



ISTITUTO ITALIANO
DI TECNOLOGIA
DYNAMIC LEGGED SYSTEMS



UNIVERSITÀ
DEGLI STUDI
DI GENOVA

Feasible Wrench Based Locomotion Strategy for Legged Robots

Abdelrahman Abdalla

Istituto Italiano di Tecnologia, Italy
Università degli Studi di Genova, Italy

Thesis submitted for the degree of:
Doctor of Philosophy (PhD)

April, 2024

Abdelrahman Abdalla

Feasible Wrench Based Locomotion Strategy for Legged Robots

Doctor of Philosophy (PhD) in Bioengineering and Robotics

Curriculum: Advanced and Humanoid Robotics

Dynamic Legged Systems (DLS) lab, Italian Institute of Technology (IIT), Italy

Tutor:

Dr. Victor Barasuol

Dynamic Legged Systems (DLS) lab, Italian Institute of Technology (IIT), Italy

Dr. Michele Focchi

Dynamic Legged Systems (DLS) lab, Italian Institute of Technology (IIT), Italy

Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Trento, Italy

Principle Advisor:

Dr. Claudio Semini

Dynamic Legged Systems (DLS) lab, Italian Institute of Technology (IIT), Italy

External Examination Committee:

Prof. Andrea Del Prete

Industrial Engineering Department, University of Trento, Trento, Italy

Dr. Stéphane Caron

Institut National de Recherche en Sciences et Technologies du Numérique, University of Modena and Reggio Emilia (INRIA), Paris, France

Abstract

The field of legged robotics has seen impressive developments in the past decade. Significant progress has been made in developing various approaches that enable robots to achieve highly dynamic and stable motions. Legged robots have been used successfully in the industry for inspection applications. However, the potential to use legged robots in interactive applications where the robot can be pushed to its limits is still considerable. In such applications, feasible locomotion under the stringent constraints of terrain interaction and the robot's actuation limits is needed. The objective of this thesis is to present a novel locomotion strategy based on base wrench requirements and feasibility guarantees, which aims to enhance the ability of legged robots to execute complex interactive maneuvers by maximizing the achievable wrenches within the robot's capabilities. The first part of this thesis analyzes different feasibility measures. It develops the *feasible region* and the *feasibility margin* to incorporate our wrench requirements, stability and contact constraints, and actuation limits. We further use machine-learning models to approximate the feasibility margin and provide algorithmic differentiation possibility to compute its gradients. In the second part of the thesis, we leverage the feasibility criteria in developing motion planning strategies. We design trajectory planning methods for the base of the robot as well as for foothold selection. We show that using such simplified criteria can be efficiently used in both improved heuristic and gradient-based optimization approaches. Simulation and experimental results, demonstrated using the Aliengo, HyQ, and HyQReal robots, validate the effectiveness of the *feasibility margin* based strategies in achieving robust locomotion under significant external disturbances and on challenging terrains. Comparisons with traditional heuristic and optimization strategies highlight the importance of considering motion and actuation feasibility in ensuring more stable and feasible locomotion.

Keywords: trajectory optimization, legged robots, locomotion planning, actuation limits

Contents

Abstract	4
Contents	5
List of Figures	7
Acronyms	13
1 Introduction	14
1.1 Motivation	14
1.2 Contributions	15
1.2.1 Feasibility Criteria	15
1.2.2 Feasible Planning	16
1.3 Outline	17
2 Related Work	19
2.1 Dynamic Models	19
2.1.1 Full Rigid Body Dynamics	19
2.1.2 Simplified Dynamics	22
2.1.3 Constraints	24
2.2 Feasibility Criteria	24
2.3 Planning	28
2.3.1 Heuristic vs. Optimization	28
2.3.2 Base and Foothold Optimization	32
2.3.3 Feasible Planning	33
2.3.4 Data-driven approaches	34
3 Feasibility Guarantees for Legged Locomotion	36
3.1 Classical Feasible Region	38

3.1.1	Model	38
3.1.2	Iterative Projection Algorithm	39
3.2	Feasible Region Extensions	43
3.2.1	Generic Plane of Projection	44
3.2.2	External wrenches	45
3.2.3	Dynamic Motions	48
3.2.4	Degenerate Feasible Regions	49
3.3	Reachable Region	51
3.4	The Improved Feasible Region	57
3.5	Feasibility Margin	59
3.6	Validation	61
3.6.1	Kinematic Feasibility	61
3.6.2	Feasibility Margin Network	63
3.7	Assumptions Summary	65
3.8	Conclusions	72
4	Feasible Wrench Based Planning	73
4.1	Introduction	74
4.2	Heuristic Base Motion Planning	74
4.2.1	CoM planning strategy	74
4.2.2	Optimization of trunk orientation to maximize joint range	78
4.3	Heuristic Foothold Planning	79
4.4	Heuristics Assumptions Summary	83
4.5	Gradient-Based Foothold Optimization	86
4.6	Simulation Results	90
4.6.1	Walk in cluttered environment	90
4.6.2	Optimization of the trunk orientation on rough terrain .	93
4.6.3	Feasibility Margin-Based Foothold Adaptation	94
4.7	Experimental Results	97
4.7.1	Walk in cluttered environment	97
4.8	Conclusions	98
5	Conclusion	102
5.1	Summary	102
5.2	Future Directions	103
Bibliography		105

List of Figures

1.1	Legged robot having to interact with the environment to traverse challenging terrain.	15
2.1	Floating base systems can be represented using traditional fixed-base models with an additional 6 DoFs to account for the base's mobility. The floating base is connected to the Inertial frame through a fictitious 6-DoFs joint [53].	20
2.2	Examples of legged robots with different numbers of DoFs and configurations.	21
a	HyQReal quadrupedal robot [58]	21
b	Spot quadrupedal robot with arm [9]	21
c	TALOS humanoid robot [47]	21
d	Atlas humanoid robot [8]	21
2.3	The Contact Wrench Cone (CWC) is a 6D set consisting of a force cone and a moment cone [12].	27
2.4	The Raibert Hopper [52].	29
3.1	Iteration of the Iterative Projection (IP) algorithm: after the LP is solved finding a new extremal \mathbf{c}_{xy}^* point along \mathbf{a}_i , this is added to the <i>inner</i> approximation while an edge with normal \mathbf{a}_i passing through \mathbf{c}_{xy}^* is added to the <i>outer</i> approximation [44].	40
3.2	Friction pyramids (shown in red) orientation with respect to the contact surface. Each pyramid base is perpendicular to the contact surface normal $\hat{\mathbf{n}}_i$. A top view of the pyramid base with respect to the tangent contact axes $\hat{\mathbf{t}}_{x,i}, \hat{\mathbf{t}}_{y,i}$ is shown (bottom right).	42

3.3	Effect of different external wrenches acting on the CoM on the (a) friction region and the (b) feasible region. Changes in size and shifting of the location of the regions can be observed. The components of the external wrench that are mentioned are applied simultaneously and the unmentioned components are set to zero. The stance feet of Hydraulically actuated Quadruped (HyQ) are shown as black points with the front feet facing right. Regions are computed for a trunk height of $c_z = 0.53m$	47
a	Friction Region (only friction considered)	47
b	Feasible Region (both friction and joint-torque limits considered)	47
3.4	The improved feasible region degenerates to a line during a trot when only two feet are simultaneously in contact with the ground. This segment is shifted forward in the same direction as the robot's acceleration. The finite width of the improved feasible region (blue) is due to the infinitesimal contact torques τ_x^{lim} and τ_y^{lim} . The robot is dynamically unstable in this scenario.	51
3.5	Different evaluations of the reachable region at different HyQ Center of Mass (CoM) heights.	58
a	Four stance feet.	58
b	Three stance feet.	58
3.6	Different evaluations of the reachable region at different HyQ roll orientations.	59
3.7	Reachable region for the <i>Aliengo</i> robot at a nominal configuration during experiments.	62
3.8	Reachable region for the <i>Aliengo</i> robot at a height and an orientation that bring the joints to its limits.	63
a	Height	63
b	Orientation	63
3.9	<i>Reachable region</i> (grey) for HyQ at different configurations. The blue arrows show the desired contact forces, while the green arrows show the actual contact forces.	64
a	Reachable pose	64
b	CoM at <i>reachable region</i> boundary	64
c	<i>Reachable region</i> violation	64
3.10	Comparison of the learned feasibility margin with the numerical margin. The green line shows the margin computed numerically, while the blue line shows the margin computed using the network.	65

a	Foot variation	65
b	Force variation	65
3.11	Friction regions using friction pyramids of different numbers of base edges. The black arrows show the contact normal to each foot.	66
a	$\mu = 0.5$	66
b	$\mu = 0.7$	66
3.12	Different orientations of the friction pyramid approximations (with eight base edges) with respect to friction cones seen from the base.	68
a	Regular octagon	68
b	22.5 deg rotated octagon	68
3.13	The effect that different orientations of the friction pyramid approximations with respect to friction cones can have on the friction region.	70
3.14	Comparison between the CoM workspace (ground-truth blue) and <i>reachable region</i> (black dashed) for HyQ at 0.49 m CoM height.	71
4.1	Block diagram of our locomotion framework. The improved feasible region helps the planner devise feasible robot trajectories.	75
4.2	Flowcharts illustrating the trajectory planning algorithms. Top: CoM FP algorithm. Bottom: Orientation FP algorithm.	76
4.3	Forward velocity and wrench tracking results for disturbance rejection task considering the HyQ robot. The dashed lines separate the different stages of the experiment: no compensation, wrench compensation, and wrench and foothold compensation.	82
a	Foward velocity	82
b	Wrench	82
4.4	Forward velocity and wrench tracking results for motion task.	84
a	Forward velocity	84
b	Wrench	84
4.5	Comparison between scaling the extended feasible region based on the centroid and using polygon offsetting.	85
4.6	Block diagram of the Feasibility Margin-Based Foothold Adaptation framework.	88
4.7	HyQreal robot performing a crawl gait using MPC-optimized footholds. Given that no joint constraints are considered in the Model Predictive Control (MPC) , it chooses a narrow stance to prioritize the base velocity task.	89

4.8	Simulation of HyQ descending a challenging 30° ramp with a 50 cm tunnel (template 1). The height of the HyQ robot is decreased from 53 cm to 40 cm in order to fit inside the tunnel. A force-controllable rope (not shown in the figure) is attached to the back of the robot's trunk to compensate for gravity.	91
4.9	Improved feasible regions and CoM planning for two instances while descending the challenging tunnel in simulation (tunnel not shown in this figure). HyQ is heading to the left (downwards) while the external force due to the rope (black arrow) is applied in a direction opposite to the motion. The regions shown above are for the future regions upon lift-off of the swing leg (<i>LF</i> in the upper plot and <i>LH</i> in the lower one): Support regions (dashed), improved feasible regions (grey), and the scaled feasible regions (black). Cubes represent the projection of the CoM target based on the FP strategy (yellow) and the SP strategy (blue) on the projection plane. The red sphere represents the projection of the current CoM . This is out of the region because the robot is still moving toward the target in the move-body phase (4 legs in stance).	92
a	Middle of the slope	92
b	Bottom of the slope	92
4.10	HyQ 's reachable region on a 15 degree ramp with (a) the robot's trunk aligned to the ramp inclination and (b) horizontal trunk. .	95
a	Aligned with ramp inclination	95
b	Horizontal configuration	95
4.11	Simulation of HyQ forced near its kinematic limits while traversing a difficult non-coplanar terrain (Template 2). The robot configurations shown are at the end of a move-body phase. Realizing orientations based on (a) the TP strategy and (b) based on the FP strategy results in different leg configurations (top). The resulting regions shown in the bottom plots are: friction regions (dashed), feasible regions (grey), and the scaled feasible regions (black). Large difference in the resulting feasible regions can be seen, in turn affecting the feasibility of the CoM trajectory (blue cube and red ball represent the projections of the CoM target and the actual CoM , respectively).	96
a	Using Heuristic-Based Planning	96
b	Using Feasible Planning	96

4.12 Comparison of optimized foothold configuration of the Nonlinear Model Predictive Control (NMPC) (<i>left</i>) vs. the Feasibility Margin-Based Foothold Adaptation (FFA) (<i>right</i>) for HyQReal performing a crawling gait with a commanded velocity of 2.5 m/s and an applied constant external force of $F = (250N, 250N, 200N)$. The blue spheres represent the footholds computed by the NMPC, and the green spheres represent the footholds computed by the FFA.	97
4.13 Feasibility Margin Comparison: Margins over the prediction horizon for FFA optimization (<i>blue</i>) versus MPC trajectory (<i>orange</i>) for two instances during the motion. The lowest value of the margins along the horizon are shown using horizontal dashed lines. The grey-shaded area marks the pre-touchdown phase. The FFA shows higher minimum margins as opposed to the MPC's margin nearing zero.	98
a Instance 1 during motion	98
b Instance 2 during motion	98
4.14 HyQReal robot traversing uneven terrain using the FFA. The selection of each foothold is constrained using the patch shown by the red spheres generated by the Vision-Based Foothold Adaptation (VFA) heightmap.	99
4.15 Box plots of feasibility margins at $k = 0$ of the optimization - current margins - taken over the whole motion using (a) the MPC and (b) the FFA footholds.	99
a MPC	99
b FFA	99
4.16 Aliengo descending a 45° ramp with a reduced robot height of 20 cm (similar to the scenario shown in simulation). A rope is attached to the back of the robot's trunk to compensate for gravity through a counterweight.	100
4.17 Experimental results showing the Right-Hind Hip Flexion-Extension (HFE) joint trajectory of Aliengo during the ramp descent. SP strategy (above): The knee starts to hit the virtual kinematic limit (red line) during the <i>move body</i> phases (shaded blue). The violations are in shaded red. FP strategy (bottom): No kinematic limit violations are observed.	100

- 4.18 Experimental results showing the **CoM** position tracking of **HyQ** in the x direction. Deterioration can be seen with the SP strategy (upper plot) due to the joint kinematic limit violations while good tracking is observed when the planning is based on the improved feasible region with the FP strategy (bottom plot). 101

Chapter 1

Introduction

1.1 Motivation

Over the past few decades, legged robots have been primarily limited to simple laboratory settings, diligently mastering the skill of walking and running. However, the main aim of legged robots is to operate in the real world, where they can perform complex tasks such as search and rescue operations, industrial inspections, and infrastructure monitoring. In such unpredictable environments, we can leverage the full capabilities of legged robots and their ability to navigate diverse and challenging terrains. Not only do these applications require mobility across complex surfaces, but they also require the ability to dynamically engage with various external disturbances.

Despite the notable progress in legged robotics, their operational capacity is limited by existing locomotion strategies that often struggle under the dynamic and unpredictable conditions of real-world terrains. For instance, due to the mechanical limitations inherent to the robots, such as torque and kinematic limits, complex terrains or unexpected obstacles along with external forces can be challenging to the robot's locomotion capabilities.

Acknowledging these constraints, a new locomotion strategy focusing on feasible wrench-based control is introduced. By optimizing achievable wrenches within the robot's inherent capabilities and addressing the robot's physical restrictions, this strategy improves the robot's performance in executing complex interactive maneuvers reliably. The goal is to expand the functional range of legged robots,

1.2. Contributions

allowing them to operate more effectively and robustly in various demanding environments and interactive circumstances.

The strategy not only addresses current challenges faced by locomotion techniques but also creates opportunities for future advancements in applications where it is useful for the robots to interact with the environment. The strategy is a further step towards integrating legged robots into real-world applications where their unique capabilities can be fully utilized.

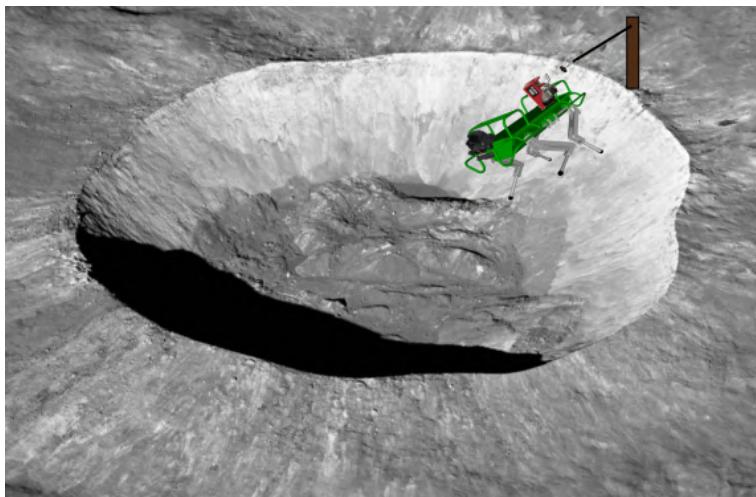


Figure 1.1: Legged robot having to interact with the environment to traverse challenging terrain.

1.2 Contributions

The contributions of this dissertation are focused on two main aspects: feasibility criteria and feasible planning for legged robots. The contributions are summarized as follows:

1.2.1 Feasibility Criteria

We introduce the *improved feasible region*, a 2D Cartesian set where dynamic balance and the satisfaction of joint-torque and joint-kinematic limits are guaranteed, whenever the projection of the CoM lies inside the admissible set. The region and margin improve upon the limitations of previous attempts where no

1.2. Contributions

consideration of dynamic balance, kinematic limit awareness, the effect of external wrenches, and the effect of plane inclination were considered. In particular, the contributions of this work are:

- Generalize the feasible region to account for the effect of external forces and torques acting on arbitrary points of the robot (see Section 3.2.2).
- We utilize the **Single Rigid Body Dynamics (SRBD)** model and relax the quasi-static assumption by considering the dynamic effects of motion, as well as the angular dynamics (see Section 3.2.3). We show that the dynamic effects and external wrenches result in a time-varying shape of the region when the robot is in motion.
- Generalize the feasible region to be defined on arbitrary plane inclinations (see Section 3.2.1).
- Embed the complete joint-kinematic limits in what we define as the *reachable region*. This presents a more accurate representation of the CoM kinematic capability than the conservative approximations used in literature (see Section 3.3). Given that the region utilizes the SRBD model, calculating it for different robots is simple, with no need for data-driven training. The region can be intersected with the joint-torque-aware region (with the aforementioned extensions) and leads to the so-called *improved feasible region* that considers friction, joint-torque, and joint-kinematic limits.
- Define the *feasibility margin* on the *improved feasible region* to guide planning in the direction of higher feasibility. We approximate the margin with a supervised model using a **Multi-Layer Perceptron (MLP)** network to provide a differentiable function that can be used efficiently in optimization (see Section 3.5).

1.2.2 Feasible Planning

We introduce planning strategies that produce **Center of Mass (CoM)**, base orientation, and footstep trajectories under extreme conditions, such as complex terrains that challenge the joint limits and/or the presence of external wrenches. We show ways to achieve such feasible planning through both heuristic and optimization approaches using the *improved feasible region* and the *feasibility margin*. In particular, the contributions of this work are:

- Design a *robust CoM* planning strategy that utilizes non-convex regions and proposes a new optimization for the trunk orientation based solely on the

1.3. Outline

improved feasible region (see Sections 4.2.1 and 4.2.2). The level of robustness can be adjusted by tuning a single parameter according to the desired level of "cautiousness" one wants to achieve in the locomotion.

- Introduce a receding-horizon foothold selection strategy that considers the *feasibility margin* to maximize the minimum margin over the horizon (see Section 4.5). This introduces robustness in the presence of external wrenches and joint limits compared to traditional approaches.
- Show simulations and hardware experiments with robots walking in scenarios that are challenging in terms of actuation and kinematic motions. The experimental results are shown on both the 90 kg HyQ robot (hydraulically actuated) and the 21 kg Aliengo (electrically actuated) robot [1] (see Sections 4.6 and 4.7).

1.3 Outline

The remainder of this dissertation is organized as follows: Chapter 2 delves into the different methods of modelling legged robots and their constraints, as well as the state-of-the-art in locomotion; Chapter 3 introduces criteria, including the improved feasible region and the feasibility margin, that can be used during planning to guarantee the feasibility of motion; Chapter 4 presents planning strategies that utilize the improved feasible region and the feasibility margin, and tests them in various challenging simulation scenarios and hardware experiments; Chapter 5 concludes the dissertation and outlines future work.

Chapter 2

Related Work

Over the years, various attempts have been carried out to model and control legged robots. We explore the spectrum of models and strategies developed in the literature to understand and control legged robots, ranging from accurate, detailed approaches to simplified but efficient ones. We review the dynamical models ranging from the ones that capture the full rigid body dynamics to simplified ones that approximate the robot as a linear inverted pendulum. These models serve as critical tools for balancing computational efficiency with motion control accuracy, which is essential for real-time robotic applications. Furthermore, we delve into different feasibility metrics that can be used to analyze the behavior of the robot and ensure the stability and practicality of the movements on the real robotic platform. The criteria are pivotal for designing robots that can reliably interact with complex environments without failing. Finally, we address the different planning strategies ranging from heuristic methods to sophisticated optimization-based and data-driven approaches that incorporate these dynamics and feasibility criteria into planning algorithms that can move the robot safely and effectively.

2.1 Dynamic Models

2.1.1 Full Rigid Body Dynamics

Legged robots can be effectively modeled as floating-base dynamical systems with multiple kinematic trees. Unlike fixed-base systems, where the base is

2.1. Dynamic Models

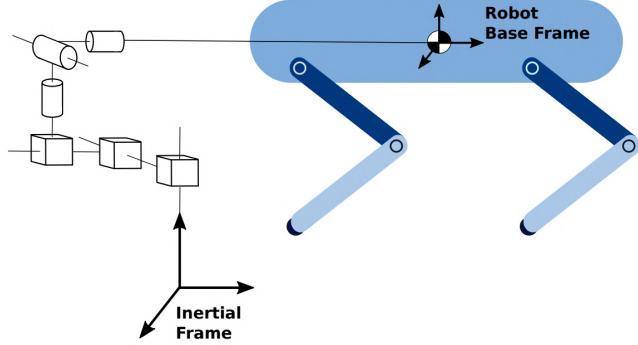


Figure 2.1: Floating base systems can be represented using traditional fixed-base models with an additional 6 DoFs to account for the base's mobility. The floating base is connected to the Inertial frame through a fictitious 6-DoFs joint [53].

stationary, the base is free to move in all directions and connects to a fictitious fixed inertial frame through a 6-Degrees of Freedom (DoF) joint. This setup is illustrated by the robot's base being linked to the inertial frame, as shown in Fig. 2.1.

Such systems are mathematically akin to more traditional fixed-base robots but include an additional 6 DoFs to account for the base's mobility (see Fig. 2.1), leading to a total of $N = n_j + 6$ degrees of freedom, where n_j is the number of actuated joints. The dynamic model of these robots can be derived using the Euler-Lagrange equation, extended to accommodate the floating base's dynamics.

The motion of a legged robot with a floating base and n_j actuated joints (similar to the ones in Fig. 2.2) can be described by the following equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}\boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^T \mathbf{F} \quad (2.1)$$

where $\mathbf{M} \in \mathbb{R}^{(6+n_j) \times (6+n_j)}$ is the joint-space inertia matrix, $\mathbf{q} \in \mathbb{R}^{(6+n_j)}$ is the generalized floating base velocity, and vector $\mathbf{h} \in \mathbb{R}^{(6+n_j)}$ includes Coriolis, centrifugal, and gravitational effects. The selection matrix $\mathbf{S} \in \mathbb{R}^{(6+n_j) \times n_j}$ directs the torque $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ to the actuated joints, and the contact Jacobian \mathbf{J}_c maps the contact forces $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_{n_c}) \in \mathbb{R}^{3n_c}$ at the n_c end-effectors to generalized forces.

The floating-base system comprises an actuated part (i.e., legs) and an unactuated part (i.e., base/torso). Therefore, Eq. (2.1) can be further clarified by

2.1. Dynamic Models



Figure 2.2: Examples of legged robots with different numbers of DoFs and configurations.

decomposing it as:

$$\begin{bmatrix} \mathbf{M}_b(\mathbf{q}) & \mathbf{M}_{bj}(\mathbf{q}) \\ \mathbf{M}_{bj}^T(\mathbf{q}) & \mathbf{M}_j(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_b \\ \ddot{\mathbf{q}}_j \end{bmatrix} + \begin{bmatrix} \mathbf{h}_b(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{h}_j(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{6 \times n_j} \\ \mathbf{I}_{n_j \times n_j} \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{J}_{cb}(\mathbf{q})^T \\ \mathbf{J}_{cj}(\mathbf{q})^T \end{bmatrix} \mathbf{F} \quad (2.2)$$

In this formulation, $\mathbf{M}_b \in \mathbb{R}^{6 \times 6}$ represents the inertia matrix for the unactuated base, and $\mathbf{M}_j \in \mathbb{R}^{n_j \times n_j}$ for the actuated joints. The matrices \mathbf{M}_{bj} and \mathbf{M}_{jb} capture the coupling between actuated and unactuated components, illustrating the complex interdependencies within the robot's dynamics. $\mathbf{q}_j \in \mathbb{R}^{n_j}$ represents the joint configuration and $\mathbf{q}_b \in \mathbb{R}^6$ represents the 6-DoF pose of the floating base. The vectors \mathbf{h}_b and \mathbf{h}_j encompass the Coriolis, centrifugal, and gravita-

2.1. Dynamic Models

tional forces acting on the base and joints, respectively. $\mathbf{F} \in \mathbb{R}^{m \times n_c}$ represents the contact wrenches with the ground, where n_c is the number of contacts with the environment. If the feet can only exert forces as assumed for point feet, then $m = 3$, while typically, for flat feet, both forces and torques are exerted, making $m = 6$. The number of contacts n_c changes continuously as robots step, leading to a system with switching dynamics. The contact wrenches play a crucial role in the dynamics equation, affecting the robot's movement and stability. The contact Jacobians \mathbf{J}_{cb} and \mathbf{J}_{cj} translate forces applied at the contact points into effects on the robot's overall motion. Finally notice that the torques τ are only applied directly to the actuated joint of the robot.

Note that the joint-space inertia matrix \mathbf{M} , the force vector \mathbf{h} , and the contact Jacobian \mathbf{J}_c , are nonlinear in the joint configuration \mathbf{q}_j and the generalized velocity $\dot{\mathbf{q}}$. Furthermore, the coupling between the actuated and un-actuated dynamics in Eq. (2.2) additionally contribute to the nonlinearity. As shown, the dynamics of legged systems are highly dimensional, nonlinear, and non-convex, making the planning and control problem complicated.

2.1.2 Simplified Dynamics

2.1.2.1 Centroidal Dynamics

We can simplify the modelling problem by focusing our analysis on the parts of the dynamics that capture the essential behavior of the system. Simplifying the rigid body dynamics model often involves focusing on the robot's behavior as projected onto its [Center of Mass \(CoM\)](#).

The [CoM](#) of the robot is a crucial point as it defines where the robot's entire mass is concentrated and, therefore, describes the most crucial part of the robot's dynamics. The dynamics of the robot's [CoM](#) are characterized by the unactuated (top) part of Eq. (2.2). If the change of momentum is expressed in a centroidal frame, we achieve the *centroidal dynamics*

$$\begin{aligned} m(\ddot{\mathbf{c}} + \mathbf{g}) &= \sum_{i=1}^n \mathbf{f}_i \\ \mathbf{I}\dot{\omega} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{M}_{bj}\ddot{\mathbf{q}} &= \sum_{i=1}^n \mathbf{p}_{cf_i} \times \mathbf{f}_i \end{aligned} \tag{2.3}$$

where $m \in \mathbb{R}$ is the total mass of the robot, $\mathbf{c} \in \mathbb{R}^3$ is the position of the [CoM](#), $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the base, $\mathbf{g} \in \mathbb{R}^3$ is the gravity vector, $\mathbf{I} \in \mathbb{R}^{3 \times 3}$

2.1. Dynamic Models

represents the robot inertia lumped at the **CoM**, and \mathbf{p}_{cf_i} is the position of the i^{th} foot.

Not focusing on the dynamics of the legs reduces the number of equations from $6+n$ to 6 and decouples the problem of finding centroidal trajectories from the problem of finding appropriate joint torques. However, the model remains highly nonlinear due to the presence of the joint state.

2.1.2.2 Single Rigid Body Dynamics (SRBD)

Many of the bipedal and quadrupedal robots have a mass distribution such that the base contributes to most of the robot's mass; I.e., the leg masses can be considered negligible w.r.t to the base. Therefore, the inertia \mathbf{I} does not change significantly with different leg configurations, and the leg inertia does not contribute significantly to the angular momentum of the robot. Taking such an assumption, we can simplify the dynamics further by assuming the robot to be a single rigid body,

$$\begin{aligned} m(\ddot{\mathbf{c}} + g) &= \sum_{i=1}^n \mathbf{f}_i \\ \mathbf{I}(\Phi)\dot{\omega} + \boldsymbol{\omega} \times \mathbf{I}(\Phi)\boldsymbol{\omega} &= \sum_{i=1}^n \mathbf{p}_{cf_i} \times \mathbf{f}_i \end{aligned} \quad (2.4)$$

The inertia tensor \mathbf{I} is assumed to be of the robot given some nominal joint positions and is now a function of only the base orientation Φ . Therefore, the nonlinear dependence on the joint angles is completely avoided. However, the model is still nonlinear and non-convex, due to the presence of the Coriolis term $(\boldsymbol{\omega} \times \mathbf{I}(\Phi)\boldsymbol{\omega})$ in the rotational dynamics.

In many cases, additional assumptions can be taken to further simplify the dynamics. Namely, if the robot will not be rotating at fast angular speeds, we can assume that the angular momentum of the robot will be zero. In fact, during human walking, the angular momentum has been empirically shown to be close to zero [49]. In such case, Eq. (2.4) simplifies to a *point mass* dynamics

$$\begin{aligned} m(\ddot{\mathbf{c}} + g) &= \sum_{i=1}^n \mathbf{f}_i \\ 0 &= \sum_{i=1}^n \mathbf{p}_{cf_i} \times \mathbf{f}_i \end{aligned} \quad (2.5)$$

2.2. Feasibility Criteria

2.1.2.3 Linear Inverted Pendulum (LIP) Model

The LIP model introduces further simplification over the SRBD model and provide the advantages of having a linear model. From the SRBD dynamics, by expressing the equations in an inertial frame and computing the Euler angular moment w.r.t. to a generic point \mathbf{o} , we can rewrite Eq. (2.4) as if we take the first half of the equations of motion, and we can express the dynamics of the base throw the *centroidal dynamics*, i.e.:

$$\mathbf{I}(\mathbf{c} - \mathbf{o}) \times (\ddot{\mathbf{c}} + \mathbf{g}) + \dot{\mathbf{L}} = \sum_{i=1}^n (\mathbf{p}_i - \mathbf{o}) \times \mathbf{f}_i$$

where \mathbf{L} is the angular momentum of the robot and \mathbf{p}_i is the position of the i^{th} foot. We can simplify the above equation by choosing \mathbf{o} such that

$$\sum_{i=1}^n (\mathbf{p}_i - \mathbf{o}) \times \mathbf{f}_i = \mathbf{0} \quad (2.6)$$

This coincides with commonly denoted Zero Moment Point (ZMP) [63]; the point with respect to which the moment of the contact forces is zero. Assuming a flat horizontal ground, we can project the above equation on the x and y axes as,

$$\frac{c^z}{\ddot{c}^z + g^z} \ddot{\mathbf{c}}^{x,y} = \mathbf{c}^{x,y} - \mathbf{z}^{x,y} + \frac{\mathbf{S} \dot{\mathbf{L}}^{x,y}}{M(\ddot{c}^z + g^z)}$$

where \mathbf{z} is the ZMP and \mathbf{S} is a selection matrix. Furthermore, if we consider the case where the CoM height is constant, and the derivative of the angular momentum $\dot{\mathbf{L}}^{x,y}$ is negligible, we can obtain the LIP model

$$\ddot{\mathbf{c}}^{x,y} = \frac{g^z}{c^z} (\mathbf{c}^{x,y} - \mathbf{z}^{x,y}) \quad (2.7)$$

2.1.3 Constraints

2.2 Feasibility Criteria

Given a model chosen from Section 2.1 and the existence of constraints on the robot (Section 2.1.3), we need to define corresponding criteria that the robot's motion must satisfy in order for it to be feasible. In this section, we introduce various criteria that can be used to guarantee the stability and physical feasibility of the robot's motion. The most important and fundamental question to

2.2. Feasibility Criteria

ask when planning a motion for a legged robot is "Will the robot remain stable during this motion?".

ZMP

The most common way to answer this question is to ensure that for a given fixed point on the robot, there exist ground reaction forces that satisfy the robot's dynamics without slipping [66]. The simplest approach to do this is to define a reference point along with a 2D region in which the projection of the reference point must lie inside in order to guarantee the existence of the feasible ground reaction forces. A reference point could be any generic point that connects with the motion of the robot. One of the most traditionally used ground reference points is the **ZMP** due to its natural emergence in the **LIP** model as discussed in section 2.1.2.3. As mentioned, the **ZMP** is defined as the point on the ground where the sum of the moment of the contact forces is zero. On flat ground, this coincides with the **Center of Pressure (CoP)** and from Eq. (2.6), we can define the **ZMP** as

$$\mathbf{z} = \frac{\sum_{i=1}^n \mathbf{p}_i \mathbf{f}_i}{\sum_{i=1}^n \mathbf{f}_i} \quad (2.8)$$

One could easily observe that if all f_i^z are positive (*unilaterality of the contact forces* condition), the **ZMP** must lies inside the convex combination of the contact points. The region where the **ZMP** lies is referred to as the *support polygon* and is the convex hull of the contact surfaces between the robot and the ground. Such condition is known as the *non-tilting condition*, i.e., as long as the **ZMP** lies inside the support polygon, the robot feet are not tilting.

$$\mathbf{z} \in \text{ConvHull}(\mathbf{p}_1, \dots, \mathbf{p}_n) \quad (2.9)$$

Note that the usage of the non-tilting condition places no constraint on the **CoM**. In other words, at any given configuration, the robot can be stable as long as the **ZMP** lies inside the support polygon, regardless of the **CoM** position. In fact, during dynamic walking, the **CoM** can be outside the support polygon, and the robot can still be *dynamically stable*.

It is also important to note that the definition of the **ZMP** and the stability condition of 2.9 only holds under the assumption that the contacts are on flat ground. In the case of non-flat terrains, the concept of the **CoP** is not defined anymore. In such case, separate **CoPs** can be computed for each contact foot (local **ZMPs**), providing a stability condition for each contact, but cannot provide a global stability guarantee for the whole robot [42, 56]. Extensions of the

2.2. Feasibility Criteria

ZMP to non-flat terrains (non-coplanar contacts) have been discussed in [14, 56].

Support Region

The simplest example of this is a robot standing on a slope; the combination of the robot’s posture, inclination of the slope, and the friction between the robot’s feet and the ground must be such that the robot does not slip. In the simplest case when we assume a *point mass* model of the robot and the robot is stationary on a flat horizontal surface, we can determine stability by ensuring that the projection of the robot’s CoM lies within its base of support. In such simple cases, the base of support is the convex hull of the points of contact with the ground, and the term *support polygon* is often used to describe this region. The support polygon is a 2D region in the plane that the robot’s CoM projection must lie within in order to ensure balance and static stability.

A more general concept of the support region was introduced by Bretl *et al.* to account for non-coplanar contacts and non-vertical orientation of the surface normals, and was coined the *support region* [11]. The resulting region is not necessarily a polygon and can lie outside of the convex hull of the feet. Borrowing ideas from computational geometry, an iterative cutting-edge algorithm [33] was introduced to accurately compute the projection of a nonlinear convex set, defined by the reduced model (balance equations) and the contact constraints (i.e., contact friction), onto the horizontal plane. Given the robot’s contact configuration, contact normals orientations, and friction coefficients, the resulting region defines the set of CoM projections that have corresponding feasible ground reaction forces.

Feasible Region

While the models introduced in section 2.1 reduced the complexity of the analysis and planning problem, it is clear they lack descriptiveness. In particular, the independence of the CoM dynamics from the joint information, prevents the addition of explicit constraints on the joint. The iterative algorithm introduced in [11] introduced a method of adding such descriptiveness to the reduced models by projecting the defined higher-order constraints to the centroidal space. This can be achieved by mapping friction limits (defined at the contact level) and joint torque and kinematic limits (defined at the joint level) to the 2D space where the CoM exists.

2.2. Feasibility Criteria

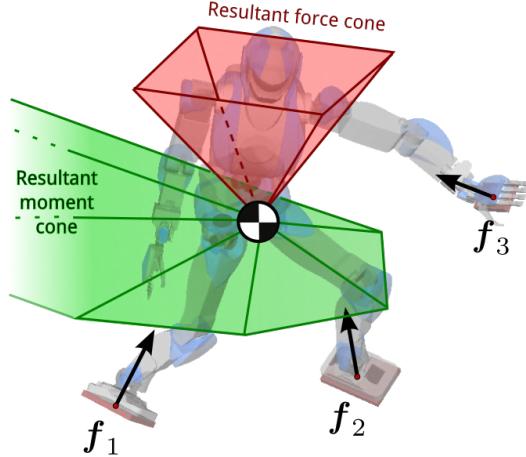


Figure 2.3: The [CWC](#) is a 6D set consisting of a force cone and a moment cone [12].

Orsolino *et. al.*, presented a modified version of the [Iterative Projection \(IP\)](#) algorithm to compute the *feasible region*, a convex region where both friction and joint-torque limits were considered [44]. The approach utilizes the leg Jacobians to transform the torque constraints to ground reaction forces, achieving consistency with the support region projection algorithm. Because of the utilization of the leg Jacobians, the resulting feasible region is configuration-dependent. The ability to formulate higher-order constraints in simplified models using the efficient [IP](#) algorithm makes the *support region* and the *feasible region* intuitive yet powerful methods to plan feasible trajectories on arbitrary terrains *online*. In fact, the feasible region could be enforced as constraints on the [CoM](#) trajectory in a [Trajectory Optimization \(TO\)](#) problem.

Contact Wrench Cone (CWC) and Feasible Wrench Polytope (FWP)

The simplified criteria introduced previously provide information on the robot’s stability and feasibility by constraining a reference point to a plane. The effect of the robot’s orientation on stability/feasibility is not considered. A more general approach to stability and feasibility is to consider the set of wrenches the robot can apply that satisfy the robot’s dynamics and constraints. The [CWC](#) has been introduced to define the set of total admissible contact wrenches that satisfy the dynamic constraints and contact friction constraints [13, 31].

2.3. Planning

To compute the **CWC**, the contact friction cones of all the contact points are summed and mapped to the centroidal wrench space. This mapping results in a 6D polytope (as illustrated in Fig. 2.3) that represents the set of admissible wrenches for which the above-mentioned constraints are satisfied. For the robot to remain stable, the **Gravito-Inertial Wrench (GIW)** [13] must lie inside the **CWC**. Similarly, the **FWP** was defined by Orsolino et al., when both the friction and joint-torque limits are taken into account [45]. Enforcing the polytopes as constraints on the centroidal wrench (or accelerations) in a **TO** problem results in feasible trajectories for the **CoM**. Unfortunately, despite the promising results, the introduction of the joint-torque limits made the computation prohibitively expensive. In fact, increasing the number of contacts dramatically increases the computation time. This makes these polytopes hard to compute *online* without accepting strong approximations on kinematics [45].

2.3 Planning

Various approaches have been studied to plan trajectories for the motion of the robot. The approaches differ in whether they are heuristic or optimization-based, model-based or model-free, hierarchical or holistic, and differ in their objectives.

2.3.1 Heuristic vs. Optimization

Heuristics

Heuristic approaches are often used to generate trajectories for the robot to move. These approaches employ simplified models and stability criteria and use a hierarchical approach to planning the motion. This hierarchical nature comes from the decoupling of the foothold planning and body motion generation. Typically, footholds are first selected using a high-level heuristic criterion, and then body trajectories are computed. Heuristic approaches reduce the complexity of planning and the computational complexity considerably. Often, practical experience is used to design the heuristic criteria.

One of the earliest uses of heuristics in legged locomotion was the *Raibert Hopper* [52]. The robot (shown in Fig. 2.4) was designed to hop in place and used a simple heuristic to plan the motion. The heuristic defines a *neutral point*, i.e., the point where when the robot places its foot on, the robot will move on a symmetric trajectory and keep its forward velocity unchanged. The neutral

2.3. Planning

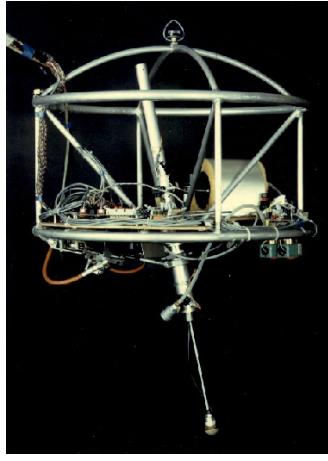


Figure 2.4: The Raibert Hopper [52].

point is defined as

$$\mathbf{p}_n = \mathbf{h} + \dot{\mathbf{c}} \frac{T_{st}}{2} \quad (2.10)$$

where \mathbf{h} is the known position of the hip on the ground, T_{st} is the stance time. An additional term can be further added to control the speed of the robot:

$$\mathbf{p}_n = \mathbf{h} + \dot{\mathbf{c}} \frac{T_{st}}{2} + k_r (\dot{\mathbf{c}} - \dot{\mathbf{c}}_d) \quad (2.11)$$

where k_r is a gain, and $\dot{\mathbf{c}}_d$ is the desired horizontal velocity of the robot.

The heuristic has been extended to other gaits where multiple feet move simultaneously in a periodic manner [51], where those legs are considered to be a single *virtual leg*.

Similarly, heuristic approaches have been developed for a crawling gait on quadrupedal robot to traverse rough terrains in [26]. The selected foothold and the step frequency were computed as a function of the desired velocity of the robot and a maximum step length (to avoid kinematic limits). After the foothold is selected and executed, a body trajectory is computed to place the robot in a favorable configuration and maintain stability with respect to the *support polygon*. Additional heuristics were used to compute the desired body orientation and the leg swing trajectory. For instance, the body orientation was adapted to be parallel to an estimated *terrain plane* inclination. For terrains where uneven contact points are present, the terrain inclination was estimated by fitting an averaging plane through the stance feet.

2.3. Planning

Another heuristic used to generate trotting gaits is inspired by animals using the concept of Central Pattern Generators (CPGs) [36]. CPGs are neuronal circuits that, when activated, can produce rhythmic motor patterns. A CPG-inspired Cartesian trajectory generator was designed in [5] to provide trajectories for the feet. Coupled with a reactive controller to compute the body motion, a fast-flying trot was achieved.

The power of heuristic tools lies in their simplicity and their computational efficiency. Decoupling the locomotion problem, using simplified models, and steering away from complex optimization techniques proved to be effective for online planning. However, due to their dependence on human experience and tuning, they quickly fail in conditions that were not taken into account during the design process. For a start, each gait often needs different heuristic approaches. Furthermore, they lack robustness when the terrain conditions change or when the robot model changes. Finally, they often lack the guarantees that the generated motion is feasible and stable.

Optimization

Optimization-based techniques aim to solve the locomotion planning problem through formulating it as an **Optimal Control Problem (OCP)** and using numerical techniques to find the optimal optimization variables. The **OCP** defines the high-level locomotion task through using cost functions taking into account the robot's dynamics as constraints

$$\min_{\mathbf{x}^*(t), \mathbf{u}^*(t)} \int_0^T \ell_c(\mathbf{x}(t), \mathbf{u}(t)) + \ell_{c_T}(\mathbf{x}(T)) \quad (2.12a)$$

$$\text{s.t. } \mathbf{x}(0) = \mathbf{x}_0, \quad (2.12b)$$

$$\mathbf{x}(t) = f_c(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [0, T], \quad (2.12c)$$

$$h_c(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \quad t \in [0, T], \quad (2.12d)$$

A user-defined cost functional (2.12a) represents a metric to be minimized, such as a deviation from a desired robot velocity or energy consumption. Particularly, ℓ_c and ℓ_{c_T} represent a running cost and a terminal cost, respectively. Equations (2.12b) defines the initial condition of the state trajectory $\mathbf{x}(t)$, (3.22) introduces the dynamics of the robot that the solution needs to respect — where $\mathbf{u}(t)$ is the control input — and Eq. (2.12d) represents path constraints on the state and control input.

To achieve online replanning, the initial condition (2.12b) can be updated at each

2.3. Planning

time step with the current state of the robot. Furthermore, the choice of the model of the dynamics (as introduced in Section 2.1) and the path constraints types play a big role in the accuracy and the computational complexity of the optimization problem and the solution obtained.

The [OCP](#) (2.12) is a continuous time problem, i.e., it has an infinitely-dimensional variables and constraints. Therefore, the problem is discretized in order to be solved numerically. The conversion of the continuous [OCP](#) to a discrete-time problem is known as *transcription problem*, leading to a finite-dimensional optimization problem, commonly known as the [Nonlinear Programming \(NLP\)](#) problem. The choice of the transcription method affects both accuracy and numerical stability of the result. The main methods used for transcription in robotic problems are the *direct collocation* and the *shooting* methods (see [50] for more details). To define the [NLP](#), we discretize the continuous time t into N time steps of equal lengths. Defining \mathbf{x}_k and \mathbf{u}_k as the state and control input at time step k , respectively, the [NLP](#) problem can be expressed as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} & \sum_{k=0}^{N-1} \ell_c(\mathbf{x}_k, \mathbf{u}_k) + \ell_{c_T}(\mathbf{x}_N) \\ \text{s.t. } & \mathbf{x}_0 = \mathbf{x}_0, \\ & \mathbf{x}_{k+1} = f_c(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \dots, N-1, \\ & h_c(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad k = 0, \dots, N-1. \end{aligned} \tag{2.13}$$

where we define $\mathbf{x} := \{\mathbf{x}_0, \dots, \mathbf{x}_N\}$, $\mathbf{u} := \{\mathbf{u}_0, \dots, \mathbf{u}_N\}$, and the continuous cost (2.12a) is approximated by the finite sum of the cost evaluated at each sampling time k . The dynamics (3.22) are approximated through an integration scheme, such as the explicit Euler scheme, and the path constraints (2.12d) are enforced instead at each time step k .

To solve the [NLP](#) problem, we can use a variety of numerical optimization techniques. In the case that the system dynamics are linear and the cost function is quadratic, the problem is known as a [Quadratic Program \(QP\)](#) problem and can be solved efficiently using specialized solvers. For nonlinear problems, the choice of the optimization technique can greatly affect the stability and the computational complexity of the solution.

Three of the most common techniques used are [Sequential Quadratic Programming \(SQP\)](#), [Interior-Point \(IP\)](#), and [Differential Dynamic Programming \(DDP\)](#). The [SQP](#) method iteratively solves local quadratic approximation of the problem and updates the solution until convergence is reached. At each

2.3. Planning

iteration, the problem is approximated with a local model that is represented by a simpler [QP](#). [SQP](#) is known to be computationally efficient and provides stable solutions.

The [IP](#) method works by navigating from within the feasible region — the set of all points that satisfy the problem’s constraints (not to be confused with the Cartesian *feasible region* of section 2.2) — towards the optimum. The [IP](#) method modifies the objective function by adding a barrier term that prevents the solution from prematurely reaching the constraint boundaries, ensuring smoother convergence. This approach involves solving large, sparse linear systems, making it computationally intensive but provide stable and robust solutions.

[DDP](#) is a direct optimization method that uses a second-order Taylor approximation of the cost function and the system dynamics, and iteratively improve a guess. Given an initial candidate trajectory, the method linearizes the problem around the trajectory and then computes an optimal state-feedback controller that minimizes the total cost. [DDP](#) handles complex dynamics by breaking down the problem into manageable subproblems, solving them iteratively, and then combining the solutions to find the optimal trajectory. However, [DDP](#) relies on a good initial guess and an accurate system model, as a poor starting point or model errors can lead to suboptimal solutions.

As opposed to heuristics, which are mainly reactive, optimization-based techniques posses a predictive nature, allowing the robot to plan ahead and achieve more complex tasks. Furthermore, while heuristics used a hierarchical approach, the use of gradient-based optimization allows for simultaneous optimization of the footholds and the body motion.

2.3.2 Base and Foothold Optimization

The use of optimization-based techniques to plan trajectories for the robot base and footholds has been widely studied in the literature. Different approaches have mainly differed in the complexity of the model used and the constraints considered.

The use of simplified models has gained a lot of prevalence in the literature due to the difficulty of designing planners for the full dynamics of the robot in real-time, at the price of a suboptimal solution and lower accuracy. This is because reduced models are often written in a reduced set of state variables and capture the main dynamics of the robot during locomotion but typically neglect the joint dynamics. As a further result, constraints at the joint variables (e.g., torque or

2.3. Planning

kinematic limits) cannot be explicitly formulated in the planning problem (i.e., they lack descriptiveness). However, the use of reduced models results in smaller optimization problems and shorter computation times.

The fast computational advantage of reduced models allows for online replanning in an [Model Predictive Control \(MPC\)](#) fashion. For instance, Oriolo et al. developed a linear [MPC](#) based on the [LIP](#) model in [57] to generate stable [CoM](#) trajectories with [ZMP](#) utilized as a control variable. Other approaches utilized the [SRBD](#) model to compute dynamic centroidal trajectories through an [MPC](#) [7, 21, 62]. To keep the linearity of the problem, the effect of the precession and nutation term is neglected. In such cases, the [MPC](#) computes optimal values of the [CoM](#), base orientation, and [Ground Reaction Forces \(GRFs\)](#). The seminal work by Winkler et al. in [67] introduced a nonlinear trajectory optimizer that additionally computes optimal footholds at the cost of a longer computation time.

To incorporate joint kinematic constraints in the planning problem, later approaches utilized the so-called *kinodynamic* models. In such cases, a simplified model such as the [SRBD](#) model is used alongside the forward kinematics to incorporate the joint positions [18, 29]. Recently, C. Mastalli et al. utilized [DDP](#) and efficient solver optimization to generate full model trajectories using the full dynamic model of the robot [38]. This allows further consideration of the joint torque limits.

2.3.3 Feasible Planning

Despite their remarkable achievements, the proposed approaches either avoid considering joint-torque and kinematic limits or perform conservative approximations. For instance, in [67], to avoid kinematic limits violation, box constraints were placed on the foot position as an approximate of the leg workspace. Similarly, Fankhauser et al. [23] optimized the orientation of the robot to ensure static stability and kinematic limits, by solving a non-linear optimization problem. However, the kinematic limits were roughly approximated by setting bounds on the leg length. The usage of higher-order models such as the kinodynamic model in [29] and the full dynamics in [38] allows for the explicit consideration of the joint limits. However, for the problem to be computationally feasible in real-time, the limits are not enforced as hard constraints; instead, relaxed barrier functions are used.

Some attempts have been made to enforce kinematic constraints in an efficient

2.3. Planning

manner. Carpentier et al. [16] focused on incorporating the kinematic constraints via learning proxy constraints. [24, 60] constrain the position of the **CoM** with respect to the contact points. However, these kinematic constraints are only approximated to maintain the convexity of the problem. Nozawa et al. iteratively solved a **QP** using the inverse kinematics and joint-kinematic limits to find a kinematically valid **CoM** target close to the original target chosen solely on the stability region in [43]. In the context of manipulators that move assembly objects, other approaches [39, 41] present a way to find all the orientations that satisfy static stability. Yet, the objects were fixed.

As mentioned in section 2.2, different attempts of simplifying the representation of joint-torque limits were made through the concept of feasibility criteria. The **FWP** was introduced in [45] to represent the set of admissible wrenches that satisfy the friction and joint-torque limits. The **FWP** was enforced as constraints on the **CoM** trajectory in a **TO** problem. However, the computation of the **FWP** is slow and prohibitive for online planning. Boussema et al. [10] introduced a leg impulse capability depending on the contact friction and joint torque limits. Such criterion was used to choose appropriate footholds and provide emergent gait transitions dependent on the robot’s speed. To achieve faster computation times than in [45], Orsolino et. al., instead utilized the lower dimensional feasible region in [44] to represent the set of admissible **CoM** projections that satisfy the friction and joint-torque limits. Feasible **CoM** trajectories were generated by selecting **CoM** targets that lie within the *feasible region*. A trajectory optimizer was developed in [46] using a stability criterion with the help of the **Instantaneous Capture Point (ICP)** to generate dynamically feasible trajectories that respect the *feasible region* in an efficient manner. No joint-kinematic limits were considered in the above approaches.

2.3.4 Data-driven approaches

In the domain of legged locomotion, a growing segment of the research community is increasingly focused on leveraging data-driven methods like Machine Learning (ML) and Reinforcement Learning (RL). These techniques are employed to develop algorithms that teach robots to walk, jump, and run, showing promising results in simulations, as highlighted in recent studies by DeepMind [30] and Abbeel et al. [48]. These approaches may be entirely unsupervised, where the algorithm independently explores the policy space through trial and error, or partially supervised, utilizing initial successful examples to guide training.

2.3. Planning

However, these learning algorithms sometimes generate fragile policies that perform poorly in scenarios that are not represented in the training data, a significant concern for deployment on real-world robotic platforms. In such environments, unmodeled dynamics and unforeseen events can adversely affect task execution.

To address these challenges, researchers have developed a number of hybrid strategies that enhance locomotion behaviors using machine learning, building on the more reliable foundations provided by model-driven trajectory optimization. This hybrid approach tends to be more directly applicable to actual hardware than attempts to learn the entire locomotion tasks from scratch. For instance, Villarreal et al. [35] describes a method where for a quadruped robot with predefined base trajectories and foot sequences, a self-supervised Convolutional Neural Network (CNN) is trained using heuristics to determine optimal foot placement on varying terrains based on morphology, slip conditions, and the potential for shin collisions.

Moreover, several strategies that integrate model-based optimization with machine learning involve optimization problems where constraints are derived from offline, data-driven studies. Such constraints often represent simplified, low-dimensional versions of more complex feasibility constraints, such as joint kinematic limits, avoidance of self-collisions, and interactions with complex environmental geometries [16, 17, 34].

Chapter 3

Feasibility Guarantees for Legged Locomotion

Incorporates material from the following publication:

Abdelrahman Abdalla, Michele Focchi, Claudio Semini. “An Efficient Paradigm for Feasibility Guarantees in Legged Locomotion”. IEEE Transactions on Robotics (T-RO) 2023.

Paper: <https://ieeexplore.ieee.org/document/10149812>

Video: <https://www.youtube.com/watch?v=vJpNWnMh0wE&t>

Incorporates material from the following publication:

Abdelrahman Abdalla, Victor Barasuol, Claudio Semini. “Feasible Wrench Based Locomotion Planning for Legged Robots”. (under preparation).

We present a feasibility criterion that allows to design feasible Center of Mass (CoM) and body trajectories in an efficient manner. In previous work [44], the notion of the 2D *feasible region* was introduced, where static balance and the satisfaction of joint torque limits were guaranteed, whenever the projection of the CoM lied inside the proposed admissible region. In this chapter we propose a general formulation of the *feasible region* that guarantees *dynamic* balance alongside the satisfaction of both joint-torque and kinematic limits in an efficient manner. To incorporate the feasibility of the kinematic limits, we introduce an algorithm that computes the *reachable region* of the CoM. We introduce a new region, the *improved feasible region*, that incorporates all the above.

To simplify the analysis, a few assumptions were adopted in [44] during the computation of the feasible region:

1. the only external wrench acting on the robot is gravity;
2. inertial accelerations (quasi-static assumption) and angular dynamics are neglected; this means that the model used to build the region is a point mass model with contact forces;
3. kinematic limits are not considered;
4. the region is always constructed on a plane perpendicular to gravity, making it not general enough to plan trajectories in planes with different inclinations (e.g., when climbing ramps).

Because of assumption (1), the feasible region presented in [44] is incapable of capturing the effects of the application of an external wrench to the robot; external wrenches usually cause a shift in the region as well as a change in its shape and size (as will be shown in Section 3.2.2). Therefore, any planning strategy based on this region would be inaccurate and can lead to unfeasible plans when external disturbances are applied. Such a feature is also needed when an external wrench is *intentionally* applied to the robot. This is the case when a load is pulled or when a rope is used for locomotion. In fact, having a feasibility metric that takes into account the effect of external wrenches would open many research opportunities in rope-aided locomotion and load-pulling applications. External disturbances are incorporated in a Model Predictive Control (MPC) in [59] and [37] to plan stable Zero Moment Point (ZMP) trajectories. The method, however, utilizes the more simplified Linear Inverted Pendulum (LIP) model and is not suitable for non coplanar contacts. Furthermore, the restricting effect of the joint-torque limits on the CoM planning, in the presence of counteracting

3.1. Classical Feasible Region

disturbances, was not considered.

Assumption (2) limits the applicability of the feasible region to quasi-static gaits. If applied to more dynamic gaits, having a trajectory computed under a statically stable assumption may induce falling due to the changes in the velocity of the robot. Recently, Audren et al. [3] incorporated the dynamics, proposing a robust static stability region that accounts for possible **CoM** accelerations. No other feasibility measures such as joint torque and kinematic limits were considered. In contrast, Nozawa et al. [43] compute a dynamic stability region for the **CoM** based on specified linear and angular accelerations. In both approaches, however, only friction guarantees were considered in the regions.

In addition, not accounting for kinematic limits in assumption (3) can be problematic when the robot climbs up and down high obstacles or is forced to walk in confined environments. In such situations, the mandatory adjustments in *height* and *orientation* may push the robot to violate its kinematic limits.

3.1 Classical Feasible Region

For a better understanding of the proposed *improved feasible region*, let us first briefly recap the *feasible region* presented in [44].

3.1.1 Model

We first consider a legged robot walking on uneven terrain with n_c contact points. Assuming quasi-static conditions, we can use the *point mass* model when expressed in an inertial frame to describe static balance if the **CoM** satisfies

$$\begin{aligned} m\mathbf{g} &= \sum_{i=1}^{n_c} \mathbf{f}_i \\ \mathbf{c} \times m\mathbf{g} &= \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i \end{aligned} \tag{3.1}$$

Equation (3.1) can also be represented in terms of wrenches as

$$\mathbf{w}_g = \mathbf{w}_c \tag{3.2}$$

where \mathbf{w}_g is the *gravitational wrench* and defined as

$$\mathbf{w}_g = \begin{bmatrix} m\mathbf{g} \\ \mathbf{c} \times m\mathbf{g} \end{bmatrix} \tag{3.3}$$

3.1. Classical Feasible Region

and \mathbf{w}_c is the *contact wrench* and defined as

$$\mathbf{w}_c = \begin{bmatrix} \sum_{i=1}^{n_c} \mathbf{f}_i \\ \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i \end{bmatrix} \quad (3.4)$$

For each contact to be stable, each force \mathbf{f}_i must satisfy the friction constraints of the contact. Assuming Coulomb friction, the contact forces thus must lie inside the friction cones defined by

$$\left\| \left(I - \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i^T \right) \mathbf{f}_i \right\| \leq \mu_i \hat{\mathbf{n}}_i^T \mathbf{f}_i \quad (3.5)$$

where $\hat{\mathbf{n}}_i$ and μ_i are the contact normal and the friction coefficient of the i^{th} foot, respectively.

Furthermore, the contact forces \mathbf{f} are related to the joint torques $\boldsymbol{\tau}$. The relation for each leg i is described by the actuated components of the full dynamics equation (2.2) as

$$\mathbf{M}_{bi}^T(\mathbf{q}) \ddot{\mathbf{q}}_b + \mathbf{M}_i(\mathbf{q}) \ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}_i) = \mathbf{J}_i^T(\mathbf{q}) \mathbf{f}_i + \boldsymbol{\tau}_i \quad (3.6)$$

where \mathbf{J}_i is the Jacobian of the i^{th} leg, and $\mathbf{c}_i(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}(\mathbf{q}_i)$ are the Coriolis and gravitational terms of the i^{th} leg, respectively.

The joint torques $\boldsymbol{\tau}_i$ are bounded by the joint torque limits, i.e., bounded by a hypercube described as

$$\mathcal{Z}_{\boldsymbol{\tau}} = \{ \boldsymbol{\tau} \in \mathbb{R}^n \mid \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}} \} \quad (3.7)$$

where $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$ are the lower and upper bounds of the joint torques, respectively, and n is the number of actuated joints of the robot. Therefore, from Eq. (3.6) and Eq. (3.7), the contact force for each leg i must lie within the force polytope defined by

$$\begin{aligned} \mathcal{F}_i = \{ \mathbf{f}_i \in \mathbb{R}^{mn_c} \mid \exists \boldsymbol{\tau}_i \in \mathbb{R}^{n_l} \text{ s.t. } & \mathbf{M}_{bi}^T \ddot{\mathbf{q}} + \mathbf{M}_i \ddot{\mathbf{q}}_i + \mathbf{c}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \\ & \mathbf{g}(\mathbf{q}_i) = \boldsymbol{\tau}_i + \mathbf{J}(\mathbf{q}_i)^T \mathbf{f}_i, \quad \underline{\boldsymbol{\tau}}_i \leq \boldsymbol{\tau}_i \leq \bar{\boldsymbol{\tau}}_i \} \end{aligned} \quad (3.8)$$

where n_l is the number of joints of the i^{th} leg.

3.1.2 Iterative Projection Algorithm

The *feasible region* was generated using an [Iterative Projection \(IP\)](#) algorithm described in Algorithm 3.1 (in black). The extensions of the region to generate

3.1. Classical Feasible Region

the improved feasible region are marked in blue and described in detail in Section 3.2.

The algorithm considers the convex constraints imposed on a legged robot (described in the previous section) and projects them onto a 2D linear subspace. This is done by building an inner and outer approximation of the projected region, via iteratively solving a sequence of [Linear Program \(LP\)](#) programs while satisfying the convex constraints (shown in step (III) of Algorithm 3.1). Namely, we considered the static stability constraints (III.a), frictional constraints on the contact feet (III.b), and the joint-torque constraints (III.c).

The solution of each LP problem, \mathbf{c}_{xy}^* , is an extremal CoM position along a certain direction (represented by the unit vector \mathbf{a}_i), that still satisfies the constraints, i.e., a vertex on the boundary of the feasible region. This optimization is performed iteratively along various directions \mathbf{a}_i that span along a circle, building the inner approximation of the region as the convex hull of all the solutions \mathbf{c}_{xy}^* (see Fig. 3.1).

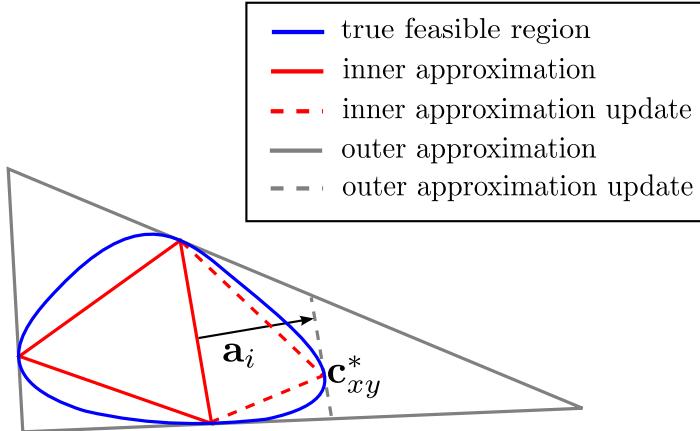


Figure 3.1: Iteration of the LP algorithm: after the LP is solved finding a new extremal \mathbf{c}_{xy}^* point along \mathbf{a}_i , this is added to the *inner* approximation while an edge with normal \mathbf{a}_i passing through \mathbf{c}_{xy}^* is added to the *outer* approximation [44].

Constraint (III.a) ensures the static balance of the robot (force and moment balance). $\mathbf{A}_1 \in \mathbb{R}^{6 \times mn_c}$ is the grasp matrix of the n_c contact points $\mathbf{p}_i \in \mathbb{R}^3$ and m depends on the nature of the contact (i.e., $m = 3$ for point contact, $m = 6$ for full contact). \mathbf{A}_1 is summing up the contact wrenches (pure forces in case of point feet) $\mathbf{f} \in \mathbb{R}^{mn_c}$ and is expressing them at the origin of the world frame.

3.1. Classical Feasible Region

Algorithm 3.1 Feasible Region IP algorithm (with external wrenches).

Input: \mathbf{c}_{xy} , \mathbf{c}_z , ${}^W R_{\mathcal{B}}$, $\mathbf{p}_1, \dots, \mathbf{p}_{n_c}, \mathbf{n}_1, \dots, \mathbf{n}_{n_c}, \mu_1, \dots, \mu_{n_c}$,
 $\underline{\tau}_1, \dots, \underline{\tau}_{n_c}, \bar{\tau}_1, \dots, \bar{\tau}_{n_c}, \mathbf{w}_{ext}$

Result: local feasible region \mathcal{Y}_{fa}

Initialization: \mathcal{Y}_{outer} and \mathcal{Y}_{inner}

while $area(\mathcal{Y}_{outer}) - area(\mathcal{Y}_{inner}) > \epsilon$ **do**

I) compute the edges of \mathcal{Y}_{inner}

II) pick \mathbf{a}_i based on the edge cutting off the largest fraction of \mathcal{Y}_{outer}

III) solve the LP:

$$\mathbf{c}_{xy}^* = \arg \max_{\mathbf{c}_{xy}, \mathbf{f}} \quad \mathbf{a}_i^T \mathbf{c}_{xy}$$

such that :

$$(III.a) \quad \mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{u}$$

$$(III.b) \quad \mathbf{B} \mathbf{f} \leq \mathbf{0}$$

$$(III.c) \quad \mathbf{G} \mathbf{f} \leq \mathbf{d}$$

IV) update the outer approximation \mathcal{Y}_{outer}

V) update the inner approximation \mathcal{Y}_{inner}

end while

$\mathbf{u} \in \mathbb{R}^6$ is the wrench due to gravity force (acting on the CoM) and \mathbf{A}_2 computes the angular component of the gravity wrench, whenever this is expressed at the origin of the world frame:

$$\begin{aligned} \mathbf{A}_1 &= [\bar{\mathbf{A}}_1 \quad \dots \quad \bar{\mathbf{A}}_{n_c}] \in \mathbb{R}^{6 \times mn_c}, \\ \mathbf{A}_2 &= \begin{bmatrix} \mathbf{0} \\ -mg \times \mathbf{P}_{xy}^T \end{bmatrix} \in \mathbb{R}^{6 \times 2}, \quad \mathbf{P}_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \mathbf{u} &= \begin{bmatrix} -mg \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{g} = [0, 0, -g]^T. \end{aligned} \quad (3.9)$$

\mathbf{P}_{xy} is the selection matrix selecting the horizontal components x, y of the CoM and $\bar{\mathbf{A}}_i$ is such that:

$$\bar{\mathbf{A}}_i = \begin{cases} \begin{bmatrix} \mathbf{I}_3 \\ [\mathbf{p}_i]_{\times} \end{bmatrix} \in \mathbb{R}^{6 \times 3} & \text{if } m = 3 \\ \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ [\mathbf{p}_i]_{\times} & \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{6 \times 6} & \text{if } m = 6 \end{cases}$$

where $[\cdot]_{\times}$ is the skew-symmetric matrix associated to the cross product.

3.1. Classical Feasible Region

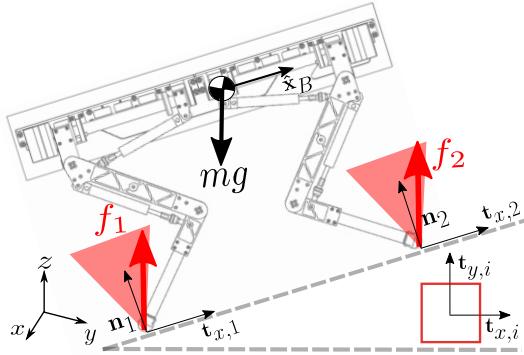


Figure 3.2: Friction pyramids (shown in red) orientation with respect to the contact surface. Each pyramid base is perpendicular to the contact surface normal $\hat{\mathbf{n}}_i$. A top view of the pyramid base with respect to the tangent contact axes $\hat{\mathbf{t}}_{x,i}, \hat{\mathbf{t}}_{y,i}$ is shown (bottom right).

Constraint (III.b) ensures the friction constraints are met. These require the contact forces to be inside inner pyramidal (conservative) approximations of the friction cones. Approximating the friction cones with a low number of linear approximations results in a smaller computation time [15, 32, 61]. The number of edges chosen to represent the pyramid with reasonable accuracy can be chosen based on the complexity of the terrain and the friction coefficient. For each contact, we can define an orthonormal reference frame composed of the contact surface normal $\hat{\mathbf{n}}_i \in \mathbb{R}^3$, and tangent vectors $\hat{\mathbf{t}}_{x,i}, \hat{\mathbf{t}}_{y,i} \in \mathbb{R}^3$ such that $\hat{\mathbf{t}}_{x,i} = \hat{\mathbf{n}}_i \times \hat{\mathbf{x}}_B \times \hat{\mathbf{n}}_i$, where $\hat{\mathbf{x}}_B$ is the unit vector along the X-axis of the base of the robot. Each pyramid is oriented along $\hat{\mathbf{n}}_i$ (with the base of the pyramid parallel to the contact surface) as shown in Fig. 3.2. The constraint matrix $\mathbf{B} \in \mathbb{R}^{4n_c \times 3n_c}$ can then be represented as:

$$\begin{aligned} \mathbf{B} &= \text{diag}(\mathbf{b}_1, \dots, \mathbf{b}_{n_c}), \\ \mathbf{b}_i &= \begin{bmatrix} (\hat{\mathbf{t}}_{x,i} - \mu_i \hat{\mathbf{n}}_i)^T \\ (\hat{\mathbf{t}}_{y,i} - \mu_i \hat{\mathbf{n}}_i)^T \\ -(\hat{\mathbf{t}}_{x,i} + \mu_i \hat{\mathbf{n}}_i)^T \\ -(\hat{\mathbf{t}}_{y,i} + \mu_i \hat{\mathbf{n}}_i)^T \end{bmatrix} \in \mathbb{R}^{4 \times 3} \end{aligned} \quad (3.10)$$

Finally, constraint (III.c) ensures that the torque at each joint does not exceed its limit. These limits are mapped to the end-effector (feet) space by means of the inverse-transpose of the Jacobian (This is true for a non-redundant leg, where the Jacobian is a square matrix). This yields to the definition of force

3.2. Feasible Region Extensions

polytopes that represent the sets of admissible contact forces that respect joint-torque limits, which, from Eq. (3.8), can be rewritten as

$$\mathcal{F}_i = \{\mathbf{f}_i \in \mathbb{R}^{mn_c} \mid \underbrace{\mathbf{D}(\mathbf{q}) + \underline{\tau}_i}_{\underline{\mathbf{d}}_i} \leq \mathbf{J}_i^T \mathbf{f}_i \leq \underbrace{\mathbf{D}(\mathbf{q}) + \bar{\tau}_i}_{\bar{\mathbf{d}}_i}\} \quad (3.11)$$

where $\mathbf{D}(\mathbf{q}) = \mathbf{M}_{bi}^T \ddot{\mathbf{q}} + \mathbf{M}_i \dot{\mathbf{q}}_i + \mathbf{c}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}(\mathbf{q}_i)$. The polytopes can be described in the constraint (III.c) using their half-plane description of such force polytopes, which is represented by $\mathbf{G} \in \mathbb{R}^{2n_{lin_c} \times mn_c}$ and $\mathbf{d} \in \mathbb{R}^{2n_{lin_c}}$:

$$\begin{aligned} \mathbf{G} &= \text{diag} \left(\begin{bmatrix} \mathbf{J}(\mathbf{q}_1)^T \\ -\mathbf{J}(\mathbf{q}_1)^T \end{bmatrix}, \dots, \begin{bmatrix} \mathbf{J}(\mathbf{q}_{n_c})^T \\ -\mathbf{J}(\mathbf{q}_{n_c})^T \end{bmatrix} \right), \\ \mathbf{d} &= \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{n_c} \end{bmatrix}, \mathbf{d}_i = \begin{bmatrix} \bar{\mathbf{d}}_i \\ \underline{\mathbf{d}}_i \end{bmatrix} \end{aligned} \quad (3.12)$$

Because \mathbf{G} and \mathbf{d} are configuration-dependent, the force polytopes and the resulting feasible region are, thus, only locally valid in a neighbourhood of the considered instantaneous configuration. The consequence of this is that the feasible region is accurate only in a neighborhood of the considered robot configuration. Therefore, for every change in the CoM position due to a change in the joint configuration, the feasible region should be recomputed.

With this, we can formally define the *feasible region* encompassing all the CoM positions \mathbf{c}_{xy} that satisfy the friction constraints and the joint-torque constraints simultaneously as:

$$\mathcal{Y}_{fa} = \left\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \mid \exists \mathbf{f}_i \in \mathbb{R}^{mn_c}, \text{ s.t. } (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A} \right\} \quad (3.13)$$

where $\mathcal{C} \cap \mathcal{A}$ is the set of contact forces and CoM positions (projected on an XY plane) satisfying both friction and joint-torque constraints:

$$\mathcal{C} \cap \mathcal{A} = \left\{ \mathbf{f}_i \in \mathbb{R}^{mn_c}, \mathbf{c}_{xy} \in \mathbb{R}^2 \mid \begin{array}{l} \mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{u} \\ \mathbf{B} \mathbf{f} \leq \mathbf{0}, \quad \mathbf{G} \mathbf{f} \leq \mathbf{d} \end{array} \right\} \quad (3.14)$$

3.2 Feasible Region Extensions

In this section, we propose an extension of the feasible region to arbitrary plane inclinations (Section 3.2.1). We then proceed to incorporate external wrenches

3.2. Feasible Region Extensions

(Section 3.2.2), and dynamic effects (Section 3.2.3). The changes on the algorithm are highlighted in blue in Algorithm 3.1.

3.2.1 Generic Plane of Projection

Under the sole influence of gravity and considering only *friction* constraints, the static equilibrium constraints in [11] are only affected by the horizontal position of the CoM¹. Therefore, the high dimensional constraints were naturally projected on a plane perpendicular to gravity (i.e., the horizontal plane). In such case, for a given set of contacts, checking feasibility for a CoM trajectory with a varying height is still appropriate with respect to the projected region. However, when used for planning purposes, computing the region in a plane consistent with the planned motion can be of convenience. One would then simply need to find a feasible 2D CoM trajectory in the plane of reference. Therefore it is important to have the possibility to choose the plane of interest where the region is computed.

More importantly, as will be explained further in Section 3.2.2, under the influence of external and inertial wrenches on the CoM (and when including joint torque and kinematic constraints), the CoM vertical position can alter the feasible region. Therefore, for a given set of contacts, the feasible region will be dependent on the height of the robot; in this case, planning a CoM motion defined in a plane *inconsistent* with the one used for the computation of the region, could result in infeasibility. Thus, to compute the region, it is important to project the high dimensional constraints on the plane where the expected CoM trajectory will lie.

For instance, for a robot climbing a ramp, the planned CoM trajectory can be expected to follow the inclination of the ramp [26][27]. In general, the orientation of the projection plane depends on the planning strategy: choosing a plane of projection consistent with the terrain inclination and with the CoM trajectory ensures a constant CoM height when expressed with respect to such plane. Note that the projection the IP algorithm is in fact a mapping of the high dimensional constraints from the wrench space (or in the case of the kinematic constraints, the joint space) to a Euclidean plane. The Euclidean plane can be chosen to be expressed with respect to the frame of our choice as explained above.

¹The only dependence on the CoM position is due to $\mathbf{c} \times \mathbf{mg} = m\|\mathbf{g}\| [-c_y \quad c_x \quad 0]^T$ in the moment balance constraints. The zero in the last row shows the independence from the vertical coordinate of the CoM.

3.2. Feasible Region Extensions

The inclination of a generic plane of interest Π can be described through a free vector $\hat{\mathbf{n}}$ normal to it (expressed with respect to the world frame). Constraints (III) can be projected on to the plane of interest Π by applying the following change of coordinates:

$$\mathbf{c} = {}^W\mathbf{R}_\Pi \hat{\mathbf{c}} \quad (3.15)$$

where $\mathbf{c} = [\mathbf{c}_{xy}^T \ c_z]^T$ and $\hat{\mathbf{c}} = [\hat{\mathbf{c}}_{\hat{x}\hat{y}}^T \ \hat{c}_{\hat{z}}]^T$ are the CoM position expressed with respect to the world frame W and a frame attached to the plane of interest Π , respectively. ${}^W\mathbf{R}_\Pi$ is the rotation matrix representing the orientation of the plane of interest Π with respect to the world frame W , and is defined as:

$${}^W\mathbf{R}_\Pi = [\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}] \quad (3.16)$$

The \hat{z} -axis of Π is aligned with $\hat{\mathbf{n}}$. $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ are unit vectors (expressed in W frame and forming the \hat{x}, \hat{y} -axes of Π frame) chosen such that they form, together with $\hat{\mathbf{z}}$, a right-handed coordinate system. With the change of coordinates in (3.15), the IP algorithm can be written in terms of $(\hat{\mathbf{c}}_{\hat{x}\hat{y}}, \hat{c}_{\hat{z}})$ and solved for the new coordinates $\hat{\mathbf{c}}_{\hat{x}\hat{y}}$. In the remainder of this dissertation, not to overload the notation, we express the CoM position in the world frame \mathbf{c} in all related equations, without any loss of generality.

3.2.2 External wrenches

Consider an external wrench, $\mathbf{w}_{ext} = [\mathbf{f}_{ext}, \boldsymbol{\tau}_{ext}]^T \in \mathbb{R}^6$, applied on the CoM of a legged robot. For the robot to be in *static* equilibrium, the point mass balance equation (3.1) becomes

$$m\mathbf{g} + \mathbf{f}_{ext} = \sum_{i=1}^{n_c} \mathbf{f}_i \quad (3.17)$$

$$\mathbf{c} \times (m\mathbf{g} + \mathbf{f}_{ext}) + \boldsymbol{\tau}_{ext} = \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i \quad (3.18)$$

The equivalent wrench balance equation becomes

$$\mathbf{w}_g + \mathbf{w}_{ext} = \mathbf{w}_c \quad (3.19)$$

As mentioned in the previous section, with only the gravity \mathbf{g} acting on the robot, the dependence on the CoM in (3.18) only comes from its horizontal positions \mathbf{c}_{xy} . However, with the presence of an external force, \mathbf{f}_{ext} , a dependence on the

3.2. Feasible Region Extensions

CoM vertical position c_z can clearly exist from the term $-\mathbf{f}_{ext} \times \mathbf{c}$ (unless \mathbf{f}_{ext} is aligned with gravity).

To incorporate the effect of \mathbf{w}_{ext} on Algorithm 3.1, the constraint (III.a) can be rewritten by redefining \mathbf{A}_2 and \mathbf{u} to be:

$$\begin{aligned}\mathbf{A}_2 &= \begin{bmatrix} \mathbf{0} \\ -[mg + \mathbf{f}_{ext}] \times \mathbf{P}_{xy}^T \end{bmatrix} \in \mathbb{R}^{6 \times 2} \\ \mathbf{u} &= \begin{bmatrix} -mg - \mathbf{f}_{ext} \\ [\mathbf{f}_{ext}] \times \mathbf{P}_z^T c_z - \boldsymbol{\tau}_{ext} \end{bmatrix} \in \mathbb{R}^{6 \times 1}\end{aligned}\quad (3.20)$$

Therefore, \mathbf{A}_2 computes the moments due to gravity and external forces (acting on the robot CoM²), about the origin of the world frame.

To better appreciate the effect of an external wrench \mathbf{w}_{ext} on the projected region we can further inspect its direct influence on \mathbf{c}_{xy} . \mathbf{c}_{xy} characterizes the set of all the projected feasible CoM positions, given the existence of feasible contact forces \mathbf{f} . From the first two equations in (3.18), \mathbf{c}_{xy} can be determined as [14]:

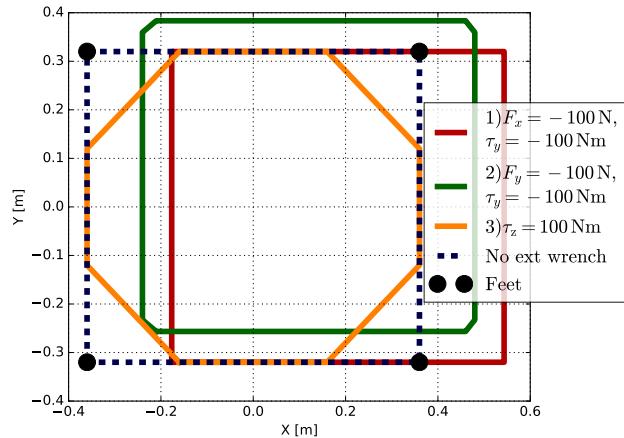
$$\begin{aligned}\mathbf{c}_{xy} &= \frac{1}{-mg + f_{ext,z}} \left([0 \ 0 \ 1]^T \times \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i - c_z \mathbf{f}_{ext,xy} + [-\tau_{ext,y} \ \tau_{ext,x}]^T \right) \quad (3.21) \\ &= -\mathbf{h}(\mathbf{f}, f_{ext,z}) + \mathbf{m}(f_{ext}, \boldsymbol{\tau}_{ext}, c_z)\end{aligned}$$

From the offset function \mathbf{m} , one could observe that an external wrench applied on the robot, combined with the CoM vertical position, results in a shift in the location of the projected CoM positions (i.e., projected region).

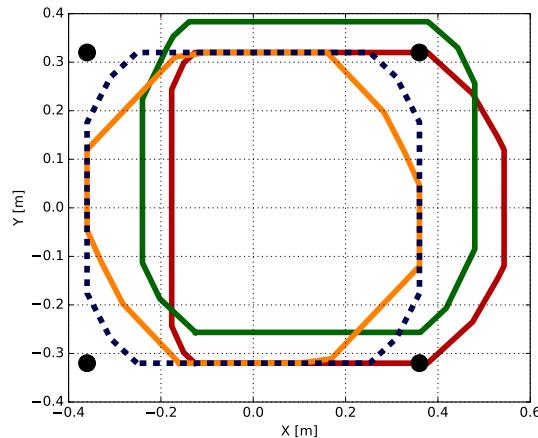
The change in shape of the region, can be intuitively understood, considering that the set of contact forces resulting from the action of the external wrench could become infeasible due to the additional effort needed to compensate for the external wrench. For example, in case of a significantly retracted leg, because the joint-torques are propagated through the leg to the foot via the Jacobian, the CoM positions closer to the contact feet are more likely to be infeasible. Furthermore, a CoM projection located near a specific foot, further loads that foot (while reducing the load on the other feet). This drives the joints of that leg closer to their torque limits making this CoM position more likely to be infeasible. This explains why an external wrench applied on the robot, such as an additional load, results in smaller feasible regions as opposed to the case when only the weight of the robot has to be supported.

²If a pure force is applied in a different point of the robot the equivalent wrench at CoM should be computed.

3.2. Feasible Region Extensions



(a) Friction Region (only friction considered)



(b) Feasible Region (both friction and joint-torque limits considered)

Figure 3.3: Effect of different external wrenches acting on the CoM on the (a) friction region and the (b) feasible region. Changes in size and shifting of the location of the regions can be observed. The components of the external wrench that are mentioned are applied simultaneously and the unmentioned components are set to zero. The stance feet of [Hydraulically actuated Quadruped \(HyQ\)](#) are shown as black points with the front feet facing right. Regions are computed for a trunk height of $c_z = 0.53m$

3.2. Feasible Region Extensions

Figure 3.3 illustrates examples of the resulting friction and feasible regions for different external disturbance wrench cases applied at the **CoM** of the **HyQ** robot at $c_z = 0.53m$. Case 1 (red) and 2 (green) show a shift both in the friction and in the feasible regions in the opposite direction to the external wrench. A reduction in the size of the friction region (e.g., obtained only considering friction constraints (III.b)) can also be seen for an external torque $\tau_{ext,z}$ (orange). This is illustrated by the clipping of the corners of the region, where no admissible set of contact forces could withstand such external wrench without slipping.

3.2.3 Dynamic Motions

To ensure stability/feasibility, it is necessary that the chosen reference point remains inside the admissible region that was computed for it. To evaluate dynamic stability, it is common to consider the **ZMP** as specified reference point. Because the **ZMP** already explicitly considers the horizontal acceleration of the robot's body, this does not have to be considered in the computation of the admissible region: this region, therefore can be obtained for dynamic conditions and, on flat terrains (if only friction cone constraints are considered), it simplifies to the convex hull of the contact points. Therefore, we underline that the choice of a reference point and its admissible region are tightly coupled and that any arbitrary reference point could be used provided that the employed admissible region is specifically formulated in accordance to it. As long as this point is inside the corresponding computed region, we are sure that the constraints that have been considered when building the region are satisfied. Therefore, conforming to the previous sections, we keep using the **CoM** as the reference point and proceed to incorporate the dynamic effects (dropping the static assumptions) in the feasible region (constraints III.a in Algorithm 3.1). In fact, it could even happen that the **ZMP** is outside of the computed region, yet dynamic stability is ensured, and the robot configuration is feasible as long as the **CoM** projection is inside it.

Note that, including dynamic effects requires that we express the Newton-Euler equations in the inertial frame. This means that the moment balance should be done with respect to the origin of the inertial frame, that in general is not coincident with the **CoM**. Then the expression of Newton-Euler equations becomes:

3.2. Feasible Region Extensions

$$\begin{aligned} m(\ddot{\mathbf{c}} - \mathbf{g}) &= \sum_{i=1}^{n_c} \mathbf{f}_i \\ \mathbf{I}_G \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}_G \boldsymbol{\omega} + \mathbf{c} \times m(\ddot{\mathbf{c}} - \mathbf{g}) &= \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i \end{aligned} \quad (3.22)$$

where $I_G \in \mathbb{R}^{3 \times 3}$ is the moment of inertia about the center of mass, $\ddot{\mathbf{c}}$ the CoM Euclidean acceleration, and $\dot{\boldsymbol{\omega}}, \boldsymbol{\omega}$ the angular acceleration and velocity of the robot base, respectively. Adding the effect of external wrenches, the equivalent wrench balance equation becomes:

$$\mathbf{w}_i + \mathbf{w}_g + \mathbf{w}_{ext} = \mathbf{w}_c \quad (3.23)$$

where

$$\mathbf{w}_i = \begin{bmatrix} m\ddot{\mathbf{c}} \\ \mathbf{I}_G \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}_G \boldsymbol{\omega} + \mathbf{c} \times m\ddot{\mathbf{c}} \end{bmatrix} \quad (3.24)$$

is the commonly denoted *inertial wrench*. By inspecting (3.22) one can see that to incorporate the dynamic effects, the matrix \mathbf{A}_1 remains unchanged while \mathbf{A}_2 and \mathbf{u} in constraint (III.a) should be redefined as:

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{0} \\ -m(\mathbf{g} - \ddot{\mathbf{c}}) \times \mathbf{P}_{xy}^T \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} m(\ddot{\mathbf{c}} - \mathbf{g}) \\ \mathbf{I}_G \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}_G \boldsymbol{\omega} \end{bmatrix}, \quad (3.25)$$

Note that now the *point mass* model becomes the *Single Rigid Body Dynamics (SRBD)* model as the angular dynamics is also taken into account. Moreover, the stability enforced in constraint (III.a) can be considered to be fully *dynamic*. As a result of the effect of the inertial accelerations, the computed region can "move" (e.g., forward or backward) according to the direction of the instantaneous body acceleration. With the dependence of the feasible region on the acceleration of the robot, one can utilize the desired body accelerations in the computation of the region to plan dynamically feasible motions.

3.2.4 Degenerate Feasible Regions

It is possible to further extend the feasible region to dynamic gaits in quadrupeds (e.g., a trot or pace) where only one or two point contacts are established with the ground at the same time. In these cases, the classical support polygon collapses to a line connecting the two point feet in case of double stance or to a point in the case of a single stance. As a result, the possible solution space becomes

3.2. Feasible Region Extensions

infeasible in the absence of contact moments. This extension of the feasible region to degenerate cases is made numerically possible by assuming the presence of infinitesimal contact torques at the feet as constraints on the problem to render it feasible. In particular, we assume that the feet can exert a small torque component tangential to the contact surface plane τ_x and τ_y , but no contact torque orthogonal to the plane τ_z . This corresponds to the case of feet with a small non-zero surface, able to adjust the location of the [Center of Pressure \(CoP\)](#) within the contact surface. Such assumption is a numerical (heuristic) assumption introduced solely for the feasibility of the problem. We include these wrench components in the constraint (III.b) of Algorithm 3.1: we update the matrix \mathbf{B} in (3.10) to embed, for each contact i , not just the constraints on the contact forces (i.e., linearized friction cone constraint $\mathbf{b}_i^{cone} \in \mathbb{R}^{4 \times 3}$) but also a box constraint $\mathbf{b}_i^{box} \in \mathbb{R}^{4 \times 2}$ on the contact torques τ_x, τ_y . The values $\tau_x^{lim}, \tau_y^{lim}$ represent the infinitesimal limits of the box constraint on the contact torque tangential to the surface plane in the foot location:

$$\begin{aligned}\mathbf{b}_i^{cone} &= \begin{bmatrix} (\mathbf{t}_{1,i} - \mu_i \mathbf{n}_i)^T \\ (\mathbf{t}_{2,i} - \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{1,i} + \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{2,i} + \mu_i \mathbf{n}_i)^T \end{bmatrix}, \quad \mathbf{b}_i^{box} = \begin{bmatrix} \tau_x^{lim} & 0 \\ 0 & \tau_y^{lim} \\ -\tau_x^{lim} & 0 \\ 0 & -\tau_y^{lim} \end{bmatrix} \\ \mathbf{B} &= \text{diag} \left(\begin{bmatrix} \mathbf{b}_1^{cone} & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{4 \times 3} & \mathbf{b}_1^{box} \end{bmatrix} \dots \begin{bmatrix} \mathbf{b}_{n_c}^{cone} & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{4 \times 3} & \mathbf{b}_{n_c}^{box} \end{bmatrix} \right) \in \mathbb{R}^{8n_c \times 5n_c}\end{aligned}\quad (3.26)$$

Because of the non-zero values of the contact torque limits τ_x^{lim} and τ_y^{lim} , the feasible region portrayed in Fig. 3.4 appears as a narrow stripe with finite area, although it should be regarded as a one-dimensional segment. Indeed, the feasible region in this double point-contact case corresponds to a segment whose length is determined by the robot's actuation limits. In presence of external wrenches acting on the platform, this segment will move away from the line connecting the two feet along the projection plane.

In case of a single point contact, the feasible region will degenerate to a point which represents the only possible value of [CoM](#) projection where the robot could balance the load acting on its trunk. Note that if the dynamic effects are considered, the feasible line will move back/forth when the robot accelerates backwards/forward, according to what is explained in Section 3.2.3. This is exemplified in Fig. 3.4, which shows the feasible region during a trotting motion. The region is a straight segment and is shifted forward with respect to the

3.3. Reachable Region

supporting line because the robot is accelerating forward. The ZMP (green point), instead, moves backward in the opposite direction to the acceleration.

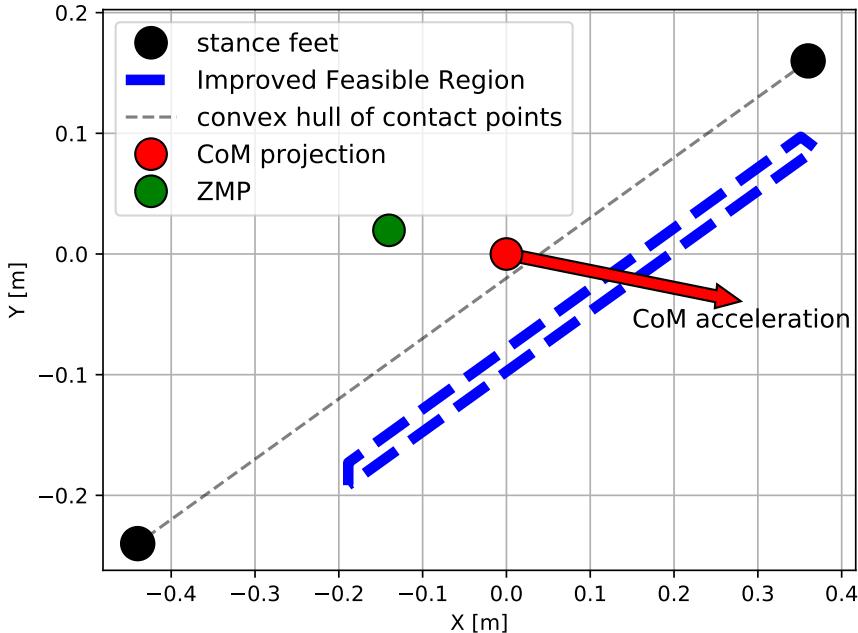


Figure 3.4: The improved feasible region degenerates to a line during a trot when only two feet are simultaneously in contact with the ground. This segment is shifted forward in the same direction as the robot's acceleration. The finite width of the improved feasible region (blue) is due to the infinitesimal contact torques τ_x^{lim} and τ_y^{lim} . The robot is dynamically unstable in this scenario.

3.3 Reachable Region

So far, the feasible region was defined as a region for which the frictional stability of the robot can be ensured without violating the joint-torque limits. The inclusion of the effect of the joint-torque limits has proved to be important in many cases. Once the torque-limits are considered, the limited leg workspace remains the next major restrictive factor for motion planning. This is particularly

3.3. Reachable Region

true in complex terrains, where the robot needs to have complex configurations that may result in joint-kinematic limits violations or leg singularities. Kinematic limits are common, for instance, in linear actuators used in hydraulic quadrupeds, such as [HyQ](#), where the piston stroke is limited. One type of singularity that could be of crucial importance to determine the workspace is related to the loss of mobility due to the complete extension or retraction of one of the legs (e.g., humanoid climbing stairs). In fact, as it will be shown in this section, it often happens that, even if the *feasible region* is sufficiently large, yet the robot [CoM](#) has a very limited reachable workspace. Parallel robots, in general, inherently suffer from such an unfavorable workspace.

We, therefore, seek to extend the definition of the *feasible region* to further incorporate the joint-kinematic limits and the manipulability of the robot. We first introduce the *reachable region*, a two-dimensional level area representing the [CoM](#) reachable workspace. We present a simplified numerical approach that computes a conservative approximation of the region. The method is designed to be *efficient* and therefore allows for *online* motion planning and optimization. Given a desired orientation, we determine the *constant orientation workspace*: namely, the set of all possible [CoM](#) locations that can be reached with a specified orientation without violating the joint-kinematic limits [40]. To simplify the nomenclature, we refer to this set as the *reachable region*. Given the kinematic nature of the problem, we can utilize the forward kinematic relations to map the kinematic constraints of the robot (defined in the joint space) to the task-space (defined in the Cartesian space of the [CoM](#)). Typically, the forward kinematics for each branch in contact (i.e., leg) is defined as:

$${}^{\mathcal{B}}\mathbf{x}_{f_i} = f_i(\mathbf{q}_i), \quad \forall i = 1, \dots, n_c \quad (3.27)$$

mapping the joint angles $\mathbf{q}_i \in \mathbb{R}^{n_i}$ of branch i to the position of the foot ${}^{\mathcal{B}}\mathbf{x}_{f_i} \in \mathbb{R}^3$ (expressed with respect to the body frame). Assuming that the foot position with respect to the world frame ${}^W\mathbf{x}_{f_i}$ is known, ${}^{\mathcal{B}}\mathbf{x}_{f_i}$ can be simply computed as

$${}^{\mathcal{B}}\mathbf{x}_{f_i} = {}^{\mathcal{B}}\mathbf{R}_W({}^W\mathbf{x}_{f_i} - \mathbf{c}) + {}^{\mathcal{B}}\mathbf{c} \quad (3.28)$$

where ${}^{\mathcal{B}}\mathbf{c}$ is the offset of the [CoM](#) with respect to the body frame, and \mathbf{c} is the [CoM](#) position with respect to the world frame. Combining (3.27) and (3.28) and rewriting for \mathbf{c} , we obtain:

$$\mathbf{c} = \mathbf{F}_i(\mathbf{q}_i, {}^W\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_W), \quad \forall i = 1, \dots, n \quad (3.29)$$

where \mathbf{F}_i is defined as:

$$\mathbf{F}_i(\mathbf{q}_i, {}^W\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_W) = {}^W\mathbf{x}_{f_i} - {}^W\mathbf{R}_{\mathcal{B}}(f_i(\mathbf{q}_i) - {}^{\mathcal{B}}\mathbf{c}) \quad (3.30)$$

3.3. Reachable Region

Therefore, for a given foot position ${}^W\mathbf{x}_{f_i}$ and trunk orientation ${}^W\mathbf{R}_B$, (3.29) provides a relationship between the joint-space angles of each leg and the CoM task-space position. We assume the feet do not move during contact. This is enforced by the Whole-Body Control (WBC) used in our framework (see Fig. 4.1) [22]. Therefore, for a CoM position ${}^W\mathbf{x}_{com}$ to be reachable, there must exist joint angles \mathbf{q}_i , satisfying (3.29), for each leg i such that:

1. $\underline{\mathbf{q}}_i \leq \mathbf{q}_i \leq \bar{\mathbf{q}}_i$
2. $J_i(\mathbf{q}_i) = [\partial f_i(\mathbf{q}_i)/\partial \mathbf{q}_i]$ is full rank

where $\underline{\mathbf{q}}_i$ and $\bar{\mathbf{q}}_i$ are the minimum and maximum joint angle limits, respectively, and \leq is an element-wise relational operator.

We can therefore utilize (3.29) (we drop the explicit dependence on ${}^W\mathbf{x}_{f_i}$ and ${}^W\mathbf{R}_B$ that are input parameters, to lighten the notation), along with conditions (1) and (2) defined above, to define the *reachable region* as:

$$\mathcal{Y}_r = \left\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \mid \exists \mathbf{q}_i \in \mathbb{R}^{n_i} \text{ s.t. } (\mathbf{c}_{xy}, \mathbf{q}_i) \in Q \right\} \quad (3.31)$$

where:

$$Q = \left\{ \mathbf{q}_i \in \mathbb{R}^{n_i}, \mathbf{c}_{xy} \in \mathbb{R}^2 \mid \begin{array}{l} \text{s.t. } \mathbf{c}_{xy} = \mathbf{P}_{xy}\mathbf{F}_i(\mathbf{q}_i), \\ \underline{\mathbf{q}}_i \leq \mathbf{q}_i \leq \bar{\mathbf{q}}_i, \quad \text{row-rank}(J_i(\mathbf{q}_i)) = n_l \quad \forall i = 1, \dots, n_c \end{array} \right\} \quad (3.32)$$

where only the legs in contact are considered. It is important to note that such a set can be composed from the intersection of pairs of concentric circles [28]. This, in general, results in a non-convex set. The problem of finding such a set accurately is difficult and time-consuming. Various techniques have been proposed to determine the workspace of manipulators by using analytic, geometric, or numerical approaches. Most analytic and geometric methods can make the analysis of the geometry very complex or can be specific to only one platform. We, therefore, employ a numerical approach that provides an approximation of the region smartly designing it to remain efficient for any generic platform. Numerical methods mostly either sample the joint-space and utilize the forward kinematics or, conversely, sample the task-space and utilize the inverse kinematics. In the case of quadrupeds, the dimension of the joint-space can be large (12-dimensional in the case of most robots). Therefore, we choose to utilize the inverse kinematics to determine the reachable region.

Algorithm 3.2 describes the procedure developed to compute the region. A similar algorithm was developed in [20] and was used to evaluate the workspace of

3.3. Reachable Region

a Stewart platform based machine tool. We further apply a modification to increase the robustness and the performance. Inspired by ray-casting algorithms, a discretized search is done iteratively in ordered directions along polar coordinates (ρ, θ) starting from the current **CoM** projection. This generates a 2D polygon whose vertices are ordered and belong to the boundary of the reachable region, therefore representing a *polygonal* approximation of the said region. For the sake of simplicity, for the remainder of this dissertation, we will refer to the reachable region \mathcal{Y}_r as its polygonal approximation.

Each ray in Algorithm 3.1, along some direction \mathbf{a}_i , finds the farthest point \mathbf{v}_{xy}^* that still belongs to the region. By construction, this point belongs to the boundary of the region and the problem of computing it can be stated, utilizing the inverse kinematics, as:

$$\max_{\mathbf{v}_{xy}} \mathbf{a}_i^T \mathbf{v}_{xy} \quad (3.33)$$

s.t. $\forall i = 1, \dots, n_c$:

$$\mathbf{q}_i = \bar{\mathbf{F}}_i(\mathbf{v}_{xy}) \quad (3.34)$$

$$\underline{\mathbf{q}}_i < \mathbf{q}_i < \bar{\mathbf{q}}_i \quad (3.35)$$

$$\sigma_{\min} \left\{ J \left(\mathbf{q}^k \right) \right\} > \sigma_0 \quad (3.36)$$

The relation (3.34) represents the kinematic constraint in (3.32) reformulated in terms of the inverse kinematics. $\bar{\mathbf{F}}_i$, therefore, is defined as:

$$\bar{\mathbf{F}}_i(\mathbf{v}_{xy}) = f_i^{-1} [\mathcal{B} \mathbf{R}_W (\mathcal{W} \mathbf{x}_{f_i} - \mathbf{P}_{xy}^T \mathbf{v}_{xy} - \mathbf{P}_z^T c_z) + \mathcal{B} \mathbf{c}] \quad (3.37)$$

where f_i^{-1} refers to the inverse kinematics mapping. It is important to note from (3.37) that for specific feet positions, the location of each \mathbf{v}_{xy}^* (and accordingly the resultant region) is influenced by the height c_z and the orientation $\mathcal{W} \mathbf{R}_B$ of the robot. A simple check for the presence of a singularity is done in (3.36), where σ_{\min} is the smallest singular value and σ_0 is a small value of choice. Due to the non-linearity of constraints (3.34) and (3.36) the problem cannot be casted as a linear program (LP), and we employ a ray-casting approach for the solution. A bisection search could be utilized to speed up the search for \mathbf{v}_{xy}^* . We first perform an evenly distributed search along the selected direction \mathbf{a}_i , with steps $\Delta\rho$, to find both the last point inside the region and the first point outside. These correspondingly generate the interval $[\rho - \Delta\rho, \rho]$ where \mathbf{v}_{xy}^* lies within. A fast bisection search is then executed on this interval to find \mathbf{v}_{xy}^* while making sure it is within an error of $[0, -\Delta\rho_{\min}]$ from the boundary of the actual

3.3. Reachable Region

workspace. The function $\text{isReachable}(\rho)$, used in Algorithm 3.2, computes the inverse kinematics of a CoM position and checks if that position is reachable:

isReachable(ρ):

```

 $v_{xy} \leftarrow c_{xy} + \rho \mathbf{a}$ 
 $\mathbf{q}_i = \bar{\mathbf{F}}_i(v_{xy})$ 
return true if  $\mathbf{q}_i$  satisfies (3.35) & (3.36)

```

Each vertex v_{xy}^* is added to the vertex description \mathcal{Y}_r such that the (non-convex) hull of the ordered set of vertex becomes an approximation of the real reachable region (see Fig. 3.5 and 3.6)³. The algorithm stops when a step smaller than $\Delta\rho_{\min}/2$ set by the user is reached. $\Delta\theta = 10^\circ$ and $\Delta\rho_{\min} = 0.03m$ are used for the shown figures. For a faster computation during planning, $\Delta\theta = 20^\circ$ was sufficient for the simulation and experimental results.

A key assumption taken in the algorithm is that the center of the *reachable region* is the *current CoM* location. This speeds up a necessary first step of searching for an approximate center from which to start the algorithm from. Moreover, this provides better boundary precision when determining the boundary of the region that is closer to the *CoM* position, presenting a safer analysis. As a consequence, the dependence of the algorithm from c_{xy} , only influences the *accuracy* of the generated region. A disadvantage of such a choice is the inability to compute the region if the robot is already in an out-of-reach configuration. Nevertheless, given that the locomotion planning shall be done in coherence with the reachable region (see Section 4.2.1), the trajectory of the *CoM* shall always remain inside the region. Section 3.7 provides a further discussion of the nature of the *CoM* workspace and the reachable region.

On the other hand, it is important also to consider the effect of the robot height c_z and orientations ${}^W\mathbf{R}_{\mathcal{B}}$ on the reachable region. In fact, different evaluations of the reachable region, presented in Fig. 3.5 and 3.6, show that the size, positioning, shape, and convexity of the reachable region can differ greatly at different c_z and ${}^W\mathbf{R}_{\mathcal{B}}$. Unsurprisingly, one can observe that the region tends to become smaller at high and low heights since the legs have, in general, less mobility when fully extended or retracted. Furthermore, a deviation from the default horizontal orientation results in smaller regions and could additionally skew the shape of the region towards one side. In both cases, at certain config-

³The large peaks in the edges of the reachable regions in Fig. 3.5 and 3.6 are due to the physical nature of the workspace. Smaller rough edges along the boundary of the regions are due to the discretization step used.

3.3. Reachable Region

Algorithm 3.2 Iterative discretized ray-casting algorithm

```

1: Input:  $\mathbf{c}_{xy}, c_z, {}^W\mathbf{R}_B, \mathbf{p}_1, \dots, \mathbf{p}_{n_c}, \underline{\mathbf{q}}_1, \dots, \underline{\mathbf{q}}_{n_c}, \bar{\mathbf{q}}_1, \dots, \bar{\mathbf{q}}_{n_c}$ 
2: Result: reachable region  $\mathcal{Y}_r$ 
3: Initialization:  $\mathbf{v}_{xy} = \mathbf{c}_{xy}$ ,  $\mathcal{Y}_r \leftarrow \{\}$ 
4: for  $\theta = 0$  to  $2\pi$  do
5:   Compute direction:  $\mathbf{a}_i = [\cos \theta \ \sin \theta \ 0]^T$ 
```

Find the first bisection interval:

```

6:   while  $isReachable(\rho)$  do
7:      $\rho \leftarrow \rho + \Delta\rho$ 
8:   end while
```

Bisection search:

```

9:    $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$ 
10:  while  $\Delta\rho \geq \Delta\rho_{min}/2$  do
11:    if  $isReachable(\rho)$  then
12:       $\rho \leftarrow \rho + \Delta\rho$ 
13:    else
14:       $\rho \leftarrow \rho - \Delta\rho$ 
15:    end if
16:     $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$ 
17:  end while
```

```

18:  if last  $\mathbf{v}_{xy}$  not  $isReachable(\rho)$  then
19:     $\rho \leftarrow \rho - \Delta\rho_{min}$ 
20:     $\mathbf{v}_{xy} \leftarrow \mathbf{c}_{xy} + \rho \mathbf{a}$ 
21:  end if
22:   $\mathcal{Y}_r \cup \{\mathbf{v}_{xy}^*\}$ 
23: end for
24: return  $\mathcal{Y}_r$ 
```

urations, the convexity of the region can be significantly affected. Such insight is greatly useful in situations where planning needs to be performed in rough terrains.

3.4 The Improved Feasible Region

The reachable region can be seen as a projection of the high-dimensional convex set \mathcal{Q} (defined in section 3.3) onto a 2D subspace. Henceforth, with the feasible region and the reachable region defined on the same plane, one could extend the definition of the feasible region to further include the **CoM** positions that are also reachable. In other words, this would present a comprehensive 2D region of all the feasible **CoM** positions \mathbf{c}_{xy} that satisfy the friction constraints, the joint-torque constraints, and the joint-kinematic constraints simultaneously. We can therefore define an *improved feasible region* as:

$$\mathcal{Y}_{far} = \left\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \mid \exists \mathbf{f}_i \in \mathbb{R}^{mn_c}, \mathbf{q}_i \in \mathbb{R}^{n_l} \text{ s.t.} \right. \\ \left. (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A}, \quad (\mathbf{c}_{xy}, \mathbf{q}_i) \in \mathcal{Q} \right\} \quad (3.38)$$

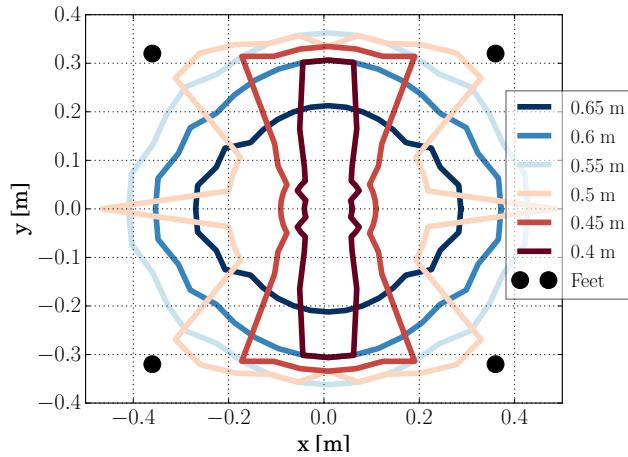
The *improved feasible region* can simply be obtained by intersecting the *feasible region* with the *reachable region*. This is in contrast with the case of attempting to obtain the *feasible region* \mathcal{Y}_{fa} by the simple intersection of the friction region \mathcal{Y}_f and the actuation region (centroidal mapping of joint-torques) \mathcal{Y}_a as explained in [44]. In general, since \mathcal{C} and \mathcal{A} are defined on the same space, the intersection of the two sets (e.g., stacking both friction and joint-torque constraints) must be carried out first before projecting the resulting set. The converse is inaccurate since the intersection and projection operators do not commute. In the case of the reachable region, the constraints are defined not on contact forces but on joints' angular positions, so this issue does not exist.

Therefore, we can define the *improved feasible region* as:

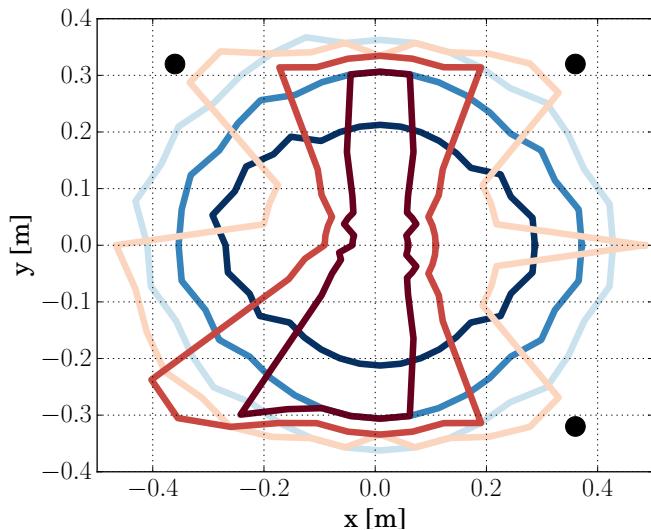
$$\mathcal{Y}_{far} = \mathcal{Y}_{fa} \cap \mathcal{Y}_r \quad (3.39)$$

Finally, differently from the \mathcal{Y}_{fa} region, that took into account only friction and joint-torque constraints, the improved feasible region \mathcal{Y}_{far} will be non-convex because the reachable region is non-convex (given that the set produced from the intersection between a convex set, and a non-convex set is non-convex). In Tab. 3.1, we summarize the type of regions introduced together with the correspondent constraints.

3.4. The Improved Feasible Region



(a) Four stance feet.



(b) Three stance feet.

Figure 3.5: Different evaluations of the reachable region at different HyQ CoM heights.

3.5. Feasibility Margin

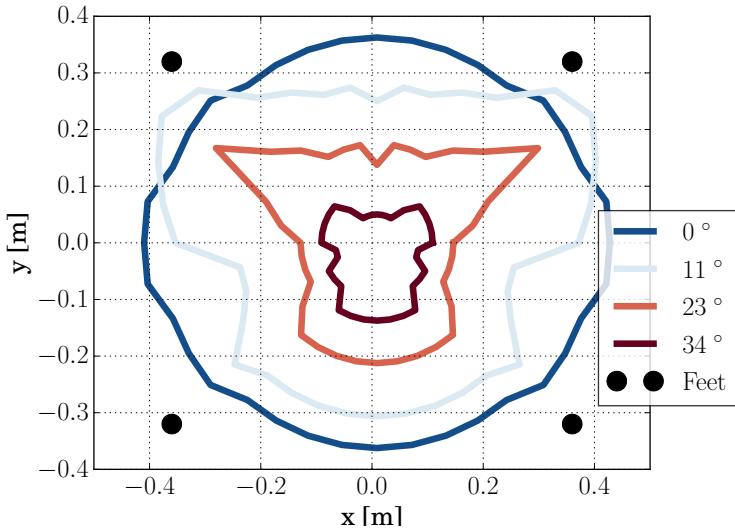


Figure 3.6: Different evaluations of the reachable region at different HyQ roll orientations.

Name	Symbol	Constraints
Friction R. ([11])	\mathcal{Y}_f	Friction
Feasible R. ([44])	\mathcal{Y}_{fa}	Friction / Joint-torque
Reachable R. (this thesis)	\mathcal{Y}_r	Kinematic
Improved Feasible R. (this thesis)	\mathcal{Y}_{far}	Friction / Joint-torque / Kinematic

Table 3.1: Types of regions

3.5 Feasibility Margin

One of the advantages of using the *improved feasible region* is that it provides an intuitive method of planning the CoM trajectory in the Cartesian space. Therefore, one can define simple 2D measures of feasibility in such space. In this section, we introduce the *feasibility margin*. The margin is considered to be a scalar value that provides a measure of the safety of the robot's configuration with respect to the feasibility constraints.

We define the margin as the distance between the current CoM position and the closest boundary of the feasible region. The margin is computed based on the desired wrench (desired trajectory and external wrench compensation), therefore taking into account the locomotion goals of the robot. Unlike to the stability

3.5. Feasibility Margin

margin developed in [46], the feasibility margin considers the effect of external wrenches and incorporates with the joint-kinematic limits. Furthermore, computation of the [Instantaneous Capture Point \(ICP\)](#) is not needed.

The computation of the *feasibility margin* is achieved through the following steps:

- 1) compute the *improved feasible region*
- 2) find the minimum distance between the CoM and the edges of the improved feasible region.

Given that the improved feasible region is nonconvex, we cannot compute the margin using the half-plane description of the edges of the region. Instead, we iteratively compute the margins of all the edges and find the minimum value.

We will use the margin to devise a selection criterion that prioritizes footholds, maximizing the margin throughout the desired trajectory (see Section 4.3). To use the feasibility margin in a gradient-based optimization problem, we would need to compute its gradient. Due to the nature of the region, this would be achievable solely through the finite-difference of the numerical computation. This is hugely inefficient as the computation of one region can range from 30-40 ms (with the gradient requiring over 1 sec). Instead, we proceed to approximate the feasibility margin through training a [Multi-Layer Perceptron \(MLP\)](#) network. The network output is the scalar value of the feasibility margin while the input to the [MLP](#) can be given by

$$\mathbf{x} = (\mathbf{e}_g, \ddot{\mathbf{c}}, \dot{\boldsymbol{\omega}}, \mathbf{f}_{\text{ext}}, \boldsymbol{\tau}_{\text{ext}}, \mathbf{p}_1, \dots, \mathbf{p}_{N_c}, \mu, \mathbf{n}_1, \dots, \mathbf{n}_{N_c})$$

$\mathbf{e}_g \in \mathbb{R}^3$ is a unit vector representing the gravity axis in the base frame B , i.e., the third row of the robot's base orientation matrix $R \in SO(3)$. $\ddot{\mathbf{c}} \in \mathbb{R}^3$ is the linear acceleration of the robot's CoM whereas $\dot{\boldsymbol{\omega}} \in \mathbb{R}^3$ is the angular acceleration of the robot's base. $\mathbf{f}_{\text{ext}}, \boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^3$ compose the wrench of an external disturbance applied on the robot's base. $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{n}_i \in \mathbb{R}^3$ represent the feet positions and the contact normals, respectively. All the variables are represented in the base frame B .

Note that the input of the network includes neither the [CoM](#) position nor the contact states, even though they are required to compute the region and the feasibility margin. The [CoM](#) position is implicitly represented in the input given that the feet positions (and the vector \mathbf{e}_g) are represented in the base frame. Defining the contact configuration as $c_1 c_2 c_3 c_4$, we choose to remove the

3.6. Validation

dependence on the contact states $c_i \in \{0, 1\}$ for $i = \{1, \dots, N_c\}$ from the architecture of the network (as an input) and instead train different MLP networks $\mathbf{FM}_{c_1 c_2 c_3 c_4}$ for each contact configuration. The contact states c_i are instead used as switches to activate which network to be used in our optimization given the contact configuration. For instance, for a crawling gait where only one foot can be in swing at a given time, the networks that will be active periodically are: \mathbf{FM}_{1111} , \mathbf{FM}_{0111} , \mathbf{FM}_{1011} , \mathbf{FM}_{1101} , and \mathbf{FM}_{1110} . For a trotting gait, this would instead be: \mathbf{FM}_{1001} and \mathbf{FM}_{0110} . This simplifies the network architecture as it possesses less inputs. Furthermore, a better model accuracy can be achieved with the same number of sampled data.

The architecture of the networks is composed of 3 hidden layers, with the first layer having 512 neurons, the second layer having 256 neurons, and the third layer having 128 neurons. The activation function used in the hidden layers is the ReLU function, while the output layer has no activation function. Depending on the number of contacts, the input layer is composed of 40 (full stance), 34 (three feet in stance), or 28 (two feet in stance) neurons. The networks are trained using the Adam optimizer, and the loss function used is the mean squared error.

The networks are trained on a dataset of $5 \cdot 10^6$ samples. The dataset is generated by sampling the input variables from a combination of uniform and normal distributions depending on the nature of the variable. Table 3.2 shows the distribution of the parameters used to generate the dataset. The corresponding feasibility margin is computed using the iterative projection algorithm.

3.6 Validation

In this section, we validate the kinematic feasibility introduced in the improved feasible region and the accuracy of the feasibility margin.

3.6.1 Kinematic Feasibility

We first show the reachable region associated with the *Aliengo* robot as the robot changes its height and orientation and the corresponding effect on the robot joint configurations. Figure 3.7 shows the reachable region for the *Aliengo* robot at a nominal configuration. Since the robot possesses liberal joint ranges, the region is large in size at nominal configuration. However, when the base height or orientation of the robot is significantly altered, the **Knee Flexion-Extension (KFE)** joints of the robot are pushed to their limits. As a result,

3.6. Validation

Parameter	Distribution
Base roll and pitch angles	$\alpha, \beta \sim U(-\pi/4, \pi/4)$
COM linear acceleration	$\ddot{c}_x, \ddot{c}_y \sim U(-4, 4)$ $\ddot{c}_z \sim U(-3, 3)$
Base angular acceleration	$\dot{\omega}_x, \dot{\omega}_y \sim \mathcal{N}(0, 0.5)$ $\dot{\omega}_z \sim \mathcal{N}(0, 0.25)$
Friction coefficient	$\mu \sim U(0.3, 0.8)$
Feet deviation from nominal position	$p_x, p_z \sim U(-0.35, 0.35)$ $p_y, p_z \sim U(-0.225, 0.225)$
Contact normal rotations	$\alpha, \beta \sim \mathcal{N}(0, \pi/8)$
External force	$F_x, F_y, F_z \sim \mathcal{N}(0, 300)$
External torque	$\tau_x, \tau_y, \tau_z \sim \mathcal{N}(0, 150)$

Table 3.2: Parameter distributions for creating the dataset of HyQReal robot.

the reachable region decreases in size, and the CoM projection approaches the boundary as shown in Fig. 3.8a and 3.8b. The robot at such configuration has limited mobility in its **Hip Flexion-Extension (HFE)** joints to move its CoM in the horizontal plane.

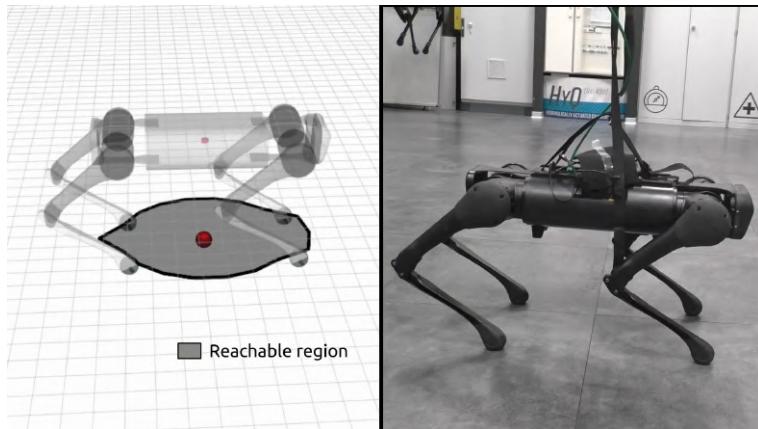


Figure 3.7: Reachable region for the *Aliengo* robot at a nominal configuration during experiments.

Figures 3.9a, 3.9b, and 3.9c show the consequence of moving the CoM outside the reachable region using the HyQ robot in simulation. As the CoM moves outside the region, the limits of the robot joints are violated. At that point, the

3.6. Validation

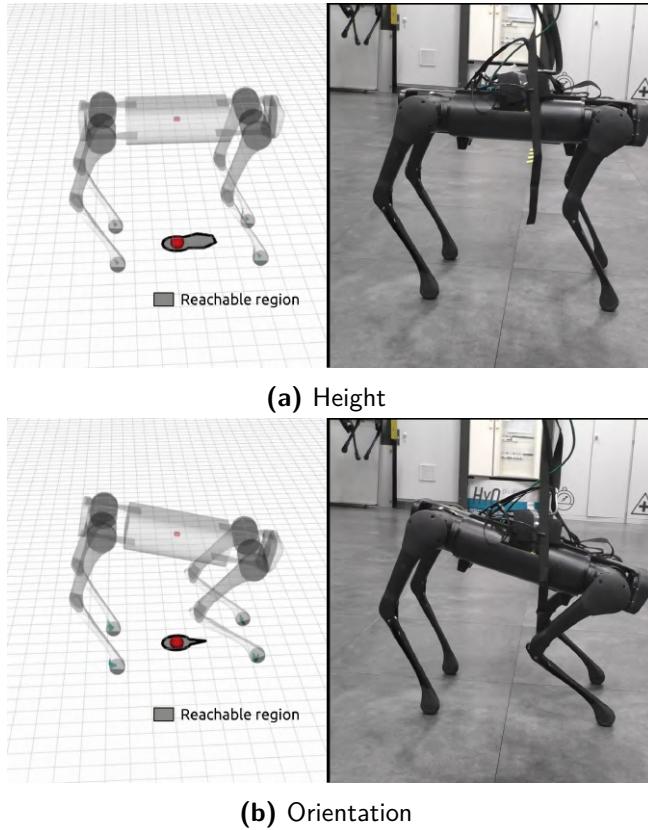


Figure 3.8: Reachable region for the *Aliengo* robot at a height and an orientation that bring the joints to its limits.

WBC loses control of the contact forces, as shown by the mismatch of desired forces illustrated with the blue arrows and the actual forces shown with the green arrows. As a result, control over the CoM is lost, and furthermore, foot slippage occurs. We further show the degradation of the CoM trajectory tracking during experiments in Section 4.7.

3.6.2 Feasibility Margin Network

Furthermore, we validate the learnt feasibility margin network by analyzing the accuracy of network training. Figure 3.10 shows the comparison of the margin computed numerically and the margin computed using the network. In Fig. 3.10a, the margin is computed for the [Hydraulically actuated Quadruped Real \(HyQReal\)](#) during a 1011 stance configuration when varying the x and y

3.6. Validation

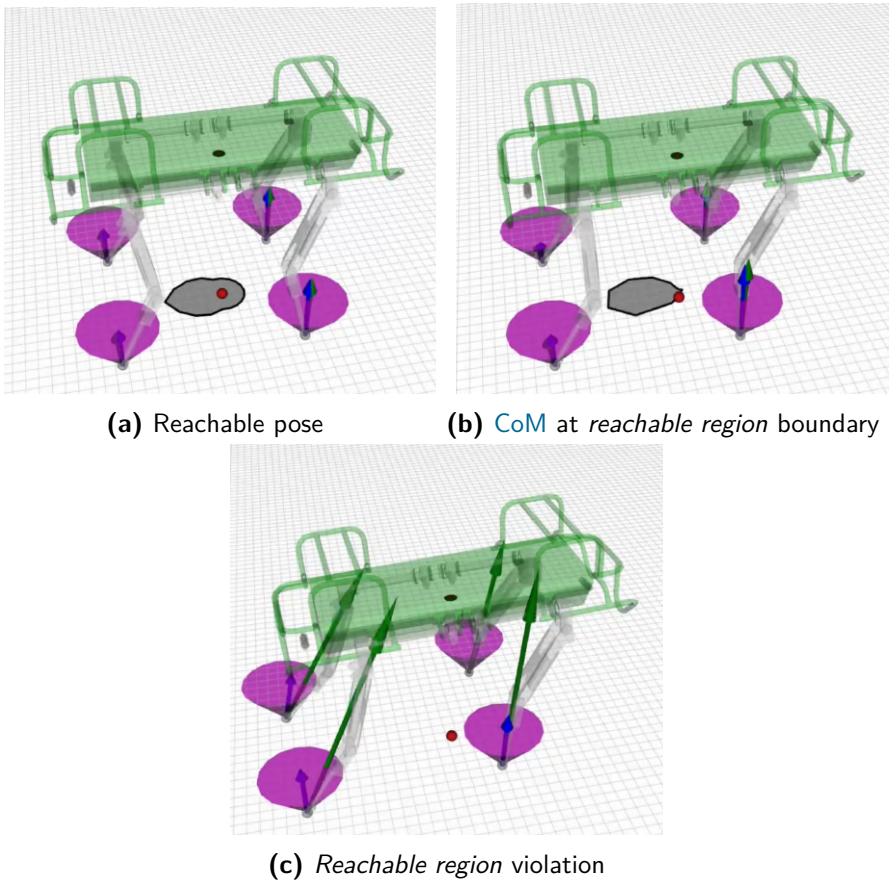


Figure 3.9: *Reachable region* (grey) for **HyQ** at different configurations. The blue arrows show the desired contact forces, while the green arrows show the actual contact forces.

positions of the **Left-Front (LF)** foot. All other parameters are kept in their nominal values. The plot shows that the margin correctly increases as the foot moves forward and to the right, as this moves the diagonal of the region away from the **CoM** (the diagonal boundary is the closest margin to the **CoM** in this configuration). At some point, however, the margin starts to decrease as the foot moves too far and the joints are pushed to their limits. Furthermore, we show the margin computed for the **HyQReal** during a 0111 stance configuration in Fig. 3.10b for a range of F_x and F_y external forces. The plot shows that the margin correctly decreases as the external force is increased in the x and y directions. This is due to the fact that the push in the positive x and y directions shifts the region in the opposite direction, bringing the diagonal boundary closer

3.7. Assumptions Summary

to the CoM.

The plots show that the network is able to accurately predict the margin with a negligible error between the numerical and learned region in most of the input range and a maximum error of 0.015m. Furthermore, for the network to be utilized in a foothold optimization, it is more crucial that the location of the optima of the margin is accurate given the foothold position. In fact, the maximum error in Fig. 3.10a is 0.02m.

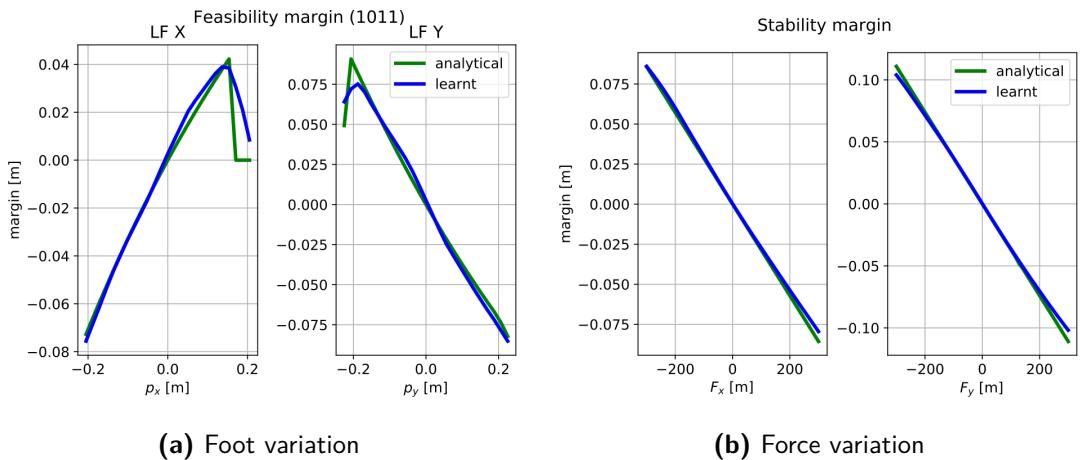


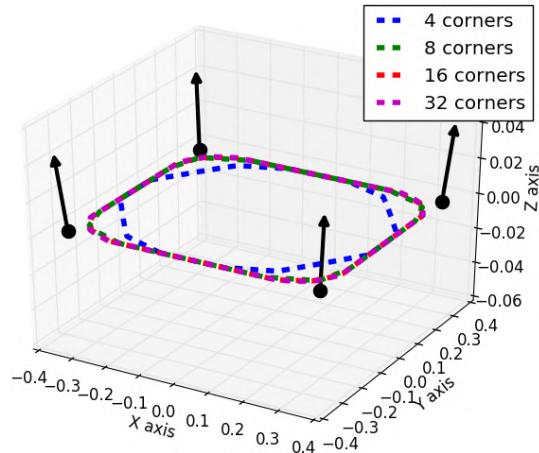
Figure 3.10: Comparison of the learned feasibility margin with the numerical margin. The green line shows the margin computed numerically, while the blue line shows the margin computed using the network.

3.7 Assumptions Summary

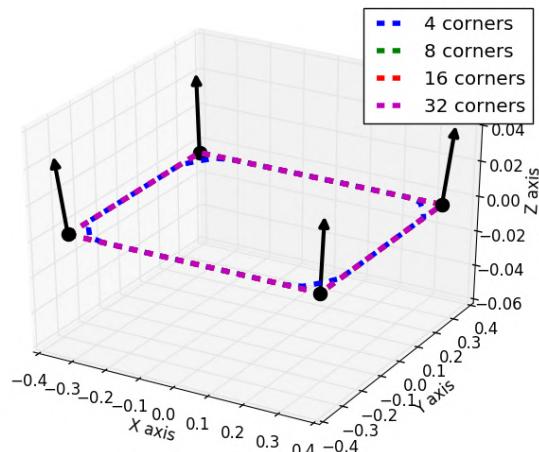
As mentioned in the last sections, several assumptions were made during the computation of the *feasible region* and the *feasibility margin*. Table 4.1 provides a summary of these assumptions, their types and purposes, and where it was discussed in the dissertation. We provide further discussion of the assumptions:

- Assumption 1: This refers to the conservative approximation of the friction cones with inner pyramids. Friction constraints (III.b) in the IP algorithm 3.1 can therefore be expressed linearly. This changes the problem from a second-order cone programming problem to an LP one, resulting in a smaller computation time.

3.7. Assumptions Summary



(a) $\mu = 0.5$



(b) $\mu = 0.7$

Figure 3.11: Friction regions using friction pyramids of different numbers of base edges. The black arrows show the contact normal to each foot.

3.7. Assumptions Summary

	Assumption	Type	Purpose	Sections
1	Friction pyramids	M	CE	3.1.2
2	Tangential contact moments \neq zero	M	SF	3.2.4
3	Approximate center of reachable region	A	CE	3.3
4	$\omega \times \mathbf{I}_G \omega = 0$ for <i>feasibility margin</i>	M	CE	3.5

Table 3.3: List of assumptions used in the calculation of the improved feasible region and during planning, along with their types and purposes. The abbreviation of the types are as follows: M - Modelling and A - Algorithmic. The abbreviations of the purposes are as follows: CE - Computation Efficiency and SF - Solution Feasibility.

Given the usage of polyhedra for the approximation, two design considerations arise. The first is the number of edges used to approximate the friction cone. We compared various friction regions computed using friction polyhedra with increasing number of base edges (to approximate the friction cones). Depending on the complexity of the terrain and the coefficient of friction, we found that the use of friction pyramids of 4-8 base edges can be a sufficient approximation of the friction cone. Figure 3.11(a) shows the friction regions computed for a friction coefficient of 0.5 in a complex terrain. For such a friction coefficient, the usage of a friction pyramid with four base edges captures most of the friction region with some loss in accuracy. Given that the robots used in our work have rubber feet, we also examine the friction regions for a friction coefficient of 0.7, which is typical of rubber feet contacts with many materials [25]. We found that in such a case, using an approximation of 4 base edges affects accuracy negligibly and is sufficient for our purposes (see Fig. 3.11(b)).

The second consideration is the orientation of the friction pyramids. The resulting friction regions can be different according to the orientation of the pyramids. Figure 3.12 shows two different orientations of a friction pyramid base with respect to the cone base and Fig. 3.13 shows the effect of this on the friction region for pyramid bases of four and 8 (for a friction coefficient of 0.5). This difference becomes negligible when increasing the number of edges to approximate the cones (e.g. eight sides or more). Again, we saw negligible differences due to the different orientations for a friction coefficient of 0.7.

In addition to the two considerations above, note the pyramidal approxima-

3.7. Assumptions Summary

tions are conservative. This is due to the fact that they are inner approximations. Therefore, any loss of accuracy does not affect the feasibility/stability of the motion.

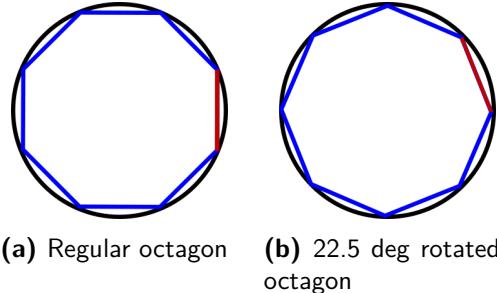


Figure 3.12: Different orientations of the friction pyramid approximations (with eight base edges) with respect to friction cones seen from the base.

- Assumption 2: With the classical support polygon collapsing to a point (line) during single (double) support phases, the possible solution space becomes infeasible in the absence of contact moments. Therefore, we introduced a numerical (heuristic) assumption by allowing infinitesimal contact moments as constraints on the problem to render it feasible. This corresponds to the case of feet with a small non-zero surface, which are able to adjust the location of the Center of Pressure within the contact surface. As a result, the **IP** algorithm can directly be applied to single or double contact phases similar to phases with more contacts.
- Assumption 3: In order to efficiently compute a reachable region suitable for planning, an approximation of the full **CoM** workspace was needed. The full workspace can be comprised of disjoint sets (e.g., [20]) which would not be captured by the algorithm. Figure 3.14 shows the **CoM** workspace for a height of 0.49 m for HyQ robot, where the green and red points show the kinematically feasible and infeasible locations, respectively, obtained using a brute force approach where each point in a two-dimensional grid (grid point distance of 2.5 cm) was tested and marked accordingly.

As evident in Fig. 3.14, a special case can arise at some configurations where the workspace is non-convex, and some disjoint sets appear. Given the nature of ray casting algorithms, in such cases the method only determines the reachable region in the range of rays casted from an initial point. The black

3.7. Assumptions Summary

dashed boundary in the figure shows the output of the algorithm when started from the center of the workspace. In fact, the disjoint regions on the right and left sides are undetected by the algorithm. A solution for this can be to start the algorithm from different parts of the workspace and attempt to rebuild the workspace. However this introduces needless complexity because disjoint regions are nevertheless infeasible for planning, given that no continuous trajectory can be constructed.

- Assumption 4: To include the inertial effects due to dynamic motions when computing the improved feasible region, the angular velocity, and linear and angular accelerations are needed. Consequently, the three quantities are needed as inputs to the [MLP](#) network used for the *feasibility margin*. To simplify the architecture of the network, and decrease the amount of data needed for the network training to achieve good accuracy, we can assume that the term due to the Coriolis effects $\omega \times \mathbf{I}_G \omega$ can be neglected. This is a reasonable assumption given that the robot is not expected to be rotating at fast angular speeds [62]. As a result, the network can be trained to estimate the feasibility margin without the need for the angular velocity and the size of the input is reduced.

3.7. Assumptions Summary

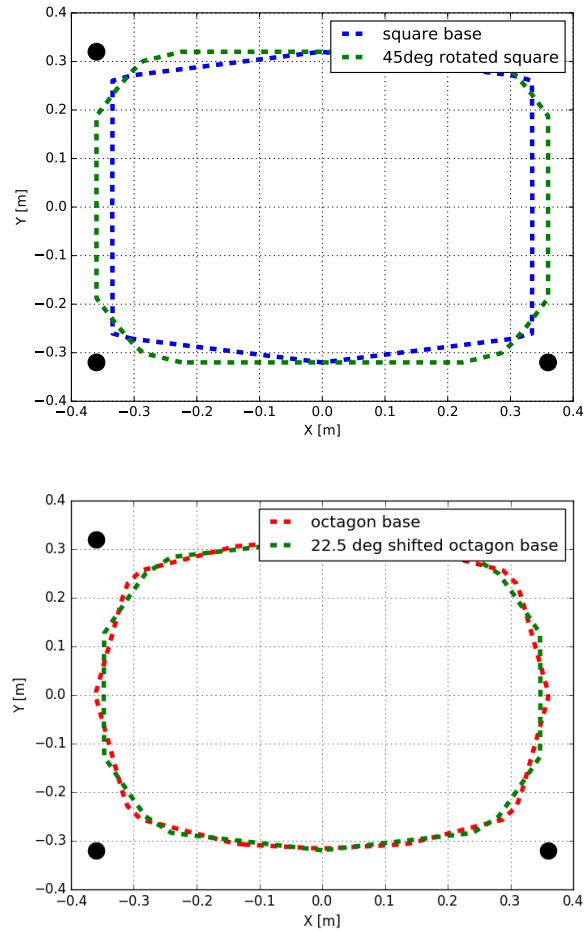


Figure 3.13: The effect that different orientations of the friction pyramid approximations with respect to friction cones can have on the friction region.

3.7. Assumptions Summary

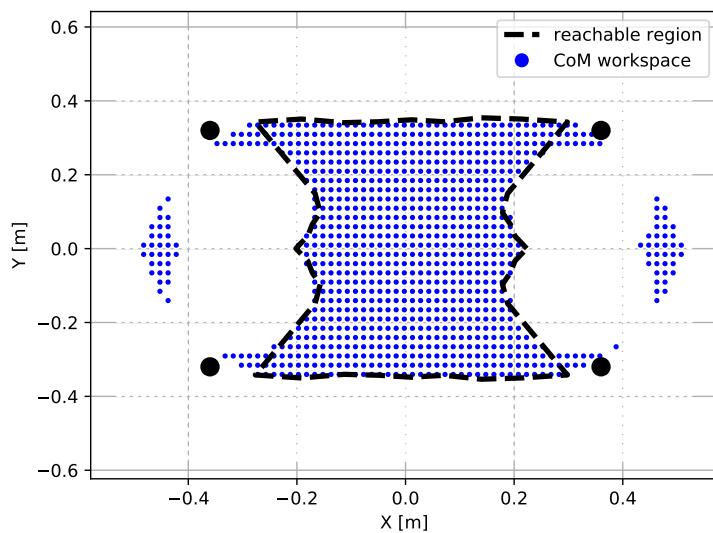


Figure 3.14: Comparison between the CoM workspace (ground-truth blue) and *reachable region* (black dashed) for HyQ at 0.49 m CoM height.

3.8 Conclusions

This chapter presented a comprehensive formulation for defining and computing feasibility criteria based on desired wrench for legged robots, namely, the *improved feasible region* and the *feasibility margin*. The improved feasible region provides a tool to ensure dynamic balance and represents physical constraints like joint torques and kinematic limits in an efficient and intuitive manner. The feasibility margin provides a simple measure, utilizing the feasible region of the robot's feasibility with respect to the feasibility constraints. The margin can be further approximated using a data-driven model, making it more suitable for gradient-based optimal planning. The improved feasible region and the feasibility margin provide an efficient method for planning motion trajectories over complex terrain, taking into consideration dynamic conditions and external disturbances.

Chapter 4

Feasible Wrench Based Planning

Incorporates material from the following publication:

Abdelrahman Abdalla, Michele Focchi, Claudio Semini. “An Efficient Paradigm for Feasibility Guarantees in Legged Locomotion”. *IEEE Transactions on Robotics (T-RO) 2023.*

Paper: <https://ieeexplore.ieee.org/document/10149812>

Video: <https://www.youtube.com/watch?v=vJpNWnMh0wE&t>

Incorporates material from the following publication:

Abdelrahman Abdalla, Victor Barasuol, Claudio Semini. “Feasible Wrench Based Locomotion Planning for Legged Robots”. (under preparation).

4.1 Introduction

Traditional planning strategies utilize heuristics and optimization to plan motion trajectories for legged robots. The approaches mainly use simplified models of the robot and focus on generating trajectories that prioritize the robot's dynamic stability. However, the feasibility of the motion is not always guaranteed, especially when the robot is required to achieve a desired wrench over a complex terrain or when interacting with the environment. To achieve the desired wrenches on the real robot, it is crucial that motion feasibility be considered in the planning process.

Using feasibility criteria, we can design a feasibility-based locomotion strategy for legged robots that provides a generic approach to maximize the desired wrench achievability. Our approach provides a simple way of integrating locomotion and interaction tasks in a single generic manner. For instance, a robot going through a complex terrain while simultaneously needing to pull an object or there is an onboard arm interacting with the environment.

This chapter first uses the *improved feasible region* for planning center of mass (CoM) and body orientation trajectories. This approach ensures that the base motion of the robot is within its capability to be executed without risking stability or violating actuation limits. Similarly, achieving desired wrenches in a feasible manner can significantly depend on footholds chosen at touchdown. Therefore, we further develop a foothold selection criterion that is aware of both the desired wrench and the feasibility of the motion. We showcase the power of adding the feasibility criteria to simple heuristics, therefore achieving simple planning strategies that are more robust and applicable on real robots, as well as the possibility of integrating them with more complex optimization-based planning strategies to achieve more efficiently optimal behavior.

4.2 Heuristic Base Motion Planning

4.2.1 CoM planning strategy

In this subsection, we improve the [Center of Mass \(CoM\)](#) planning strategy developed for *crawl* gaits described in our previous work [26], by exploiting the proposed definition of the improved feasible region (see Fig. 4.1 and 4.2). We will denote the method used in [26] as the *stable planning (SP)* strategy and our enhanced one as the *feasible planning (FP)* strategy. To simplify the planning

4.2. Heuristic Base Motion Planning

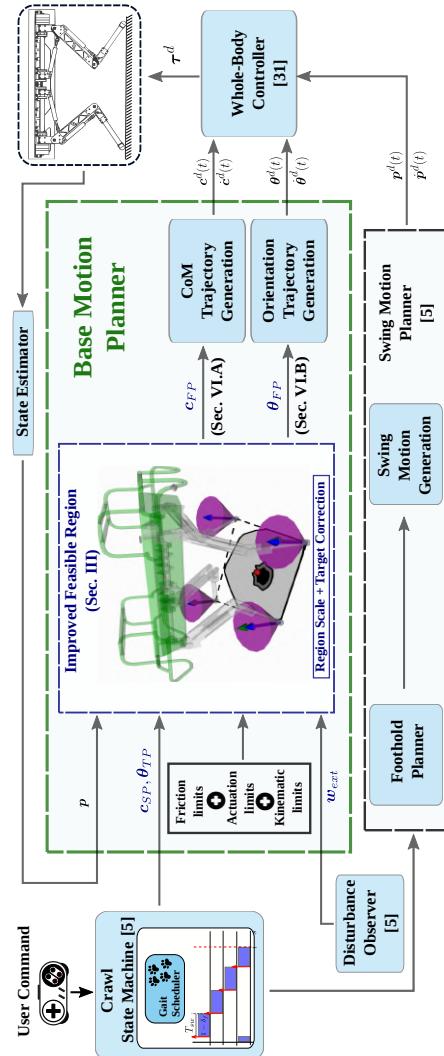


Figure 4.1: Block diagram of our locomotion framework. The improved feasible region helps the planner devise feasible robot trajectories.

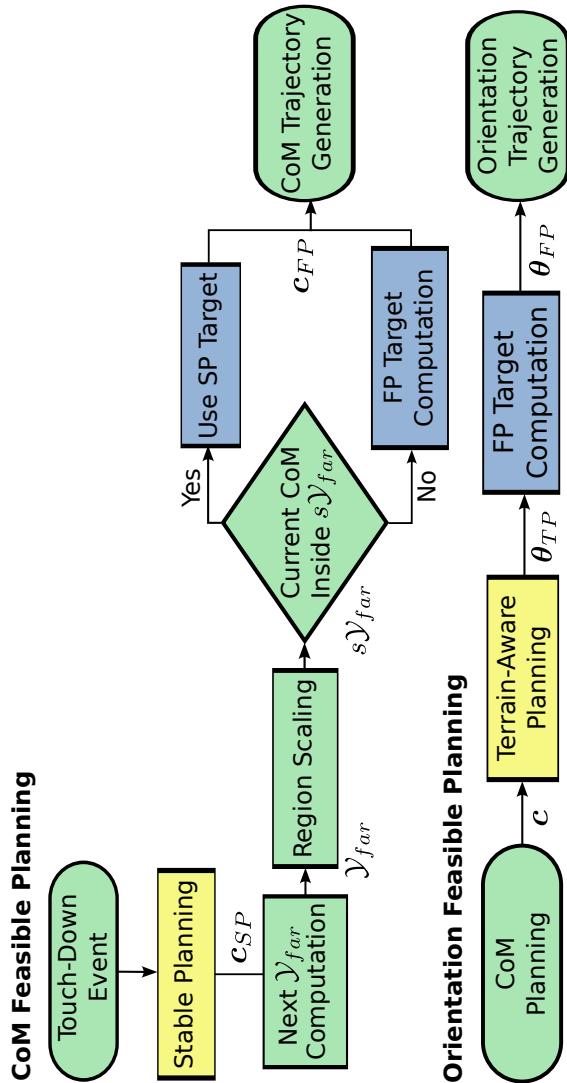


Figure 4.2: Flowcharts illustrating the trajectory planning algorithms. Top: CoM FP algorithm. Bottom: Orientation FP algorithm.

4.2. Heuristic Base Motion Planning

framework, we assume a quasi-static motion (inertial wrenches \mathbf{w}_i are negligible): during a crawl cycle, the robot motion is divided into *swing phases*, where only one foot is allowed to swing while the robot trunk is kept stationary, and *move-body phases*, where all feet are in stance and the trunk is moved to a target location and orientation. A pre-defined foot sequence is used¹. Note that it is possible to extend the strategy to a more dynamic gait by designing **CoM** trajectories that are consistent with the extension shown in Section 3.2.3. Therefore, one would need to ensure the trajectory is consistent with the dynamic feasible region and that the accelerations are consistent with the desired ones for which the region was computed. Through the use of the improved feasible region, we will improve the SP behavior adding guarantees on the physical feasibility. The feasible region is utilized to plan a **CoM** trajectory for the move-body phase such that in the following swing phase, i.e., when only three feet are in stance, the **CoM** target remains feasible.

The three-contact phase is the most critical in terms of stability (the friction region is typically the smallest) and actuation capability, as only three legs support the whole robot's weight and the other possible external wrenches. After each touch-down (i.e., at the start of a move-body phase), the next feasible region \mathcal{Y}_{far} is computed based on the future three stance legs (known from the foot sequence). An FP target **CoM** position, using the criterion explained below, is then chosen. In such a manner, the feasibility is ensured when the next swing foot is lifted, and the robot is only supported by three feet. A quintic polynomial trajectory for the **CoM** is generated, linking the current **CoM** position with the chosen target, and is tracked during the move-body phase in progress. Figure 4.2 provides a flowchart of the planning algorithms. As mentioned in Section 3.1.2, the Jacobians used to evaluate the force polytopes of the contact legs make the feasible region configuration-dependent. Therefore, the region should be recomputed for each CoM location along the planned trajectory. To simplify the planning problem, we instead evaluate the region using the leg Jacobians computed at a configuration corresponding to the SP CoM target. The use of a single Jacobian can be further justified by the analysis done in [45], which showed that the variation of the Jacobian around a specific configuration has negligible effect on the contact forces. To introduce a level of robustness against uncertainties, the planning of the target is done considering a *scaled* version of the feasible region $s\mathcal{Y}_{far}$ with a tunable scaling coefficient $s \in (0, 1)$.

¹The default locomotion sequence for the crawl is: Right-Hind (RH), Right-Front (RF), Left-Hind (LH), Left-Front (LF)

4.2. Heuristic Base Motion Planning

The procedure is devised as follows: if the current **CoM** projection \mathbf{c}_{xy} (onto the region plane)² is inside $s\mathcal{Y}_{far}$, feasibility is already guaranteed and the target **CoM** position is chosen to be the current one to minimize unnecessary motion. Otherwise, we proceed to select the point on the boundary of $s\mathcal{Y}_{far}$, that is closest to the target computed using the SP. This allows the motion to: (1) be as close as possible to the SP target, thus benefiting from its proven reliable, practical effectiveness [26]; (2) formally fulfill the feasibility requirements; and (3) achieve a desired level of robustness (tunable by the shrinkage factor s). Remaining close to the SP target, also allows to (4) maintain the *local* validity of the *feasible region* (the Jacobian was evaluated for the SP target position).

Furthermore, the scaling of a convex polygon can be performed through an affine transformation with respect to the Chebyshev center or the centroid (see [44]). For non-convex polygons, this problem is harder. One solution is to use inward polygon offsetting. However, this is not yet fast enough for online planning and we have noticed that, for this purpose, scaling the feasible region with respect to its centroid provides satisfactory results. For a more detailed discussion, refer to section 4.4.

4.2.2 Optimization of trunk orientation to maximize joint range

Upon planning a **CoM** trajectory, our previous planning approach [26] also computes a target trunk orientation (roll and pitch) to be attained during the move-body phase. This target is chosen to be aligned with the inclination of the *terrain plane* which is estimated in [26] via fitting an averaging plane through the stance feet. We will denote the method of [26] as the *terrain-based planning (TP)* strategy and our improved one as the *feasible planning (FP)* strategy.

This TP strategy aims at bringing the legs as close as possible to the middle of their workspace in order to avoid violating the kinematic limits. For instance, if the robot walks up a ramp, keeping a horizontal posture will lead the back legs to extend and the front ones to retract, risking a kinematic limit violation in some of the joints. However, for rough terrains, where the feet are located on distant non-coplanar surfaces, this might not be sufficient. In such cases, it can happen that some legs become more extended/retracted than others, as will be illustrated in Section 4.6.2.

²In the accompanying video, the projected regions are illustrated at the foot level just for visualization purposes. However, the computation of the regions has been performed at the level of the **CoM**.

4.3. Heuristic Foothold Planning

Examining the effect of the trunk orientation on the region in Section 3.3, we can exploit the region to guide the choice of the orientation that best encloses the whole CoM trajectory chosen in Section 4.2.1. In particular, we choose to optimize the orientation to maximize the *minimum* distance between the trajectory and the boundary of the region during the move-body phase. This not only attempts to ensure the inclusion of the *whole* trajectory in the region but also tries to keep it away from the boundary as much as possible, thus increasing robustness. In case multiple orientations result in similar distances, we opt for the one that maximizes the area of the region. Optimizing for the orientation allows the robot to be less conservative in its movements and to achieve more complex configurations on rough terrains. In other words, we make sure that *each leg has a minimum distance from the limits of its workspace*, as opposed to the previous TP approach that handles all the legs collectively to better match the terrain inclination estimate.

To reduce the size of the problem, it is necessary to initialize the search space around some solution. As mentioned above, the TP orientation provides an elementary yet satisfactory behavior in many cases. Accordingly, we choose to sample the orientation space around the TP orientation. Furthermore, we only optimize for the pitch and roll angles since the yaw angle is computed to keep the base aligned with the locomotion direction.

Note that this orientation planning strategy aims to improve upon the CoM planning strategy described in Section 4.2.1 and does not necessarily guarantee feasibility on its own; a CoM target that is highly unfeasible for the default orientation is very likely to remain unfeasible for any other possible *better* orientation. For this reason, we choose to perform the CoM planning strategy in Section 4.2.1 (computed at the default orientation) *before* optimizing for the orientation.

4.3 Heuristic Foothold Planning

Given the simplicity of the feasible region, we can use an intuitive heuristic to plan for footholds. Given a desired wrench, the region can be used to choose footholds that maximize the feasibility with respect to the desired wrench and the corresponding CoM trajectory.

In fact, rather than computing instantaneous wrenches in Eq. (3.23) based on

4.3. Heuristic Foothold Planning

the robot state, we can instead provide the desired wrenches to the criterion

$$\mathbf{w}_{i_{des}} + \mathbf{w}_g + \mathbf{w}_{ext_{des}} = \mathbf{w}_c \quad (4.1)$$

where $\mathbf{w}_{i_{des}}$ is the desired inertial wrench to generate *locomotion* and $\mathbf{w}_{ext_{des}}$ is the desired wrench to reject disturbances or to generate *interaction forces*.

We first show a use case of *compensating* the footholds planned for locomotion to maximize the feasibility of the wrench needed when an external disturbance is applied to the robot. We consider a case when the robot is trotting in place. Similar to [5], we first compute nominal footholds based on the *Raibert's heuristic* [52] that are needed for the trotting gait:

$$\mathbf{p}_{n_i} = \mathbf{h} + \dot{\mathbf{c}} \frac{t_{st}}{2} + k_r (\dot{\mathbf{c}} - \dot{\mathbf{c}}_d) \quad (4.2)$$

where \mathbf{p}_{n_i} is the nominal foothold for the i^{th} foot (corresponding with the neutral point), \mathbf{h} is the projection of the hip position of the i^{th} leg on the ground, t_{st} is the stance time, $\dot{\mathbf{c}}_d$ is the desired CoM velocity, and k_r is a gain. A desired centroidal wrench is generated by the [Whole-Body Control \(WBC\)](#) to track a mass-damper system behavior

$$\mathbf{w}_{wbc} = m_b \ddot{\mathbf{c}}_d + \mathbf{k}_d (\dot{\mathbf{c}}_d - \dot{\mathbf{c}}) \quad (4.3)$$

where k_d is a velocity gain.

When a disturbance \mathbf{w}_{ext} is applied to the robot, the centroidal wrench to be generated by the [WBC](#) can be appended to further reject the disturbance:

$$\mathbf{w}_{wbc} = m_b \ddot{\mathbf{c}}_d + \mathbf{k}_d (\dot{\mathbf{c}}_d - \dot{\mathbf{c}}) - \mathbf{w}_{ext} \quad (4.4)$$

In some cases, if the disturbance is not large and the robot is in stance, the wrench applied by the [CoM](#) can be rejected completely. However, as the robot trots, the robot becomes underactuated, and the disturbance might not be rejected completely. In such cases, the robot might need to adjust the touchdown locations of the feet to maximize the feasibility of the wrench needed to reject the disturbance. Therefore, in addition to compensating for the external disturbances in Eq. (4.4), we incorporate the wrench needed to compensate for the external disturbances through the computation of the feasible region in Eq. (4.1). To isolate the effect of this compensation from the locomotion, we compute the inertial wrench in Eq. (4.1) based on the actual robot state and not the desired locomotion wrench. The resulting feasible region is one that describes

4.3. Heuristic Foothold Planning

the feasibility of the robot given its current dynamic state and given an external disturbance. Therefore, the closer the **CoM** is to the feasible region, the more feasible the wrench is. Given that during trotting the feasible region becomes a narrow stripe, a shift in the feasible region due to an external disturbance (as shown in Section 3.2.2) can result in a significant reduction in feasibility. We, therefore, propose to shift the desired foothold location by a distance similar to the difference between the CoM position and the centroid of the feasible region. Therefore, the corrected desired touchdown locations can be computed as

$$\mathbf{p}_{des}^{xy} = \mathbf{p}_n^{xy} + (\mathbf{c}^{xy} - \mathbf{c}_{FR}(\mathbf{w}_{ext_{des}})) \quad (4.5)$$

where \mathbf{p}_n^{xy} is the nominal touchdown location using Raibert's heuristic and \mathbf{c}_{FR} is the centroid of the feasible region (which is a function of $\mathbf{w}_{ext_{des}}$).

Fig. 4.3 shows the results of the proposed heuristic for a disturbance rejection task. To showcase the effectiveness of the proposed heuristic, we perform the compensation using multiple stages. First, a disturbance of 100 N along the x -axis is applied to the HyQ robot as it trots in place. The first section of the plot shows the robot's response with no wrench or foothold compensation. The disturbance results in a forward velocity (deviating from the desired zero velocity). To stop the motion, the **WBC** estimates a wrench of about -100 N (shown by the red curve) is required to counteract the forward velocity. However, given the nominal footholds, the controller is poorly able to track such a wrench (as shown by oscillations in the green curve). In the second stage, a compensating wrench of -100 N is applied to the robot using the **WBC** without changing the footholds. The rejecting wrench causes the base velocity to successfully decrease closer to zero. Nevertheless, this alone is not sufficient to stop the robot that remains with a residual velocity. The poor tracking of the wrench remains, given that the realized nominal footholds do not maximize the feasibility of the wrench needed to reject the disturbance. In the third stage, the proposed heuristic is applied to adjust the touchdown locations according to Eq. (4.5). As a result, the wrench tracking is significantly improved. and the robot is able to reject disturbance and stop motion.

Besides the disturbance task, we also propose to use the heuristic to choose the footholds for the motion task. The feasible region is, therefore, to be computed using the wrench model in Eq. (4.1), i.e., based on the desired wrench needed for locomotion as well as the desired wrench needed for disturbance rejection. The desired locomotion wrench needs to be consistent with the one desired by a centroidal planner. In our case, this is similar to the wrench enforced by the

4.3. Heuristic Foothold Planning

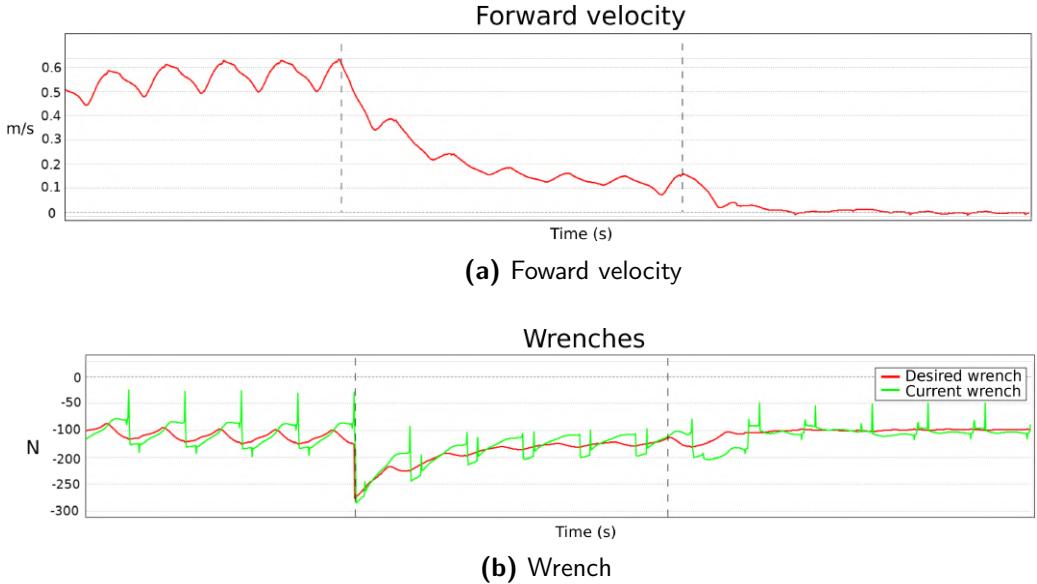


Figure 4.3: Forward velocity and wrench tracking results for disturbance rejection task considering the HyQ robot. The dashed lines separate the different stages of the experiment: no compensation, wrench compensation, and wrench and foothold compensation.

WBC during the full stance period

$$\mathbf{w}_{i_{des}} = m\ddot{\mathbf{c}}_d + \mathbf{k}_d(\dot{\mathbf{c}}_d - \dot{\mathbf{c}}) \quad (4.6)$$

The desired wrench in Eq. (4.6), however, is computed based on the instantaneous CoM position and velocity. In other words, the feasible region is computed based on the desired wrench for the current robot state and does not take into account the wrench needed for the full stance period. In fact, a foothold achieving the desired wrench at the beginning of the stance period might be different to the one achieving the desired wrench at the end of the stance period. During stance, the WBC can apply the continuously changing wrench through the full period. However, the choice of the foothold is a discrete decision that needs to be made at the beginning of the stance period. One proposed solution is to incorporate the average of the desired wrench during the full next stance period in the feasible region. As such, the selected foothold would attempt to maximize the achievability of the wrench during this period. To compute the average wrench, we first analyze the evolution of the desired model behavior of

4.4. Heuristics Assumptions Summary

Eq. (4.3):

$$\dot{\mathbf{c}}(t) = \dot{\mathbf{c}}(0) \cdot e^{-\gamma t} \quad (4.7)$$

where $\mathbf{c}(0)$ is the predicted initial [CoM](#) position at touchdown, m_b is the mass of the base, $\gamma = \mathbf{k}_d/m_b$, and t is the time. The average velocity, therefore, is

$$\dot{\mathbf{c}}_{avg} = \frac{\dot{\mathbf{c}}(0)}{\gamma T_{st}} [1 - e^{-\gamma T_s}] \quad (4.8)$$

From (4.6) and (4.8), the average desired wrench can be computed as

$$\mathbf{w}_{i_{avg}} = m_b \ddot{\mathbf{c}}_d + \mathbf{k}_d (\dot{\mathbf{c}}_d - \dot{\mathbf{c}}_{avg}) \quad (4.9)$$

Because the model already embeds the locomotion wrench, the inclusion of Raibert's heuristic in Eq. (4.5) is not necessary anymore. The corrected touchdown locations can instead be computed directly as

$$\mathbf{p}_{des}^{xy} = \mathbf{c}^{xy} - \mathbf{c}_{FR}(\mathbf{w}_{i_{avg}}, \mathbf{w}_{ext_{des}}) \quad (4.10)$$

Figure 4.4 shows the results of the proposed heuristic for a motion task. The robot is commanded to trot forward with a desired velocity of 0.2 m/s . The first stage of the plot shows the robot's response using the Raibert heuristic for the footholds. Similar to the case of disturbance rejection, the robot is unable to track the desired wrenches in a satisfactory manner. As a result, the robot's average forward velocity falls below the desired value. In the second stage, the proposed heuristic is applied to adjust the touchdown locations according to Eq. (4.10). However, the robot's velocity overshoots the desired value, and the robot is unable to track the desired wrenches. Note that the desired wrench in Eq. (4.6) attempts to track a simplified second-order model with no knowledge of the contact forces and contact constraints. Therefore, an accurate prediction of the achievable wrench is limited. This explains the poor behavior of the feasible region heuristic as well as Raibert's heuristic for the motion task. In contrast, during the disturbance rejection task, the heuristic achieves superior performance due to the accurate knowledge of the needed wrench.

In Section 4.5, we use a [Model Predictive Control \(MPC\)](#) that employs the [Single Rigid Body Dynamics \(SRBD\)](#) model to predict and provide the feasible region with more accurate desired wrenches for the motion task.

4.4 Heuristics Assumptions Summary

As mentioned in the last sections, several assumptions were made for the purpose of planning with the *improved feasible region*. Table 4.1 provides a summary of

4.4. Heuristics Assumptions Summary

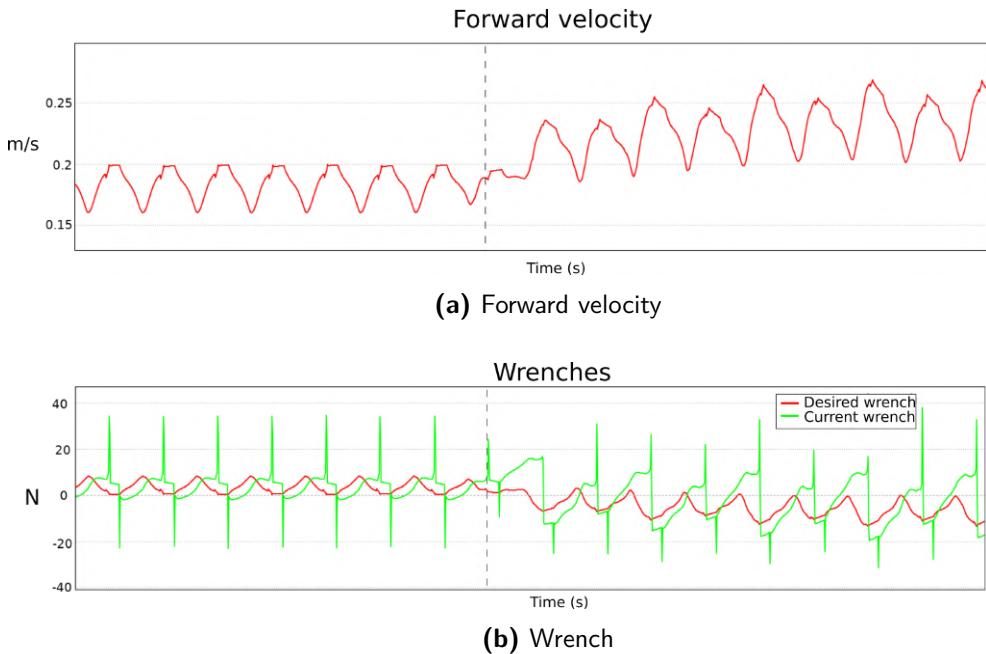


Figure 4.4: Forward velocity and wrench tracking results for motion task.

these assumptions, their types and purposes, and where they were discussed in the dissertation.

	Assumption	Type	Purpose	Sections
1	Scaling of region based on centroid	A	CE	4.2.1
2	Single leg Jacobians for planning	M	PS	4.2.1

Table 4.1: List of assumptions taken during planning, along with their types and purposes. The abbreviation of the types are as follows: M - Modelling and A - Algorithmic. The abbreviation of the purposes are as follows: CE - Computation Efficiency and PS - Planning Simplicity

- Assumption 1: Scaling a non-convex polygon through an *affine* transformation with respect to a reference point (e.g., the Chebyshev center or the centroid) could result in a scaled region with parts outside the original one. On the other hand, inward *polygon offsetting* algorithms guarantee that the scaled polygon always remains inside the original one. One downside of this

4.4. Heuristics Assumptions Summary

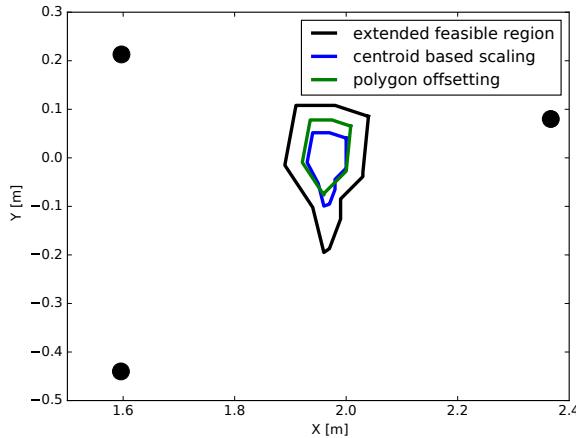


Figure 4.5: Comparison between scaling the extended feasible region based on the centroid and using polygon offsetting.

algorithm is that the scaled polygon can suffer from topological changes (e.g., some edges might contract until they vanish [19]). Furthermore, the scaling during the offsetting procedure is defined by a distance. The centroid-based scaling, on the other hand, characterizes the scaling in terms of a percentage. This allows the algorithm to be directly scalable and consistent with robots of different dimensions. Although offsetting non-convex polygons is still a hard problem in itself, [65] proposed a solution for non-convex polygons.

In Fig. 4.5, we compare the result achieved using the polygon offsetting algorithm to the output of the affine scaling with respect to the centroid. An improved feasible region (shown in black) is computed for a three-contacts phase with a height of 0.37 m for the HyQ robot. We set a scaling factor $s = 0.5$ for the affine scaling (black) and an offset of $r = 0.03\text{ m}$ for the polygon offsetting (green). A small difference in the area of the scaled polygons and a slight shift can be observed, with the offset polygon changing its topology due to the already small size of the original polygon. We chose to utilize the affine scaling for the direct scalability and its efficiency (an average computation time of 0.005 ms as compared to 5 ms for the polygon offsetting). To guarantee that the scaled region would strictly be a member of the original one, an additional efficient step of using the intersection of the original and the scaled region can be performed.

4.5. Gradient-Based Foothold Optimization

- Assumption 2: Given the dependence of the feasible region on the configuration of the legs and therefore, the local validity of the region, we must evaluate the feasible region based on a specified robot configuration (corresponding to a specified CoM location). Therefore, the region should be recomputed for each CoM location along the planned trajectory. To simplify the planning problem, we instead evaluate the region using the leg Jacobians computed at a configuration corresponding to the heuristic CoM target. Indeed, the planning during the three-contacts phase is done with respect and in the vicinity of heuristic CoM target at that phase. The use of a single Jacobian can be further justified by the analysis done in [45], which showed that the variation of the Jacobian, around a specific configuration, has a negligible effect on the contact forces.

4.5 Gradient-Based Foothold Optimization

As shown in Section 4.3, the heuristic approach to planning footholds is limited in its ability to adapt the footholds to improve the feasibility of the desired wrench. The approach is based on a heuristic that shifts the footholds based on the difference between the CoM position and the centroid of the feasible region. This heuristic is not guaranteed to provide the optimal footholds for the desired wrench. Furthermore, an accurate prediction of the achievable wrench is needed.

To better adapt the footholds according to a desired wrench, we propose the [Feasibility Margin-Based Foothold Adaptation \(FFA\)](#): a gradient-based optimization framework that utilizes the *feasibility margin* introduced in Section 3.5. The approach attempts to find the footholds that maximize the margin throughout a predicted wrench trajectory. An accurate prediction of the achievable wrench is obtained from a [Nonlinear Model Predictive Control \(NMPC\)](#) that utilizes a [SRBD](#) model. The gradient of the margin with respect to the footholds is computed using backpropagation of the [Multi-Layer Perceptron \(MLP\)](#). The optimization problem is formulated as a nonlinear optimization problem that utilizes the [MLP](#) to compute optimized footholds. We first formulate the problem as a maximin optimization problem where the cost function is defined such that the footholds maximize the minimum margin along the whole trajectory:

$$\max_{\mathbf{p}_i^{xy}(t_k)} \min \left[\text{FM}(\mathbf{p}_i^{xy}(t_k)) \right] \quad \text{for } k = 1, 2, \dots, N_k \quad (4.11)$$

This strategy increases the feasibility of the motion in a robust manner as it focuses on finding a solution that improves the worst-case scenario along the

4.5. Gradient-Based Foothold Optimization

motion. However, solving this maximin problem directly is complex due to the nonlinear and nonconvex nature of the problem. To simplify the problem and achieve a more rapid convergence to a solution, we introduce an auxiliary variable Z and instead formulate the minimax problem as

$$\begin{aligned} & \max_{\mathbf{p}_i^{xy}(t_k), Z} && Z \\ & \text{s.t.} && \\ & && Z \leq \text{FM}(\mathbf{p}_i^{xy}(t_k)) \quad \text{for } k = 1, 2, \dots, N_k \end{aligned} \tag{4.12}$$

Notice that no explicit representation of the full or simplified models of the robot is needed to maintain stability and physical consistency, as the *feasibility margin* embeds the dynamics implicitly by definition.

The problem attempts to find the footholds that optimize the *feasibility margin* along the prediction horizon for our desired wrench. Given our generic approach, we consider wrenches needed for motion tracking and disturbance rejection collectively, as shown in Section 4.3. The wrench needs to be evaluated for the whole prediction horizon. Therefore, we utilize an **NMPC** to provide such wrench, considering both a user-desired base velocity and interaction forces (whether that is from a desired pulling/pushing force or rejecting an external force). Unlike in Section (4.3), the **NMPC** uses a **SRBD** model and computes base trajectories, foothold positions, along with corresponding ground reaction force trajectories for a horizon of N_m steps. The only constraint imposed in the problem is the constraint on the ground reaction forces to be within the friction cones. The cost minimized by the **NMPC** involves the following terms: 1) A tracking cost associated with the reference base velocity, 2) a tracking cost associated with the reference pitch and roll angles of the base to match the terrain, 3) a cost introducing a bias to the footholds towards a nominal one (computed through heuristics introduced by Raibert [52]), and a cost penalizing the ground reaction forces. Figure 4.6 shows the framework we use for the foothold adaptation.

The bias to the nominal foothold can be modified using a weight. Notice that the **MPC** only considers the friction constraints and physical consistency as constraints in its formulation. In fact, if a low biasing weight is used for the foothold optimization in the **MPC**, the foothold solution tends to converge towards the center of the robot (i.e., below its CoM) as can be seen in Fig. 4.7. This is because during a crawling gait, such footholds would provide greater stability during phases when only three feet are in stance (diagonal of support polygon

4.5. Gradient-Based Foothold Optimization

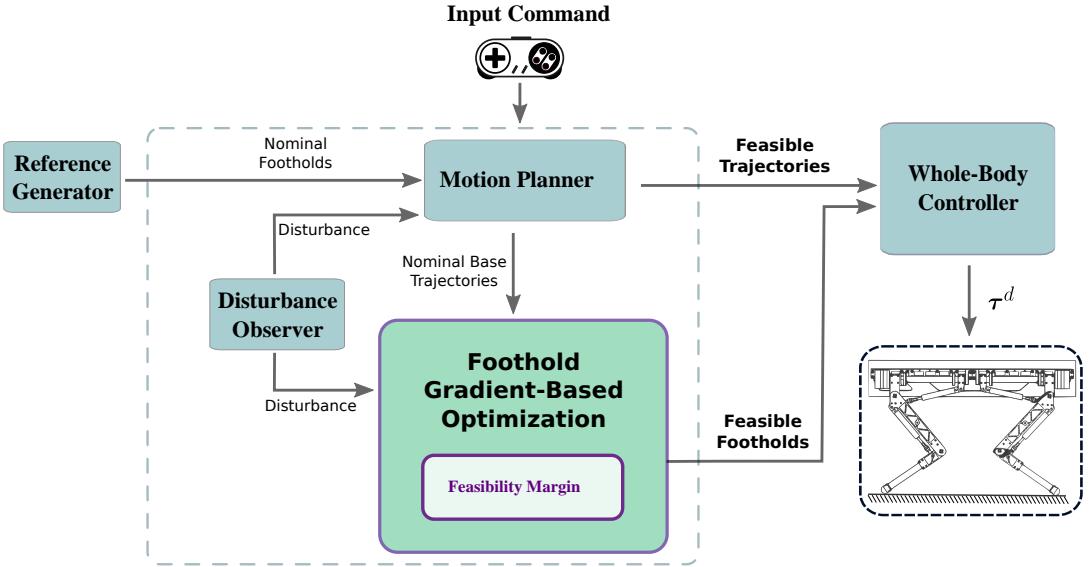


Figure 4.6: Block diagram of the Feasibility Margin-Based Foothold Adaptation framework.

covers the trajectory of the CoM), while no consideration of kinematic or torque limits are taken. However, most robot legs, indeed would operate very close to their kinematic limits at such configuration. Furthermore, it requires more torque from the actuators. The figure shows the HyQReal robot walking with footholds optimized by the MPC when no actuation constraints are taken. As such, a bias towards a nominal configuration (such as that of Raibert) is necessary for MPC formulations that have no awareness of the kinematics/dynamics of the legs [54]. The FFA solves this deficiency by adapting the footholds to the desired trajectory while taking into consideration the additional constraints.

To use the FFA over uneven terrain, we need to provide terrain height information. We use heightmaps computed using the **Vision-Based Foothold Adaptation (VFA)** in [35] to estimate the height of the terrain. The VFA maps are incorporated as additional constraints in the **NMPC** problem to generate desired trajectories that are consistent with the terrain. Given the desired trajectories, we can constrain the foothold choice to patches in the **VFA** maps that are suitable for stepping. The FFA is then used to optimize the footholds based on the

4.5. Gradient-Based Foothold Optimization

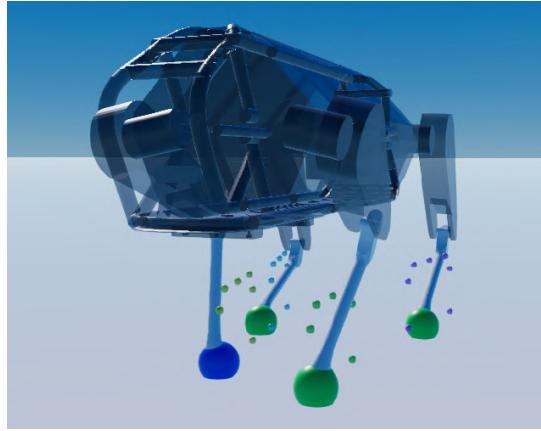


Figure 4.7: HyQreal robot performing a crawl gait using MPC-optimized footholds. Given that no joint constraints are considered in the MPC, it chooses a narrow stance to prioritize the base velocity task.

possible footholds, and the optimization problem can be defined as

$$\begin{aligned}
 & \max_{\mathbf{p}_i^{xy}(t_k), Z} && Z \\
 & \text{s.t.} && \\
 & && Z \leq \text{FM}(\mathbf{p}_i^{xy}(t_k)) \\
 & && \mathbf{p}_i^z(t_k) \leq \mathbf{P}_{VFA}(\mathbf{p}_i^{xy}(t_k)) \quad \text{for } k = 1, 2, \dots, N_k
 \end{aligned} \tag{4.13}$$

where $\mathbf{p}_i^z(t_k)$ is the z-component of the i^{th} foot position at time t_k precomputed by the NMPC. \mathbf{P}_{VFA} defines a patch on the VFA heightmap with a constant height of $\mathbf{p}_i^z(t_k)$.

We use the L4CasADI library [55] for the integration of the PyTorch learned models with CasADI [2] efficiently. To achieve faster optimization times, an approximation of the network is achieved using local second-order Taylor approximations. The approximation is done around the nominal MPC trajectory. The Interior Point nonlinear solver IPOPT [64] is used to solve the nonlinear problem. While no awareness of the MPC footholds is needed in the feasibility margin based optimization, we provide it as a warm start for the solver. The solver requires on average eight iterations to converge to a solution, with an average total solution time of 70 ms.

4.6 Simulation Results

We devised several challenging scenarios for the robot to traverse, designed to best illustrate the utility of the developed feasibility criteria (Chapter 4) and planning methods (Chapter 3).

The generation of the projected regions is done in Python. Table 4.2 shows a summary of the computation times of the different stages of the planning³. The computer is equipped with an i7-8700 3.2 GHz processor and 16GB of RAM. Whenever a multitude of regions needs to be computed (as in the case of the optimization of the trunk orientation) we make use of the parallelism capabilities of our CPU using the *multi-processing* module in Python. The regions are sent via a ROS node to our locomotion planner, that runs in a ROS environment. The WBC runs at 250 Hz.

Stage	Computation Time		
	2-contacts	3-contacts	4-contacts
Feasible region	5 ms	9 ms	14 ms
Reachable region	25 ms	28 ms	29 ms
Intersection \mathcal{Y}_{far}	0.3 ms		
Region scaling	0.005 ms		
Target planning	0.03 ms		
Total feasible planning	55 ms (worst case)		
Whole-body controller	4 ms (worst case)		

Table 4.2: Average computation time for each stage of planning, using the SP target as an initialization for the required stages.

4.6.1 Walk in cluttered environment

We first show the superior performance of using the CoM heurisitc-planning based on the improved feasible region compared to the heuristics of [26]. In this simulation, we assess the influence of an *external wrench* acting on the robot, combined with a *reduced* robot height necessary to walk in confined places. This challenging task consists of the HyQ robot descending a 30° ramp while being

³We expect a decrease in the computation time upon performing the computation in C++, e.g., using Cython [6].

4.6. Simulation Results

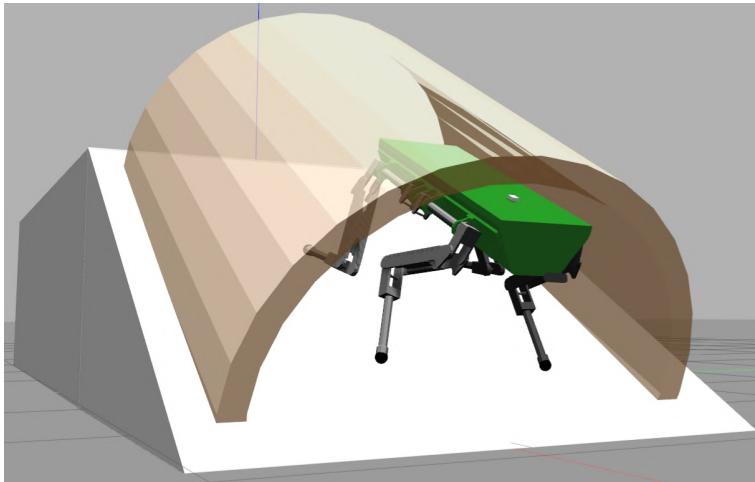


Figure 4.8: Simulation of [Hydraulically actuated Quadruped \(HyQ\)](#) descending a challenging 30° ramp with a 50 cm tunnel (template 1). The height of the HyQ robot is decreased from 53 cm to 40 cm in order to fit inside the tunnel. A force-controllable rope (not shown in the figure) is attached to the back of the robot's trunk to compensate for gravity.

attached to a rope to explore a low tunnel (see Fig. 4.8). This can be a typical scenario that a robot needs to face in oil rig inspection assignments. A rope (not shown in the simulation software) connects the back of the robot to an anchor. The aid of the rope results in regulated locomotion down the steep slope (e.g., the same way a climber is *rappelling* down a wall)⁴. The role of the rope is to allow the contact forces to better satisfy friction constraints (i.e., be more in the middle of the friction cones) when walking on highly inclined terrains [4]. Indeed, on a slope with high inclination, the robot eventually creates a tangential force on the terrain that surpasses the friction force that is needed to prevent slippage. An additional advantage of using a rope is that the robot can keep a more natural configuration without the need to lean back or forth to keep stability, thus keeping the joints in a more kinematically advantageous configuration. As an additional difficulty, the restricted height of the tunnel places a risk of collision with the trunk of [HyQ](#). The robot is therefore forced to *crouch walk* down the tunnel. For this reason, we reduce the robot height from

⁴Experimentally, it is possible to attach the robot to an anchor where a torque-controlled electrically-driven hoist releases the rope while maintaining the required pulling force (i.e., the component of gravity force parallel to the sagittal axis of the trunk).

4.6. Simulation Results

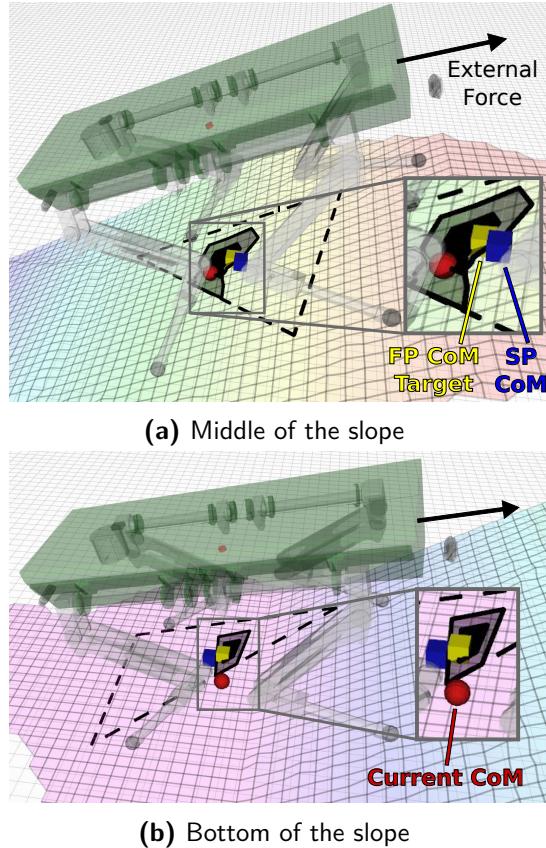


Figure 4.9: Improved feasible regions and **CoM** planning for two instances while descending the challenging tunnel in simulation (tunnel not shown in this figure). HyQ is heading to the left (downwards) while the external force due to the rope (black arrow) is applied in a direction opposite to the motion. The regions shown above are for the future regions upon lift-off of the swing leg (*LF* in the upper plot and *LH* in the lower one): Support regions (dashed), improved feasible regions (grey), and the scaled feasible regions (black). Cubes represent the projection of the **CoM** target based on the FP strategy (yellow) and the SP strategy (blue) on the projection plane. The red sphere represents the projection of the current **CoM**. This is out of the region because the robot is still moving toward the target in the move-body phase (4 legs in stance).

4.6. Simulation Results

the default value of 53cm to 40cm .⁵ This places the robot joints considerably close to their kinematic limits and, in turn, results in a restricted feasible region throughout the motion. In addition, the feasible region will be shifted due to the influence of the external force (equivalent to 440 N applied to the back of the robot) coming from the rope.

The above-mentioned effects on the friction region and on the feasible region can be seen in Fig. 4.9 for two instances in the simulation. The regions are computed on the plane fitted through the stance legs [26]. This is parallel to the plane expressed by the robot trunk orientation where the **CoM** planning is done. In both situations, a shift in the friction and feasible regions, opposite to the external wrench on the robot, could be observed. Furthermore, the low height imposed on the robot results in a big shrinkage of the feasible region. Under these conditions, the **CoM** target (blue) planned with the SP strategy is located outside the region. Conversely, the **CoM** planner based on the improved feasible region (FP), computes a feasible target (yellow) that is on the boundary of the scaled feasible region and closest to the SP target. It is interesting to remark that even though the friction region is shifted, thus giving the robot more freedom to lean forward if desired, the improved feasible region is inhibiting such courageous motions due to joint-torque restrictions and to the limited reachable region.

4.6.2 Optimization of the trunk orientation on rough terrain

To illustrate the effectiveness of the orientation optimization strategy proposed in Section 4.2.2, we test it separately from the **CoM** planning strategy developed in Section 4.2.1. For this reason, the optimization of the orientation will be based on the **CoM** target computed by the SP approach. As mentioned before, even if this does not necessarily guarantee feasibility, it allows us to compare clearly the improvements of the FP strategy over the TP strategy.

To begin with, we consider the trivial case of examining the behavior of the strategy in comparison to the heuristic approach on a ramp.

As expected, the planner chooses the heuristic orientation (or one in the vicinity of it) and rejects more horizontal orientations, further validating the insights behind the heuristic strategy. Figure 4.10 shows the resulting reachable regions for the two orientations for the robot standing on a 15° ramp.

⁵The robot height is defined as the distance between the **CoM** and the terrain plane along its normal \mathbf{n} .

4.6. Simulation Results

In fact, we can see that the reachable region for the horizontal trunk orientation (Fig. 4.10 (b)) is smaller compared to the one where the trunk of the robot is aligned with the ramp (Fig. 4.10 (a)). We can also see that, because of the smaller area of the reachable region, forward trunk motions are significantly impaired in the former case.

While climbing up a ramp, it is typical to move the torso forward [26, 27] in order to have the **CoM** projection position closer to the middle of the support polygon, thus increasing the stability margin. Therefore, aligning the trunk with the terrain inclination has the advantage of a superior feasible region and consequently, an ability to achieve a higher stability margin. The case of the rough terrain shown in Fig. 4.11, is particularly challenging in terms of kinematic limits. One of the legs can be forced to overly extend/retract during the move-body phase even though the other legs are possibly far from their limits. In fact, adopting an orientation based solely on the TP strategy results in infeasible trajectories in multiple locations of the terrain (Fig. 4.11(a) bottom). The TP approach would not capture the difficulty given by the "lateral asymmetry" of this scenario. Indeed, it would result in a trunk with the hips being equally distant from the left and the right feet. In the example shown, a pitch angle of 9.7° (estimated averaging terrain plane), is selected by the TP approach. This results in a hyper-extension of the **RH** leg and a kinematic violation at the **Knee Flexion-Extension (KFE)** joint (Fig. 4.11(a) top). Note that since we model the kinematic limits in our simulator, the **CoM** will not be allowed to go out of the boundary of the region. The same **CoM** trajectories could instead be feasible if the orientation is planned based on the FP strategy, with an optimized pitch angle of -0.3° (see Fig. 4.11(b)). The optimized pitch angle maximizes the distance of the trajectory from the boundary of the region (i.e., the margin), as well as the area of the region, thus resulting in a safer joint configuration.

4.6.3 Feasibility Margin-Based Foothold Adaptation

We show the results of using the **FFA** during a crawling gait with the HyQReal robot. We command the robot to move with a velocity of 2.5 m/s and simultaneously apply a constant external disturbance to the robot as a force $F = (250N, 250N, 200N)$. We compare the realized motion due to the required motion and disturbance rejection wrenches using footholds realized using only the **NMPC** and using the **NMPC** with our feasibility margin-based optimization. Figure 4.12 shows the difference in the realized foot placement using the two strategies. One could observe that the **FFA** has configured the right front (RF)

4.6. Simulation Results

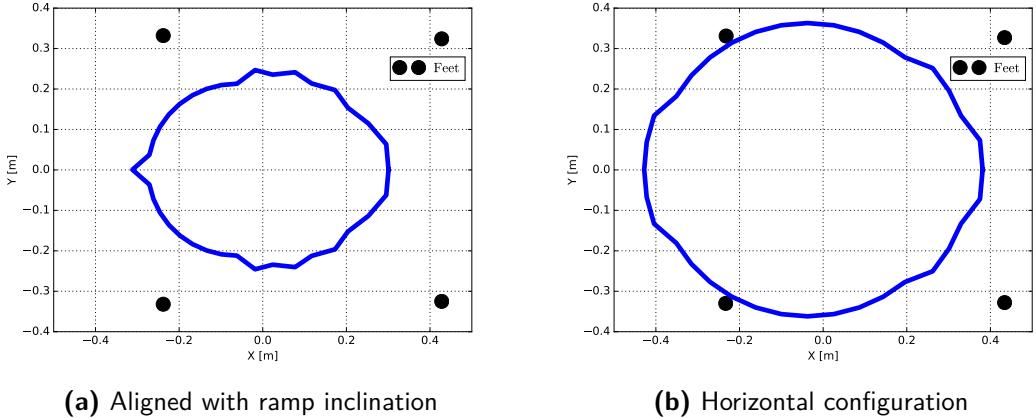


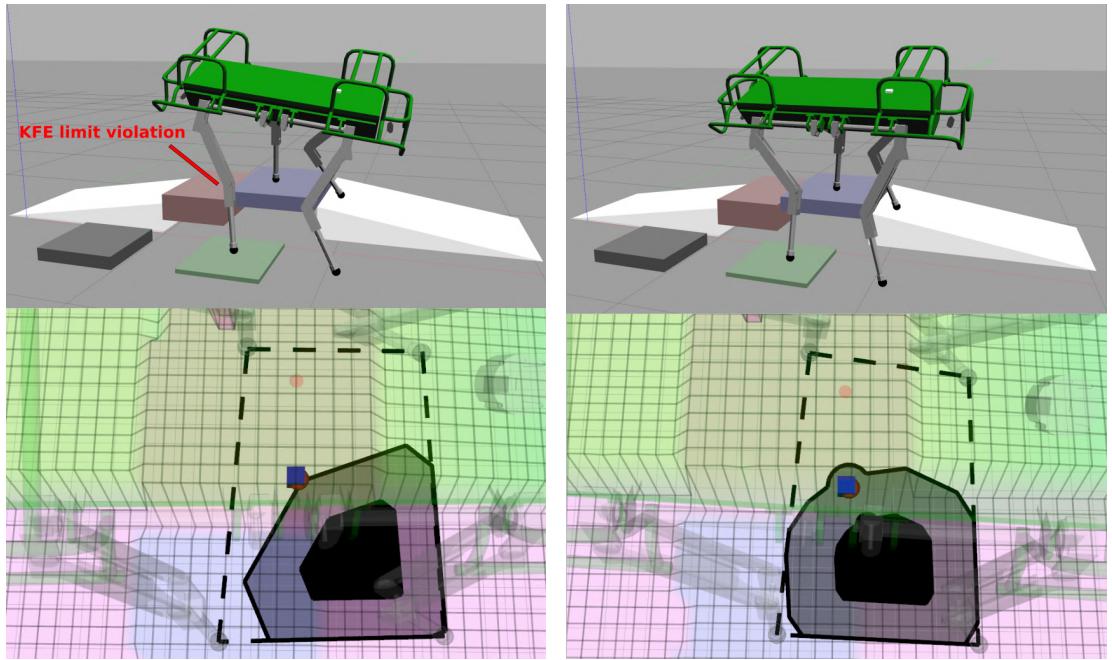
Figure 4.10: HyQ’s reachable region on a 15 degree ramp with (a) the robot’s trunk aligned to the ramp inclination and (b) horizontal trunk.

and left hind (LH) feet to be further in the direction of the force, anticipating the reduction in feasibility margin during the swinging phase of the LF foot due to the effect of the force. Shifting the feet in the direction of the force increases the feasibility margin and, therefore, provides increased stability and robustness against changes in the force. The right hind (RH) foot takes a wider configuration for an increased feasibility margin during full stance phases. Figure 4.13 shows the *feasibility margins* along the prediction horizon of a single FFA optimization and compares it to the margins along the horizon computed for the MPC trajectory. The phase before the first touchdown point is shaded in grey, as the foothold optimizer has no effect on the margin during that duration. While the MPC’s lowest margin during the horizon is close to 0 (the CoM is on the boundary of the *improved feasible region*), the FFA chooses footholds that increase the minimum margin, resulting in higher robustness against disturbances and model mismatches.

Finally, we also test the performance of the FFA on uneven terrain by commanding the robot to walk over the terrain shown in Fig. 4.14 using the FFA and using the MPC separately. The blue spheres represent the footholds computed by the NMPC, and the green spheres represent the footholds computed by the FFA.

Figure 4.15 shows the box plots of the feasibility margins at $k = 0$ of the optimization recorded over the whole motion. The FFA successfully achieves higher minimum margins compared to the MPC footholds, providing increased robust-

4.6. Simulation Results



(a) Using Heuristic-Based Planning

(b) Using Feasible Planning

Figure 4.11: Simulation of HyQ forced near its kinematic limits while traversing a difficult non-coplanar terrain (Template 2). The robot configurations shown are at the end of a move-body phase. Realizing orientations based on (a) the TP strategy and (b) based on the FP strategy results in different leg configurations (top). The resulting regions shown in the bottom plots are: friction regions (dashed), feasible regions (grey), and the scaled feasible regions (black). Large difference in the resulting feasible regions can be seen, in turn affecting the feasibility of the CoM trajectory (blue cube and red ball represent the projections of the CoM target and the actual CoM, respectively).

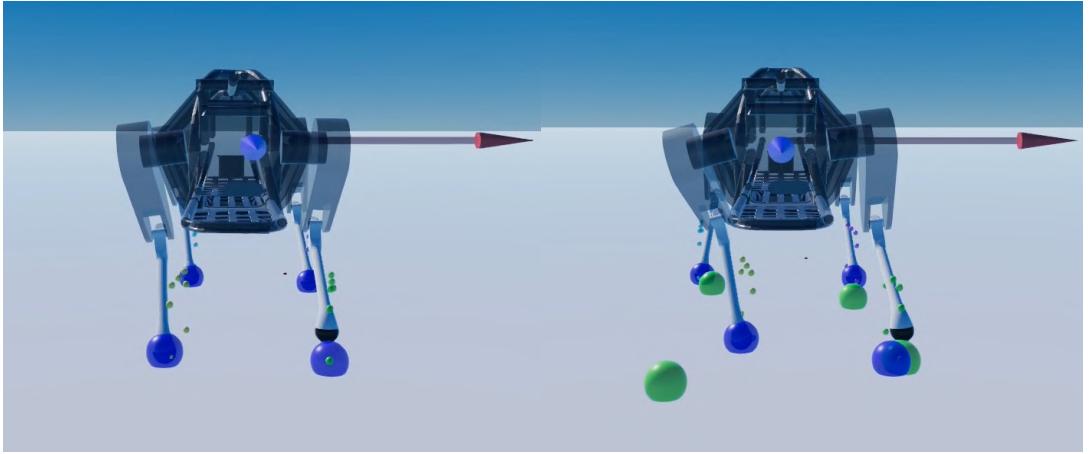


Figure 4.12: Comparison of optimized foothold configuration of the [NMPC](#) (*left*) vs. the [FFA](#) (*right*) for HyQReal performing a crawling gait with a commanded velocity of 2.5 m/s and an applied constant external force of $F = (250N, 250N, 200N)$. The blue spheres represent the footholds computed by the [NMPC](#), and the green spheres represent the footholds computed by the [FFA](#).

ness with respect to the feasibility of our motion task. This is consistent with the goal of the optimization problem (4.13).

4.7 Experimental Results

4.7.1 Walk in cluttered environment

We implement the simulation example shown in Section 4.6 on real hardware using the 21 kg Aliengo robot (for safety reasons) as shown in Fig. 4.16. The robot is commanded to walk down a steeper slope of 45° with a robot height of 20 cm . A pulley, rope and counterweight (a mass of about 10 kg was used) are used to introduce the external force needed to pull the robot backward. The kinematic limits of Aliengo are virtually lowered to simulate that of robots actuated with hydraulic cylinders like HyQ at low robot heights.

The plots of the [KFE](#) joint trajectory during the experiments are reported in Fig. 4.17. A [CoM](#) target based on the [SP](#) strategy would result in multiple violations of the kinematic limits (upper plot), while the one based on the [FP](#)

4.8. Conclusions

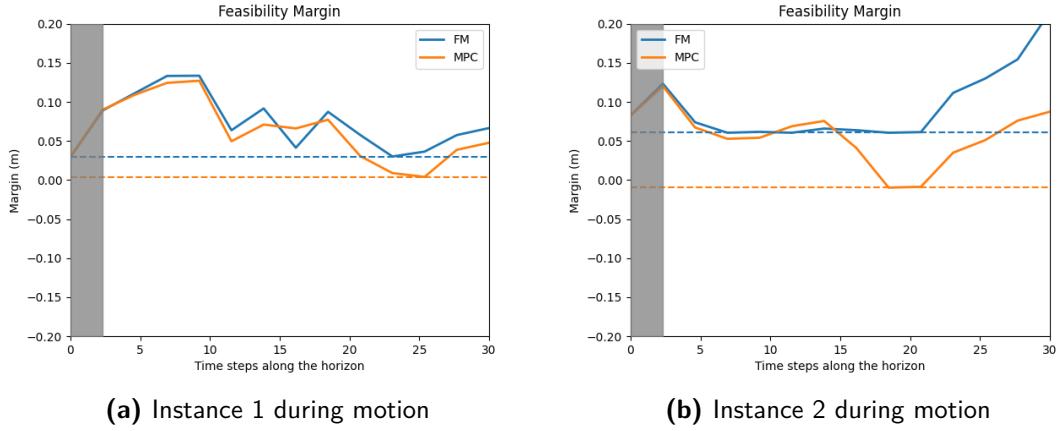


Figure 4.13: Feasibility Margin Comparison: Margins over the prediction horizon for FFA optimization (blue) versus MPC trajectory (orange) for two instances during the motion. The lowest value of the margins along the horizon are shown using horizontal dashed lines. The grey-shaded area marks the pre-touchdown phase. The FFA shows higher minimum margins as opposed to the MPC’s margin nearing zero.

strategy has no violations (lower plot).

Additionally, to show the effect kinematic violations can have on the performance of the robot, we perform experiments with the 90 kg HyQ robot platform walking on flat ground with a reduced height of 43 cm. Fig. 4.18 shows that such kinematic violations result in a deterioration of the tracking of the CoM trajectory computed using the SP strategy as opposed to the FP strategy.

4.8 Conclusions

In this chapter, we presented a wrench-based efficient approach to improve the feasibility of motion planning for legged robots. We utilized the *feasible region* to plan trajectories for the CoM and the orientation of the robot’s base. Furthermore, we employed the *feasibility margin* to optimize the footholds of the robot in a gradient-based optimization framework. Finally, we demonstrated through simulations and experiments the effectiveness of the proposed approach in challenging scenarios where the robot is forced to operate near its kinematic limits and when external disturbances are present.

4.8. Conclusions

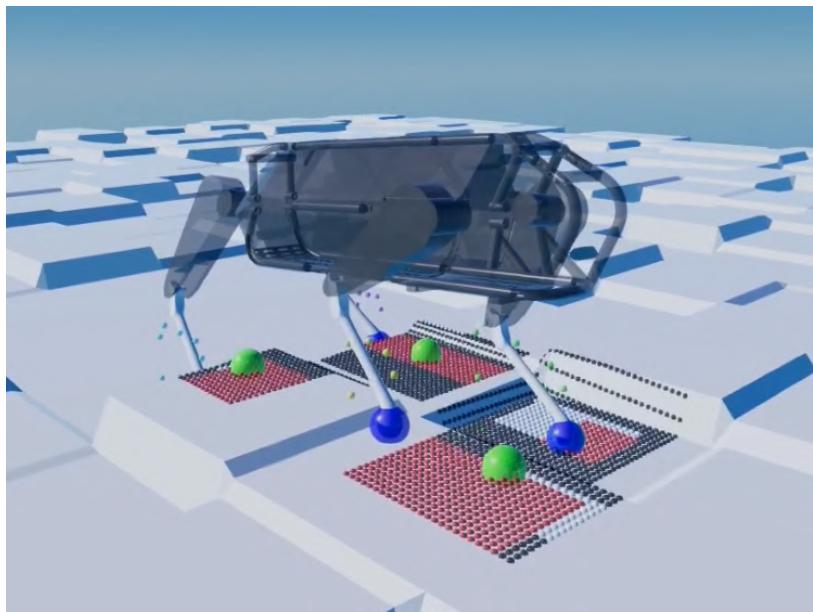


Figure 4.14: HyQReal robot traversing uneven terrain using the [FFA](#). The selection of each foothold is constrained using the patch shown by the red spheres generated by the [VFA](#) heightmap.

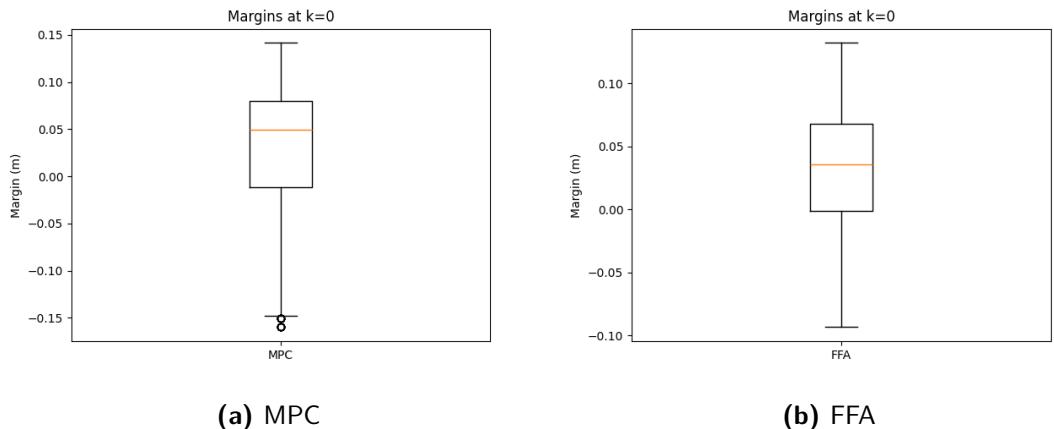


Figure 4.15: Box plots of feasibility margins at $k = 0$ of the optimization - current margins - taken over the whole motion using (a) the [MPC](#) and (b) the [FFA](#) footholds.

4.8. Conclusions

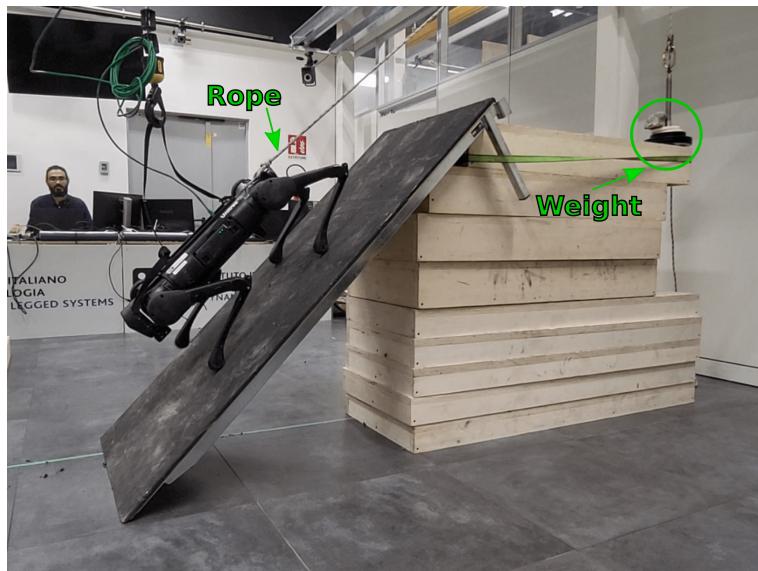


Figure 4.16: Aliengo descending a 45° ramp with a reduced robot height of 20 cm (similar to the scenario shown in simulation). A rope is attached to the back of the robot's trunk to compensate for gravity through a counterweight.

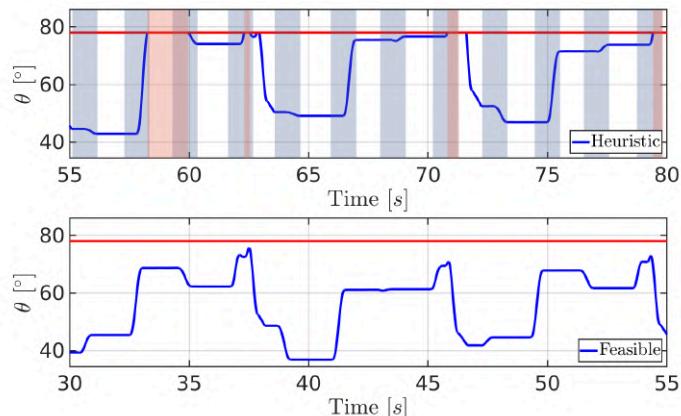


Figure 4.17: Experimental results showing the Right-Hind **Hip Flexion-Extension (HFE)** joint trajectory of Aliengo during the ramp descent. SP strategy (above): The knee starts to hit the virtual kinematic limit (red line) during the *move body* phases (shaded blue). The violations are in shaded red. FP strategy (bottom): No kinematic limit violations are observed.

4.8. Conclusions

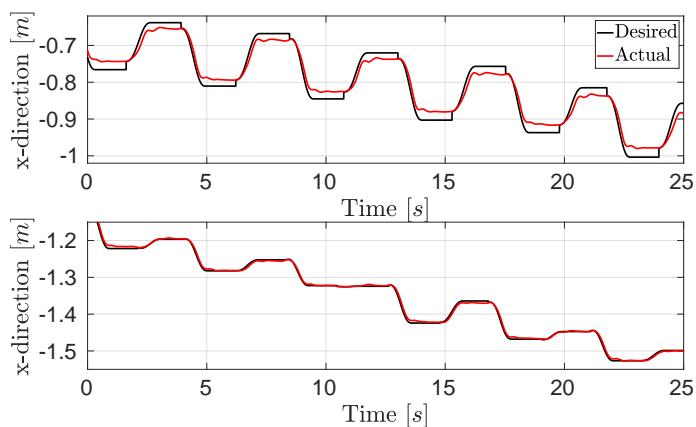


Figure 4.18: Experimental results showing the **CoM** position tracking of **HyQ** in the x direction. Deterioration can be seen with the SP strategy (upper plot) due to the joint kinematic limit violations while good tracking is observed when the planning is based on the improved feasible region with the FP strategy (bottom plot).

Chapter 5

Conclusion

5.1 Summary

This dissertation has developed strategies for legged robots that aim to enhance their locomotion in complex environments and scenarios. This is achieved by first developing measures for legged robots that analyze the feasibility of their motion in terms of locomotion stability and actuator limits. The feasibility measures are then used to devise different methods to plan trajectories for the **Center of Mass (CoM)**, base orientation, and footholds, given a desired wrench task.

In Chapter 3, we introduced the concept of the *improved feasible region*. This 2D Cartesian non-convex region provides a set for the CoM such that the robot remains stable and within actuator limits. In particular, we focused on incorporating the dynamic stability of the reduced model of the robot, the crucial effect of external wrench disturbances on the robot, and actuators' kinematic and torque limits. The region provides an intuitive and efficient way to assess the feasibility of a configuration.

Building on the *improved feasible region*, we also introduced the *feasibility margin*, which introduces a measure of feasibility based on the distance of the CoM from the boundary of the region. This margin plays a critical role in planning by steering trajectories in a manner that increases the achievement of desired wrenches while adding robustness against uncertainties. We further increase in the efficiency of using the margin in real-time and computing its gradient by ap-

5.2. Future Directions

proximating it with a supervised model using a **Multi-Layer Perceptron (MLP)** network.

In Chapter 4, we presented different planning strategies that makes use of the feasibility measures. First, a heuristic **CoM** planning strategy was developed utilizing the *improved feasible region*. This heuristic method corrects **CoM** targets that only consider stability by ensuring that the target is within the feasible region. A simple way of increasing the robustness of the feasibility through scaling the non-convex region is shown.

A heuristic method to plan the orientation of the robot’s base was also presented. This method focuses on exploring the different orientations of the robot’s base to increase the distance of the **CoM** trajectory to the boundary and maximize the area of the feasible region. This ensures that the robot joints are far from their limits, especially on uneven terrains where each leg can have a vastly different configuration.

Furthermore, we presented two foothold selection strategies utilizing the *feasibility margin*. The first strategy attempts to minimize the distance between the **CoM** to the center of the *improved feasible region* by shifting the footholds with a similar distance. The second strategy uses a gradient-based optimization to find the optimal footholds that maximizes the minimum *feasibility margin* for the predicted trajectory, further increasing the robustness of the motion.

We showed the effectiveness of the planning strategies in various simulation scenarios and experimental setups and compared them to more traditional methods that prioritize stability only. The results show that the proposed methods provide more feasible and robust motions, especially in challenging terrains. Using the feasibility measures and the feasibility-based planning strategies, we provide a unified approach that ensures that the robot locomotion is stable (also against external disturbances) and respects the actuator limits.

5.2 Future Directions

The work presented in this dissertation opens up several avenues for future research.

- The *feasibility margin* can be further utilized to optimize the contact sequence of the robot. In fact, despite the planning strategies presented in Chapter 4 aiming to maximize the feasibility of achieving a desired wrench,

5.2. Future Directions

the wrench might be infeasible to achieve due to the contact sequence. A contact planner based on the *feasibility margin* can instead be designed to maximize the achievability of such a wrench.

- The *feasibility margin* can be utilized to obtain a [Reinforcement Learning \(RL\)](#) policy in a more efficient manner. Given the usually high-dimensional state space of the robot, the margin can be used as a metric of the robot's performance in the reward function, therefore adding density to the reward function and making the learning process more efficient.

Bibliography

- [1] Unitree robotics. aliengo. <https://m.unitree.com/products/aliengo>. Accessed: 15 October 2022. (cit. on p. 17.)
- [2] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. (cit. on p. 89.)
- [3] Hervé Audren and Abderrahmane Kheddar. 3-d robust stability polyhedron in multicontact. *IEEE Transactions on Robotics*, PP, 02 2018. (cit. on p. 38.)
- [4] Masahiro Bando, Masaki Murooka, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Walking on a Steep Slope Using a Rope by a Life-Size Humanoid Robot. *IEEE International Conference on Intelligent Robots and Systems*, pages 705–712, 2018. (cit. on p. 91.)
- [5] Victor Barasoul, Jonas Buchli, Claudio Semini, Marco Frigerio, Edson R De Pieri, and Darwin G Caldwell. A reactive controller framework for quadrupedal locomotion on challenging terrain. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2554–2561, 2013. (cit. on pp. 30 and 80.)
- [6] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, march 2011. (cit. on p. 90.)
- [7] Gerardo Bledt, Patrick M. Wensing, and Sangbae Kim. Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4102–4109, 2017. (cit. on p. 33.)

BIBLIOGRAPHY

- [8] Boston Dynamics. Atlas: The world’s most dynamic humanoid. Boston Dynamics, 2024. Accessed: 2024-04-21. (cit. on pp. 7 and 21.)
- [9] Boston Dynamics. Spot: The agile mobile robot. Boston Dynamics, 2024. Accessed: 2024-04-21. (cit. on pp. 7 and 21.)
- [10] C. Boussema, M. J. Powell, G. Bledt, A. J. Ijspeert, P. M. Wensing, and S. Kim. Online gait transitions and disturbance recovery for legged robots via the feasible impulse set. *IEEE Robotics and Automation Letters*, 4(2):1611–1618, 2019. (cit. on p. 34.)
- [11] Timothy Bretl and Sanjay Lall. Testing static equilibrium for legged robots. *IEEE Transactions on Robotics*, 2008. (cit. on pp. 26, 44, and 59.)
- [12] Stephane Caron. Contact wrench cone. Website, 2024. Accessed: 2024-04-21. (cit. on pp. 7 and 27.)
- [13] Stéphane Caron, Quang-cuong Pham, and Yoshihiko Nakamura. Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics. In *Robotics: Science and Systems (RSS)*, Rome, Italy, 2015. (cit. on pp. 27 and 28.)
- [14] Stéphane Caron, Quang-Cuong Pham, and Yoshihiko Nakamura. Zmp support areas for multi-contact mobility under frictional constraints. *IEEE Transactions on Robotics*, 33(1):67–80, February 2017. (cit. on pp. 26 and 46.)
- [15] Stéphane Caron, Quang-Cuong Pham, and Yoshihiko Nakamura. Zmp support areas for multicontact mobility under frictional constraints. *IEEE Transactions on Robotics*, 33(1):67–80, 2017. (cit. on p. 42.)
- [16] J Carpentier, R Budhiraja, and N Mansard. Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots. *Robotics Science and Systems (RSS)*, 2017. (cit. on pp. 34 and 35.)
- [17] Justin Carpentier and Nicolas Mansard. Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018. (cit. on p. 35.)
- [18] Oguzhan Cebe, Carlo Tiseo, Guiyang Xin, Hsiu-chin Lin, Joshua Smith, and Michael Mistry. Online dynamic trajectory optimization and control for a quadruped robot. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12773–12779, 2021. (cit. on p. 33.)

BIBLIOGRAPHY

- [19] Siu-Wing Cheng and Antoine Vigneron. Motorcycle graphs and straight skeletons. *Algorithmica*, 47(2):159–182, Feb 2007. (cit. on p. 85.)
- [20] Joseph P Conti, Charles M Clinton, Guangming Zhang, and Albert J Wavering. Workspace variation of a hexapod machine tool. Technical Report NIST IR 6135, National Institute of Standards and Technology, Gaithersburg, MD, 1998. (cit. on pp. 53 and 68.)
- [21] Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. *IEEE International Conference on Intelligent Robots and Systems*, pages 7440–7447, 2018. (cit. on p. 33.)
- [22] Shamel Fahmi, Carlos Mastalli, Michele Focchi, and Claudio Semini. Passive Whole-body Control for Quadruped Robots: Experimental Validation over Challenging Terrain. *IEEE Robotics and Automation Letters*, 4(3):2553–2560, July 2019. (cit. on p. 53.)
- [23] Peter Fankhauser, Marko Bjelonic, C. Dario Bellicoso, Takahiro Miki, and Marco Hutter. Robust Rough-Terrain Locomotion with a Quadrupedal Robot. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5761–5768, 2018. (cit. on p. 33.)
- [24] P. Fernbach, S. Tonneau, and M. Taïx. Croc: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018. (cit. on p. 34.)
- [25] Michele Focchi, Andrea del Prete, Ioannis Havoutis, Roy Featherstone, Darwin G. Caldwell, and Claudio Semini. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots*, 41(1):259–272, 2017. (cit. on p. 67.)
- [26] Michele Focchi, Romeo Orsolino, Marco Camurri, Victor Barasuol, Carlos Mastalli, Darwin G. Caldwell, , and Claudio Semini. Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality. *Springer Track in Advanced Robotics series, (ECHORD++: Innovation from LAB to MARKET)*:page143–183, 2020. (cit. on pp. 29, 44, 74, 78, 90, 93, and 94.)

BIBLIOGRAPHY

- [27] Christian Gehring, C. Dario Bellicoso, Stelian Coros, Michael Bloesch, Peter Fankhauser, Marco Hutter, and Roland Siegwart. Dynamic Trotting on Slopes for Quadrupedal Robots. *International Conference on Intelligent Robots and Systems*, 2015. (cit. on pp. 44 and 94.)
- [28] C Gosselin. Determination of the Workspace of 6-DOF Parallel Manipulators. *Journal of Mechanical Design*, 112(3):331–336, 1990. (cit. on p. 53.)
- [29] Ruben Grandia, Fabian Jenelten, Shaohui Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model-predictive control. *IEEE Transactions on Robotics*, 39(5):3402–3421, 2023. (cit. on p. 33.)
- [30] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin Riedmiller, and David Silver. Emergence of locomotion behaviours in rich environments, 2017. (cit. on p. 34.)
- [31] H Hirukawa, K Kaneko, S Hattori, F Kanehiro, K Harada, K Fujiwara, S Kajita, and M Morisawa. A Universal Stability Criterion of the Foot Contact of Legged Robots - Adios ZMP. *IEEE ICRA*, 2006. (cit. on p. 27.)
- [32] Imin Kao, Kevin Lynch, and Joel W. Burdick. *Contact Modeling and Manipulation*, pages 647–669. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. (cit. on p. 42.)
- [33] J E Kelley Jr. The Cutting-Plane Method for Solving Convex Programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960. (cit. on p. 26.)
- [34] Yu-Chi Lin, Brahayam Ponton, Ludovic Righetti, and Dmitry Berenson. Efficient humanoid contact planning using learned centroidal dynamics prediction, 2019. (cit. on p. 35.)
- [35] Octavio Antonio Villarreal Magaña, Victor Barasuol, Marco Camurri, Luca Franceschi, Michele Focchi, Massimiliano Pontil, Darwin G. Caldwell, and Claudio Semini. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *IEEE Robotics and Automation Letters*, 4(2):2140–2147, 2019. (cit. on pp. 35 and 88.)

BIBLIOGRAPHY

- [36] Eve Marder and Dirk Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986–R996, Nov 2001. (cit. on p. 30.)
- [37] Sean Mason, Nicholas Rotella, Stefan Schaal, and Ludovic Righetti. An mpc walking framework with external contact forces. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1785–1790, 2018. (cit. on p. 37.)
- [38] Carlos Mastalli, Wolfgang Merkt, Josep Martí-Saumell, Henrique Ferrolho, Joan Solà, Nicolas Mansard, and Sethu Vijayakumar. A feasibility-driven approach to control-limited ddp. *Autonomous Robots*, 46(8):985–1005, Dec 2022. (cit. on p. 33.)
- [39] R Mattikalli, D Baraff, and P Khosla. Finding all stable orientations of assemblies with friction. *IEEE Transactions on Robotics and Automation*, 12(2):290–301, apr 1996. (cit. on p. 34.)
- [40] J P Merlet. Parallel Robots: J.P. Merlet. (cit. on p. 52.)
- [41] H Mosemann, F Rohrdanz, and F M Wahl. Stability analysis of assemblies considering friction. *IEEE Transactions on Robotics and Automation*, 13(6):805–813, 1997. (cit. on p. 34.)
- [42] Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. icub whole-body control through force regulation on rigid non-coplanar contacts. *Frontiers in Robotics and AI*, 2, 2015. (cit. on p. 25.)
- [43] S. Nozawa, M. Kanazawa, Y. Kakiuchi, K. Okada, T. Yoshiike, and M. Inaba. Three-dimensional humanoid motion planning using com feasible region and its application to ladder climbing tasks. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 49–56, 2016. (cit. on pp. 34 and 38.)
- [44] Romeo Orsolino, Michele Focchi, Stéphane Caron, Gennaro Raiola, Victor Barasuol, and Claudio Semini. Feasible region: an actuation-aware extension of the support region. *IEEE Transactions on Robotics (TRO)*, 2020. (cit. on pp. 7, 27, 34, 37, 38, 40, 57, 59, and 78.)

BIBLIOGRAPHY

- [45] Romeo Orsolino, Michele Focchi, Carlos Mastalli, Hongkai Dai, Darwin G. Caldwell, and Claudio Semini. Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots. *IEEE Robotics and Automation Letters*, 3(4):3363–3370, 2018. (cit. on pp. 28, 34, 77, and 86.)
- [46] Romeo Orsolino, Siddhant Gangapurwala, Oliwier Melon, Mathieu Geisert, Ioannis Havoutis, and Maurice Fallon. Rapid stability margin estimation for contact-rich locomotion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8485–8492, 2021. (cit. on pp. 34 and 60.)
- [47] PAL Robotics. Talos: Advanced humanoid robot. PAL Robotics, 2024. Accessed: 2024-04-21. (cit. on pp. 7 and 21.)
- [48] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4):1–14, July 2018. (cit. on p. 34.)
- [49] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: biomechanics and control. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 3, pages 2405–2411 Vol.3, 2004. (cit. on p. 23.)
- [50] P. Puchaud, F. Bailly, and M. Begon. Direct multiple shooting and direct collocation perform similarly in biomechanical predictive simulations. *Computer Methods in Applied Mechanics and Engineering*, 414:116162, 2023. (cit. on p. 31.)
- [51] M. Raibert, M. Chepponis, and H. Brown. Running on four legs as though they were one. *IEEE Journal on Robotics and Automation*, 2(2):70–82, 1986. (cit. on p. 29.)
- [52] Marc H. Raibert and Ernest R. Tello. Legged robots that balance. *IEEE Expert*, 1:89–89, 1986. (cit. on pp. 7, 28, 29, 80, and 87.)
- [53] Niraj Rathod. *Design, Modeling and Control of Compliance in Robotic Systems*. Ph.d. thesis, IMT School for Advanced Studies Lucca, Lucca, Italy, 2019. (cit. on pp. 7 and 20.)

BIBLIOGRAPHY

- [54] Niraj Rathod, Angelo Bratta, Michele Focchi, Mario Zanon, Octavio Villarreal, Claudio Semini, and Alberto Bemporad. Model predictive control with environment adaptation for legged locomotion. *IEEE Access*, 9:145710–145727, 2021. (cit. on p. 88.)
- [55] Tim Salzmann, Jon Arrizabalaga, Joel Andersson, Marco Pavone, and Markus Ryll. Learning for casadi: Data-driven models in numerical optimization. 2023. (cit. on p. 89.)
- [56] 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*, 34(5):630–637, 2004. (cit. on pp. 25 and 26.)
- [57] N. Scianca, M. Cognetti, 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, 2016. (cit. on p. 33.)
- [58] Claudio Semini, Victor Barasuol, Michele Focchi, Chundri Boelens, Mohamed H Emara, Salvatore Casella, Octavio Villarreal, Romeo Orsolino, Geoff Fink, Shamel Fahmi, Gustavo A. Medrano-Cerda, Darwin Gordon Caldwell, Dhinesh Sangiah, Jack Lesniewski, Kyle Fulton, Michel Donadon, and Mike Baker. Brief introduction to the quadruped robot hyqreal. 2020. (cit. on pp. 7 and 21.)
- [59] Filippo M. Smaldone, Nicola Scianca, Valerio Modugno, Leonardo Lanari, and Giuseppe Oriolo. Gait generation using intrinsically stable mpc in the presence of persistent disturbances. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 651–656, 2019. (cit. on p. 37.)
- [60] Steve Tonneau, Pierre Fernbach, Andrea Del Prete, Julien Pettré, and Nicolas Mansard. 2pac: Two-point attractors for center of mass trajectories in multi-contact scenarios. *ACM Trans. Graph.*, 37(5), October 2018. (cit. on p. 34.)
- [61] J. C. Trinkle, J. S. Pang, S. Sudarsky, and G. Lo. On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction. *Zeitschrift Angewandte Mathematik und Mechanik*, 77(4):267–279, January 1997. (cit. on p. 42.)

BIBLIOGRAPHY

- [62] Octavio Villarreal, Victor Barasuol, Patrick M. Wensing, Darwin G. Caldwell, and Claudio Semini. Mpc-based controller with terrain insight for dynamic legged locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2436–2442, 2020. (cit. on pp. 33 and 69.)
- [63] MIOMIR VUKOBRAZOVIĆ and BRANISLAV BOROVAC. Zero-moment point — thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01):157–173, 2004. (cit. on p. 24.)
- [64] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006. (cit. on p. 89.)
- [65] Ron Wein. Exact and approximate construction of offset polygons. *Computer Aided Design*, 39(6):518–527, June 2007. (cit. on p. 85.)
- [66] Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. *Modeling and Control of Legged Robots*, pages 1203–1234. Springer International Publishing, Cham, 2016. (cit. on p. 25.)
- [67] Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018. (cit. on p. 33.)