the robot is able to stop before a collision occurs. If a collision is inevitable, at least the robot will be at rest when that happens. No collision would happen if everybody behaved that way, so in a sense the robot will have done its share. Capturability was used to successfully control the movement of a humanoid robot in a crowd and to guarantee both fall avoidance and **PS** in a single **MPC** scheme.

Most existing walking strategies propose to re-plan the walking motion, adapting to changing environments only once at every step. In contrast, the **MPC** scheme outlined above re-plans the walking motion not only at each step initiation but also in between (8 times per step). In Chapter 5, we show that we can favor to re-plan the walking motion to adapt to changing environments once per step, instead of 2, 4 or 8 times per step. We thus save computational power without deteriorating the robot's motion safety capability. But re-planning should only happen just before initiating the next step, or during the time that the robot has only one step in contact with the ground.

PS has been criticized because while the robot makes sure to always have time to stop before a collision happens, this time might not be enough for people around to actually react and avoid the collision once the robot has stopped. Based on the criticism of **PS**, we conveyed

that people around the robot could potentially attempt to react and avoid collisions if they have enough time to do so. For this reason, we claimed the following: more time to react for the surrounding environment reduces the number of collisions. In Chapter 6, we proposed to control the movement of a humanoid robot in a crowd that guarantee fall avoidance while aiming to leave as much time to react as possible for the people (instead of guarantee PS) in a single MPC scheme, that we call *Collision Mitigation (CM)*. When controlling the movement of a humanoid robot in a crowd, we can favor CM instead of PS to reduce the number of collisions and postpones these collisions later in time.

We considered the case where a humanoid robot moves in a crowd and the members of this crowd are equally divided in robots and people. In Chapter 7 we proposed a new MPC scheme guarantees fall avoidance and while aiming to leave as much time to react as possible for the people and then, if possible, for the other robots. When this revised version of CM takes priorities into account, *e.g.* prioritizing people, the simulation results reflect these priorities. The biped robot collides always later with people and reduces the number of collisions with them. For the other robots instead, collisions happen earlier and more often.

We then considered another situation for the robot moving in a crowd (populated with only people). We suppose the robot must reach a target location at the utmost important, and this time people might obstruct the motion of the robot. In this situation we proposed another new MPC scheme guarantees fall avoidance while enabling the robot to jostle people of the crowd if necessary to reach the target location. We compare this scheme in the situation outlined above with a robot that does not enable jostling but only aims to reach the target location as soon as possible. Our results show that the robot collides less and collisions are postponed in time when we enable jostling. However, If we increase the density of the crowd, both robots converges to equally perform in terms of collisions. This is because the robot might not have enough space to jostle among people.

8.2 Future Work

We discussed few research directions Section 3.7 and 5.5. We conclude by proposing an additional research direction for strong recursive feasibility in MPC of biped walking.

The notion of SRF (Definition 2.4.1) demands that for any feasible state-control pair, a feasible state is reached. In the literature of MPC, we can find a “stronger” notion called *Strong Forward Invariance (SFI)* [Grüne 2012, Section 5]: for any feasible state-motion pair, a feasible state is reached. In between, we can introduce M -actions Strong Recursive Feasibility, or M -SRF in short (where M is the *planning period* introduced in Chapter 5).

Definition 8.2.1 (M -actions Strong Recursive Feasibility). *An MPC scheme is M -actions strongly recursive feasible if and only if*

$$\begin{aligned} \forall i, \forall (\mathbf{x}_i, \{\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_M\}) \in \mathcal{K}_i^{M|N}, \\ \mathbf{x}_{i+M} = \mathbf{A}^M \mathbf{x}_i + \sum_{j=1}^M \mathbf{A}^{M-j} \mathbf{B} \boldsymbol{\kappa}_j \in \mathcal{X}_{i+M}^N. \end{aligned} \quad (8.1)$$

This definition introduces another projection of the feasible set of solution \mathcal{F}_i^N :

$$\mathcal{K}_i^{M|N} \triangleq \{(\mathbf{x}_i, \{\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_M\}) \mid \exists \boldsymbol{\pi}_i^N, (\mathbf{x}_i, \boldsymbol{\pi}_i^N) \in \mathcal{F}_i^N, \boldsymbol{\kappa}_j = \mathbf{u}_{(i+j-1|i)}, j \in [1, M]\}, \quad (8.2)$$

the set of state and M -actions pairs of a feasible motion. Definition 8.2.1 corresponds to the standard SRF when $M=1$, and corresponds to SFI when $M=N$. The following lemma shows SFI implies SRF, and in general: when the MPC is SFI, then it is also $(N-1, \dots, 1)$ -SRF.

Lemma 8.2.1 (M -SRF implies $(M - 1)$ -SRF). *if an MPC scheme is M -actions strongly recursive feasible then it is also $(M - 1)$ -actions strongly recursive feasible, where $M \in \{2, 3, \dots, N\}$.*

Proof. From any pair $(\mathbf{x}_i, \{\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_M\}) \in \mathcal{K}_i^{M|N}$ execute the first $M - 1$ actions, and reach a state \mathbf{x}_{i+M-1} . This state is feasible ($\mathbf{x}_{i+M-1} \in \mathcal{X}_{i+M-1}^N$) and the MPC scheme is $(M - 1)$ -SRF since there exists at least one feasible $\boldsymbol{\pi}_{i+M-1}^N$ associated with it. For example

$$\boldsymbol{\pi}_{i+M-1}^N = \underbrace{\{\mathbf{u}_{(i+M|i)}, \mathbf{u}_{(i|i+M)}\}}_{\in \boldsymbol{\pi}_i^N}, \underbrace{\{\mathbf{u}_{(i+1|i+M)}, \dots, \mathbf{u}_{(i+N-2|i+M)}\}}_{\in \boldsymbol{\pi}_{i+M}^N}, \quad (8.3)$$

in which $\boldsymbol{\pi}_{i+M}^N$ exists since the MPC scheme is M -SRF. \square

It is then preferable SFI property over SRF for the MPC of biped walking. When the robot is planning to stop SFI is guaranteed by *construction*, but the guarantee is lost when the robot actually consider to make a new step at the end of the horizon. This is the exact same issue as explained in Chapter 3 and it could be investigated numerically with our proposed method. As a consequence, SFI would probably depend on the length of the planning horizon.

Appendix A

Subtleties on the implementation of MPC for biped robots

A.1 Introduction

In this Appendix, we focus on 3 aspects about the implementation of *Model Predictive Control* (MPC) for biped robots. The first aspect (Section A.3) is taken from [Sherikov 2016, Section 4.2.2.1]. We discuss about the variation of the CoP during the sampling interval. This variation depends on the discrete model that describes the CoM dynamics. The second and third aspect (Section A.4 and A.5) relate to the work in [Herdt 2010]: the MPC with automatic footstep placement. We discuss how the step selection matrices cycle for the automatic footstep placement and how many steps are planned based on the choice of the planning horizon. Last, we show the effect of the DS kinematic constraints in our MPC scheme.

A.2 Derivation of the Discrete Time CoM Model

The analysis of this section is valid for both $\{x, y\}$ coordinates, but for sake of simplicity we consider only one. We want to calculate the matrix exponential

$$\mathbf{A} = e^{\mathbf{G}T} \quad (\text{A.1})$$

where

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix}. \quad (\text{A.2})$$

We expand the exponential up to the third term as

$$e^{\mathbf{G}T} = \mathbf{I} + \mathbf{A}T + \frac{(\mathbf{A}T)^2}{2!} + \frac{(\mathbf{A}T)^3}{3!} + \frac{(\mathbf{A}T)^4}{4!} + \dots \quad (\text{A.3})$$

and we have:

$$(\mathbf{A}T)^2 = \mathbf{A}T\mathbf{A}T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix} T^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \omega^2 & 0 \\ 0 & 0 & \omega^2 \end{bmatrix} T^2 = \begin{bmatrix} 0 & 0 & T^2 \\ 0 & \omega^2 T^2 & 0 \\ 0 & 0 & \omega^2 T^2 \end{bmatrix}. \quad (\text{A.4})$$

and

$$(\mathbf{A}T)^3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \omega^2 & 0 \\ 0 & 0 & \omega^2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix} T^3 = \begin{bmatrix} 0 & \omega^2 & 0 \\ 0 & 0 & \omega^2 \\ 0 & \omega^4 & 0 \end{bmatrix} T^3 = \begin{bmatrix} 0 & \omega^2 T^3 & 0 \\ 0 & 0 & \omega^2 T^3 \\ 0 & \omega^4 T^3 & 0 \end{bmatrix}. \quad (\text{A.5})$$

and

$$(\mathbf{AT})^4 = \begin{bmatrix} 0 & \omega^2 T^3 & 0 \\ 0 & 0 & \omega^2 T^3 \\ 0 & \omega^4 T^3 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \omega^2 & 0 \end{bmatrix} T^4 = \begin{bmatrix} 0 & 0 & \omega^2 \\ 0 & \omega^4 & 0 \\ 0 & 0 & \omega^4 \end{bmatrix} T^4 = \begin{bmatrix} 0 & 0 & \omega^2 T^4 \\ 0 & \omega^4 T^4 & 0 \\ 0 & 0 & \omega^4 T^4 \end{bmatrix}. \quad (\text{A.6})$$

Back to the exponential:

$$e^{\mathbf{GT}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & T & 0 \\ 0 & 0 & T \\ 0 & \omega^2 T & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{T^2}{2} \\ 0 & \frac{\omega^2 T^2}{2} & 0 \\ 0 & 0 & \frac{\omega^2 T^2}{2} \end{bmatrix} + \begin{bmatrix} 0 & \frac{\omega^2 T^3}{3!} & 0 \\ 0 & 0 & \frac{\omega^2 T^3}{3!} \\ 0 & \frac{\omega^4 T^3}{3!} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{\omega^2 T^4}{4!} \\ 0 & \frac{\omega^4 T^4}{4!} & 0 \\ 0 & 0 & \frac{\omega^4 T^4}{4!} \end{bmatrix} + \dots \quad (\text{A.7})$$

$$e^{\mathbf{GT}} = \begin{bmatrix} 1 & T + \frac{\omega^2 T^3}{3!} + \dots & \frac{T^2}{2} + \frac{\omega^2 T^4}{4!} + \dots \\ 0 & 1 + \frac{\omega^2 T^2}{2} + \frac{\omega^4 T^4}{4!} + \dots & T + \frac{\omega^2 T^3}{3!} + \dots \\ 0 & \omega^2 T + \frac{\omega^4 T^3}{3!} + \dots & 1 + \frac{\omega^2 T^2}{2} + \frac{\omega^4 T^4}{4!} + \dots \end{bmatrix} \quad (\text{A.8})$$

Let's analyse the second element in the first row: (1,2), of (A.8). We have

$$T + \frac{\omega^2 T^3}{3!} + \dots = \frac{1}{\omega} \left(\omega T + \frac{\omega^3 T^3}{3!} + \dots \right). \quad (\text{A.9})$$

Based on the definition of the exponential expansion (A.3), we have

$$\begin{aligned} T + \frac{\omega^2 T^3}{3!} + \dots &= \frac{1}{\omega} \left(\frac{e^{T\omega} - e^{-T\omega}}{2} \right) \\ &= \frac{1}{\omega} \sinh(T\omega). \end{aligned} \quad (\text{A.10})$$

Consider now (1,3) of (A.8). We have

$$\begin{aligned} \frac{T^2}{2} + \frac{\omega^2 T^4}{4!} + \dots &= \frac{1}{\omega^2} \left(\frac{T^2}{2} + \frac{\omega^2 T^4}{4!} + \dots \right) \\ &= \frac{1}{\omega^2} \left(1 + \frac{T^2}{2} + \frac{\omega^2 T^4}{4!} + \dots - 1 \right) \\ &= \frac{1}{\omega^2} \left(\frac{e^{\omega T} + e^{-\omega T}}{2} - 1 \right) \\ &= \frac{1}{\omega^2} (\cosh(T\omega) - 1) \\ &= \frac{\cosh(T\omega)}{\omega^2} - \frac{1}{\omega^2}. \end{aligned} \quad (\text{A.11})$$

Consider now (2,2) or (3,3) of (A.8). We have

$$\begin{aligned} 1 + \frac{\omega^2 T^2}{2} + \frac{\omega^4 T^4}{4!} + \dots &= \frac{e^{\omega T} + e^{-\omega T}}{2} \\ &= \cosh(T\omega). \end{aligned} \quad (\text{A.12})$$

Consider now (2,3) of (A.8). We have

$$\begin{aligned} T + \frac{\omega^2 T^3}{3!} + \dots &= \frac{1}{\omega} \left(\omega T + \frac{\omega^3 T^3}{3!} + \dots \right) \\ &= \frac{1}{\omega} \left(\frac{e^{\omega T} - e^{-\omega T}}{2} \right) \\ &= \frac{1}{\omega} \sinh(T\omega). \end{aligned} \quad (\text{A.13})$$

Consider now (3,2) of (A.8). We have

$$\begin{aligned} \omega^2 T + \frac{\omega^4 T^3}{3!} + \dots &= \omega \left(\omega T + \frac{\omega^3 T^3}{3!} + \dots \right) \\ &= \omega \left(\frac{e^{\omega T} - e^{-\omega T}}{2} \right) \\ &= \omega \sinh(T\omega). \end{aligned} \tag{A.14}$$

We ends up with

$$e^{\mathbf{G}T} = \begin{bmatrix} 1 & \sinh(T\omega)/\omega & \cosh(T\omega)/\omega^2 - 1/\omega^2 \\ 0 & \cosh(T\omega) & \sinh(T\omega)/\omega \\ 0 & \sinh(T\omega)\omega & \cosh(T\omega) \end{bmatrix}. \tag{A.15}$$

Now we calculate the following:

$$\mathbf{B} = \left(\int_0^T e^{\mathbf{G}t} dt \right) \mathbf{H}, \tag{A.16}$$

where

$$\mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ -\omega^2 \end{bmatrix}. \tag{A.17}$$

We need to simply integrate the matrix $e^{\mathbf{G}T}$ componentwise.

$$\begin{aligned} \int_0^T e^{\mathbf{G}t} dt &= \\ &= \int_0^T \begin{bmatrix} 1 & \sinh(t\omega)/\omega & \cosh(t\omega)/\omega^2 - 1/\omega^2 \\ 0 & \cosh(t\omega) & \sinh(t\omega)/\omega \\ 0 & \sinh(t\omega)\omega & \cosh(t\omega) \end{bmatrix} dt \\ &= \begin{bmatrix} \int_0^T 1 & \int_0^T \sinh(t\omega)/\omega & \int_0^T (\cosh(t\omega)/\omega^2 - 1/\omega^2) \\ \int_0^T 0 & \int_0^T \cosh(t\omega) & \int_0^T \sinh(t\omega)/\omega \\ \int_0^T 0 & \int_0^T \sinh(t\omega)\omega & \int_0^T \cosh(t\omega) \end{bmatrix} dt \end{aligned} \tag{A.18}$$

Since \mathbf{H} is zero apart from the third component, we can focus only on the last column of the matrix. We have

$$\begin{aligned} \int_0^T (\cosh(t\omega)/\omega^2 - 1/\omega^2) dt &= \frac{1}{\omega^2} \left(\int_0^T (\cosh(t\omega) - 1) dt \right) \\ &= \frac{1}{\omega^2} \left(\int_0^T \cosh(t\omega) dt \right) - \frac{1}{\omega^2} \left(\int_0^T 1 dt \right) \\ &= \frac{\sinh(T\omega)}{\omega^3} - \frac{T}{\omega^2}, \end{aligned} \tag{A.19}$$

and

$$\int_0^T \sinh(t\omega)/\omega = \cosh(T\omega)/\omega^2 - 1/\omega^2, \tag{A.20}$$

and

$$\int_0^T \cosh(t\omega) = \sinh(T\omega)/\omega. \tag{A.21}$$

We ends up with

$$\begin{aligned}
\left(\int_0^T e^{\mathbf{G}t} dt\right) \mathbf{H} &= \begin{bmatrix} \star & \star & \sinh(T\omega)/\omega^3 - T/\omega^2 \\ \star & \star & \cosh(T\omega)/\omega^2 - 1/\omega^2 \\ \star & \star & \sinh(T\omega)/\omega \end{bmatrix} \mathbf{H} \\
&= \begin{bmatrix} \star & \star & \sinh(T\omega)/\omega^3 - T/\omega^2 \\ \star & \star & \cosh(T\omega)/\omega^2 - 1/\omega^2 \\ \star & \star & \sinh(T\omega)/\omega \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\omega^2 \end{bmatrix} \\
&= \begin{bmatrix} T - \sinh(T\omega)/\omega \\ 1 - \cosh(T\omega) \\ -\omega \sinh(T\omega) \end{bmatrix}.
\end{aligned} \tag{A.22}$$

A.3 Center of Pressure position during the time interval

The CoM dynamics (2.16) has been originally proposed as a triple integrator by [Kajita 2003] where CoM jerk, \ddot{c} , is the control input. This was later employed and extended in [Diedam 2008, Herdt 2010, Agravante 2016] and many other works. The triple integrator implies a smooth variation of CoM acceleration. This results in a smooth change of CoP position (2.16), which helps the realization of robot locomotion. This model however leads to some subtle difficulties in terms of realization of robot locomotion.

The analysis of this section is valid for both $\{x, y\}$ coordinates, but for sake of simplicity we consider only one. The triple integrator model controlled by the CoM jerk in discrete time has the following linear form:

$$\begin{bmatrix} c_{i+1} \\ \dot{c}_{i+1} \\ \ddot{c}_{i+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} c_i \\ \dot{c}_i \\ \ddot{c}_i \end{bmatrix} + \mathbf{B} \ddot{c}_i \tag{A.23}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}. \tag{A.24}$$

And the relationship with the CoP is

$$\mathbf{p}_i = \mathbf{D} \begin{bmatrix} c_i \\ \dot{c}_i \\ \ddot{c}_i \end{bmatrix} \tag{A.25}$$

where

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & -\frac{1}{\omega^2} \end{bmatrix}. \tag{A.26}$$

Satisfaction of the constraints on \mathbf{p}_i and \mathbf{p}_{i+1} in the models controlled by the CoM acceleration or jerk does not guarantee their satisfaction during the i -th sampling interval. Let $[c_i, \dot{c}_i, \ddot{c}_i]$ be an initial state, \ddot{c}_i – the constant jerk applied during T , $[c_t, \dot{c}_t, \ddot{c}_t]$ – the state of the system at some $t \in [0, T]$. Position of the CoP during the sampling interval can be found as

$$\mathbf{p}_t = \mathbf{D} \begin{bmatrix} c_t \\ \dot{c}_t \\ \ddot{c}_t \end{bmatrix} = \mathbf{D} \left(\mathbf{A}_t \begin{bmatrix} c_i \\ \dot{c}_i \\ \ddot{c}_i \end{bmatrix} + \mathbf{B}_t \ddot{c}_i \right), \tag{A.27}$$

where

$$\mathbf{A}_t = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_t = \begin{bmatrix} t^3/6 \\ t^2/2 \\ t \end{bmatrix}. \tag{A.28}$$

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

$$p_t = \frac{\ddot{c}_i}{6}t^3 + \frac{\dot{c}_i}{2}t^2 + \left(\dot{c}_i - \ddot{c}_i \frac{1}{\omega^2}\right)t - \dot{c}_i \frac{1}{\omega^2} + c_i. \quad (\text{A.29})$$

Similarly, this dependence is quadratic in the case of a second order model controlled by the **CoM** acceleration [Bohórquez 2017]. The constant jerk \ddot{c}_i applied during T could lead to an overshoot of the **CoP** that violates the limit of the support area. Therefore, satisfaction of the **CoP** constraints at time 0 and T , as is usually enforced by **MPC** schemes, does not guarantee their satisfaction at $t \in (0, T)$. This problem, however, is typically not critical, since the support areas are intentionally shrunk due to the addition of *safety margins* [Wieber 2016]. The size of these margins can be estimated by computing maxima of the polynomial equation (A.29).

The systems controlled by the **CoP** position or its velocity are not subject to this problem. Take the system controlled with **CoP** velocity (2.23) and use the procedure as (A.27), where

$$\mathbf{A}_t = \begin{bmatrix} 1 & \sinh(t\omega)/\omega & \cosh(t\omega)/\omega^2 - 1/\omega^2 \\ 0 & \cosh(t\omega) & \sinh(t\omega)/\omega \\ 0 & \sinh(t\omega)\omega & \cosh(t\omega) \end{bmatrix}, \quad \mathbf{B}_t = \begin{bmatrix} t - \sinh(t\omega)/\omega \\ 1 - \cosh(t\omega) \\ -\omega \sinh(t\omega) \end{bmatrix}. \quad (\text{A.30})$$

During the sampling interval, the **CoP** position at time t depends *linearly* on time t :

$$p_t = p_i + \dot{p}_i t, \quad (\text{A.31})$$

which guarantees the **CoP** does not violate the support area during the sampling interval.

Only at the instant t_{sDS} , the robot is planning 2 steps ahead. The robot starts the **DS** phase and the current step \mathbf{s}^c is considered the contact with the ground supporting the weight of the robot, the planned step ahead are: \mathbf{s}^{pc} and \mathbf{s}_{sDS}^f . For all the other samples of the step cycle, the robot plans only 1 step ahead. The step $\mathbf{s}^c = \mathbf{s}^{pc}$ is considered the contact with the ground supporting the weight of the robot, and \mathbf{s}^f is the planned step ahead.

The number of planned steps ahead along horizon In this implementation, with the simplification made in Section 2.2.3, the step cycle is sampled in (s_d/T) time instants. When $N = (s_d/T)k + j$ with $k \in \mathbb{N}$ and $j \in \{1, \dots, (s_d/T)\}$, the robot plans $\lceil (NT/s_d) \rceil - 1$ steps ahead for $(s_d/T) - j$ time instants along the step cycle, $\lceil (NT/s_d) \rceil$ otherwise.

A.5 Kinematic constraints during the DS phase

Figure A.1a shows the effect of imposing

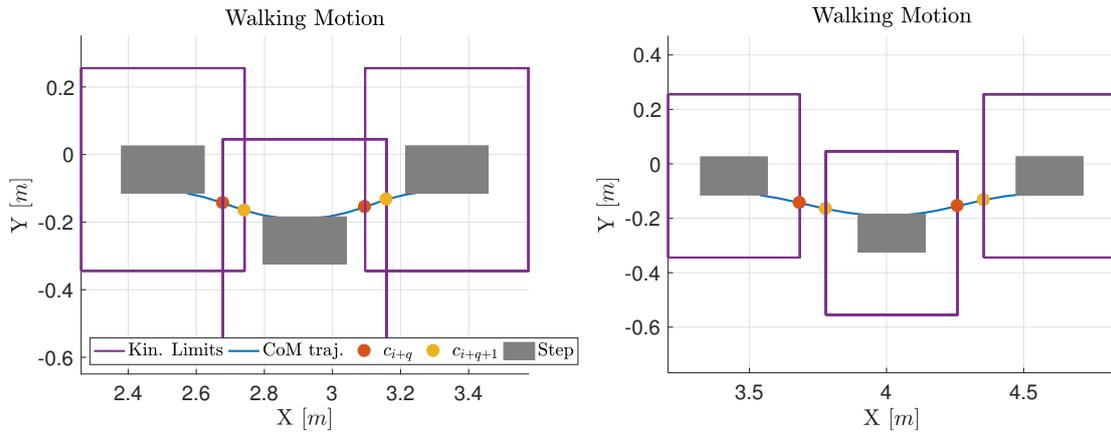
$$\begin{cases} \mathbf{c}_{i+q} \in \mathcal{C}(\mathbf{s}_{j+1}), \\ \mathbf{c}_{i+q+1} \in \mathcal{C}(\mathbf{s}_j) \end{cases} \quad \text{DS Kinematic constraints (Section 2.4.2),} \quad (\text{A.41})$$

along the walking motion to make sure the maximum length of the legs is respected during the **DS**. Where t_{i+q} is the instant the robot starts the **DS** phase, and t_{i+q+1} is the instant the robot ends the **DS** phase.

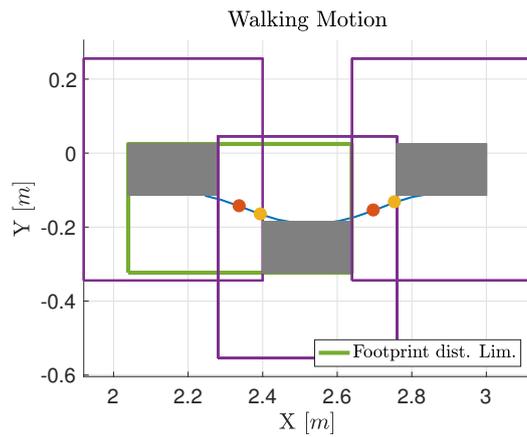
Figure A.1b shows that when we remove these constraints, the maximum length of the legs is not respected during the **DS** phases. We can decide to limit the distance between footprints. The limit acts as (A.41) (Figure A.1c) but it requires hand tuning. This tuning can over-constrain the kinematic capabilities of the biped robot, so we favor for the **DS** Kinematic constraints.

Figure A.1: The figures represent the top view of the same biped robot walking for few footsteps. When we impose the DS kinematic constraints (A.41) along the walking motion of the robot (Figure A.1a), when we remove those constraints (Figure A.1b-A.1c) but we limit the distance between footprints (Figure A.1c).

- (a) DS kinematic constraints ensure the maximum length of the legs is respected during the DS.
- (b) Without the DS kinematic constraints, the maximum length of the legs is *not* respected during the DS.



- (c) Without the DS kinematic constraints, we limit the distance between footprints that ensures the maximum length of the legs is respected during the DS, but they might overconstrain the kinematic capabilities of the biped robot. The figure shows only the limit for the first two footsteps.



Bibliography

- [Agravante 2016] D. J. Agravante, A. Sherikov, P. Wieber, A. Cherubini and A. Kheddar. *Walking pattern generators designed for physical collaboration*. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1573–1578, May 2016. (Cited on pages [10](#), [13](#), [19](#), [32](#), and [81](#).)
- [Agrawal 2017] A. Agrawal and K. Sreenath. *Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation*. In Robotics: Science and Systems, 2017. (Cited on page [45](#).)
- [Al Homsy 2016] S. Al Homsy, A. Sherikov, D. Dimitrov and P. Wieber. *A hierarchical approach to minimum-time control of industrial robots*. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 2368–2374, May 2016. (Cited on pages [41](#) and [61](#).)
- [Althoff 2010] D. Althoff, M. Althoff, D. Wollherr and M. Buss. *Probabilistic collision state checker for crowded environments*. In 2010 IEEE International Conference on Robotics and Automation, pages 1492–1498, May 2010. (Cited on page [37](#).)
- [Althoff 2012] D. Althoff, J. J. Kuffner, D. Wollherr and M. Buss. *Safety assessment of robot trajectories for navigation in uncertain and dynamic environments*. Autonomous Robots, vol. 32, no. 3, pages 285–302, Apr 2012. (Cited on page [35](#).)
- [AN2 2018] *World’s First Autonomous Offshore Robot*. Available: <https://www.anybotics.com/worlds-first-autonomous-offshore-robot/>, 2018. (Cited on page [2](#).)
- [AR2 2019] *Agility robotics and ford motor company announce partnership*. Available: <https://www.agilityrobotics.com/ford-partnership>, 2019. (Cited on page [2](#).)
- [Arechavaleta 2008] G. Arechavaleta, J. 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 [35](#).)
- [Aubin 2009] J.-P. Aubin. *Viability theory*. Springer Science & Business Media, 2009. (Cited on page [17](#).)
- [Batkovic 2018] I. Batkovic, M. Zanon, N. Lubbe and P. Falcone. *A Computationally Efficient Model for Pedestrian Motion Prediction*. In 2018 European Control Conference (ECC), pages 374–379, June 2018. (Cited on page [35](#).)
- [Bautin 2010] A. Bautin, L. Martinez-Gomez and T. Fraichard. *Inevitable Collision States: A probabilistic perspective*. In 2010 IEEE International Conference on Robotics and Automation, pages 4022–4027, May 2010. (Cited on page [37](#).)

- [BDA 2020] *Exploring the potential of robotics in the oil and gas industry.* Available: <https://www.akerbp.com/en/exploring-the-potential-of-robotics-in-the-oil-and-gas-industry/>, 2020. (Cited on page 2.)
- [Bertsimas 1997] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997. (Cited on page 25.)
- [Bissacco 2009] A. Bissacco and S. Soatto. *Hybrid Dynamical Models of Human Motion for the Recognition of Human Gaits.* *International Journal of Computer Vision*, vol. 85, no. 1, pages 101–114, Oct 2009. (Cited on page 35.)
- [Blumentals 2017] A. Blumentals and P.-B. Wieber. *An MPC Approach to Safe Trajectory Generation for an Autonomous Bus*, 2017. (Cited on page 61.)
- [Boccia 2014] A. Boccia, L. Grüne and K. Worthmann. *Stability and feasibility of state constrained MPC without stabilizing terminal constraints.* *Systems & control letters*, vol. 72, pages 14–21, 2014. (Cited on page 30.)
- [Bohorquez 2016] N. Bohorquez, A. Sherikov, D. Dimitrov and P. Wieber. *Safe navigation strategies for a biped robot walking in a crowd.* In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 379–386, Nov 2016. (Cited on pages 3, 17, 19, 24, 35, 38, 45, 49, and 56.)
- [Bohórquez 2017] N. Bohórquez and P. Wieber. *Adaptive step duration in biped walking: A robust approach to nonlinear constraints.* In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pages 724–729, Nov 2017. (Cited on pages 13 and 82.)
- [Bohorquez 2018a] N. Bohorquez. *Design of safe control laws for the locomotion of biped robots.* Theses, Université Grenoble Alpes, December 2018. (Cited on pages 5, 33, and 43.)
- [Bohórquez 2018b] N. Bohórquez and P.-B. Wieber. *Adaptive step rotation in biped walking.* In IROS 2018 - IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 720–725, Madrid, Spain, October 2018. IEEE. (Cited on page 13.)
- [Borwein 1987] J. M. Borwein and P. B. Borwein. *Pi and the agm: A study in analytic number theory and computational complexity.* Wiley, New York, 1987. (Cited on page 25.)
- [Bouguerra 2019] M. Bouguerra, T. Fraichard and M. Fezari. *Viability-Based Guaranteed Safe Robot Navigation.* *Journal of Intelligent and Robotic Systems*, vol. 95, no. 2, pages 459–471, August 2019. (Cited on page 45.)
- [Bouraine 2014] S. Bouraine, T. Fraichard, O. Azouaoui and H. Salhi. *Passively safe partial motion planning for mobile robots with limited field-of-views in unknown dynamic environments.* In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 3576–3582, May 2014. (Cited on pages 3, 17, 35, 37, 38, 45, and 49.)
- [Boyd 2004] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004. (Cited on pages 18, 25, and 41.)

- [Brasseur 2015] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov and P. 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), pages 595–601, Nov 2015. (Cited on page 16.)
- [Brossette 2017] S. Brossette and P. Wieber. *Collision avoidance based on separating planes for feet trajectory generation*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pages 509–514, Nov 2017. (Cited on page 41.)
- [Brscić 2015] D. Brscić, H. Kidokoro, Y. Suehiro and T. Kanda. *Escaping from Children’s Abuse of Social Robots*. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI ’15, pages 59–66, New York, NY, USA, 2015. ACM. (Cited on page 35.)
- [Cao 2019] C. Cao, P. Trautman and S. Iba. *Dynamic Channel: A Planning Framework for Crowd Navigation*. In 2019 International Conference on Robotics and Automation (ICRA), pages 5551–5557, May 2019. (Cited on pages 35 and 36.)
- [Caron 2015] S. Caron, Q. Pham and Y. Nakamura. *Stability of surface contacts for humanoid robots: Closed-form formulae of the Contact Wrench Cone for rectangular support areas*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 5107–5112, May 2015. (Cited on page 16.)
- [Caron 2019] S. Caron, A. Kheddar and O. Tempier. *Stair Climbing Stabilization of the HRP-4 Humanoid Robot using Whole-body Admittance Control*. In 2019 International Conference on Robotics and Automation (ICRA), pages 277–283, May 2019. (Cited on page 49.)
- [Carver 2009] S. G. Carver, N. J. Cowan and J. M. Guckenheimer. *Lateral stability of the spring-mass hopper suggests a two-step control strategy for running*. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 2, page 026106, 2009. (Cited on page 3.)
- [CBP 2018] *CrowdBot project*. Available: <http://crowdbot.eu>, 2018. (Cited on page 2.)
- [Chen 2016] Y. Chen, M. Liu, S.-Y. Liu, J. Miller and J. P. How. *Predictive modeling of pedestrian motion patterns with bayesian nonparametrics*. 2016. (Cited on page 36.)
- [Chen 2017] Y. F. Chen, M. Everett, M. Liu and J. P. How. *Socially aware motion planning with deep reinforcement learning*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1343–1350, Sep. 2017. (Cited on page 43.)
- [Chen 2018] Y. Chen, H. Peng and J. Grizzle. *Obstacle Avoidance for Low-Speed Autonomous Vehicles With Barrier Function*. *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pages 194–206, Jan 2018. (Cited on page 44.)
- [Chestnutt 2005] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins and T. Kanade. *Footstep Planning for the Honda ASIMO Humanoid*. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pages 629–634, April 2005. (Cited on pages 3 and 47.)
- [Choudhury 2014] S. Choudhury, S. Arora and S. Scherer. *The Planner Ensemble and Trajectory Executive: A High Performance Motion Planning System with Guaranteed Safety*. In Proceedings of AHS 70th Annual Forum, Montreal, Quebec, Canada, May 2014. (Cited on page 46.)

- [Ciocca 2017] M. Ciocca, P.-B. Wieber and T. Fraichard. *Strong recursive feasibility in model predictive control of biped walking*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pages 730–735, Nov 2017. (Not cited.)
- [Ciocca 2019] M. Ciocca, P.-B. Wieber and T. Fraichard. *Effect of Planning Period on MPC-based Navigation for a Biped Robot in a Crowd*. In IROS 2019 - IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1–8, Macau, China, November 2019. IEEE. (Not cited.)
- [COL 2017] *Colossus*. Available: <https://robots.ieee.org/robots/colossus/>, 2017. (Cited on page 69.)
- [Dafarra 2018] S. Dafarra, G. Nava, M. Charbonneau, N. Guedelha, F. Andrade, S. Traversaro, L. Fiorio, F. Romano, F. Nori, G. Metta and D. Pucci. *A Control Architecture with Online Predictive Planning for Position and Torque Controlled Walking of Humanoid Robots*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1–9, Oct 2018. (Cited on pages viii, 3, and 49.)
- [Diedam 2008] H. Diedam, D. Dimitrov, P. Wieber, K. Mombaur and M. Diehl. *Online walking gait generation with adaptive foot positioning through Linear Model Predictive control*. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1121–1126, Sep. 2008. (Cited on pages 10, 55, and 81.)
- [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 pages 61 and 62.)
- [DRC 2015] *DARPA Robotics Challenge (DRC)*. Available: https://en.wikipedia.org/wiki/DARPA_Robotics_Challenge, 2015. (Cited on page 2.)
- [Dune 2011] C. Dune, A. Herdt, E. Marchand, O. Stasse, P.-B. Wieber and E. Yoshida. *Vision based control for Humanoid robots*. In IROS Workshop on Visual Control of Mobile Robots (ViCoMoR), pages 19–26, San Francisco, USA, United States, September 2011. (Cited on pages 19, 24, 32, and 49.)
- [Ellis 2009] D. Ellis, E. Sommerlade and I. Reid. *Modelling pedestrian trajectory patterns with Gaussian processes*. In 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pages 1229–1234, Sep. 2009. (Cited on page 36.)
- [Englsberger 2011] J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schäffer and G. Hirzinger. *Bipedal walking control based on Capture Point dynamics*. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4420–4427, Sep. 2011. (Cited on pages 10, 11, and 20.)
- [Englsberger 2014] J. Engelsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid and A. Albu-Schäffer. *Overview of the torque-controlled humanoid robot TORO*. In 2014 IEEE-RAS International Conference on Humanoid Robots, pages 916–923, Nov 2014. (Cited on page 11.)
- [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, vol. 33, no. 7, pages 1006–1028, 2014. (Cited on page 61.)

- [Ethics Commission 2017] Ethics Commission. *Automated and Connected Driving*. Report, Federal Minister of Transport and Digital Infrastructure, 2017. (Cited on page 57.)
- [Faverjon 1987] B. Faverjon and P. Tournassoud. *A local based approach for path planning of manipulators with a high number of degrees of freedom*. Rapport technique RR-0621, INRIA, February 1987. (Cited on page 44.)
- [Feng 2016] S. Feng, X. Xinjilefu, C. G. Atkeson and J. Kim. *Robust dynamic walking using online foot step optimization*. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5373–5378, Oct 2016. (Cited on page 10.)
- [Ferreau 2017] H. Ferreau, A. Potschka and C. Kirches. *qpOASES webpage*. <http://www.qpOASES.org/>, 2007–2017. (Cited on page 19.)
- [Fraichard 2003] T. Fraichard and H. Asama. *Inevitable collision states. A step towards safer robots?* In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), volume 1, pages 388–393 vol.1, Oct 2003. (Cited on pages 17, 45, and 58.)
- [Fraichard 2006] T. Fraichard. *A Short Report about Motion Safety*. Research report, 2006. (Cited on pages 2, 35, 36, and 45.)
- [FUK 2011] *Fukushima Daiichi nuclear disaster*. Available: https://en.wikipedia.org/wiki/Fukushima_Daiichi_nuclear_disaster, 2011. (Cited on page 2.)
- [Fukuda 1996] K. Fukuda and A. Prodon. *Double description method revisited*. In Combinatorics and Computer Science, pages 91–111. Springer Berlin Heidelberg, 1996. (Cited on pages 25 and 31.)
- [Garimort 2011] J. Garimort, A. Hornung and M. Bennewitz. *Humanoid navigation with dynamic footstep plans*. In 2011 IEEE International Conference on Robotics and Automation, pages 3982–3987, May 2011. (Cited on pages 3 and 47.)
- [GMI 2016] *Humanoid Robot Market Size By Product (Wheel Drive, Biped), By Application (Retail, Hospitality, Education & Scientific Research, Healthcare, Residential, Military & Defense, Construction, Underwater Systems), Industry Analysis Report, Regional Outlook, Growth Potential, Price Trends, Competitive Market Share & Forecast, 2017-2024*. Available: <https://www.gminsights.com/industry-analysis/humanoid-robot-market>, 2016. (Cited on page 2.)
- [Gouaillier 2010] D. Gouaillier, C. Collette and C. Kilner. *Omni-directional closed-loop walk for NAO*. In 2010 10th IEEE-RAS International Conference on Humanoid Robots, pages 448–454, Dec 2010. (Cited on page 49.)
- [Grasso 1998] R. Grasso, P. Prévost, Y. P. Ivanenko and A. Berthoz. *Eye-head coordination for the steering of locomotion in humans: an anticipatory synergy*. Neuroscience Letters, vol. 253, no. 2, pages 115 – 118, 1998. (Cited on page 35.)
- [Grüne 2012] L. Grüne. *NMPC without terminal constraints*. IFAC Proceedings Volumes, vol. 45, no. 17, pages 1 – 13, 2012. 4th IFAC Conference on Nonlinear Model Predictive Control. (Cited on pages 30 and 76.)
- [Helbing 1995] D. Helbing and P. Molnár. *Social force model for pedestrian dynamics*. Phys. Rev. E, vol. 51, pages 4282–4286, May 1995. (Cited on page 43.)

- [Herceg 2013] M. Herceg, M. Kvasnica, C. Jones and M. Morari. *Multi-Parametric Toolbox 3.0*. In Proc. of the European Control Conference, pages 502–510, Zürich, Switzerland, July 17–19 2013. <http://control.ee.ethz.ch/~mpt>. (Cited on page 31.)
- [Herdt 2010] 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, vol. 24, no. 5-6, pages 719–737, 2010. (Cited on pages 10, 13, 19, 24, 43, 49, 78, and 81.)
- [HRP 2020] *HRP-2*. Available: <https://robots.ieee.org/robots/hrp2/>, 2020. (Cited on pages viii and 7.)
- [Hun-ok Lim 2002] Hun-ok Lim, Y. Kaneshima and A. Takanishi. *Online walking pattern generation for biped humanoid robot with trunk*. In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), volume 3, pages 3111–3116 vol.3, May 2002. (Cited on page 17.)
- [Hurmuzlu 2004] Y. Hurmuzlu, F. Génot and B. Brogliato. *Modeling, stability and control of biped robots-a general framework*. Automatica, no. 10, pages 1647 – 1664, 2004. (Cited on pages 8 and 9.)
- [IE 2020] *IEEE Xplore*. Available: <https://ieeexplore.ieee.org/Xplore/home.jsp>, 2020. (Cited on page 1.)
- [ISO 2014] ISO. *Robots and robotic devices - Safety requirements for personal care robots*. Standard, International Organization for Standardization (ISO), Geneva, CH, 2014. (Cited on page 46.)
- [Jones 2004] C. Jones, E. C. Kerrigan and J. Maciejowski. *Equality Set Projection: A new algorithm for the projection of polytopes in halfspace representation*. page 45, 2004. (Cited on page 25.)
- [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 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), volume 2, pages 1620–1626 vol.2, Sep. 2003. (Cited on pages 3, 10, 24, 32, 43, and 81.)
- [Kajita 2010] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi. *Biped walking stabilization based on linear inverted pendulum tracking*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4489–4496, Oct 2010. (Cited on pages viii, 3, and 10.)
- [Kajita 2014] S. Kajita, H. Hirukawa, K. Harada and K. Yokoi. Introduction to humanoid robotics, volume 101. Springer, 2014. (Cited on pages viii and 11.)
- [Kanehiro 2008] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida and J.-P. Laumond. *A local collision avoidance method for non-strictly convex polyhedra*. Proceedings of robotics: science and systems IV, 2008. (Cited on page 45.)
- [Kaneko 2004] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi and T. Isozumi. *Humanoid robot HRP-2*. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, volume 2, pages 1083–1090 Vol.2, April 2004. (Cited on pages viii, 7, 11, and 24.)

- [Kaneko 2009] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. Nakaoka and S. Kajita. *Cybernetic human HRP-4C*. In 2009 9th IEEE-RAS International Conference on Humanoid Robots, pages 7–14, Dec 2009. (Cited on page 11.)
- [Kaneko 2011] K. Kaneko, F. Kanehiro, M. Morisawa, T. Tsuji, K. Miura, S. Nakaoka, S. Kajita and K. Yokoi. *Hardware improvement of Cybernetic Human HRP-4C for entertainment use*. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4392–4399, Sep. 2011. (Cited on page 13.)
- [Karkowski 2016] P. Karkowski and M. Bennewitz. *Real-time footstep planning using a geometric approach*. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1782–1787, May 2016. (Cited on pages 3 and 47.)
- [Keller 2014] C. G. Keller and D. M. Gavrila. *Will the Pedestrian Cross? A Study on Pedestrian Path Prediction*. IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 2, pages 494–506, April 2014. (Cited on page 36.)
- [Kerrigan 2001] E. C. Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2001. (Cited on pages 18 and 20.)
- [Khatib 1985] O. Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. In Proceedings. 1985 IEEE International Conference on Robotics and Automation, volume 2, pages 500–505, March 1985. (Cited on pages 43 and 44.)
- [Kheddar 2019] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Engelsberger, M. A. Roa, P. Wieber, F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro and P. Rabaté. *Humanoid Robots in Aircraft Manufacturing: The Airbus Use Cases*. IEEE Robotics Automation Magazine, vol. 26, no. 4, pages 30–45, Dec 2019. (Cited on page 2.)
- [Kimmel 2017] M. Kimmel and S. Hirche. *Invariance Control for Safe Human–Robot Interaction in Dynamic Environments*. IEEE Transactions on Robotics, vol. 33, no. 6, pages 1327–1342, Dec 2017. (Cited on page 44.)
- [Kooij 2014] J. F. P. Kooij, N. Schneider, F. Flohr and D. Gavrila. *Context-Based Pedestrian Path Prediction*. In ECCV, 2014. (Cited on page 36.)
- [Koolen 2012] T. Koolen, T. de Boer, J. Rebula, A. Goswami and J. Pratt. *Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models*. The International Journal of Robotics Research, vol. 31, no. 9, pages 1094–1113, 2012. (Cited on pages 3 and 17.)
- [Krause 2012] M. Krause, J. Engelsberger, P.-B. Wieber and C. Ott. *Stabilization of the Capture Point Dynamics for Bipedal Walking based on Model Predictive Control*. IFAC Proceedings Volumes, vol. 45, no. 22, pages 165 – 171, 2012. 10th IFAC Symposium on Robot Control. (Cited on page 11.)
- [Kruse 2013] T. Kruse, A. K. Pandey, R. Alami and A. Kirsch. *Human-aware robot navigation: A survey*. Robotics and Autonomous Systems, vol. 61, no. 12, pages 1726 – 1743, 2013. (Cited on page 35.)
- [Kudruss 2015] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres and K. Mombaur. *Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations*. In 2015 IEEE-RAS 15th

- International Conference on Humanoid Robots (Humanoids), pages 684–689, Nov 2015. (Cited on page 8.)
- [Kuehn 2016] D. Kuehn. *Design and development of a hominid robot with local control in its adaptable feet to enhance locomotion capabilities*. PhD thesis, Robotics Innovation Center Robert-Hooke-Straße 1 28359 Bremen, Germany, 12 2016. (Cited on page 8.)
- [Kushleyev 2009] A. Kushleyev and M. Likhachev. *Time-bounded lattice for efficient planning in dynamic environments*. In 2009 IEEE International Conference on Robotics and Automation, pages 1662–1668, May 2009. (Cited on pages 36 and 37.)
- [Laumond 2017] J. Laumond, M. Benallegue, J. Carpentier and A. Berthoz. *The Yoyo-Man*. The International Journal of Robotics Research, vol. 36, no. 13-14, pages 1508–1520, 2017. (Cited on page 35.)
- [Lefèvre 2014] S. Lefèvre, D. Vasquez and C. Laugier. *A survey on motion prediction and risk assessment for intelligent vehicles*. ROBOMECH Journal, vol. 1, no. 1, page 1, 2014. (Cited on page 57.)
- [Lerner 2007] A. Lerner, Y. Chrysanthou and D. Lischinski. *Crowds by example*. In Computer graphics forum, volume 26, pages 655–664. Wiley Online Library, 2007. (Cited on page 35.)
- [Liu 2017] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, 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), pages 1351–1357, Sep. 2017. (Cited on pages 35, 37, and 38.)
- [Löfberg 2012] J. Löfberg. *Oops! I cannot do it again: Testing for recursive feasibility in MPC*. Automatica, vol. 48, no. 3, pages 550–555, mar 2012. (Cited on page 25.)
- [Macek 2008] K. Macek, D. Vasquez, T. Fraichard and R. Siegwart. *Safe Vehicle Navigation in Dynamic Urban Scenarios*. In 2008 11th International IEEE Conference on Intelligent Transportation Systems, pages 482–489, Oct 2008. (Cited on pages 4, 46, 57, 59, and 67.)
- [Mayne 2000] D. Mayne, J. Rawlings, C. Rao and P. Scokaert. *Constrained model predictive control: Stability and optimality*. Automatica, vol. 36, no. 6, pages 789 – 814, 2000. (Cited on page 19.)
- [Mayne 2005] D. Mayne, M. Seron and S. Raković. *Robust model predictive control of constrained linear systems with bounded disturbances*. Automatica, vol. 41, no. 2, pages 219 – 224, 2005. (Cited on page 19.)
- [Mayne 2014] D. Q. Mayne. *Model predictive control: Recent developments and future promise*. Automatica, vol. 50, no. 12, pages 2967 – 2986, 2014. (Cited on page 20.)
- [Mercy 2016] T. Mercy, W. Van Loock and G. Pipeleers. *Real-time motion planning in the presence of moving obstacles*. In 2016 European Control Conference (ECC), pages 1586–1591, June 2016. (Cited on page 41.)
- [Mitsch 2013] S. Mitsch, K. Ghorbal and A. Platzer. *On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles*. In P. Newman, D. Fox and D. Hsu, editors, Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24 - June 28, 2013, Berlin, Germany, 2013. (Cited on pages 4 and 60.)

- [Morisawa 2005] M. Morisawa, S. Kajita, K. Harada, K. Fujiwara, F. Kanehiro, K. Kaneko and H. Hirukawa. *Emergency stop algorithm for walking humanoid robots*. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2109–2115, Aug 2005. (Cited on page 46.)
- [MuM 2016] *MuMMER (MultiModal Mall Entertainment Robot)*. Available: <http://www.mummer-project.eu>, 2016. (Cited on page 2.)
- [Muske 1993] K. R. Muske and J. B. Rawlings. *Model predictive control with linear models*. AICHE Journal, vol. 39, no. 2, pages 262–287, 1993. (Cited on page 21.)
- [Nagasaka 2004] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh and J. Yamaguchi. *Integrated motion control for walking, jumping and running on a small bipedal entertainment robot*. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, volume 4, pages 3189–3194 Vol.4, April 2004. (Cited on page 17.)
- [Nava 2018] G. Nava, L. Fiorio, S. Traversaro and D. Pucci. *Position and Attitude Control of an Underactuated Flying Humanoid Robot*. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), pages 1–9, Nov 2018. (Cited on page 8.)
- [Naveau 2017] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Souères. *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 pages 13, 24, and 49.)
- [Nocedal 2006] J. Nocedal and S. Wright. Numerical optimization. Springer Science & Business Media, 2006. (Cited on pages 18 and 44.)
- [Ogata 1995] K. Ogata *et al.* Discrete-time control systems, volume 2. Prentice Hall Englewood Cliffs, NJ, 1995. (Cited on page 12.)
- [Pajon 2019] A. Pajon and P. Wiebe. *Safe 3D Bipedal Walking through Linear MPC with 3D Capturability*. In 2019 International Conference on Robotics and Automation (ICRA), pages 1404–1409, May 2019. (Cited on page 33.)
- [Pellegrini 2009] S. Pellegrini, A. Ess, K. Schindler and L. van Gool. *You'll never walk alone: Modeling social behavior for multi-target tracking*. In 2009 IEEE 12th International Conference on Computer Vision, pages 261–268, Sep. 2009. (Cited on page 35.)
- [Petti 2005] S. Petti and T. Fraichard. *Safe motion planning in dynamic environments*. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2210–2215, Aug 2005. (Cited on page 2.)
- [Pratt 2012] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson and P. Neuhaus. *Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid*. The International Journal of Robotics Research, vol. 31, no. 10, pages 1117–1133, 2012. (Cited on page 14.)
- [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, Oct 1958. (Cited on page 56.)

- [Romualdi 2018] G. Romualdi, S. Dafarra, Y. Hu and D. Pucci. *A Benchmarking of DCM Based Architectures for Position and Velocity Controlled Walking of Humanoid Robots*. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), pages 1–9, Nov 2018. (Cited on page 10.)
- [Rose 2005] J. Rose and J. Gamble. *Human walking*. Lippincott Williams and Wilkins, 2005. (Cited on page 13.)
- [Sardain 2004] P. Sardain and G. Bessonnet. *Forces acting on a biped robot. Center of pressure-zero moment point*. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 34, no. 5, pages 630–637, Sep. 2004. (Cited on page 10.)
- [Schneider 2013] N. Schneider and D. M. Gavrila. *Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study*. In J. Weickert, M. Hein and B. Schiele, editors, *Pattern Recognition*, pages 174–183, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. (Cited on page 35.)
- [Scianca 2016] N. Scianca, M. Cagnetti, D. De Simone, L. Lanari and G. Oriolo. *Intrinsically stable MPC for humanoid gait generation*. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 601–606, Nov 2016. (Cited on pages 10, 17, and 24.)
- [Scianca 2020] N. Scianca, D. De Simone, L. Lanari and G. Oriolo. *MPC for Humanoid Gait Generation: Stability and Feasibility*. IEEE Transactions on Robotics, 2020. (Cited on page 33.)
- [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 page 10.)
- [Shea] R. N. Shea. *Collaborative Robot Technical Specification ISO/TS 15066 Update*. Available: <http://dept.me.umn.edu/courses/me5286/robotlab/Resources/12-TR15066Overview-SafetyforCollaborativeApplications-RobertaNelsonShea.pdf>. (Cited on page 65.)
- [Sherikov 2015] A. Sherikov, D. Dimitrov and P. Wieber. *Balancing a humanoid robot with a prioritized contact force distribution*. In 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pages 223–228, Nov 2015. (Cited on page 61.)
- [Sherikov 2016] A. Sherikov. *Balance preservation and task prioritization in whole body motion control of humanoid robots*. Theses, Université Grenoble Alpes, May 2016. (Cited on pages 3, 8, 9, 11, 13, 15, 19, 49, 55, and 78.)
- [Shimmyo 2013] S. Shimmyo, T. Sato and K. Ohnishi. *Biped Walking Pattern Generation by Using Preview Control Based on Three-Mass Model*. IEEE Transactions on Industrial Electronics, vol. 60, no. 11, pages 5137–5147, Nov 2013. (Cited on page 10.)
- [SHS 2015] *SecondHands: A robot assistant for industrial maintenance*. Available: <https://secondhands.eu>, 2015. (Cited on page 2.)

- [Silva 2018] G. Silva, A. Olivier, A. Crétual, J. Pettré and T. Fraichard. *Human Inspired Effort Distribution During Collision Avoidance in Human-Robot Motion*. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 1111–1117, Aug 2018. (Cited on page 35.)
- [Sontges 2018] S. Sontges, M. Koschi and M. Althoff. *Worst-case Analysis of the Time-To-React Using Reachable Sets*. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1891–1897, 2018. (Cited on page 57.)
- [Sugihara 2002] T. Sugihara and Y. Nakamura. *Whole-body cooperative balancing of humanoid robot using COG Jacobian*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 3, pages 2575–2580 vol.3, Sep. 2002. (Cited on page 15.)
- [Tajima 2009] R. Tajima, D. Honda and K. Suga. *Fast running experiments involving a humanoid robot*. In 2009 IEEE International Conference on Robotics and Automation, pages 1571–1576, May 2009. (Cited on page 17.)
- [Takanishi 1989] A. Takanishi, M. Tochizawa, H. Karaki and I. Kato. *Dynamic Biped Walking Stabilized With Optimal Trunk And Waist Motion*. In Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications, pages 187–192, Sep. 1989. (Cited on page 17.)
- [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 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1084–1091, Oct 2009. (Cited on pages 11 and 17.)
- [Trautman 2010] P. Trautman and A. Krause. *Unfreezing the robot: Navigation in dense, interacting crowds*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 797–803, Oct 2010. (Cited on page 35.)
- [Tsagarakis 2017] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settini, A. Rocchi and A. Ajoudani. *WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments*. Journal of Field Robotics, vol. 34, no. 7, pages 1225–1259, 2017. (Cited on page 49.)
- [Tsai 2014] C. Tsai, J. Hu and M. Tomizuka. *Ensuring safety in human-robot coexistence environment*. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4191–4196, Sep. 2014. (Cited on page 44.)
- [Vasquez Govea 2009] D. A. Vasquez Govea, T. Fraichard and C. Laugier. *Incremental Learning of Statistical Motion Patterns with Growing Hidden Markov Models*. IEEE Transactions on Intelligent Transportation Systems, 2009. (Cited on page 36.)
- [Villa 2019a] N. A. Villa, J. Engelsberger and P.-B. Wieber. *Sensitivity of Legged Balance Control to Uncertainties and Sampling Period*. In IROS 2019 - IEEE/RSJ International Conference on Intelligent Robots and Systems, MACAO, China, November 2019. IEEE. (Cited on pages 10, 11, and 56.)

- [Villa 2019b] N. A. Villa. *Managing Uncertainties in Legged Robots*. PhD thesis, Université Grenoble Alpes, 2019. (Cited on pages 11 and 33.)
- [Wagner 2018] S. Wagner, K. Groh, T. Kuhbeck, M. Dorfel and A. Knoll. *Using Time-to-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving*. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1521–1528, 2018. (Cited on page 57.)
- [Wieber 2002] P.-B. Wieber. *On the stability of walking systems*. In Proceedings of the International Workshop on Humanoid and Human Friendly Robotics, Tsukuba, Japan, 2002. (Cited on pages 2 and 17.)
- [Wieber 2006a] P. Wieber. *Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations*. In 2006 6th IEEE-RAS International Conference on Humanoid Robots, pages 137–142, Dec 2006. (Cited on pages 2, 10, and 43.)
- [Wieber 2006b] 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: Optimization and Feedback Control*, pages 411–425. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. (Cited on pages 8 and 9.)
- [Wieber 2008] P.-B. Wieber. *Viability and Predictive Control for Safe Locomotion*. In IROS 2008 - IEEE-RSJ International Conference on Intelligent Robots & Systems, pages 1103–1108, Nice, France, September 2008. IEEE. (Cited on page 18.)
- [Wieber 2016] P.-B. Wieber, R. Tedrake and S. Kuindersma. *Modeling and Control of Legged Robots*. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1203–1234. Springer International Publishing, Cham, 2016. (Cited on pages 8, 16, and 82.)
- [Wieber 2018] P.-B. Wieber. *Model predictive control*, pages 1–21. Springer Netherlands, Dordrecht, 2018. (Cited on page 18.)
- [Winter 1998] D. A. Winter, A. E. Patla, F. Prince, M. Ishac and K. Gielo-Periczak. *Stiffness control of balance in quiet standing*. *Journal of neurophysiology*, vol. 80, no. 3, pages 1211–1221, 1998. (Cited on page 10.)
- [Wu 2012] A. Wu and J. P. How. *Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles*. *Autonomous Robots*, vol. 32, no. 3, pages 227–242, Apr 2012. (Cited on page 35.)
- [Zaytsev 2015] P. Zaytsev, S. J. Hasaneini and A. Ruina. *Two steps is enough: No need to plan far ahead for walking balance*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 6295–6300, May 2015. (Cited on page 3.)
- [Zheng 2020] P. Zheng, P.-B. Wieber and O. Aycard. *Online optimal motion generation with guaranteed safety in shared workspace*. In 2020 IEEE International Conference on Robotics and Automation, May 2020. (Cited on page 41.)