SAPIENZA
UNIVERSITÀ DI ROMA

# Motion Feasibility Analysis in Legged Robots

Abdelrahman Abdalla

Supervisors: Michele Focchi, Claudio Semini, and Alessandro De Luca

October 16, 2020

# Contents


**Contents** **2**

**1 Introduction** **5**
1.1 Proposed Approach and Contribution . . . . . . . . . . . . . 10
1.2 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**2 Recap on classical feasible region** **12**

**3 Improvements on the Feasible Region** **17**
3.1 Generic Plane of Projection . . . . . . . . . . . . . . . . . . 17
3.2 External wrenches . . . . . . . . . . . . . . . . . . . . . . . 19
3.3 Dynamic Motions . . . . . . . . . . . . . . . . . . . . . . . 22
3.4 Degenerate Feasible Regions . . . . . . . . . . . . . . . . . . 23
3.5 Reachable Region . . . . . . . . . . . . . . . . . . . . . . . 25
3.6 The Improved Feasible Region . . . . . . . . . . . . . . . . . 33

**4 Trajectory Planning** **35**
4.1 CoM planning strategy . . . . . . . . . . . . . . . . . . . . . 35
4.2 Optimization of trunk orientation to maximize joint range . 37

**5 Simulations** **40**
5.1 Walk in cluttered environment . . . . . . . . . . . . . . . . . 41
5.2 Optimization of the Trunk Orientation on very rough terrain 44

**6 Experiments** **47**
6.1 Walk with low height (army crawl) . . . . . . . . . . . . . . 47

**7 Conclusions and future works** **49**

**Bibliography** **51**

**Abstract**

Developing feasible body trajectories for legged systems on arbitrary terrains is a challenging task. Given some contact points, the trajectories for the Center of Mass (CoM) and body orientation, designed to move the robot, must satisfy crucial constraints to maintain *balance*, and to not violate physical *actuation* and *kinematic* limits. In this thesis, a paradigm that allows the design of feasible trajectories in an efficient manner is presented. In continuation to the work done in [1], we extend the notion of the 2D *feasbile region*, where static balance and the satisfaction of actuation limits were guaranteed whenever the projection of the CoM lies inside the proposed admissible region. We further develop a general formulation of the feasible region to guarantee *dynamic* balance alongside the satisfaction of both actuation and kinematic limits for arbitrary terrains in an efficient manner. To incorporate the feasibility of the kinematic limits, an algorithm that computes the *reachable region* of the CoM is introduced. Furthermore, an efficient planning strategy that utilizes the *feasible region* to design feasible CoM and body orientation trajectories is proposed. Finally, the capabilities of the feasible region and the proposed planning strategy are validated using simulations and experiments on the Hydraulically actuated Quadruped (HyQ) robot and comparing them to a previously developed heuristic approach. Various scenarios and terrains that mimic confined and challenging environments are used for the validation.

# Chapter 1

# Introduction

The central ambition in legged robots development, is the ability to traverse unstructured environments. This will allow the use of legged robots in difficult applications such as nuclear plants decommissioning, search and rescue missions, and space crater explorations. Due to the complexity of the terrain and the variety of obstacles encountered during such operations, challenging demands are posed on the robot joints in terms of required actuation efforts and range of motion.

Therefore, planning trajectories that are *feasible* becomes crucial for the success of the locomotion task. A feasible trajectory in this manuscript is defined to be one that fulfills physical constraints in terms of *contact stability*, *actuation* and *kinematic* limits. As the complexity of the terrain increases, the robot is forced to work close to these limits, and hence designing feasible trajectories becomes even more critical.

A powerful tool that is often utilized to devise feasible trajectories is numerical optimization. Due to their computational intensity, optimization-based approaches are usually difficult to implement on a real machine where the on-board computer typically has a limited computing capacity.

However, in recent years, the availability of increased computational power and the formulation of more efficient algorithms, allowed implementations that are compatible with real-time requirements [2, 3]. Nonetheless, despite their remarkable achievements, all the proposed approaches employ simplified models that usually avoid considering joint actuation and kinematic limits or perform conservative approximations.

On the other hand, heuristic approaches with some or no predictive capabilities were used to successfully address rough terrains through blind locomotion [4] or by employing visual feedback to construct (online) the map of the environment [5]. Their advantage is the light computational complexity that enabled to easily implement them online on a real robotic platform. However, these heuristic approaches fail to provide any guarantee on the feasibility of the computed trajectories.

Other optimization approaches employ approximate (*i.e.,* reduced) models to reduce the number of states and achieve on-line re-planning in a Model Predictive Control (MPC) fashion. Namely, a Linear Inverted Pendulum (LIP) model was adopted by Bellicoso et al. [6] for quadrupeds and by Scianca et al. [7] for humanoids, while Di Carlo et al. [8] employed the *linearized Centroidal Dynamics*. Indeed, re-planning is an important feature to avoid accumulation of errors especially in non-flat terrains [5].

The use of reduced models results in smaller optimization problems and shorter computation times at the price of a lower accuracy. This is because reduced models are often written in a reduced set of the state variables and capture the main dynamics of the robot during locomotion, but typically neglect the joint dynamics. Therefore, with these models, constraints at the joint variables (*e.g.,* torque or kinematic limits) cannot be explicitly formulated in the planning problem. With respect to the LIP, the centroidal dynamics is more accurate because: 1) it captures the angular dynamics, 2) it is applicable to uneven terrains (*e.g.,* non-coplanar feet), and 3) it allows to encode friction constraints.

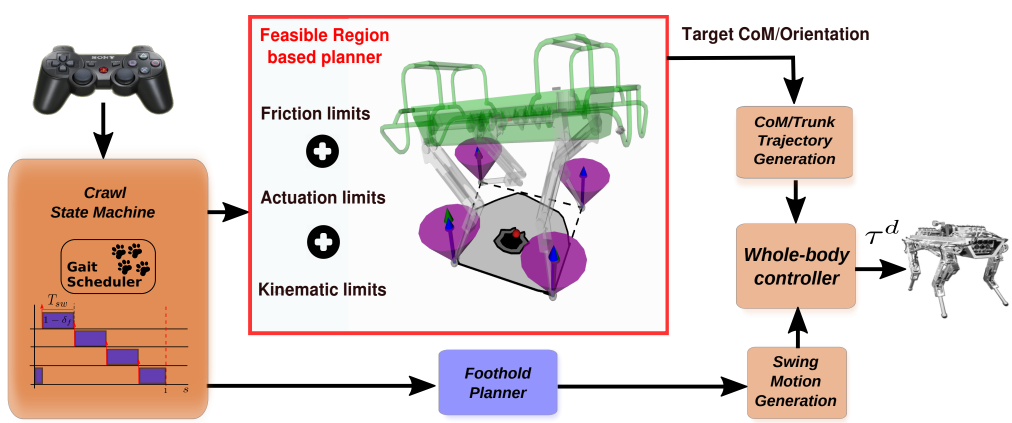



Figure 1.1: Block diagram of our locomotion framework. The feasible region is an aid for the planner to devise feasible robot postures.

Borrowing ideas from computational geometry, researchers succeeded in adding more descriptiveness to the centroidal dynamics model without explicitly optimizing for joint torques nor for contact forces. This can be achieved by mapping friction limits (defined at the contact level) and actuation limits (defined at the joint level) to the 6D space where the centroidal wrench exist.

These mappings result in 6D polytopes that represent the set of admissible wrenches for which the above-mentioned constraints are not violated. Namely, the Contact Wrench Cone (CWC) is defined when only friction constraints are considered [9, 10], while the Feasible Wrench Polytope (FWP) is defined when both the friction and actuation limits are taken into account [11]. Enforcing the polytopes as constraints on the centroidal wrench (or accelerations) in a Trajectory Optimization (TO) results in feasible trajectories for the CoM.[1]

Unfortunately, despite the promising results, the introduction of the actuation limits made the computation prohibitively expensive. In fact, increasing the number of contacts dramatically increases the computation time. This unfortunately makes these polytopes hard to be computed *online* without accepting strong approximations on kinematics [11].

Another approach to address the problem of feasibility is to define a reference point[2] (henceforth we will consider the CoM, even though any other point can be chosen [12]) along with a 2D **feasible region** in which the projection of the reference point must lie inside, in order to meet the requested feasibility conditions (*e.g.,* friction, actuation or kinematic). Depending on the type of the constraints that are considered, such region can be convex or non-convex, and may depend on the instantaneous position of the CoM. The euclidean distance between the 2D projection of the CoM on the plane of the feasible region and the edges of the feasible region itself can be used to evaluate the *robustness* of the robot pose in static and dynamic gaits.

Because of the above reasons, the feasible region represents an intuitive yet powerful way to plan feasible trajectories for the CoM while being favored with its computational efficiency. Indeed, these regions are efficiently generated through incremental projection algorithms [13] that achieve a

[1] FWP has been introduced more recently than CWC and allowed to create motion without slippage or hitting the torque limits.

[2] In robotics there are many "ground" reference points used to devise locomotion strategies: ICP, ZMP, CoM, etc. Here a reference point could be any generic point that is connected with the motion of the robot.

reduced computational complexity: namely, a polygonal approximation of a projection of the original 6D polytope is computed without the need of computing the full-dimensional polytope (*i.e.,* the FWP).

Bretl et al. [14] were the first to introduce an Iterative Projection (IP) algorithm for the computation of a *support region* for arbitrary terrain (*e.g.,* non coplanar contacts). We will refer to such region as the *friction region* in the remainder of this manuscript to avoid possible confusion with the support polygon, which is the convex hull of the supporting feet.

In the previous work [1], we proposed a modified version of the IP algorithm to compute the *feasible region*, a convex region where both friction and actuation limits (*i.e.,* joint torque limits) were considered. As in the case of the FWP, the *feasible region* varies with the contact condition and with the joint configuration. The advantage of this convex region with respect to the 6D wrench polytope counterpart, is that it can be computed at least 20 times faster (10 *ms*). This makes planning CoM trajectories and foothold locations on arbitrary terrains based on such region, suitable for *online* implementation.

Nonetheless, to simplify the analysis, a few assumptions were adopted during the computation of the feasible region: (1) the sole external force acting on the robot is gravity, (2) the inertial accelerations and angular dynamics are neglected (quasi-static assumption); this means that the model used to build the region is a point mass model with contact forces, (3) kinematic limits are not considered, and (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 is incapable of capturing the effects of the application of an external force to the robot; as will be shown in section 3.2. External forces usually cause a shift in the region. 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 force is *intentionally* applied to the robot. This is the case when a load is pulled or when a rope is used for locomotion. As a matter of fact, on highly inclined terrains, using a rope can significantly aid the locomotion as it helps the robot walk in a configuration that is farther away from its limits (*i.e.,* contact forces towards the middle of the friction cones, and the joints towards the middle of their range [15], see Fig. 1.2). Without a rope, a limit on the terrain inclination that allows a statically stable gait exists, which is imposed by the friction coefficient[1]. Therefore, 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.

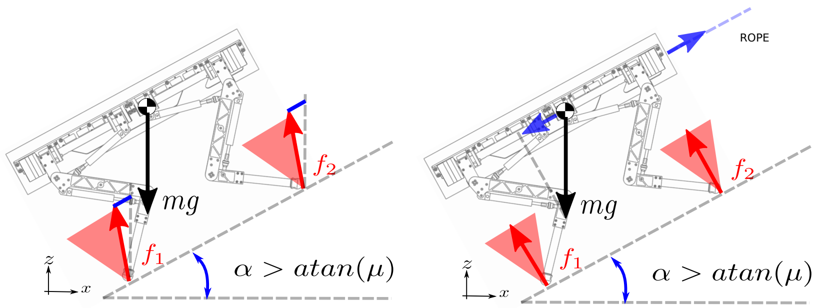



Figure 1.2: Robot climbing a steep ramp: (left) is slipping, (right) the usage of a rope to aid locomotion it increases the robustness of the contact (the contact forces are more toward the middle of the cones).

Assumption (2) limits the applicability of the 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. [16] incorporated the dynamics, proposing a robust static stability region that accounts for possible CoM accelerations. To achieve this, a limit is set on the possible CoM acceleration and accordingly, all the feasible CoM positions are consequently found. No other feasibility measures were considered. In contrast, Nozawa et al. [17] compute a dynamic stability region for the CoM based on specified linear and angular accelerations. In both approaches only friction guarantees were considered in the regions. We instead examine a *dynamic feasible region* (incorporating all the feasibility measures mentioned before) considering the dynamic balance constraints in a similar fashion to [17].

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 inconvenient

[1]If the bounds of the friction cone are both on the same side of the vertical direction, there will be a net tangential component coming from gravity that cannot be counterbalanced by the contact forces, causing slippage. This means that the ramp inclination cannot be larger than $atan(\mu)$.

adjustments in *height* and *orientation* may push the robot to violate its kinematic limits. In this respect, the seminal work of Carpentier et al. [18] focused on incorporating the kinematic constraints via learning proxy constraints. On a similar line, [19, 20] constrain the position of the CoM with respect to the contact points, however these kinematic constraints are only approximated, thus the "guarantees" that we mention for feasibility are only valid for a simplified representation of the robot. More recently, Fankhauser et al. [21] optimized the orientation to ensure static stability and kinematic limits, by solving a non-linear optimization problem (SQP). However, they used a rough approximation of the kinematic limits by setting bounds on the leg length. An SQP problem is also utilized in [17] to find a kinematically valid CoM target close to the original target chosen solely on the stability region. In the context of manipulators that move assembly objects, other approaches [22, 23] present a way to find all the orientations that satisfy static stability. Yet, the objects were fixed and not actuated. None of the previous studies evaluated a region, that is implicitly consistent with the robot kinematic limits.

## 1.1 Proposed Approach and Contribution

In this work we aim to address the above limitations and extend the descriptive capability of 2D admissible regions by introducing a redefinition of the *feasible region* initially proposed in [1]. In particular, in this extension we:

- Account for external wrenches acting on arbitrary points of the robot.
- Relax the quasi-static assumption by considering the dynamic effects, as well as the angular dynamics. Due to this, the model used to build the region from a **point mass** turns into a **centroidal dynamics** model [24]. Differently from [16] where the region was built considering the set of admissible CoM accelerations, we consider the *actual* acceleration resulting in a time-varying shape of the region when the robot is in motion.
- Embed the kinematic limits in what we call the *reachable region* (see section 3.5). This can be intersected with the actuation-aware region

and leads to an improved feasible region that considers friction, actuation and kinematic limits.

- Generalize the feasible region to be defined on arbitrary plane inclinations.
- Employ the proposed improved *feasible region* to plan *robust* CoM trajectories for the HyQ robot and propose a new optimization for the trunk orientation based solely on this quantity. The optimal orientation is obtained with a sampling-based method and aims to maximize the margin w.r.t. the joint limits for the whole CoM trajectory. 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. Being able to adjust robustness improves the quality of planning as it tolerates modeling and state estimation errors along with making the controller more resilient to external perturbations.
- We show simulations with the robot walking in scenarios that are challenging in terms of actuation and kinematic motions. We compare a planning approach based on the improved *feasible region* with our previous heuristic approach [5] that had no feasibility guarantees, showing that the former prevents violations in the actuation and kinematic limits, while with the heuristic approach, they are violated several times. We also show preliminary results on hardware where HyQ is walking on flat terrain at a significantly lower height than the nominal value.

## 1.2 Outline

The thesis is organized as follows: in Section 2 we recall the modified IP algorithm used to compute the feasible region, while in Section 3 we presents the updates to compute the new *feasible region*. Section 4 illustrates the planning strategies for CoM and orientation based on the proposed region. Simulations and experimental results with the HyQ robot are presented in Section 5 and 6. Section 7 draws the conclusions and discusses possible future developments.

# Chapter 2

# Recap on classical feasible region

The *feasible region*, previously presented in [1], was generated using a modified IP algorithm described in Algorithm 1.

The algorithm considers the convex constraints existing on a legged robot 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 LP programs while satisfying the convex constraints (shown in step (III) of Algorithm 1). Namely, the static stability constraints (III.a), frictional constraints on the contact feet (III.b), and the joint actuation constraints (III.c) were considered.

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. 2.1).

Each vertex $\mathbf{c}^*_{xy}$ also defines a half-space, as illustrated in Fig. 2.1 (dashed gray line), which, along with all the other half-spaces defined by the other vertices, outlines an outer approximation of the region. This outer approximation contains the inner approximation (polygon connecting the vertices) along with other unclassified points. The direction $\mathbf{a}_i$ is chosen, at each iteration, to minimize the amount of points needed to be classified. Furthermore, a desired precision for the algorithm can be defined by setting the minimum allowed difference (in area) between the inner and outer

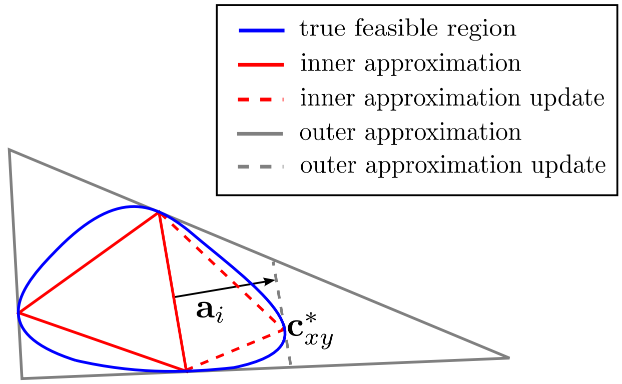



Figure 2.1: Iteration of the IP algorithm: after the LP is solved finding a new extremal $\mathbf{c}^*_{xy}$ point along $\mathbf{a}_i$, this is added to the *inner* approximation while an edge with normal $\mathbf{a}_i$ passing through $\mathbf{c}^*_{xy}$ is added to the *outer* approximation [1].

approximation.

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.*, $m = 3$ point contact, $m = 6$ 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. $\mathbf{u} \in \mathbb{R}^6$ is the linear part of 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:

$$
\begin{aligned}
&\mathbf{A}_1 = \begin{bmatrix} \bar{\mathbf{A}}_1 & \dots & \bar{\mathbf{A}}_{n_c} \end{bmatrix} \in \mathbb{R}^{6\times mn_c}, \\
&\mathbf{A}_2 = \begin{bmatrix} \mathbf{0} \\ -m\mathbf{g} \times \mathbf{P}^T \end{bmatrix} \in \mathbb{R}^{6\times 2}, \quad \mathbf{P}_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
&\mathbf{u} = \begin{bmatrix} -m\mathbf{g} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{g} = [0, 0, -g]^T.
\end{aligned}
\tag{2.1}
$$

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

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

**Algorithm 1** Feasible Region IP algorithm (with additional wrenches).

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

**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:

$$\mathbf{c}^*_{xy} = \underset{\mathbf{c}_{xy}, \mathbf{f}}{\text{argmax}} \quad \mathbf{a}_i^T \mathbf{c}_{xy}$$

such that :

$$\begin{aligned} &(\text{III.a}) \quad \mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{u} \\ &(\text{III.b}) \quad \mathbf{B}\mathbf{f} \leq \mathbf{0} \\ &(\text{III.c}) \quad \mathbf{G}\mathbf{f} \leq \mathbf{d} \end{aligned}$$

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

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

**end while**

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

Constraint (III.b) ensures the friction constraints are met. These are requiring that the contact forces are inside a pyramid (conservative) approximation of the friction cones. For contact surface normals $\mathbf{n}_i \in \mathbb{R}^3$, tangent vectors $\mathbf{t}_{x,i}, \mathbf{t}_{y,i} \in \mathbb{R}^3$, and friction coefficients $\mu_i \in \mathbb{R}$, the constraint matrix $\mathbf{B} \in \mathbb{R}^{4n_c \times 3n_c}$ is represented as:

$$\begin{aligned} \mathbf{B} &= \text{diag}(\mathbf{b}_1, \ldots, \mathbf{b}_{n_c}), \\ \mathbf{b}_i &= \begin{bmatrix} (\mathbf{t}_{x,i} - \mu_i \mathbf{n}_i)^T \\ (\mathbf{t}_{y,i} - \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{x,i} + \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{y,i} + \mu_i \mathbf{n}_i)^T \end{bmatrix} \in \mathbb{R}^{4\times 3} \end{aligned} \tag{2.2}$$

Note that in case of bilateral contacts, the friction constraints are not applied on those contacts, and the dimension of $\mathbf{B}$ is modified accordingly.

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[1]. This yields to the definition of force polytopes that represent the sets of admissible contact forces that respect actuation limits. By considering the vectors of minimum ($\underline{\boldsymbol{\tau}}_i \in \mathbb{R}^{n_l}$) and maximum ($\bar{\boldsymbol{\tau}}_i \in \mathbb{R}^{n_l}$) joint torque limits, on the $n_l$ joints of the $i$th leg, the half plane description of such force polytopes is represented by $\mathbf{G} \in \mathbb{R}^{2n_l n_c \times m n_c}$ and $\mathbf{d} \in \mathbb{R}^{2n_l n_c}$:

$$\mathbf{G} = \text{diag}\left(\begin{bmatrix} \mathbf{J}(\mathbf{q}_1)^T \\ -\mathbf{J}(\mathbf{q}_1)^T \end{bmatrix}, \dots, \begin{bmatrix} \mathbf{J}(\mathbf{q}_{n_c})^T \\ -\mathbf{J}(\mathbf{q}_{n_c})^T \end{bmatrix}\right), \mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{n_c} \end{bmatrix} \tag{2.3}$$

where $\mathbf{q}_i$ represents the vector of angular positions of the joints of the $i$-th leg in contact with the environment (cfg. [11] on how to compute $\mathbf{d}$ from $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$). Note that $\mathbf{q}_i$ are not directly provided as inputs to the Algorithm 1. However, knowing the kinematic model of the robot, the joint values can be simply computed from other inputs: the feet positions ${}^{\mathcal{W}}\mathbf{x}_{f_i}$, the CoM location $\mathbf{c}$ (both expressed with respect to the world frame) and the trunk orientation ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$. 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. 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:

$$\mathcal{Y}_{fa} = \Big\{\mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{f}_i \in \mathbb{R}^{m n_c}, \text{ s.t. } (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A}\Big\} \tag{2.4}$$

where $\mathcal{C} \cap \mathcal{A}$ is the set of contact forces and CoM positions (projected on a $X - Y$ plane) satisfying both friction and actuation constraints:

$$\begin{aligned} \mathcal{C} \cap \mathcal{A} = \Big\{\mathbf{f}_i \in \mathbb{R}^{m n_c}, \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad & \mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{u} \\ & \mathbf{B}\mathbf{f} \leq \mathbf{0}, \quad \mathbf{G}\mathbf{f} \leq \mathbf{d}\Big\} \end{aligned} \tag{2.5}$$

As mentioned before, the developed feasible region assumed the absence of external wrenches and was only suitable for quasi-static motions. Furthermore, not considering the kinematic limits of the robot in the feasibility analysis can be insufficient for complex motions.

[1] This is true for a non-redundant leg, where the Jacobian is a square matrix.

Note that the model considered is a point mass where the contact forces enter the equation (III.a), and there is no angular dynamics. In the following, these assumptions will be relaxed and a more general formulation of the feasible region will be introduced.

# Chapter 3

# Improvements on the Feasible Region

In this chapter we propose an extension of the feasible region to arbitrary plane inclinations. We then proceed to incorporate external wrenches (section 3.2), dynamic effects (section 3.3), and kinematic limits (section 3.5). The changes on the algorithm are highlighted in blue in Algorithm 1.

## 3.1 Generic Plane of Projection

Under the sole influence of gravity and considering only *friction* constraints, the static equilibrium constraints in [14] are only affected by the horizontal position of the CoM[1]. 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 stability 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.

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

More importantly, as will be explained further in Section 3.2, 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 region of feasibility. 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. Therefore, 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 [5, 25], while for a robot climbing a ladder it is expected to lie in the vertical plane. 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.

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

$$\mathbf{c} = {}^{\mathcal{W}}\mathbf{R}_{\Pi}\hat{\mathbf{c}} \tag{3.1}$$

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 $\mathcal{W}$ and a frame attached to the plane of interest $\Pi$, respectively. ${}^{\mathcal{W}}\mathbf{R}_{\Pi}$ is the rotation matrix representing the orientation of the plane of interest $\Pi$ with respect to the world frame $\mathcal{W}$, and is defined as:

$${}^{\mathcal{W}}\mathbf{R}_{\Pi} = \left[\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\right] \tag{3.2}$$

The $\hat{z}$-axis of $\Pi$ is aligned with a vector that we will call $\mathbf{n}$. $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ are unit vectors (expressed in $\mathcal{W}$ frame and forming the $\hat{x}, \hat{y}$-axes of $\Pi$ frame) chosen such that they form, together with $\mathbf{n}$, a right-handed coordinate system. With the change of coordinates in (3.1), 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 $\hat{\mathbf{c}}_{xy}$. In the remainder of this manuscript, not to overload the notation, we assume the CoM to be constrained in a plane perpendicular to gravity (parallel to the horizontal plane of the world frame), hence $\hat{\mathbf{c}} = \mathbf{c}$. Therefore, the CoM position expressed in the world frame $\mathbf{c}$ will be used extensively in all related equations, without any loss of generality.

## 3.2 External wrenches

Consider an external wrench, $\boldsymbol{w}_{ext} = [\boldsymbol{f}_{ext}, \boldsymbol{\tau}_{ext}] \in \mathbb{R}^6$, applied on the CoM of a legged robot. For the robot to be in *static* equilibrium, the wrench balance equations should satisfy:

$$\sum_{i=1}^{n_c} \mathbf{f}_i + m\mathbf{g} + \boldsymbol{f}_{ext} = 0 \tag{3.3}$$

$$\sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i - (m\mathbf{g} + \boldsymbol{f}_{ext}) \times \mathbf{c} + \boldsymbol{\tau}_{ext} = 0 \tag{3.4}$$

As mentioned in the previous section, with only the gravity $\mathbf{g}$ acting on the robot, the dependence on the CoM in the second equation only comes from its horizontal positions $\mathbf{c}_{xy}$. However, with the presence of an external force, $\boldsymbol{f}_{ext}$, a dependence on the CoM vertical position $c_z$ can clearly exist from the term $-\boldsymbol{f}_{ext} \times \mathbf{c}$ (unless $\boldsymbol{f}_{ext}$ is aligned with gravity).

To incorporate the effect of $\boldsymbol{w}_{ext}$ on Algorithm 1, the constraint (III.a) can be rewritten by redefining $\mathbf{A}_2$ and $\mathbf{u}$ to be:

$$\begin{aligned} \mathbf{A}_2 &= \begin{bmatrix} \mathbf{0} \\ -[m\mathbf{g} + \boldsymbol{f}_{ext}] \times \mathbf{P}_{xy}^T \end{bmatrix} \in \mathbb{R}^{6\times 2} \\ \mathbf{u} &= \begin{bmatrix} -m\mathbf{g} - \boldsymbol{f}_{ext} \\ [\boldsymbol{f}_{ext}] \times \mathbf{P}_z^T c_z - \boldsymbol{\tau}_{ext} \end{bmatrix} \in \mathbb{R}^{6\times 1} \end{aligned} \tag{3.5}$$

Therefore, $\mathbf{A}_2$ computes the moments due to gravity and external forces (acting on the robot CoM [1]), about the origin of the world frame.

To better appreciate the effect of an external wrench $\boldsymbol{w}_{ext}$ 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 (3.4), $\mathbf{c}_{xy}$ can be determined as:

$$\mathbf{c}_{xy} = -\mathbf{h}(\mathbf{f}) + \mathbf{m}(\boldsymbol{f}_{ext}, \boldsymbol{\tau}_{ext}, c_z) \tag{3.6}$$

where $\mathbf{h}$ is a function linear in the contact forces $\mathbf{f}$ and $\mathbf{m}$ is an offset function dependent on the external forces $\boldsymbol{f}_{ext}$, external torques $\boldsymbol{\tau}_{ext}$, and the CoM

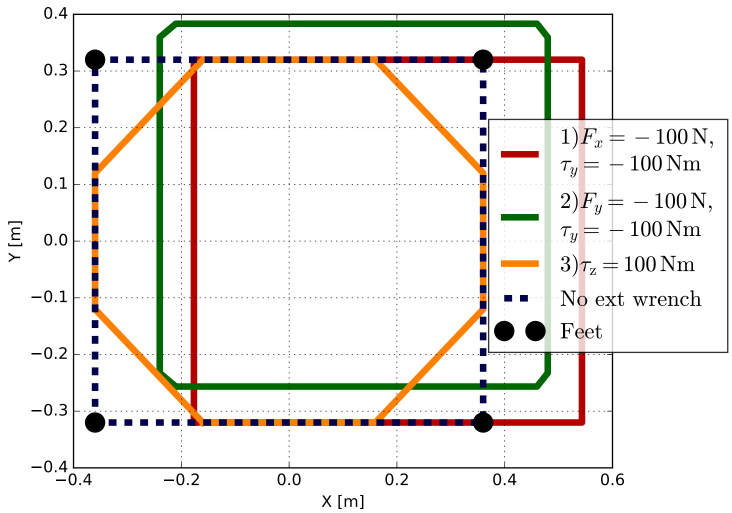
[1] If a pure force is applied in a different point of the robot the equivalent wrench at CoM should be computed.

vertical position $c_z$. From this, 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).

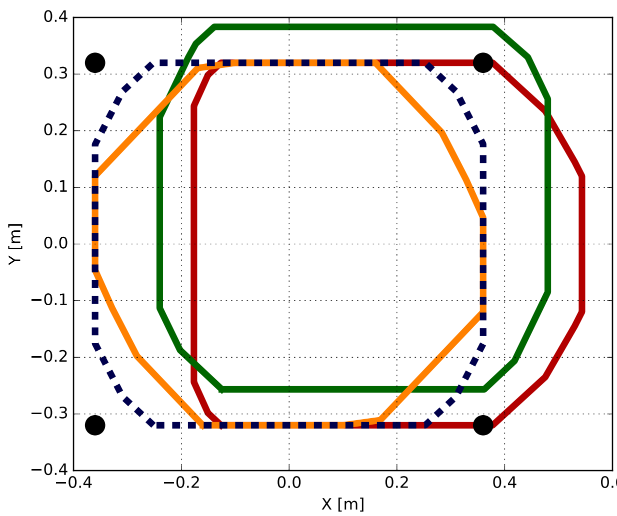
The change in shape of the region, can be intuitively understood, considering that the set of contact forces resulting from the action of the external wrench, could become infeasible due to the additional effort needed to compensate the external wrench. This usually results in smaller regions because, for extremal CoM positions, the contact forces typically become infeasible in terms of actuation or friction constraints.

For example, in case of a leg significantly retracted, because the joint-torques are propagated through the leg to the foot via the Jacobian, the CoM positions closer to the contacts feet are more likely to be infeasible. Besides that, 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.

Figure 3.1 illustrates examples of the resulting friction and feasible regions for different external wrench cases calculated for 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 force. 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. Further reduction in the size of the region can be observed on the feasible region for all cases of the external wrenches, because in that also the actuation limits are considered.




(a) Friction Region (only friction considered)




(b) Feasible Region (both friction and actuation considered)

Figure 3.1: 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 not mentioned are zero. The stance feet of HyQ are shown as black points with the front feet facing right. Regions are computed for a trunk height of $c_z = 0.53m$

## 3.3 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 Zero Moment Point (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 static 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 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 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 reference point and proceed to incorporate the dynamic effects (relaxing the static assumptions) in the feasible region (constraints III.a in Algorithm 1).

In certain cases it could even happen that the ZMP is out of the computed region (*e.g.,* because of the action of an external force) but as long as the CoM projection is inside it, the robot configuration is feasible and dynamic stability will be ensured.

Note that, including dynamic effects it requires that we express the Newton-Euler equations in the inertial frame. This means that the moment balance it should be done with respect to the origin of the inertial frame, that in general is not coincident with the CoM. As a consequence the equations get more complicated and several bias terms appear due to the derivative of vectors when a body frame is rotating [26]:

$$\underbrace{\begin{bmatrix} mI_{3\times3} & -m\mathbf{c}\times \\ m\mathbf{c}\times & I_G - \mathbf{c}\times\mathbf{c}\times \end{bmatrix}}_{I_s} \begin{bmatrix} \ddot{\mathbf{c}} \\ \dot{\omega} \end{bmatrix} + \underbrace{\begin{bmatrix} m\omega\times\omega\times\mathbf{c} \\ \omega\times I_G\omega + m\mathbf{c}\times\omega\times(\omega\times\mathbf{c}) \end{bmatrix}}_{b} = \begin{bmatrix} \sum_{i=1}^{n_c}\mathbf{f}_i + m\mathbf{g} \\ \sum_{i=1}^{n_c}\mathbf{p}_i\times\mathbf{f}_i - (m\mathbf{g})\times\mathbf{c} \end{bmatrix} \tag{3.7}$$

where $I_s \in \mathbb{R}^{6\times6}$ is the *spatial* inertia, $\ddot{\mathbf{c}}$ the CoM euclidean acceleration, and $\dot{\omega}, \omega$ the angular acceleration and velocity of the robot base, respectively. In equation (3.7) one can see the dependency on $\mathbf{c}$ is no longer linear due to the double cross-product terms $\mathbf{c} \times \mathbf{c} \times$ in the angular dynamics. However, the influence of these inertial moments is small in comparison to the one of contact forces and gravity. Therefore, as a simplification, we consider a constant value (equal to the actual CoM position) in the computation of $I_s$ and of the bias terms $b$. Therefore, if we incorporate the dynamic effects, the matrix $A_2$ remains unchanged and $u$ in (III.a) should be redefined as:

$$\mathbf{u} = \begin{bmatrix} -m\mathbf{g} \\ \mathbf{0} \end{bmatrix} + I_s \begin{bmatrix} \ddot{\mathbf{c}} \\ \dot{\omega} \end{bmatrix} + b \tag{3.8}$$

Note that now the *simple mass* model becomes a *centroidal dynamics* model as the angular dynamics is also taken into account. Moreover, the *static* stability enforced in (III.a) can be considered to be fully *dynamic*.

The dynamic effects will become visible from the fact that the region will "move" (*e.g.,* forward or backward) according to the direction of the instantaneous body acceleration (see accompanying video).

This shift in the dynamic region could be exploited for planning purposes: the region could be shifted forward when swinging the legs that are in the direction of motion, thus avoiding to move the CoM backward. We conjecture that the reason for which real quadrupeds (*e.g.,* horses) move their head forward periodically during locomotion, is to accelerate their CoM (and therefore to shift the associated region) forward in order to keep stability when swinging the front legs, avoiding unnecessary backward motion and maximizing forward motion. Even though the the CoM projection might move out of the convex hull of the contact points, it might still reside within the feasible region and thus the quadruped would still be dynamically stable.

## 3.4 Degenerate Feasible Regions

It is possible to further extend the feasible region to dynamic gaits (*e.g.,* a trot or a pace) were 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.

This extension of the feasible region to degenerate cases is made numerically possible by assuming the presence of an infinitesimal contact torques at the feet. In particular, we assume that the feet can exert a small torque component tangential to the ground $\tau_x$ and $\tau_y$, but we assume that the foot cannot perform any contact torque orthogonal to the ground $\tau_z$. We include these wrench components in the constraint (III.b) of Algorithm 1: we update the matrix $\mathbf{B}$ in Eq. 2.2 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 $\tau_x, \tau_y$. The values $\tau_x^{lim}, \tau_y^{lim}$ represent the infinitesimal limits of the box constraint on the contact torque tangential to the ground in the foot location:

$$
\begin{gathered}
\mathbf{b}_i^{cone} = \begin{bmatrix} (\mathbf{t}_{1,i} - \mu_i \mathbf{n}_i)^T \\ (\mathbf{t}_{2,i} - \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{1,i} + \mu_i \mathbf{n}_i)^T \\ -(\mathbf{t}_{2,i} + \mu_i \mathbf{n}_i)^T \end{bmatrix}, \quad \mathbf{b}_i^{box} = \begin{bmatrix} \tau_x^{lim} & 0 \\ 0 & \tau_y^{lim} \\ -\tau_x^{lim} & 0 \\ 0 & -\tau_y^{lim} \end{bmatrix} \\
\mathbf{B} = \operatorname{diag}\left( \begin{bmatrix} \mathbf{b}_1^{cone} & \mathbf{0}_{4\times 2} \\ \mathbf{0}_{4\times 3} & \mathbf{b}_1^{box} \end{bmatrix} \dots \begin{bmatrix} \mathbf{b}_{n_c}^{cone} & \mathbf{0}_{4\times 2} \\ \mathbf{0}_{4\times 3} & \mathbf{b}_{n_c}^{box} \end{bmatrix} \right) \in \mathbb{R}^{8n_c \times 5n_c}
\end{gathered}
\tag{3.9}
$$

Because of the non-zero values of the contact torque limits $\tau_x^{lim}$ and $\tau_y^{lim}$, the feasible region portrayed in Figure 3.2 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. In the likely case in which the actuators' limits are too small to allow the robot to balance on the only stance leg, the feasible region will then be undefined. Note that if the dynamic effects are considered, the feasible line will move back/forth when the robot accelerates backwards/forward, according to what explained in Section 3.3. Figure 3.2 shows that the feasible region is a straight segment during a trotting motion and it is shifted forward with respect to the supporting line, because the robot is accelerating forward. We can see that the ZMP (green point), instead, moves backwards in the opposite direction to the acceleration.

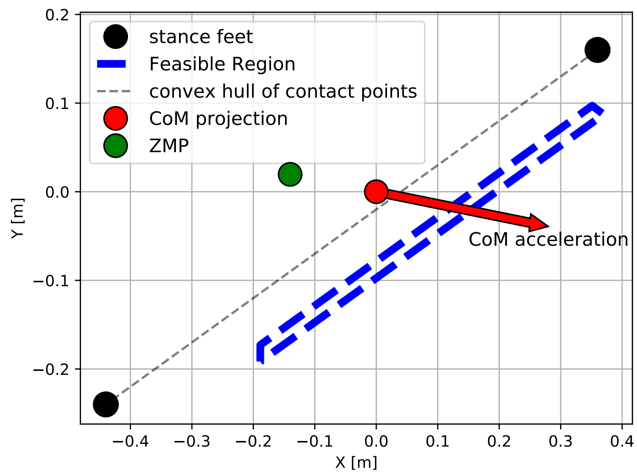



Figure 3.2: Feasible regions degenerate to a line during a trot, when only two legs are in contact. This segment is shifted forward in the same direction of the robot's acceleration.

In future works, this can be exploited to perform fast turning maneuvers to check the maximum feasible sideways inclination that can be achieved (*e.g.,* to compensate centrifugal forces).

## 3.5 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 actuation limits has proved to be important in many cases. This is particularly true, for instance, in cases where the robot is traversing steep terrains, or scenarios where there is an extra weight or an external wrench acting on the robot.

Once the torque-limits are considered, the limited legs workspace remains the next major restrictive factor for motion planning. Indeed, the feasible region makes it easier to plan admissible motion plans for on complex terrains where complex robot configurations are required and, therefore, it gets even more compelling to make sure that the obtained trajectories do not violate the joint-kinematic limits or that none of the legs approaches a singularity. Kinematic limits are common, for instance, in linear actuators used in hydraulic quadrupeds, 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 legs (*e.g.,* humanoid climbing stairs).

In principle, these singularities are already captured in the previously introduced feasible region itself, since the Jacobians used to compute the region become singular resulting in flat force polytopes (with no volume), thus impeding the feasibility for the contact force. However, the Jacobians cannot consider other limitations like the *limited range* in the joints.

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 [27]. To simplify the nomenclature, we refer to it 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:

$$ {}^{\mathcal{B}}\mathbf{x}_{f_i} = f_i(\mathbf{q}_i), \quad \forall i = 1, ..., n_c \tag{3.10} $$

mapping the joint angles $\mathbf{q}_i \in \mathbb{R}^{n_l}$ 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 ${}^{\mathcal{W}}\mathbf{x}_{f_i}$ is known, ${}^{\mathcal{B}}\mathbf{x}_{f_i}$ can be simply computed as

$$ {}^{\mathcal{B}}\mathbf{x}_{f_i} = {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}({}^{\mathcal{W}}\mathbf{x}_{f_i} - \mathbf{c}) + {}^{\mathcal{B}}\mathbf{c} \tag{3.11} $$

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 (3.10) and (3.11) and rewriting for $\mathbf{c}$, we obtain:

$$\mathbf{c} = \mathbf{F}_i(\mathbf{q}_i, {}^{\mathcal{W}}\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}), \quad \forall i = 1, ..., n \tag{3.12}$$

where $\mathbf{F}_i$ is defined as:

$$\mathbf{F}_i(\mathbf{q}_i, {}^{\mathcal{W}}\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}) = {}^{\mathcal{W}}\mathbf{x}_{f_i} - {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}(f_i(\mathbf{q}_i) - {}^{\mathcal{B}}\mathbf{c}) \tag{3.13}$$

Therefore, for a given foot position ${}^{\mathcal{W}}\mathbf{x}_{f_i}$ and trunk orientation ${}^{\mathcal{W}}\mathbf{R}_B$, (3.12) provides a relationship between the joint-space angles of each leg and the CoM task-space position. Assuming that the feet in contact do not move, for a CoM position ${}^{\mathcal{W}}\mathbf{x}_{com}$ to be reachable, there must exist joint angles $\mathbf{q}_i$, satisfying (3.12), for each leg $i$ such that:

a. $\underline{\mathbf{q}}_i \leq \mathbf{q}_i \leq \bar{\mathbf{q}}_i$

b. $J_i(\mathbf{q}_i) = [\partial f_i(\mathbf{q}_i)/\partial \mathbf{q}_i]$ 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 (3.12) (we drop the explicit dependence on ${}^{\mathcal{W}}\mathbf{x}_{f_i}$ and ${}^{\mathcal{W}}\mathbf{R}_B$ that are input parameters, to lighten the notation), along with conditions (a) and (b), to define the *reachable region* as:

$$\mathcal{Y}_r = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{q}_i \in \mathbb{R}^{n_l} \text{ s.t.} \quad (\mathbf{c}_{xy}, \mathbf{q}_i) \in \mathcal{Q} \Big\} \tag{3.14}$$

where:

$$\begin{aligned} \mathcal{Q} = \Big\{ & \mathbf{q}_i \in \mathbb{R}^{n_l}, \mathbf{c}_{xy} \in \mathbb{R}^2 | \text{ s.t.} \quad \mathbf{c}_{xy} = \mathbf{P}_{xy}\mathbf{F}_i(\boldsymbol{q}_i), \\ & \underline{\boldsymbol{q}}_i \leq \boldsymbol{q}_i \leq \bar{\boldsymbol{q}}_i, \quad \text{row-rank}(J_i(\mathbf{q}_i)) = n_l \quad \forall i = 1, ..., n_c \Big\} \end{aligned} \tag{3.15}$$

where only the legs in contact are considered. It is important to note that such 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 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 turn 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 2 describes the procedure developed to compute the region. A similar algorithm was developed in [29], and was used to evaluate the workspace of a Stewart platform based machine tool. A modification was applied to increase the robustness and the performance.

Inspired by ray-casting algorithms, a discretized search is done iteratively in ordered directions along polar coordinates $(\rho, \theta)$ 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 thesis, we will refer to the reachable region $\mathcal{Y}_r$ as its polygonal approximation.

Each ray along some direction $\mathbf{a}_i$ finds the farthest point $\boldsymbol{\nu}_{xy}^*$ that yet 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:

$$\max_{\boldsymbol{\nu}_{xy}} \mathbf{a}_i^T \boldsymbol{\nu}_{xy} \tag{3.16}$$

$$\text{s.t. } \forall i = 1, ..., n_c:$$

$$\mathbf{q}_i = \bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy}) \tag{3.17}$$

$$\underline{\mathbf{q}}_i < \mathbf{q}_i < \bar{\mathbf{q}}_i \tag{3.18}$$

$$\sigma_{\min}\left\{J\left(\mathbf{q}^k\right)\right\} > \sigma_0 \tag{3.19}$$

The relation (3.17) represents the kinematic constraint in (3.15) reformulated in terms of the inverse kinematics. $\bar{\mathbf{F}}_i$, therefore, is defined as:

$$\bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy}) = f_i^{-1}[{}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}({}^{\mathcal{W}}\mathbf{x}_{f_i} - \mathbf{P}_{xy}^T\boldsymbol{\nu}_{xy} - \mathbf{P}_z^T c_z) + {}^{\mathcal{B}}\mathbf{c}] \tag{3.20}$$

where $f_i^{-1}$ refers to the inverse kinematics mapping. It is important to note from (3.20) that for specific feet positions, the location of each $\boldsymbol{\nu}_{xy}^*$

(and accordingly the resultant region) is influenced by the height $c_z$ and the orientation ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ of the robot. A simple check for the presence of a singularity is done in (3.19), where $\sigma_{\min}$ is the smallest singular value and $\sigma_0$ is a small value of choice. Due to the non-linearity of constraints (3.17) and (3.19) 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 $\boldsymbol{\nu}^*_{xy}$. 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 $\boldsymbol{\nu}^*_{xy}$ lies in. A fast bisection search is then executed on this interval to find $\boldsymbol{\nu}^*_{xy}$ while making sure it's within an error of $[0, -\Delta\rho_{\min}]$ from the boundary of the actual workspace. The function *isReachable($\rho$)*, used in Algorithm 2, computes the inverse kinematics of a CoM position and checks if that position is reachable:

**isReachable**($\rho$):

$\boldsymbol{\nu}_{xy} \leftarrow \mathbf{c}_{xy} + \rho\mathbf{a}$
$\mathbf{q_i} = \bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy})$
**return** true if $\mathbf{q_i}$ satisfies (3.18) & (3.19)

Each vertex $\boldsymbol{\nu}^*_{xy}$ 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. 3.3 and 3.4). The algorithm stops when a step smaller than $\Delta\rho_{min}/2$ set by the user, is reached.

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. 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 $\mathbf{c}_{xy}$, only influences the *accuracy* of the generated region. A disadvantage of such 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 4.1), the trajectory of the CoM should always remain inside the region. On the other hand, it is important also to consider the effect of the robot height $c_z$ and orientations ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ on the reachable region. In fact, different evaluations of the reachable region, presented in Fig. 3.3 and 3.4, show that the size, positioning, shape, and

**Algorithm 2** Iterative discretized ray-casting algorithm

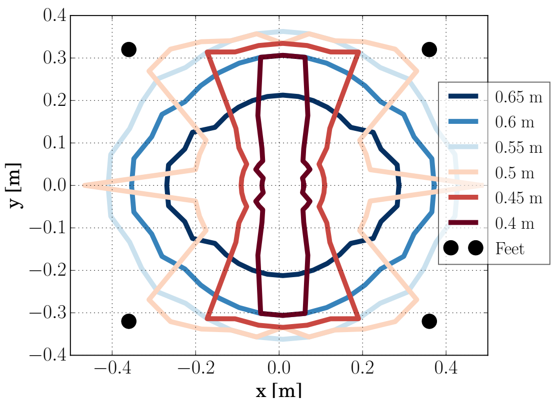
**Input:** $\mathbf{c}_{xy}, c_z, {}^{W}\mathbf{R}_B, \mathbf{p}_1, ..., \mathbf{p}_{n_c}, \underline{\mathbf{q}}_1, \bar{\mathbf{q}}_1, ..., \underline{\mathbf{q}}_{n_c}, \bar{\mathbf{q}}_{n_c}$
**Result:** reachable region $\mathcal{Y}_r$
**Initialization**: $\boldsymbol{\nu}_{xy} = \mathbf{c}_{xy}$, $\mathcal{Y}_r \leftarrow \{\}$
**for** $\theta = 0$ to $2\pi$ **do**
    Compute direction: $\mathbf{a}_i = \begin{bmatrix} \cos\theta & \sin\theta & 0 \end{bmatrix}^T$

    *Find the first bisection interval*:
    **while** *isReachable($\rho$)* **do**
        $\rho \leftarrow \rho + \Delta\rho$
    **end while**

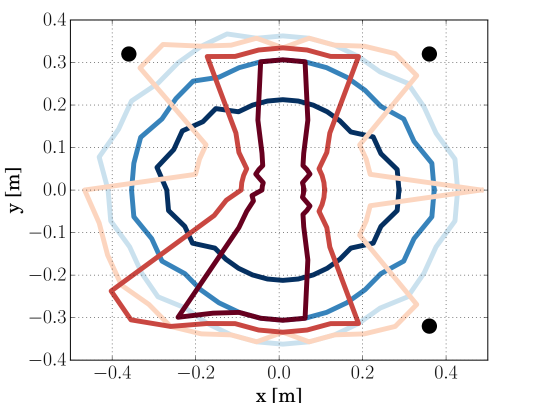
    *Bisection search*:
    $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$
    **while** $\Delta\rho \geq \Delta\rho_{\min}/2$ **do**
        **if** *isReachable($\rho$)* **then**
            $\rho \leftarrow \rho + \Delta\rho$
        **else**
            $\rho \leftarrow \rho - \Delta\rho$
        **end if**
        $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$
    **end while**

    **if** last $\boldsymbol{\nu}_{xy}$ not *isReachable($\rho$)* **then**
        $\rho \leftarrow \rho - \Delta\rho_{\min}$
        $\boldsymbol{\nu}_{xy} \leftarrow \mathbf{c}_{xy} + \rho\mathbf{a}$
    **end if**
    $\mathcal{Y}_r \cup \left\{\boldsymbol{\nu}^*_{xy}\right\}$
**end for**
**return** $\mathcal{Y}_r$

convexity of the reachable region can differ greatly at different $c_z$ and ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$. 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 configurations, the convexity of region can be significantly affected. Such insight is greatly useful in situations where planning needs to be performed in cluttered environments.




(a) Four stance feet.




(b) Three stance feet.

Figure 3.3: Different evaluations of the reachable region at different robot heights.

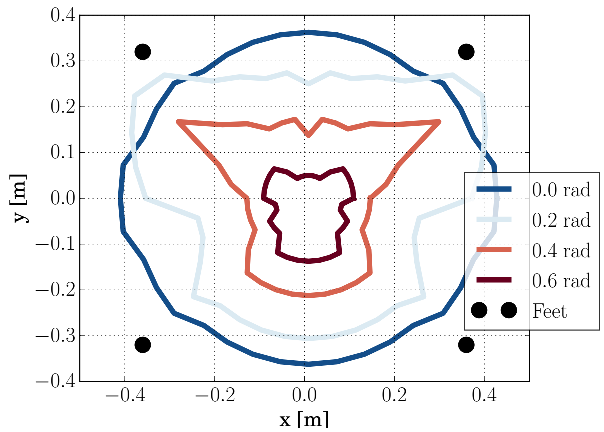



Figure 3.4: Different evaluations of the reachable region at different robot orientations.

Hereafter, we present some remarks related to the algorithm:

**Remark 1:** For legged robots with legs having redundant joints, finding a solution for the inverse kinematics can be challenging, particularly at the position level. Furthermore, internal singularity loci could appear in the workspace, introducing internal barriers and resulting in an excessively conservative reachable region [30]; considerable parts of the region become hidden to the algorithm, even though the singularities could have been avoided during the motion planning stage [31]. While this, in general, is not an issue for non-redundant legs, it is a standard difficulty for determining the reachable region of robots with redundant legs.

**Remark 2:** The full workspace can be comprised of disjoint sets (*e.g.,* [29]) which would not be captured by the algorithm. However, the only significant set for the CoM planning is the one visible from the current CoM, given that the other sets are unreachable from it.

**Remark 3:** Given the non-convexity of the region, the choice of $\Delta\theta$ and $\Delta\rho_{\min}$ should be small enough to provide a good approximation of the region. However, a higher computation time is required for such increase in precision. We observed that choosing $\Delta\theta = 20°$ and $\Delta\rho_{\min} = 0.03m$ for the HyQ robot during simulations and experiments, provides satisfactory results.

## 3.6 The Improved Feasible Region

The reachable region (3.14) can be seen as a projection of the high-dimensional convex set $\mathcal{Q}$ 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 extended *feasible region* as:

$$\mathcal{Y}_{far} = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{f}_i \in \mathbb{R}^{mn_c}, \mathbf{q}_i \in \mathbb{R}^{n_l} \text{ s.t.} \\ (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A}, \quad (\mathbf{c}_{xy}, \mathbf{q}_i) \in \mathcal{Q} \Big\} \tag{3.21}$$

Given that $\mathcal{C} \cap \mathcal{A}$ and $\mathcal{Q}$ are defined on different spaces, $\mathcal{Y}_e$ can therefore be obtained by computing the feasible region $\mathcal{Y}_{fa}$ (projecting $\mathcal{C} \cap \mathcal{A}$) and the reachable region $\mathcal{Y}_r$ (projecting $\mathcal{Q}$) separately, then considering the intersection of the two regions. Therefore we can define the new *feasible region* as:

$$\mathcal{Y}_{far} = \mathcal{Y}_{fa} \cap \mathcal{Y}_r \tag{3.22}$$

| **Name** | **Symbol** | **Constraints** |
|---|---|---|
| Friction Region | $\mathcal{Y}_f$ | Friction |
| Feasible Region | $\mathcal{Y}_{fa}$ | Friction / Actuation |
| Reachable Region | $\mathcal{Y}_r$ | Kinematic |
| Improved Feasible Region | $\mathcal{Y}_{far}$ | Friction / Actuation / Kinematic |

Table 3.1: Types of regions

Note that 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 $\mathcal{Y}_a$ as explained in [1]. 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 actuation 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. Finally, differently from the $\mathcal{Y}_{fa}$ region, that took into account only friction and actuation constraints, the new feasible region $\mathcal{Y}_{far}$ will be non-convex because the reachable region is non-convex (given that the intersection between a convex set and a non-convex set is non-convex). In Table 3.1 we summarize the type of regions introduced together with the correspondent constraints.

# Chapter 4

# Trajectory Planning

## 4.1 CoM planning strategy

In this section we improve the heuristic CoM planning strategy developed for *crawl* gaits described in our previous work [5], by exploiting the proposed definition of improved feasible region (see Fig. 1.1). This planning framework assumes a quasi-static motion: 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 used[1].

Through the use of the feasible region we will improve the heuristics behavior adding guarantees on the physical feasibility in order to obtain a certain level of robustness.

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.

As such, we ensure a certain level of robustness during the swing phase (also labeled as three-stance phase) which is the most critical in terms of stability (the friction region is typically smaller), and actuation capability as only three legs support the whole robot 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

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

stance legs (known from the foot sequence). A feasible target CoM position, using the criterion explained below, is then chosen. In such 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. As mentioned in Section 2, the Jacobians used to evaluate the force polytopes of the contact legs make the feasible region configuration-dependent. Therefore, we choose to compute them at the configuration associated to the target CoM position provided by the heuristics.

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 $s \in (0, 1)$.

The procedure is devised as follows: if the current CoM projection $\mathbf{c}_{xy}$ (onto the region plane)[1] is inside $s\mathcal{Y}_{far}$, feasibility is already guaranteed and the target CoM position is chosen to be the current one to minimize unneeded 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 heuristics. This allows the motion to: (1) be as close as possible to the heuristic target, thus benefiting from its proven reliable practical effectiveness [5], (2) formally fulfill the feasibility requirements, and (3) achieve a desired level of robustness (tunable by the shrinkage factor $s$). Remaining close to the heuristic target, also allows to (4) maintain the *local* validity of the *feasible region* (the Jacobians were evaluated for the heuristic target position).

Note that the non-convexity of $\mathcal{Y}_{far}$ presents a few complications regarding the planning procedure as opposed to a convex region: to evaluate the inclusion in a convex polygon, a simple algebraic check could be done using the half-plane description. However, with a non-convex polygon, a numerical point-in-polygon algorithm must be used instead. As a matter of fact, the non convexity of the region also prevents the formulation of a Quadratic Program (QP) where the membership is utilized as a constraint, as in [1]. To obtain the optimum target on the boundary of $s\mathcal{Y}_e$, we perform the following search algorithm:

1) Create a line segment using two consecutive vertices of $s\mathcal{Y}_{far}$.

[1] In the accompanying video, the projected regions are illustrated at the feet level just for visualization purposes. However, the computation of the regions has been performed at the level of the CoM.

2) Find the point on the line segment which is closest to the heuristic target and compute the distance between them.

3) Repeat the procedure over all vertices and choose the point associated with the smallest distance as the optimum CoM target.

Furthermore, the scaling of a convex polygon can be performed through an affine transformation with respect to the Chebyshev center or the centroid (see [1]). For non-convex polygons, this problem is harder. Scaling a polygon with respect to a reference point could result in a scaled region with parts outside the original one. One solution is to use inward polygon offsetting. Although offsetting non-convex polygons is still a hard problem itself, [32] proposed an efficient solution for non-convex polygons. However, this is not yet fast enough for online planning and we have notices that, for this purpose, scaling the feasible region with respect to its centroid provides satisfactory results. The centroid $\bar{\mathbf{c}}_{xy}$ of the region (and any non-intersecting polygon defined by $n$ vertices [33]) can be computed as:[1]

$$\bar{\mathbf{c}}_{xy} = \frac{1}{6A} \sum_{i=0}^{n-1} (\boldsymbol{\nu}_i + \boldsymbol{\nu}_{i+1})(\boldsymbol{\nu}_i \times \boldsymbol{\nu}_{i+1}),$$

where $A$ is the polygon's signed area and is defined as

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (\boldsymbol{\nu}_i \times \boldsymbol{\nu}_{i+1}).$$

## 4.2 Optimization of trunk orientation to maximize joint range

Upon planning a CoM trajectory, our previous *heuristic* planning approach computes also 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 [5] via fitting an averaging plane through the stance feet.

[1] Computing the centroid simply as the arithmetic mean of the vertices would result in a point location that depends on the distribution of the vertices. In fact, with the changing number of vertices being generated by the IP algorithm, this can result in a discontinuously shifting centroid, therefore the computation with the arithmetic mean should be avoided.

This approach aims at bringing the legs as close as possible to the middle of their workspace in order to avoid the violation of 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 very rough terrains, where the feet are located on distant non coplanar surfaces (*e.g.,* like the one in Fig. 5.3), this might not be sufficient. In such cases, it can happen that some legs become more extended/retracted than others.

More importantly, this strategy does not take whole trajectory of the CoM into consideration. Indeed, aligning the trunk with the terrain inclination attempts to achieve an overall better configuration for the legs, but their specificity is not strictly considered: for a specified CoM trajectory, some legs can be more critical (*e.g.,* with some joints closer to their limits) than others. For instance, when moving the body toward a certain leg the opposite one will be stretching and more prone to hit the kinematic end-stops. This is especially significant for very rough terrains as will be illustrated in Section 5.2.

Given that the CoM trajectory lives in the same space as the feasible region, and examining the effect of the trunk orientation on the region in Section 3.5, we can exploit the region to guide the choice of the orientation that best encloses the whole CoM trajectory chosen in Section 4.1.

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 heuristic approaches that make sure that each leg is as close as possible to the middle of its workspace (*i.e.,* the default configuration).

To reduce the size of the problem, it is necessary to initialize the search space around some solution. As mentioned above, the heuristic-based orientation planning provides an elementary, yet satisfactory, behavior in many cases. Accordingly, we choose to sample the orientation space around the heuristic solution. 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 4.1 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 4.1 (computed at the default orientation) *before* optimizing for the orientation.

# Chapter 5

# Simulations

To demonstrate the capability of the proposed improved feasible region, we devised some challenging scenarios that the robot has to traverse, designed to best illustrate the region's features. Under such scenarios, we show the superior performance of a planning based on the improved feasible region compared to the default heuristics. The terrain templates are summarized in Table 5.1:

| **Name** | **Description** | **Test Type** |
|---|---|---|
| Template 1 | Ramp of 30° inclination with 50 cm high tunnel | Kin. limits / actuation limits. |
| Template 2 | Cobblestones with different heights and inclinations | Orientation Optim. |

Table 5.1: Rough terrain templates used to benchmark the locomotion strategy based on the feasible region.

The accompanying video of the experiments can be found here[1]. The generation of the projected regions is done in Python 2.7 with a computation time of 20 $ms$ (50 $Hz$ update rate) for the improved feasible region[2].

[1] www.dropbox.com/s/xnnb50f6e07frw2/tro20abdallah.mp4

[2] We expect a decrease in the computation time upon performing the computation in C++, *e.g.,* using Cython [34].

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 whole-body controller runs at 250 $Hz$.

## 5.1 Walk in cluttered environment

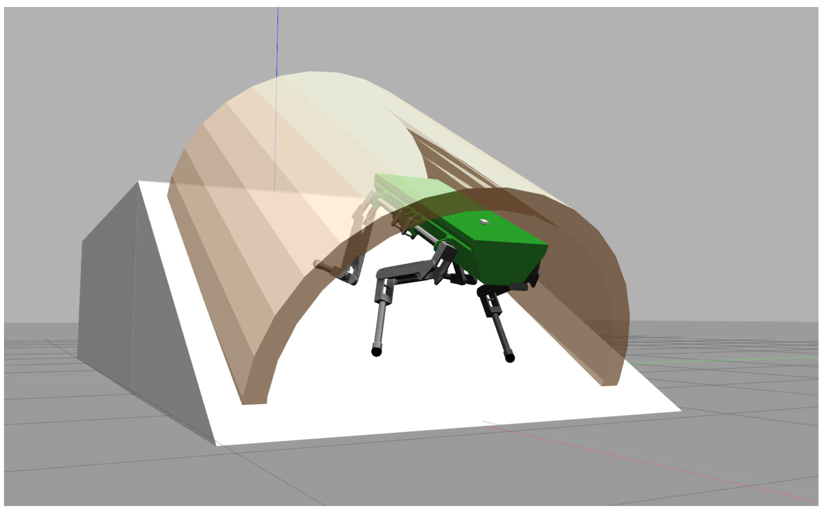
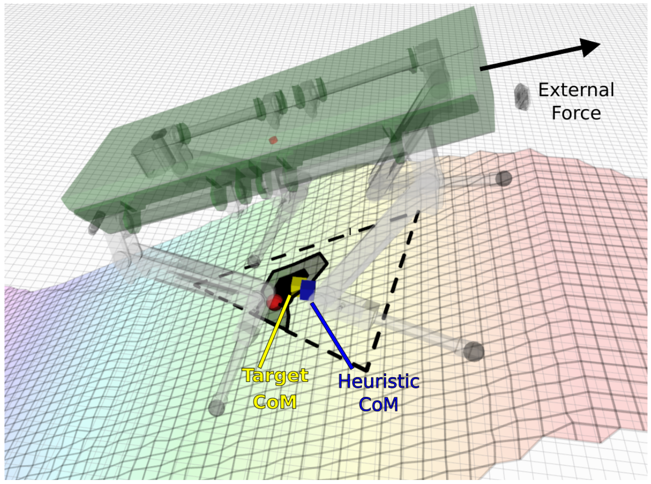


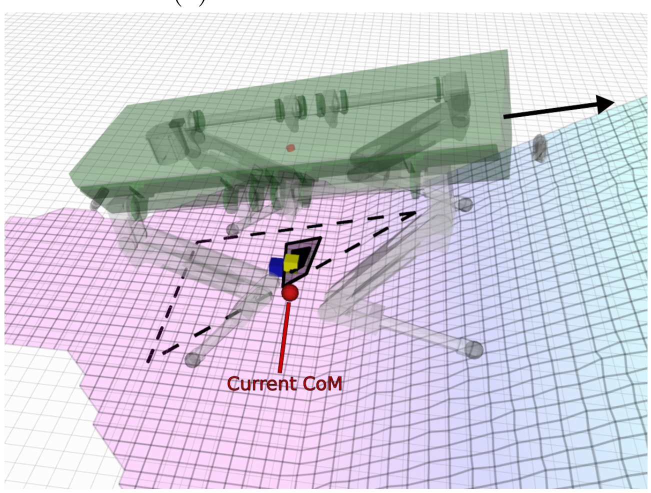
Figure 5.1: Simulation of HyQ descending down a challenging 30° ramp with a 50 $cm$ high declined tunnel (Template 1). The height of the HyQ robot is decreased from 53 $cm$ to 40 $cm$ in order to fit inside the tunnel. A controlled rope (not shown in the figure) is attached to the back of the robot's trunk to compensate for gravity.

In 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 in the HyQ robot descending a 30° ramp while being attached to a rope, to explore a low tunnel. This can be a typical scenario that a robot needs to face in oil rigs inspection assignments (see Fig. 5.1). A rope (not shown in the simulation software) connects the back of the robot to an anchor. The effect of the rope has been implemented in simulation as a constant external force compensating for the component




(a) Middle of the tunnel




(b) Bottom of the tunnel

Figure 5.2: 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), feasible regions (grey), and the scaled feasible regions (black). Cubes represent the projection of the CoM target based on the feasible region (yellow) and the heuristics (blue), on the projection plane. 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).

of the gravity force along the sagittal axis of the trunk, and applied at the back of the trunk (see Fig. 5.2). This results in regulated locomotion down the steep slope (*e.g.,* the same way a climber is *rappelling* down a wall)[1].

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 [15] (see Fig. 1.2). Indeed, in a slope with a very high inclination, the robot eventually creates a tangential force on the terrain that surpasses the friction force that is needed to prevent slippage. A rope can introduce an external force to compensate for gravity solving this issue and allowing the robot to walk on ideally any terrain inclination. We opted however to operate on an angle of not more than $30°$ just to avoid the problem of shin collision at the beginning of the ramp (although the proposed algorithm can also work for steeper inclines). 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 the default value of $53cm$ to $40cm$.[2] 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 tension coming from the rope.

The above-mentioned effects on the friction region (*i.e.,* support region in Bretl's terminology [14]) and on the feasible region can be seen in Fig. 5.2 for two instances in the simulation. The regions are computed on the plane fitted through the stance legs [5]. This is parallel to the plane expressed by the orientation of the trunk of the robot 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

[1]Experimentally, 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).

[2]The robot height is defined as the distance between the CoM and the terrain plane along its normal $\mathbf{n}$.

heuristics happens to be outside the region. Conversely, the CoM planner based on feasible region, computes a feasible target (yellow) that is on the boundary of the *scaled* feasible region and closest to the heuristic 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.

## 5.2 Optimization of the Trunk Orientation on very rough terrain

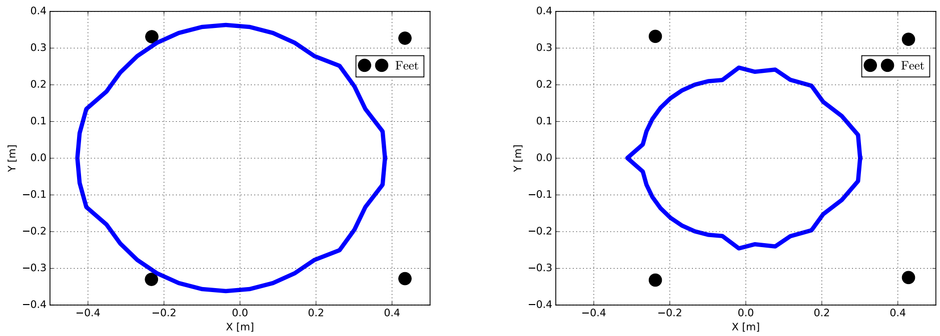
To illustrate the effectiveness of the orientation optimization strategy proposed in Section 4.2, we test it separately from the CoM planning strategy developed in Section 4.1. For this reason, the optimization of the orientation will be based on the CoM target computed by the heuristic approach. As mentioned before, even if this does not necessarily guarantee feasibility, it allows us to compare clearly the improvements of the orientation optimization over an orientation planning based on heuristics.

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 5.3 shows the resulting reachable regions for the two orientations for the robot standing on a 15° ramp.

In fact we can see that the reachable region for horizontal trunk orientation (Fig. 5.3 (b)) is smaller compared to the one where the trunk of the robot is aligned with the ramp (Fig. 5.3 (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 [25, 5] 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 advantages of a superior feasible region and consequently an ability to achieve a higher stability margin. The case of the very rough terrain shown in Fig. 5.4, is particularly

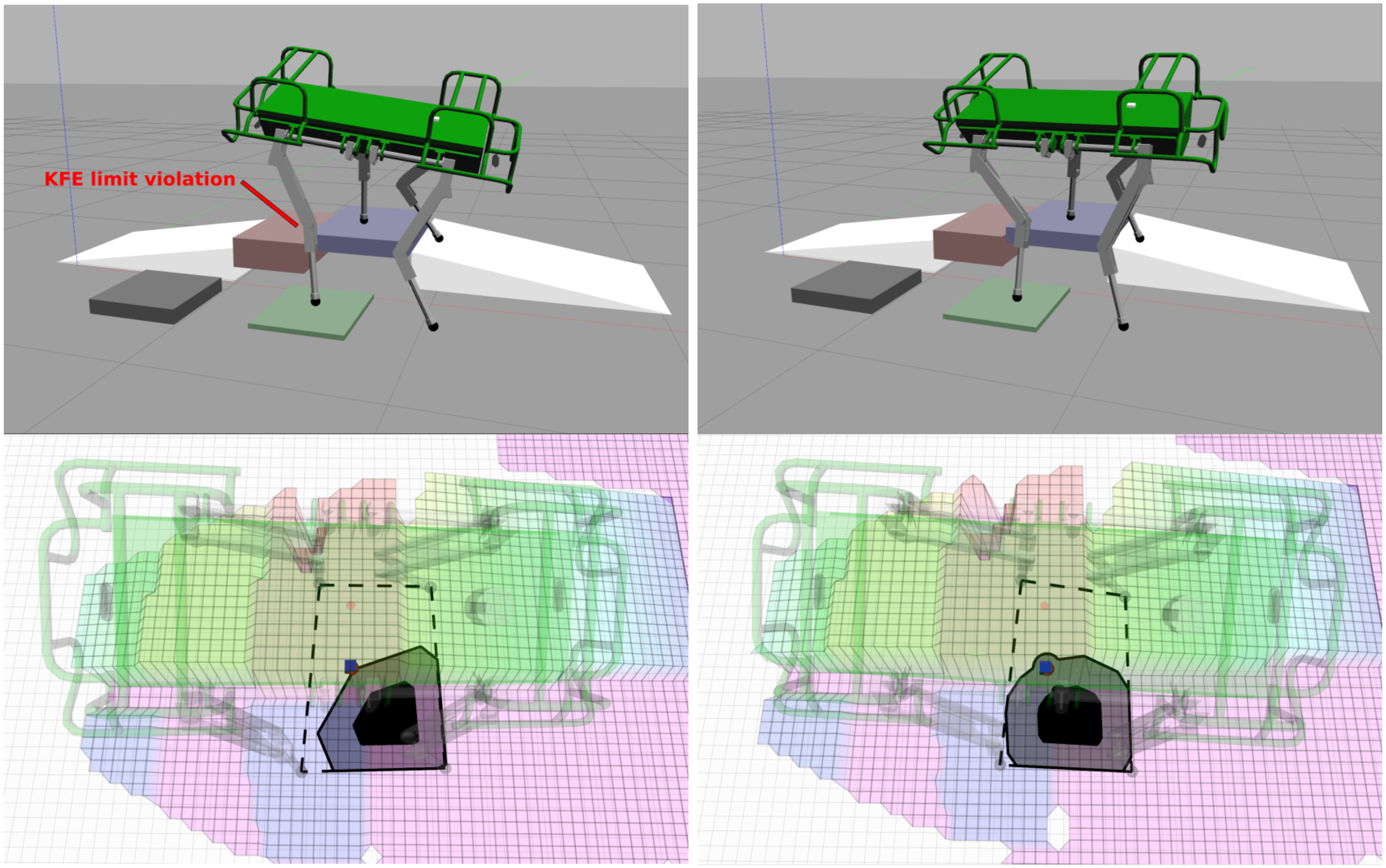



(a) Aligned with ramp (b) Horizontal configuration.

Figure 5.3: HyQ's reachable region on a 15° ramp with (a) the robot's trunk aligned to the ramp and with (b) horizontal trunk.

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 heuristics results in infeasible trajectories in multiple locations of the terrain (Fig. 5.4(a) bottom). The heuristic 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° ( equal to the estimated averaging terrain plane), is selected by the heuristic approach. This results in a hyper-extension of the RH leg and a kinematic violation at the Knee Flexion-Extension (KFE) joint (Fig. 5.4 (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 proposed improved feasible region[1], with an optimized pitch angle of $-0.3°$ (see Fig. 5.4 (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 joints' configuration.

[1]Due to the complexity of the terrain and the consequent complex robot configuration the improved feasible region presented a limited size. As a consequence it was not possible to ensure a bigger margin of robustness.




(a) Using Heuristics (b) Using Feasible Region

Figure 5.4: Simulation of HyQ forced near its kinematic limits while traversing a difficult non-coplanar terrain (Template 2). The configurations shown are at the end of a move-body phase. Realizing orientations based on (a) the heuristics and (b) based on the feasible region results in different leg configurations (top). (bottom) The resulting regions shown 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).

# Chapter 6

# Experiments

## 6.1 Walk with low height (army crawl)

To further validate the CoM planning strategy based on the feasible region, we carry out experiments with the real robot platform HyQ, focusing on the shown difficulty of walking with a reduced height. We have the robot walk at 0.03 $m/s$ with a low height of 43 $cm$. The plots of the KFE joint trajectory during the experiments are reported in Fig. 6.1. A CoM planning strategy based on the heuristics would results in multiple violations of the kinematic limits (upper plot) while the one based on the improved feasible region has no violation at all (lower plot). Additionally, Fig. 6.2 shows that such kinematic violations result in a deterioration of the tracking of the CoM trajectory. The region can also be used as a tool to check what is the optimal height that maximizes the robustness to disturbances (*i.e.,* represented by the area of the region).

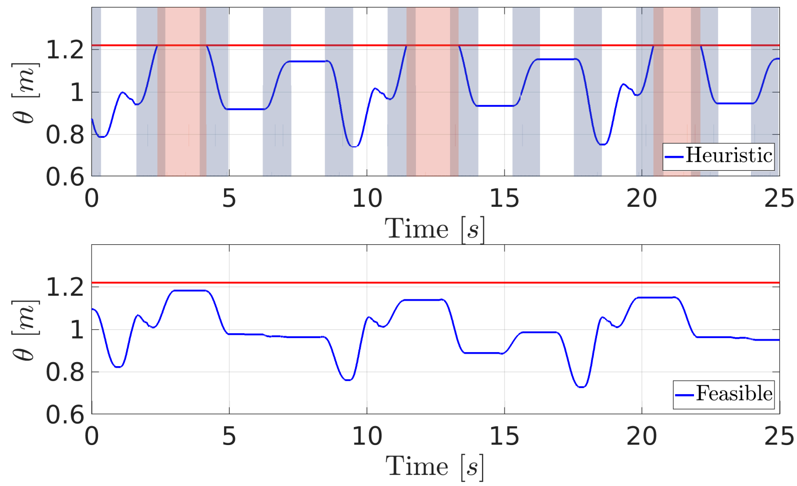



Figure 6.1: Experimental results showing the Left-Front KFE joint trajectory during a few crawl steps. Heuristic planner (above): the knee starts to hit the kinematic limit (red line) during the *move body* phases (shaded blue). The violations are in shaded red. Feasible region planner (bottom): no kinematic limit violations are observed.

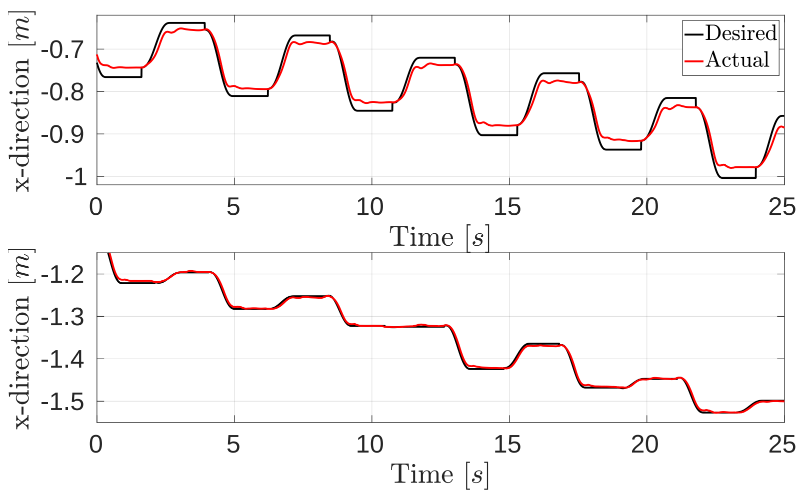



Figure 6.2: Experimental results showing the CoM position tracking in the $x$ direction. A deterioration can be seen with the heuristic planning (upper plot) due to the joint kinematic limit violations while good tracking is observed when the planning is based on the improved feasible region (bottom plot).

# Chapter 7

# Conclusions and future works

In this thesis, an improved version of the feasible region presented in previous work [1] is presented. The feasible regions are intuitive yet powerful and computational efficient tools to plan feasible trajectories for a reference point of the robot (*e.g.,* the CoM). The original feasible region, that was originally defined as the set of CoM positions where a robot is able to maintain static equilibrium without violating friction and actuation limits, was extended to take into account also kinematic limits and the presence of external wrenches acting on arbitrary points of the robot. This offers the possibility to employ the proposed motion planning strategy to new possible applications such as the mentioned rope-aided locomotion.

The consideration of the kinematic limits becomes crucial when big changes in the height and orientation of the robot are required that may push the robot to violate its kinematic limits.

We also generalized the computation of the region (originally defined in a plane perpendicular to gravity) to be projected on arbitrary plane inclinations to be consistent with the planning intention of the user. This allows to easily plan feasible trajectories on uneven terrains like when walking on ramps, climbing stairs, ladders, etc. To include the dynamic effects, the quasi-static assumption were relaxed in the iterative projection algorithm. To incorporate the feasibility of the kinematic limits, we introduced an algorithm that efficiently computes the reachable region of the robot's CoM that we intersect with the feasible region to obtain the improved feasible region. Furthermore, a planning strategy that utilizes the improved feasible region to design feasible CoM and trunk orientation trajectories was proposed. For this, we adopted a hierarchical approach that separates the planning of the CoM position and of the trunk orientation in two different sequential phases. We validated the capabilities of the feasible region and the the effectiveness of the proposed planning strategy on challenging simulations and experiments with the HyQ robot and we compared our results to a previously developed heuristic approach [5] that could not formally guarantee the feasibility of its trajectories. Instead, with a motion plan based on the feasible region all the feasibility constraints were formally verified because the projection of the CoM always lied inside the improved feasible region. The robustness of the approach could be simply tuned by a single scaling parameter of the region, adjusting the desired level of *cautiousness* one wants to achieve during the locomotion on complex geometry environments. Being able to adjust the robustness improves the quality of planning as it makes the controller more resilient to external perturbations.

As future works, focus on speeding up the computation of the region increasing its accuracy in the vicinity of the direction of motion is important. This would allow to only refine (or even only compute) the parts of the feasible region that are relevant to the direction of locomotion. Furthermore, a cost map on the region could be integrated that can then be used as a different metric than the distance from the boundaries used in section 4.1 to select the CoM target. As a matter of fact, not all the points of the reachable region have the same properties in terms of global manipulability: one generic point of the region the robot could be, for example, associated to a much lower mobility than other points in the same region.

# Bibliography


[1] R. Orsolino, M. Focchi, S. Caron, G. Raiola, V. Barasuol, and C. Semini, "Feasible region: an actuation-aware extension of the support region," *IEEE Transactions on Robotics (TRO)*, 2020.

[2] M. Neunert, M. Stauble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[3] J. Carpentier and N. Mansard, "Multicontact Locomotion of Legged Robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

[4] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on challenging terrain," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2554–2561, 2013.

[5] M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. G. Caldwell, , and C. Semini, "Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality," *Springer Track in Advanced Robotics series*, no. ECHORD++: Innovation from LAB to MARKET, pp. page143–183, 2020.

[6] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic Locomotion through Online Nonlinear Motion Optimization for Quadrupedal Robots," *IEEE Robotics and Automation Letters*, vol. 3766, no. c, pp. 1–1, 2018.

[7] 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)*, 2016, pp. 601–606.

[8] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," *IEEE International Conference on Intelligent Robots and Systems*, pp. 7440–7447, 2018.

[9] 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.

[10] S. Caron, Q.-c. Pham, and Y. 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.

[11] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3363–3370, 2018.

[12] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multi-contact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, Feb. 2017.

[13] J. E. Kelley Jr., "The Cutting-Plane Method for Solving Convex Programs," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[14] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, 2008.

[15] M. Bando, M. Murooka, S. Nozawa, K. Okada, and M. Inaba, "Walking on a Steep Slope Using a Rope by a Life-Size Humanoid Robot," *IEEE International Conference on Intelligent Robots and Systems*, pp. 705–712, 2018.

[16] H. Audren and A. Kheddar, "3D robust stability polyhedron in multi-contact," *Submitted to IEEE Transactions On Robotics (TRO)*, 2017.

[17] 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)*, 2016, pp. 49–56.

[18] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots," *Robotics Science and Systems (RSS)*, 2017.

[19] S. Tonneau, P. Fernbach, A. D. Prete, J. Pettré, and N. Mansard, "2pac: Two-point attractors for center of mass trajectories in multi-contact scenarios," *ACM Trans. Graph.*, vol. 37, no. 5, Oct. 2018.

[20] 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)*, 2018, pp. 1–9.

[21] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust Rough-Terrain Locomotion with a Quadrupedal Robot," *Proceedings - IEEE International Conference on Robotics and Automation*, no. February, pp. 5761–5768, 2018.

[22] R. Mattikalli, D. Baraff, and P. Khosla, "Finding all stable orientations of assemblies with friction," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 2, pp. 290–301, apr 1996.

[23] H. Mosemann, F. Rohrdanz, and F. M. Wahl, "Stability analysis of assemblies considering friction," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 805–813, 1997.

[24] D. E. Orin, A. Goswami, and S. H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, pp. 161–176, 2013.

[25] C. Gehring, C. D. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart, "Dynamic Trotting on Slopes for Quadrupedal Robots," *International Conference on Intelligent Robots and Systems*, 2015.

[26] B. Bona, *Dynamic Modelling of Mechatronic Systems.* CELID, 2018. [Online]. Available: https://books.google.it/books?id=gOn8ngEACAAJ

[27] J. P. Merlet, "Parallel Robots: J.P. Merlet." [Online]. Available: https://www.springer.com/gp/book/9781402003851

[28] C. Gosselin, "Determination of the Workspace of 6-DOF Parallel Manipulators," *Journal of Mechanical Design*, vol. 112, no. 3, pp. 331–336, 1990.

[29] J. P. Conti, C. M. Clinton, G. Zhang, and A. J. Wavering, “Workspace variation of a hexapod machine tool,” National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST IR 6135, 1998.

[30] O. Bohigas, M. Manubens, and L. Ros, “A Complete Method for Workspace Boundary Determination on General Structure Manipulators,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 993–1006, 2012.

[31] S. Chiaverini, G. Oriolo, and I. D. Walker, *Kinematically Redundant Manipulators*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 245–268.

[32] R. Wein, “Exact and approximate construction of offset polygons,” *Computer Aided Design*, vol. 39, no. 6, p. 518–527, Jun. 2007.

[33] G. Bashein and P. R. Detmer, *Centroid of a Polygon*. USA: Academic Press Professional, Inc., 1994, p. 3–6.

[34] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. Seljebotn, and K. Smith, “Cython: The best of both worlds,” *Computing in Science Engineering*, vol. 13, no. 2, pp. 31 –39, march 2011.