An Efficient Paradigm for Feasibility Guarantees in Legged Locomotion

Abdelrahman Abdalla[1], Michele Focchi[1], Romeo Orsolino[2] and Claudio Semini[1]

***Abstract*—Developing feasible body trajectories for legged systems on arbitrary terrains is a challenging task. In this paper, we present a paradigm that allows to design feasible Center of Mass (CoM) and body trajectories in an efficient manner. In our previous work [1], we introduced the notion of the 2D *feasible region*, where static balance and the satisfaction of joint torque limits were guaranteed, whenever the projection of the CoM lied inside the proposed admissible region. In this work we propose a general formulation of the *improved feasible region* that guarantees *dynamic* balance alongside the satisfaction of both joint-torque and kinematic limits in an efficient manner. To incorporate the feasibility of the kinematic limits, we introduce an algorithm that computes the *reachable region* of the CoM. Furthermore, we propose an efficient planning strategy that utilizes the improved feasible region to design feasible CoM and body orientation trajectories. Finally, we validate the capabilities of the improved feasible region and the effectiveness of the proposed planning strategy, using simulations and experiments on the 90 kg Hydraulically actuated Quadruped (HyQ) and the 21 kg Aliengo robots.**

***Index Terms*—planning, trajectory optimization, legged robots, locomotion, computational geometry, improved feasible region, reachable region**

I. 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, the demanding payloads, 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*, *joint-torque* and *kinematic* limits.

A powerful tool that is often utilized to devise feasible trajectories is numerical optimization. In recent years, the availability of increased computational power and the formulation of more efficient algorithms, allowed implementations of optimization-based approaches 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-torque 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 their small computational effort that enabled to easily implement them online on a real robot. 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 [6–8]. 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, constraints at the joint variables (*e.g.,* torque or kinematic limits) cannot be explicitly formulated in the planning problem (i.e. they lack descriptiveness).

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 joint-torque limits (defined at the joint level) to the 6D space where the centroidal wrench exists. These mappings result in 6D polytopes that represent the set of admissible wrenches for which the above-mentioned constraints are satisfied. 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 joint-torque limits are taken into account [11]. Enforcing the polytopes as constraints on the centroidal wrench (or accelerations) in a Trajectory Optimization (TO) problem results in feasible trajectories for the CoM. Unfortunately, despite the promising results, the introduction of the joint-torque limits made the computation prohibitively expensive. In fact, increasing the number of contacts dramatically increases the computation time. This makes these polytopes hard to compute *online* without accepting strong approximations on kinematics [11].

Another approach to address the problem of feasibility is to define a reference point[1] (henceforth we will consider the CoM, even though any other point can be chosen [12]) along

[1] The authors are with the Dynamic Legged Systems lab, Istituto Italiano di Tecnologia (IIT), Genova, Italy. name.surname@iit.it

[2] This work was carried out while Orsolino was at the Oxford Robotics Institute, University of Oxford, Oxford, UK. orso.romeo@gmail.com

[1] 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.

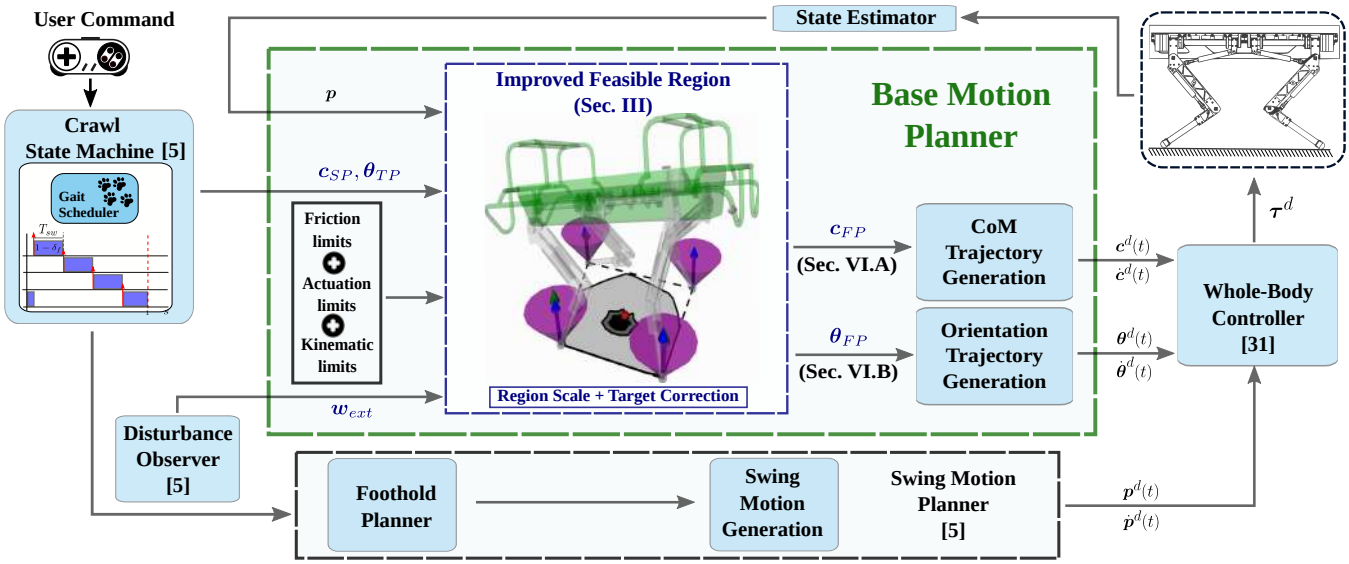


Fig. 1: Block diagram of our locomotion framework. The improved feasible region is a help for the planner to devise feasible robot postures.

with a 2D region in which the projection of the reference point must lie inside, in order to meet the requested feasibility conditions (*e.g.,* friction, joint-torque or kinematic).

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 reduced computational complexity. Similar to the CWC and FWP, the feasible region could be enforced as constraints on the CoM trajectory in a TO problem.

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 our previous work [1], we presented a modified version of the IP algorithm to compute the *feasible region*, a convex region where both friction and joint-torque 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 (15 ms on an i7-8700 3.2 GHz processor and 16GB of RAM). 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 in [1] during the computation of the feasible region: (1) the only external wrench acting on the robot is gravity, (2) 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 wrench to the robot; external wrenches usually cause a shift in the region as well as a change in its shape and size (as will be shown in section III-B). Therefore, any planning strategy based on this region would be inaccurate and can lead to unfeasible plans when external disturbances are applied. Such a feature is also needed when an external wrench is *intentionally* applied to the robot. This is the case when a load is pulled or when a rope is used for locomotion. In fact, having a feasibility metric that takes into account the effect of external wrenches would open many research opportunities in rope-aided locomotion and load-pulling applications. External disturbances are incorporated in an MPC in [15] and [16] to plan stable Zero Moment Point (ZMP) trajectories. The method, however, utilizes the more simplified Linear Inverted Pendulum (LIP) model and is not suitable for non coplanar contacts. Furthermore, the restricting effect of the joint-torque limits on the CoM planning, in the presence of counteracting disturbances, is not considered.

Assumption (2) limits the applicability of the feasible region to quasi-static gaits. If applied to more dynamic gaits, having a trajectory computed under a statically stable assumption may induce falling due to the changes in the velocity of the robot. Recently, Audren et al. [17] incorporated the dynamics, proposing a robust static stability region that accounts for possible CoM accelerations. No other feasibility measures such as joint torque and kinematic limits were considered. In contrast, Nozawa et al. [18] compute a dynamic stability region for the CoM based on specified linear and angular accelerations. In both approaches, however, only friction guarantees were considered in the regions.

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

orientation may push the robot to violate its kinematic limits. In this respect, the seminal work of Carpentier et al. [19] focused on incorporating the kinematic constraints via learning proxy constraints. On a similar line, [20, 21] constrain the position of the CoM with respect to the contact points, however, these kinematic constraints are only approximated to maintain the convexity of the problem, thus only valid for a simplified representation of the robot. More recently, Fankhauser et al. [22] optimized the orientation to ensure static stability and kinematic limits, by solving a non-linear optimization problem (SQP). The kinematic limits were roughly approximated by setting bounds on the leg length. An SQP problem is also utilized in [18] 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 [23, 24] present a way to find all the orientations that satisfy static stability. Yet, the objects were fixed and not actuated.

A. 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 we:

leftmargin=*

- Generalize the feasible region to account for the effect of external wrenches (see Section III-B). Unlike in [15] and [16], the centroidal dynamics model is used and we consider external forces and torques acting on arbitrary points of the robot.
- We relax the quasi-static assumption by considering the dynamic effects, as well as the angular dynamics (see Section III-C). Differently from [17] 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.
- Generalize the feasible region to be defined on arbitrary plane inclinations (see Section III-A).
- Embed the complete joint-kinematic limits in what we define as the *reachable region*. This presents a more accurate representation of the CoM kinematic capability than the approximations performed in [20–22]. Furthermore, the region does not need to be relearnt for different robots as in [19]. The region can be intersected with the joint-torque aware region (with the aforementioned extensions) and leads to the so-called *improved feasible region* that considers friction, joint-torque and kinematic limits.
- Design a *robust* CoM planning strategy that utilizes non-convex regions and propose a new optimization for the trunk orientation based solely on the improved feasible region. 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.
- Show simulations and hardware experiments with robots 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]. The experimental results are shown on both the 90 kg HyQ robot (hydraulically actuated) and the 21 kg Aliengo (electrically actuated) robot [25]. Feasibility of dynamic motions are validated by the feasible region in experiments with the Aliengo robot.

B. Outline

The paper is organized as follows: in Section II we recall the modified IP algorithm used to compute the feasible region [1], while in Section III we present the extensions to compute the feasible region. Section IV introduces the *reachable region* and Section V defines the intersection of the *feasible region* and the *reachable region* to define the *improved feasible region*. Section VI illustrates the planning strategies for the CoM and the orientation based on the region. Section VII summarizes the assumption made in this work. Simulations and experimental results with HyQ and Aliengo robots are presented in Section VIII and IX. Section X draws the conclusions and discusses possible future developments. Finally, the Appendix includes additional information about the reachable region and its effect on planning.

II. Recap on classical feasible region

For a better understanding of the proposed improved feasible region, let us first briefly recap the *feasible region* presented in [1]. This region was generated using an IP algorithm described in Algorithm 1 (in black). The extension of the region to generate the improved feasible region are marked in blue and described in detail in Section III.

The algorithm considers the convex constraints imposed 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, we considered the static stability constraints (III.a), frictional constraints on the contact feet (III.b), and the joint-torque constraints (III.c).

The solution of each LP problem, $\mathbf{c}^*_{xy}$, is an extremal CoM position along a certain direction (represented by the unit vector $\mathbf{a}_i$), that still satisfies the constraints, *i.e.,* a vertex on the boundary of the feasible region. This optimization is performed iteratively along various directions $\mathbf{a}_i$ that span along a circle, building the inner approximation of the region as the convex hull of all the solutions $\mathbf{c}^*_{xy}$ (see Fig. 2).

Constraint (III.a) ensures the static balance of the robot (force and moment balance). $\mathbf{A_1} \in \mathbb{R}^{6\times mn_c}$ is the grasp matrix of the n_c contact points $\mathbf{p}_i \in \mathbb{R}^3$ and m depends on the nature of the contact (*i.e.,* $m = 3$ for point contact, $m = 6$ for full contact). $\mathbf{A_1}$ is summing up the contact wrenches (pure forces in case of point feet) $\mathbf{f} \in \mathbb{R}^{mn_c}$ and is expressing them at the origin of the world frame. $\mathbf{u} \in \mathbb{R}^6$ is the linear part of the wrench due to gravity force (acting on the CoM) and

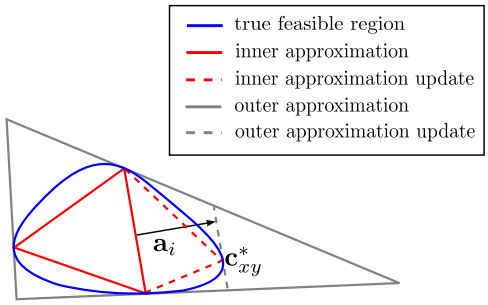


Fig. 2: Iteration of the IP algorithm: after the LP is solved finding a new extremal $\mathbf{c}^*_{xy}$ point along $\mathbf{a}_i$, this is added to the *inner* approximation while an edge with normal $\mathbf{a}_i$ passing through $\mathbf{c}^*_{xy}$ is added to the *outer* approximation [1].

Algorithm 1 Feasible Region IP algorithm (with external 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:

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

such that :

$$\text{(III.a)} \quad \mathbf{A}_1\mathbf{f} + \mathbf{A_2}\mathbf{c}_{xy} = \mathbf{u}$$
$$\text{(III.b)} \quad \mathbf{B}\mathbf{f} \leq \mathbf{0}$$
$$\text{(III.c)} \quad \mathbf{G}\mathbf{f} \leq \mathbf{d}$$

IV) update the outer approximation $\mathcal{Y}_{outer}$
V) update the inner approximation $\mathcal{Y}_{inner}$
end while

$\mathbf{A_2}$ computes the angular component of the gravity wrench, whenever this is expressed at the origin of the world frame:

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

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

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

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

Constraint (III.b) ensures the friction constraints are met. These require the contact forces to be inside inner pyramidal (conservative) approximations of the friction cones.

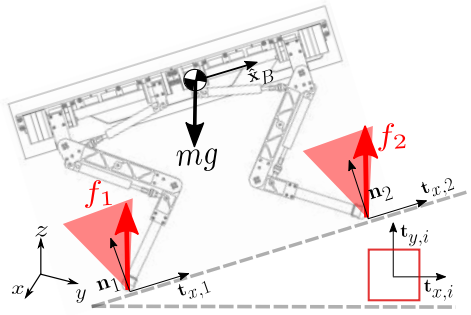


Fig. 3: Friction pyramids (shown in red) orientation with respect to the contact surface. Each pyramid base is perpendicular to the contact surface normal $\hat{\mathbf{n}}_i$. A top view of the pyramid base with respect to the tangent contact axes $\hat{\mathbf{t}}_{x,i}, \hat{\mathbf{t}}_{y,i}$ is shown (bottom right).

Approximating the friction cones with a low number of linear approximations results in a smaller computation time [26–28]. The number of edges chosen to represent the pyramid with reasonable accuracy can be chosen based on the complexity of the terrain and the friction coefficient. For each contact, we can define an orthonormal reference frame composed of the contact surface normal $\hat{\mathbf{n}}_i \in \mathbb{R}^3$, and tangent vectors $\hat{\mathbf{t}}_{x,i}, \hat{\mathbf{t}}_{y,i} \in \mathbb{R}^3$ such that $\hat{\mathbf{t}}_{x,i} = \hat{\mathbf{n}}_i \times \hat{\mathbf{x}}_B \times \hat{\mathbf{n}}_i$, where $\hat{\mathbf{x}}_B$ is the unit vector along the X-axis of the base of the robot. Each pyramid is oriented along $\hat{\mathbf{n}}_i$ (with the base of the pyramid parallel to the contact surface) as shown in Fig. 3. The constraint matrix $\mathbf{B} \in \mathbb{R}^{4n_c\times 3n_c}$ can then be represented as:

$$\mathbf{B} = \operatorname{diag}(\mathbf{b}_1, \dots, \mathbf{b}_{n_c}),$$
$$\mathbf{b}_i = \begin{bmatrix}(\hat{\mathbf{t}}_{x,i} - \mu_i\hat{\mathbf{n}}_i)^T \\ (\mathbf{t}_{y,i} - \mu_i\hat{\mathbf{n}}_i)^T \\ -(\hat{\mathbf{t}}_{x,i} + \mu_i\hat{\mathbf{n}}_i)^T \\ -(\hat{\mathbf{t}}_{y,i} + \mu_i\hat{\mathbf{n}}_i)^T\end{bmatrix} \in \mathbb{R}^{4\times 3} \quad (2)$$

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 joint-torque 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 ith leg, the half-plane description of such force polytopes is represented by $\mathbf{G} \in \mathbb{R}^{2n_ln_c\times mn_c}$ and $\mathbf{d} \in \mathbb{R}^{2n_ln_c}$:

$$\mathbf{G} = \operatorname{diag}\left(\begin{bmatrix}\mathbf{J}(\mathbf{q}_1)^T \\ -\mathbf{J}(\mathbf{q}_1)^T\end{bmatrix}, \dots, \begin{bmatrix}\mathbf{J}(\mathbf{q}_{n_c})^T \\ -\mathbf{J}(\mathbf{q}_{n_c})^T\end{bmatrix}\right), \mathbf{d} = \begin{bmatrix}\mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{n_c}\end{bmatrix} \quad (3)$$

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 $\bar{\boldsymbol{\tau}}$). 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.

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

The consequence of this is that the feasible region can be considered to be accurate only in a neighborhood of the considered robot configuration. Therefore, for every change in the CoM position due to a change in the joint configuration, the feasible region should be recomputed.

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

$$\mathcal{Y}_{fa} = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{f}_i \in \mathbb{R}^{mn_c}, \text{ s.t. } (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A} \Big\} \tag{4}$$

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

$$\mathcal{C} \cap \mathcal{A} = \Big\{ \mathbf{f}_i \in \mathbb{R}^{mn_c}, \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{u} \\ \mathbf{B}\mathbf{f} \leq \mathbf{0}, \quad \mathbf{G}\mathbf{f} \leq \mathbf{d} \Big\} \tag{5}$$

III. Feasible Region Extensions

In this section we propose an extension of the feasible region to arbitrary plane inclinations (Section III-A). We then proceed to incorporate external wrenches (Section III-B), and dynamic effects (Section III-C). The changes on the algorithm are highlighted in blue in Algorithm 1.

A. 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.

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

For instance, for a robot climbing a ramp, the planned CoM trajectory can be expected to follow the inclination of the ramp [5][29]. 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.[2]

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

$$\mathbf{c} = {}^{\mathcal{W}}\mathbf{R}_{\Pi} \hat{\mathbf{c}} \tag{6}$$

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

$${}^{\mathcal{W}}\mathbf{R}_{\Pi} = \big[\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\big] \tag{7}$$

The $\hat{z}$-axis of Π is aligned with $\hat{\mathbf{n}}$. $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ are unit vectors (expressed in $\mathcal{W}$ frame and forming the $\hat{x}, \hat{y}$-axes of Π frame) chosen such that they form, together with $\hat{\mathbf{z}}$, a right-handed coordinate system. With the change of coordinates in (6), the IP algorithm can be written in terms of $(\hat{\mathbf{c}}_{\hat{x}\hat{y}}, \hat{c}_{\hat{z}})$ and solved for the new coordinates $\hat{\mathbf{c}}_{\hat{x}\hat{y}}$. In the remainder of this manuscript, not to overload the notation, we express the CoM position in the world frame $\mathbf{c}$ in all related equations, without any loss of generality.

B. External wrenches

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

$$\sum_{i=1}^{n_c} \mathbf{f}_i + m\mathbf{g} + \boldsymbol{f}_{ext} = 0 \tag{8}$$

$$\sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i - (m\mathbf{g} + \boldsymbol{f}_{ext}) \times \mathbf{c} + \boldsymbol{\tau}_{ext} = 0 \tag{9}$$

As mentioned in the previous section, with only the gravity $\mathbf{g}$ acting on the robot, the dependence on the CoM in (9) 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:

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{0} \\ -[m\mathbf{g} + \boldsymbol{f}_{ext}] \times \mathbf{P}_{xy}^T \end{bmatrix} \in \mathbb{R}^{6\times 2} \\ \mathbf{u} = \begin{bmatrix} -m\mathbf{g} - \boldsymbol{f}_{ext} \\ [\boldsymbol{f}_{ext}] \times \mathbf{P}_z^T c_z - \boldsymbol{\tau}_{ext} \end{bmatrix} \in \mathbb{R}^{6\times 1} \tag{10}$$

[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.

[2]Note that the projection the IP algorithm is in fact a mapping of the high dimensional constraints from the wrench space (or in the case of the kinematic constraints, the joint space) to a Euclidean plane. The Euclidean plane can be chosen to be expressed with respect to the frame of our choice as explained above.

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 (9), $\mathbf{c}_{xy}$ can be determined as [12]:

$$\begin{aligned}\mathbf{c}_{xy} = & \frac{1}{-mg + f_{ext,z}} \Bigg(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i - c_z \mathbf{f}_{ext,xy} \\ & + \begin{bmatrix} -\tau_{ext,y} & \tau_{ext,x} \end{bmatrix}^T \Bigg) \\ & = -\mathbf{h}(\mathbf{f}, f_{ext,z}) + \mathbf{m}(\boldsymbol{f}_{ext}, \boldsymbol{\tau}_{ext}, c_z) \end{aligned} \tag{11}$$

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

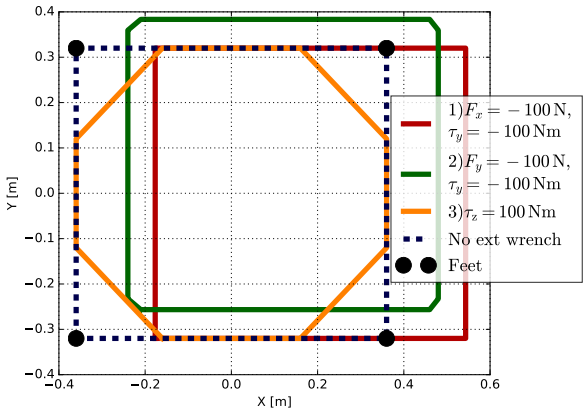
The change in shape of the region, can be intuitively understood, considering that the set of contact forces resulting from the action of the external wrench, could become infeasible due to the additional effort needed to compensate for the external wrench. For example, in case of a significantly retracted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

Figure 4 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 wrench. A reduction in the size of the friction region (*e.g.,* obtained only considering friction constraints (III.b)) can also be seen for an external torque $\tau_{ext,z}$ (orange). This is illustrated by the clipping of the corners of the region, where no admissible set of contact forces could withstand such external wrench without slipping.

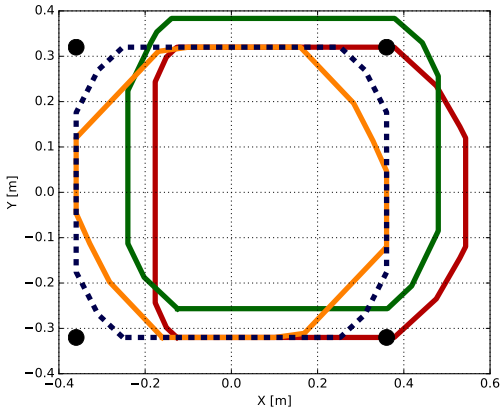
C. Dynamic Motions

To ensure stability/feasibility, it is necessary that the chosen reference point remains inside the admissible region that was computed for it. To evaluate dynamic stability, it is common to consider the ZMP as specified reference point. Because the ZMP already explicitly considers the horizontal acceleration of the robot's body, this does not have to be considered in the computation of the admissible region: this region therefore can

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



(a) Friction Region (only friction considered)



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

Fig. 4: Effect of different external wrenches acting on the CoM on the (a) friction region and the (b) feasible region. Changes in size and shifting of the location of the regions can be observed. The components of the external wrench that are mentioned are applied simultaneously and the unmentioned components are set to zero. The stance feet of HyQ are shown as black points with the front feet facing right. Regions are computed for a trunk height of $c_z = 0.53m$

be obtained for dynamic conditions and, on flat terrains, (if only friction cone constraints are considered) it simplifies to the convex hull of the contact points. Therefore, we underline that the choice of a reference point and its admissible region are tightly coupled and that any arbitrary reference point could be used provided that the employed admissible region is specifically formulated in accordance to it. As long as this point is inside the corresponding computed region, we are sure that the constraints that have been considered when building the region, are satisfied. Therefore, conforming to the previous sections, we keep using the CoM as the reference point and proceed to incorporate the dynamic effects (dropping the static assumptions) in the feasible region (constraints III.a in Algorithm 1). In fact, it could even happen that the ZMP is outside of the computed region, yet dynamic stability is ensured and the robot configuration is feasible as long as the CoM projection is inside it.

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

$$\begin{cases} m\left(\ddot{\mathbf{c}} - \mathbf{g}\right) = \sum_{i=1}^{n_c} \mathbf{f}_i \\ \mathbf{I_G}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I_G}\boldsymbol{\omega} + \mathbf{c} \times m\left(\ddot{\mathbf{c}} - \mathbf{g}\right) = \sum_{i=1}^{n_c} \mathbf{p}_i \times \mathbf{f}_i \end{cases} \tag{12}$$

where $I_G \in \mathbb{R}^{3\times3}$ is the moment of inertia about the center of mass, $\ddot{\mathbf{c}}$ the CoM Euclidean acceleration, and $\dot{\boldsymbol{\omega}}, \boldsymbol{\omega}$ the angular acceleration and velocity of the robot base, respectively. By inspecting (12) one can see that to incorporate the dynamic effects, the matrix $\mathbf{A}_1$ remains unchanged while $\mathbf{A}_2$ and $\mathbf{u}$ in constraint (III.a) should be redefined as:

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{0} \\ -m\left(\mathbf{g} - \ddot{\mathbf{c}}\right) \times \mathbf{P}_{xy}^T \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} m\left(\ddot{\mathbf{c}} - \mathbf{g}\right) \\ \mathbf{I_G}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I_G}\boldsymbol{\omega} \end{bmatrix}, \tag{13}$$

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 constraint (III.a) can be considered to be fully *dynamic*. As a result of the effect of the inertial accelerations, the computed region can "move" (*e.g.,* forward or backward) according to the direction of the instantaneous body acceleration (see accompanying video at 0:20 and 04:52). With the dependence of the feasible region on the acceleration of the robot, one can utilize the desired body accelerations in the computation of the region to plan dynamically feasible motions.

D. Degenerate Feasible Regions

It is possible to further extend the feasible region to dynamic gaits in quadrupeds (*e.g.,* a trot or 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. As a result, the possible solution space becomes infeasible in the absence of contact moments. This extension of the feasible region to degenerate cases is made numerically possible by assuming the presence of infinitesimal contact torques at the feet as constraints on the problem to render it feasible. In particular, we assume that the feet can exert a small torque component tangential to the contact surface plane τ_x and τ_y, but no contact torque orthogonal to the plane τ_z. This corresponds to the case of feet with a small non-zero surface, able to adjust the location of the Center of Pressure (CoP) within the contact surface. Such assumption is a numerical (heuristic) assumption introduced solely for the feasibility of the problem. We include these wrench components in the constraint (III.b) of Algorithm 1: we update the matrix $\mathbf{B}$ in (2) to embed, for each contact i, not just the constraints on the contact forces (*i.e.,* linearized friction cone constraint $\mathbf{b}_i^{cone} \in \mathbb{R}^{4\times3}$) but also a box constraint $\mathbf{b}_i^{box} \in \mathbb{R}^{4\times2}$ on the contact torques τ_x, τ_y. The values $\tau_x^{lim}, \tau_y^{lim}$ represent the infinitesimal limits of the box constraint on the contact torque tangential to the surface plane in the foot location:

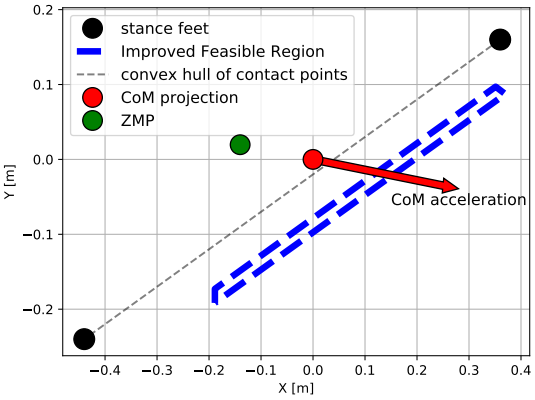


Fig. 5: The improved feasible region degenerates to a line during a trot, when only two feet are simultaneously in contact with the ground. This segment is shifted forward in the same direction of the robot's acceleration. The finite width of the improved feasible region (blue) is due to the infinitesimal contact torques τ_x^{lim} and τ_y^{lim}. The robot is dynamically unstable in this scenario.

$$\mathbf{b}_i^{cone} = \begin{bmatrix} \left(\mathbf{t}_{1,i} - \mu_i \mathbf{n}_i\right)^T \\ \left(\mathbf{t}_{2,i} - \mu_i \mathbf{n}_i\right)^T \\ -\left(\mathbf{t}_{1,i} + \mu_i \mathbf{n}_i\right)^T \\ -\left(\mathbf{t}_{2,i} + \mu_i \mathbf{n}_i\right)^T \end{bmatrix}, \quad \mathbf{b}_i^{box} = \begin{bmatrix} \tau_x^{lim} & 0 \\ 0 & \tau_y^{lim} \\ -\tau_x^{lim} & 0 \\ 0 & -\tau_y^{lim} \end{bmatrix}$$

$$\mathbf{B} = \mathrm{diag}\left(\begin{bmatrix} \mathbf{b}_1^{cone} & \mathbf{0}_{4\times2} \\ \mathbf{0}_{4\times3} & \mathbf{b}_1^{box} \end{bmatrix} \cdots \begin{bmatrix} \mathbf{b}_{n_c}^{cone} & \mathbf{0}_{4\times2} \\ \mathbf{0}_{4\times3} & \mathbf{b}_{n_c}^{box} \end{bmatrix} \right) \in \mathbb{R}^{8n_c\times5n_c} \tag{14}$$

Because of the non-zero values of the contact torque limits τ_x^{lim} and τ_y^{lim}, the feasible region portrayed in Fig. 5 appears as a narrow stripe with finite area, although it should be regarded as a one-dimensional segment. Indeed, the feasible region in this double point-contact case corresponds to a segment whose length is determined by the robot's actuation limits. In presence of external wrenches acting on the platform, this segment will move away from the line connecting the two feet along the projection plane.

In case of a single point contact, the feasible region will degenerate to a point which represents the only possible value of CoM projection where the robot could balance the load acting on its trunk. Note that if the dynamic effects are considered, the feasible line will move back/forth when the robot accelerates backwards/forward, according to what is explained in Section III-C. This is exemplified in Fig. 5 which shows the feasible region during a trotting motion. The region is a straight segment and is shifted forward with respect to the supporting line, because the robot is accelerating forward. The ZMP (green point), instead, moves backwards in the opposite direction to the acceleration. In future works, we plan to exploit this to perform fast turning maneuvers to check the maximum feasible sideways inclination that can be achieved (*e.g.,* to compensate centrifugal forces).

IV. Reachable Region

So far the feasible region was defined as a region for which the frictional stability of the robot can be ensured without violating the joint-torque limits. The inclusion of the effect

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

We, therefore, seek to extend the definition of the *feasible region* to further incorporate the joint-kinematic limits and the manipulability of the robot. We first introduce the *reachable region*, a two-dimensional level area representing the CoM reachable workspace. We present a simplified numerical approach that computes a conservative approximation of the region. The method is designed to be *efficient* and therefore allows for *online* motion planning and optimization. Given a desired orientation, we determine the *constant orientation workspace*: namely, the set of all possible CoM locations that can be reached with a specified orientation without violating the joint-kinematic limits [30]. 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:

$$^{\mathcal{B}}\mathbf{x}_{f_i} = f_i(\mathbf{q}_i), \quad \forall i = 1, ..., n_c \tag{15}$$

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

$$^{\mathcal{B}}\mathbf{x}_{f_i} = {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}(^{\mathcal{W}}\mathbf{x}_{f_i} - \mathbf{c}) + {}^{\mathcal{B}}\mathbf{c} \tag{16}$$

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 (15) and (16) and rewriting for $\mathbf{c}$, we obtain:

$$\mathbf{c} = \mathbf{F}_i(\mathbf{q}_i, {}^{\mathcal{W}}\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}), \quad \forall i = 1, ..., n \tag{17}$$

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

$$\mathbf{F}_i(\mathbf{q}_i, {}^{\mathcal{W}}\mathbf{x}_{f_i}, {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}) = {}^{\mathcal{W}}\mathbf{x}_{f_i} - {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}(f_i(\mathbf{q}_i) - {}^{\mathcal{B}}\mathbf{c}) \tag{18}$$

Therefore, for a given foot position $^{\mathcal{W}}\mathbf{x}_{f_i}$ and trunk orientation $^{\mathcal{W}}\mathbf{R}_B$, (17) provides a relationship between the joint-space angles of each leg and the CoM task-space position. We assume the feet do not move during contact. This is enforced by the Whole-Body Control (WBC) used in our framework (see Fig. 1) [31]. Therefore, for a CoM position $^{\mathcal{W}}\mathbf{x}_{com}$ to be reachable, there must exist joint angles $\mathbf{q}_i$, satisfying (17), 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 (17) (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) defined above, to define the *reachable region* as:

$$\mathcal{Y}_r = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{q}_i \in \mathbb{R}^{n_l} \text{ s.t.} \quad (\mathbf{c}_{xy}, \mathbf{q}_i) \in \mathcal{Q} \Big\} \tag{19}$$

where:

$$\mathcal{Q} = \Big\{ \mathbf{q}_i \in \mathbb{R}^{n_l}, \mathbf{c}_{xy} \in \mathbb{R}^2 | \text{ s.t.} \quad \mathbf{c}_{xy} = \mathbf{P}_{xy}\mathbf{F}_i(\boldsymbol{q}_i), \\ \underline{\boldsymbol{q}}_i \leq \boldsymbol{q}_i \leq \bar{\boldsymbol{q}}_i, \quad \text{row-rank}(J_i(\mathbf{q}_i)) = n_l \quad \forall i = 1, ..., n_c \Big\} \tag{20}$$

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 [32]. 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 [33], and was used to evaluate the workspace of a Stewart platform based machine tool. We further apply a modification to increase the robustness and the performance. Inspired by ray-casting algorithms, a discretized search is done iteratively in ordered directions along polar coordinates (ρ, θ) starting from the current CoM projection. This generates a 2D polygon whose vertices are ordered and belong to the boundary of the reachable region, therefore representing a *polygonal* approximation of the said region. For the sake of simplicity, for the remainder of this paper, 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 still belongs to the region. By construction, this point belongs to the boundary of the region and the problem of computing it can be stated, utilizing the inverse kinematics,

as:

$$\max_{\boldsymbol{\nu}_{xy}} \ \mathbf{a}_i^T \boldsymbol{\nu}_{xy} \tag{21}$$

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

$$\mathbf{q}_i = \bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy}) \tag{22}$$

$$\underline{\mathbf{q}}_i < \mathbf{q}_i < \bar{\mathbf{q}}_i \tag{23}$$

$$\sigma_{\min}\left\{J\left(\mathbf{q}^k\right)\right\} > \sigma_0 \tag{24}$$

The relation (22) represents the kinematic constraint in (20) reformulated in terms of the inverse kinematics. $\bar{\mathbf{F}}_i$, therefore, is defined as:

$$\bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy}) = f_i^{-1}[{}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}({}^{\mathcal{W}}\mathbf{x}_{f_i} - \mathbf{P}_{xy}^T\boldsymbol{\nu}_{xy} - \mathbf{P}_z^T c_z) + {}^{\mathcal{B}}\mathbf{c}] \tag{25}$$

where f_i^{-1} refers to the inverse kinematics mapping. It is important to note from (25) 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 ${}^{\mathcal{W}}\mathbf{R}_B$ of the robot. A simple check for the presence of a singularity is done in (24), where $\sigma_{\min}$ is the smallest singular value and σ_0 is a small value of choice. Due to the non-linearity of constraints (22) and (24) 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 is within an error of [0, $-\Delta\rho_{\text{min}}$] from the boundary of the actual workspace. The function *isReachable(ρ)*, used in Algorithm 2, computes the inverse kinematics of a CoM position and checks if that position is reachable:

isReachable(ρ):

$\boldsymbol{\nu}_{xy} \leftarrow \mathbf{c}_{xy} + \rho\mathbf{a}$

$\mathbf{q_i} = \bar{\mathbf{F}}_i(\boldsymbol{\nu}_{xy})$

return true if $\mathbf{q_i}$ satisfies (23) & (24)

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. 6 and 7). The algorithm stops when a step smaller than $\Delta\rho_{min}/2$ set by the user, is reached[1][2].

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 to start the algorithm from. Moreover, this provides better boundary precision when determining the boundary of the region that is closer to the CoM position, presenting a safer analysis. As a consequence, the dependence of the algorithm from $\mathbf{c}_{xy}$, 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 VI-A), the trajectory of the CoM shall always remain inside the region. The Appendix provides a further discussion of the nature of the CoM workspace and the reachable region.

[1]$\Delta\theta = 10^\circ$ and $\Delta\rho_{\text{min}} = 0.03m$ are used for the shown figures. For a faster computation during planning, $\Delta\theta = 20^\circ$ was sufficient for the simulation and experimental results.

[2]The large peaks in the edges of the reachable regions in Fig. 6 and 7 are due to the physical nature of the workspace. Smaller rough edges along the boundary of the regions are due to the discretization used.

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 2π **do**
  Compute direction: $\mathbf{a}_i = \begin{bmatrix}\cos\theta & \sin\theta & 0\end{bmatrix}^T$

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

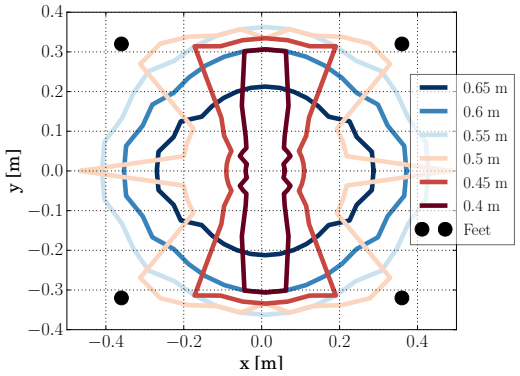
  Bisection search:
  $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$
  while $\Delta\rho \geq \Delta\rho_{\text{min}}/2$ **do**
    if *isReachable(ρ)* **then**
      $\rho \leftarrow \rho + \Delta\rho$
    else
      $\rho \leftarrow \rho - \Delta\rho$
    end if
    $\Delta\rho \leftarrow \frac{\Delta\rho}{2}$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

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

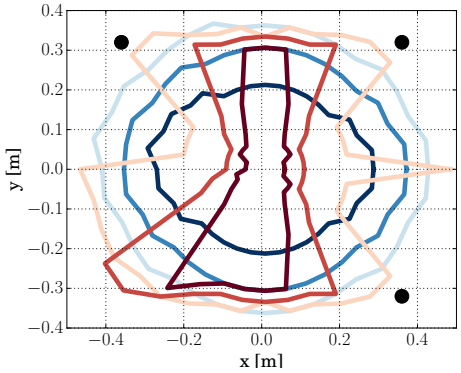
On the other hand, it is important also to consider the effect of the robot height c_z and orientations ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}$ on the reachable region. In fact, different evaluations of the reachable region, presented in Fig. 6 and 7, show that the size, positioning, shape, and convexity 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 the region can be significantly affected. Such insight is greatly useful in situations where planning needs to be performed in rough terrains.

V. The Improved Feasible Region

The reachable region can be seen as a projection of the high-dimensional convex set $\mathcal{Q}$ onto a 2D subspace. Henceforth,



(a) Four stance feet.



(b) Three stance feet.

Fig. 6: Different evaluations of the reachable region at different HyQ CoM heights.

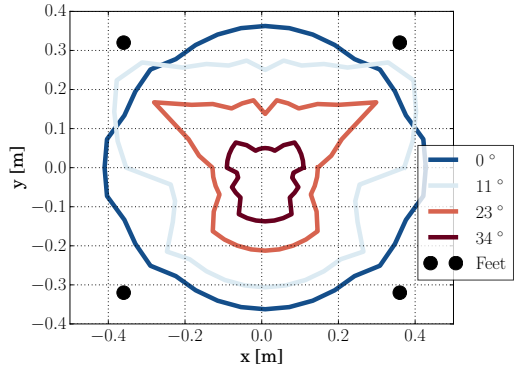


Fig. 7: Different evaluations of the reachable region at different HyQ roll orientations.

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

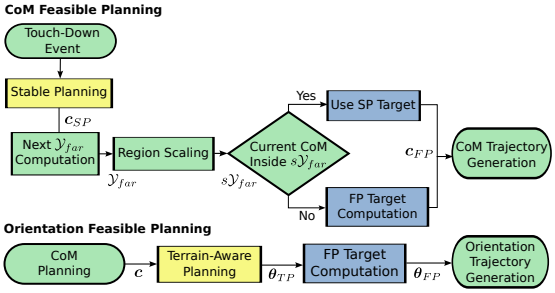


Fig. 8: Flowcharts illustrating the trajectory planning algorithms. Top: CoM FP algorithm. Bottom: Orientation FP algorithm.

define an *improved feasible region* as:

$$\mathcal{Y}_{far} = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 | \quad \exists \mathbf{f}_i \in \mathbb{R}^{mn_c}, \mathbf{q}_i \in \mathbb{R}^{n_l} \text{ s.t.} \\ (\mathbf{c}_{xy}, \mathbf{f}_i) \in \mathcal{C} \cap \mathcal{A}, \quad (\mathbf{c}_{xy}, \mathbf{q}_i) \in \mathcal{Q} \Big\} \tag{26}$$

Given that $\mathcal{C} \cap \mathcal{A}$ and $\mathcal{Q}$ are defined on different spaces, $\mathcal{Y}_{far}$ 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 (see Appendix for additional details). Therefore we can define the *improved feasible region* as:

$$\mathcal{Y}_{far} = \mathcal{Y}_{fa} \cap \mathcal{Y}_r \tag{27}$$

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

Name	Symbol	Constraints
Friction R. ([14])	$\mathcal{Y}_f$	Friction
Feasible R. ([1])	$\mathcal{Y}_{fa}$	Friction / Joint-torque
Reachable R. (this paper)	$\mathcal{Y}_r$	Kinematic
Improved Feasible R. (this paper)	$\mathcal{Y}_{far}$	Friction / Joint-torque / Kinematic

TABLE I: Types of regions

VI. Trajectory Planning

A. CoM planning strategy

In this subsection we improve the CoM planning strategy developed for *crawl* gaits described in our previous work [5], by exploiting the proposed definition of the improved feasible region (see Fig. 1 and 8). We will denote the method used in [5] as the *stable planning (SP)* strategy and our improved one as the *feasible planning (FP)* strategy. To simplify the planning framework, we assume 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]. Note that it is possible to extend the strategy to a more dynamic gait by designing CoM trajectories that are consistent with the extension shown in Section III-C. Therefore, one would need to ensure the trajectory is consistent with the dynamic feasible region, and that the accelerations are consistent with the desired ones that the region was computed for. Through the use of the improved feasible region we will improve the SP behavior adding guarantees on the physical feasibility. The feasible region is utilized to plan a CoM trajectory for the move-body phase such that in the following swing phase, *i.e.,* when only three feet are in stance, the CoM target remains feasible.

This phase (also labeled as three-contact phase) 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 stance legs (known from the foot sequence). A FP 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. Figure 8 provides a flowchart of the planning algorithms. As mentioned in Section II, the Jacobians used to evaluate the force polytopes of the contact legs make the feasible region configuration-dependent. Therefore, the region should be recomputed for each CoM location along the planned trajectory. To simplify the planning problem, we instead evaluate the region using the leg Jacobians computed at a configuration corresponding to the SP CoM target. The use of a single Jacobian can be further justified by the analysis done in [11], which showed that the variation of the Jacobian, around a specific configuration, has negligible effect on the contact forces. To introduce a level of robustness against uncertainties, the planning of the target is done considering a *scaled* version of the feasible region $s\mathcal{Y}_{far}$ with a tunable scaling coefficient $s \in (0, 1)$.

The procedure is devised as follows: if the current CoM projection $\mathbf{c}_{xy}$ (onto the region plane)[2] is inside $s\mathcal{Y}_{far}$, feasibility is already guaranteed and the target CoM position is chosen to be the current one to minimize 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 SP. This allows the motion to: (1) be as close as possible to the SP 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 SP target, also allows to (4) maintain the *local* validity of the *feasible region* (the Jacobian was evaluated for the SP target position).

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

[2]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.

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. One solution is to use inward polygon offsetting. However, this is not yet fast enough for online planning and we have noticed that, for this purpose, scaling the feasible region with respect to its centroid provides satisfactory results. For a more detailed discussion, refer to the Appendix.

B. Optimization of trunk orientation to maximize joint range

Upon planning a CoM trajectory, our previous planning approach [5] also computes a target trunk orientation (roll and pitch) to be attained during the move-body phase. This target is chosen to be aligned with the inclination of the *terrain plane* which is estimated in [5] via fitting an averaging plane through the stance feet. We will denote the method of [5] as the *terrain-based planning (TP)* strategy and our improved one as the *feasible planning (FP)* strategy.

This TP strategy aims at bringing the legs as close as possible to the middle of their workspace in order to avoid 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 rough terrains, where the feet are located on distant non coplanar surfaces, this might not be sufficient. In such cases, it can happen that some legs become more extended/retracted than others, as will be illustrated in Section VIII-B.

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

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

Note that this orientation planning strategy aims to improve upon the CoM planning strategy described in Section VI-A 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 VI-A (computed at the default orientation) *before* optimizing for the orientation.

VII. Assumptions Summary

As mentioned in the last sections, several assumptions were made during the computation of the algorithms presented in this paper. Table II provides a summary of these assumptions, their types and purposes, and where it was discussed in the paper.

	Assumption	Type	Purpose	Sections
1	Friction pyramids	M	CE	II
2	Feet in contact are stationary	M	PS	IV
3	Tangential contact moments $\neq$ zero	M	SF	III-D
4	Approximate center of reachable region	A	CE	IV, Appx.A
5	Scaling of region based on centroid	A	CE	VI-A, Appx.C
6	Single leg Jacobians for planning	M	PS	VI-A
7	Quasi-static motion planning	M	PS	VI-A

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

We classify the assumptions made to be of *modelling* or *algorithmic* type. *Modelling* assumptions relate to the model of the robot or the model of the interaction of the robot with the environment. *Algorithmic* assumptions relate to the definition of the algorithms that compute the regions or the planning target.

Each assumption was made for the purpose of either having *computational efficiency*, *planning simplicity*, or *solution feasibility*. *Computation efficiency* refers to assumptions that are made for the purpose of avoiding large computation times. Other assumptions were made to demonstrate the capabilities of the improved feasible region while attempting to avoid complicating the planning method, i.e. *planning simplicity*. *Solution feasibility* refers to assumptions that are made for the purpose of adapting a solution method to our problem.

Some assumptions can be chosen to be relaxed as required. Assumptions that are made for the purpose of computation efficiency (i.e., 1, 4, and 5) can be dropped at the cost of higher computation times (possibly becoming unsuitable for online planning). Assumptions 6 and 7 can be relaxed and subsequently a more involved planning strategy is needed. Assumption 2 is assumed to be enforced by the control architecture and Assumption 3 is necessary with our solution method.

VIII. Simulation Results

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 planning based on the improved feasible region compared to heuristics of [5].

Fig. 9: Simulation of HyQ descending a challenging 30° ramp with a 50 cm tunnel (template 1). The height of the HyQ robot is decreased from 53 cm to 40 cm in order to fit inside the tunnel. A force-controllable rope (not shown in the figure) is attached to the back of the robot's trunk to compensate for gravity.

All the presented simulations and experiments are shown in the accompanying video. The generation of the projected regions is done in Python 2.7[1]. Table III shows a summary of the computation times of the different stages of the planning[2]. The computer is equipped with an i7-8700 3.2 GHz processor and 16GB of RAM. Whenever a multitude of regions needs to be computed (as in the case of the optimization of the trunk orientation) we make use of the parallelism capabilities of our CPU using the *multi-processing* module in Python. The regions are sent via a ROS node to our locomotion planner, that runs in a ROS environment. The WBC runs at 250 Hz.

Stage	Computation Time		
	2-contacts	3-contacts	4-contacts
Feasible region	5 ms	9 ms	14 ms
Reachable region	25 ms	28 ms	29 ms
Intersection $\mathcal{Y}_{far}$	0.3 ms		
Region scaling	0.005 ms		
Target planning	0.03 ms		
Total feasible planning	18 Hz (worst case)		
Whole-body controller	250 Hz (worst case)		

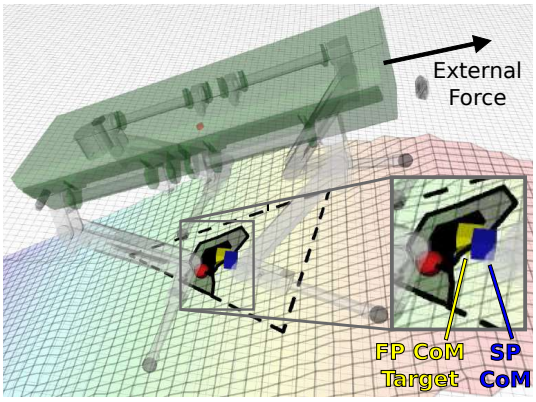
TABLE III: Average computation time for each stage of planning, using the SP target as an initialization for the required stages.

A. Walk in cluttered environment

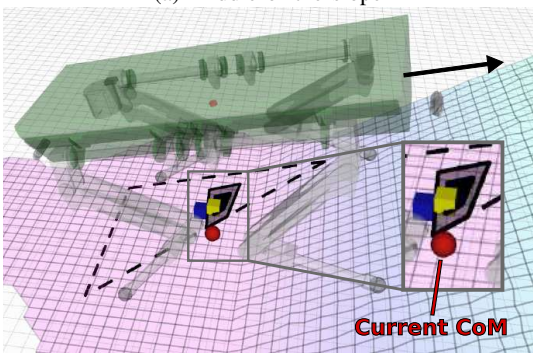
In this simulation, we assess the influence of an *external wrench* acting on the robot, combined with a *reduced* robot height necessary to walk in confined places. This challenging task consists of the HyQ robot descending a 30° ramp while being attached to a rope, to explore a low tunnel. This can be a typical scenario that a robot needs to face in oil rigs inspection assignments (see Fig. 9). A rope (not shown in the simulation software) connects the back of the robot to an anchor. The aid of the rope results in regulated locomotion down the steep slope (*e.g.,* the same way a climber is *rappelling* down a

[1]Source code available at github.com/abdelrahman-h-abdalla/jet-leg.

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



(a) Middle of the slope



(b) Bottom of the slope

Fig. 10: Improved feasible regions and CoM planning for two instances while descending the challenging tunnel in simulation (tunnel not shown in this figure). HyQ is heading to the left (downwards) while the external force due to the rope (black arrow) is applied in a direction opposite to the motion. The regions shown above are for the future regions upon lift-off of the swing leg (LF in the upper plot and LH in the lower one): support regions (dashed), improved feasible regions (grey), and the scaled feasible regions (black). Cubes represent the projection of the CoM target based on FP strategy (yellow) and the SP strategy (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).

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 [35]. Indeed, in a slope with high inclination, the robot eventually creates a tangential force on the terrain that surpasses the friction force that is needed to prevent slippage. An additional advantage of using a rope is that the robot can keep a more natural configuration, without the need to lean back or forth to keep stability, thus keeping the joints in a more kinematically advantageous configuration. As an additional difficulty, the restricted height of the tunnel places a risk of collision with the trunk of HyQ. The robot is therefore forced to *crouch walk* down the tunnel. For this reason, we reduce the robot height from 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 force (equivalent to 440 N applied to the back of the robot) coming from the rope.

The above-mentioned effects on the friction region and on the feasible region can be seen in Fig. 10 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 robot trunk orientation where the CoM planning is done. In both situations, a shift in the friction and feasible regions, opposite to the external wrench on the robot, could be observed. Furthermore, the low height imposed on the robot results in a big shrinkage of the feasible region. Under these conditions, the CoM target (blue) planned with the SP strategy is located outside the region. Conversely, the CoM planner based on the improved feasible region (FP), computes a feasible target (yellow) that is on the boundary of the *scaled* feasible region and closest to the SP target. It is interesting to remark that even though the friction region is shifted, thus giving the robot more freedom to lean forward if desired, the improved feasible region is inhibiting such courageous motions due to joint-torque restrictions and to the limited reachable region.

B. Optimization of the Trunk Orientation on rough terrain

To illustrate the effectiveness of the orientation optimization strategy proposed in Section VI-B, we test it separately from the CoM planning strategy developed in Section VI-A. For this reason, the optimization of the orientation will be based on the CoM target computed by the SP approach. As mentioned before, even if this does not necessarily guarantee feasibility, it allows us to compare clearly the improvements of the FP strategy over the TP strategy. While climbing up a ramp, it is typical to move the torso forward [5, 29] 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 rough terrain shown in Fig. 11, is particularly challenging in terms of kinematic limits. One of the legs can be forced to overly extend/retract during the move-body phase even though the other legs are possibly far from their limits. In fact, adopting an orientation based solely on the TP strategy results in infeasible trajectories in multiple locations of the terrain (Fig. 11(a) bottom). The TP approach would not capture the difficulty given by the "lateral asymmetry" of this scenario. Indeed, it would result in a trunk with the hips being equally distant from the left and the right feet. In the example shown, a pitch angle of $9.7°$ estimated averaging terrain plane), is selected by the TP approach. This results in a hyper-extension of the RH leg and a kinematic violation at the Knee Flexion-Extension (KFE) joint (Fig. 11(a) top). Note that since we

[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 $\mathbf{n}$.

model the kinematic limits in our simulator, the CoM will not be allowed to go out of the boundary of the region. The same CoM trajectories could instead be feasible if the orientation is planned based on the FP strategy, with an optimized pitch angle of -0.3° (see Fig. 11(b)). The optimized pitch angle maximizes the distance of the trajectory, from the boundary of the region (*i.e.,* the margin), as well as the area of the region, thus resulting in a safer joint configuration.

IX. Experimental Results

A. Walk in cluttered environment

We implement the simulation example shown in Section VIII.A on real hardware using the robot Aliengo (for safety reasons) as shown in Fig. 12. The robot is commanded to walk down a steeper slope of 45° with a robot height of 20 cm. A pulley, rope and counterweight (a mass of about 10 kg was used) are used to introduce the external force needed to pull the robot backwards. The kinematic limits of Aliengo are virtually lowered to simulate that of hydraulic robots like HyQ at low robot heights.

The plots of the KFE joint trajectory during the experiments are reported in Fig. 13. A CoM target based on the SP strategy would result in multiple violations of the kinematic limits (upper plot) while the one based on the FP strategy has no violations (lower plot).

Additionally, to show the effect kinematic violations can have on the performance of the robot, we perform experiments with the 90 kg HyQ robot platform walking on flat ground with a reduced height of 43 cm. Fig. 14 shows that such kinematic violations result in a deterioration of the tracking of the CoM trajectory computed using the SP strategy as opposed to the FP strategy.

B. Dynamic Motions

In this section we show an experiment with the 21 kg robot Aliengo performing a dynamic trotting gait developed in an earlier work [4]. The improved feasible region for the motion is computed, encompassing the dynamic effects as explained in Section III-C. The purpose of this is two-fold: 1) demonstrate the inertial effects that dynamic motions have on the improved feasible region, and 2) to show how the solution we proposed in Section III-D to deal with "degenerate" regions works effectively. We report the results in the accompanying video showing that the CoM projection remains inside the feasible region.

X. Conclusion

In this paper we introduced an improved version of the feasible region presented in our previous work [1]. 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 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 (through the newly defined *reachable region*) and the presence of external wrenches acting on arbitrary points of the robot. This, along with projecting the region on arbitrary plane inclinations that are consistent with the planning intention of the user, offers the opportunity to employ the proposed motion planning strategy to new possible applications. One such application has been demonstrated in this work, i.e. rope-aided locomotion, while the same approach can be applied to load-pulling/pushing applications. To include the dynamic effects of motion we relaxed the quasi-static assumption in the iterative projection algorithm.

Furthermore, we proposed a planning strategy that utilizes the improved feasible region to design feasible CoM and trunk orientation trajectories. We validated the capabilities of the improved feasible region and the effectiveness of the proposed planning strategy on challenging simulations and experiments with the HyQ and Aliengo robots and we compared our results to a previously developed approach [5] that is not able to formally guarantee the kinematic feasibility of its trajectories. We validated the extensions of the feasible region to be compatible with dynamic motions using experiments with the Aliengo robot and demonstrated the effect that the height and orientation of the robot have on the reachable region.

As future works, we intend to incorporate the feasible region in an centroidal momentum MPC controller. This will result in a CoM constraint that is suitable for non coplanar contacts and would add descriptiveness to the MPC formulation (employing a reduced model) with joint-based constraints (e.g. torque and/or kinematic constraints that are embedded in the region). Other ongoing works are focused on speeding up the computation of the region increasing its accuracy in the vicinity of the direction of motion. This would allow us to only refine (or even only compute) the parts of the feasible region that are relevant to the locomotion direction.

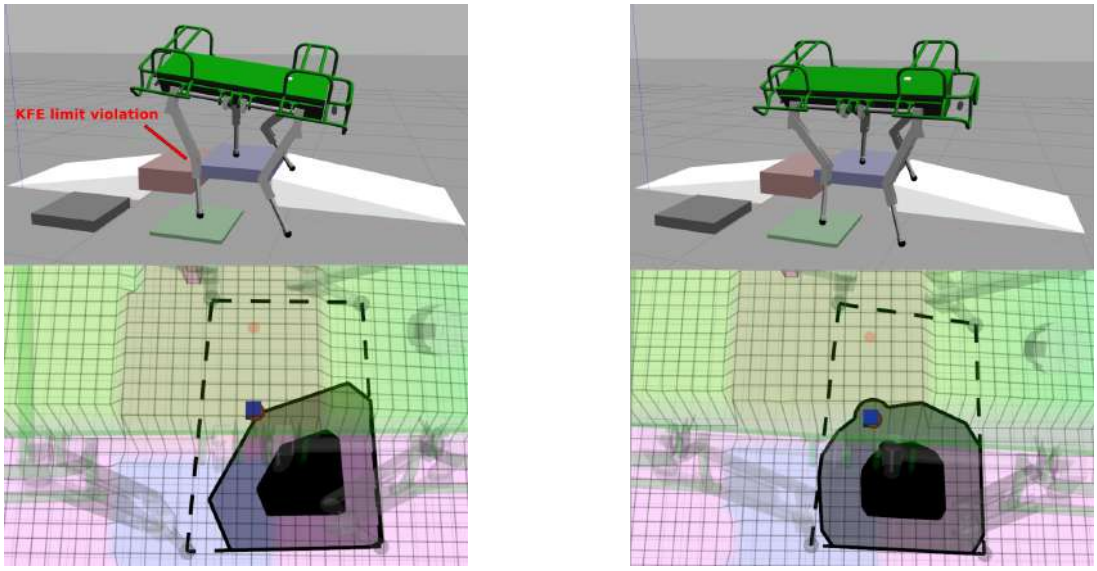
Appendix

This appendix provides more theoretical analysis on some aspects of the reachable and improved feasible region, as well as provide additional implementation details to the interested readers.

A. Reachable region and CoM workspace

It is useful to further illustrate the effect of the ray casting algorithm on the produced reachable region as compared to the full workspace of the CoM. The full workspace can be comprised of disjoint sets (*e.g.,* [33]) which would not be captured by the algorithm. Figure 15 shows the CoM workspace for a height of 0.49 m for the HyQ robot, where the green and red points show the kinematically feasible and infeasible locations, respectively, obtained using a brute force approach where each point in a two-dimensional grid (grid point distance of 2.5 cm) was tested and marked accordingly.

As evident in Fig. 15, a special case can arise at some configurations where the workspace is non-convex and some disjoint sets appear. Given the nature of ray casting algorithms, in such cases the method only determines the reachable region in the range of rays casted from an initial point. The black dashed boundary in the figure shows the output of the



(a) Using Terrain-Based Planning

(b) Using Feasible Planning

Fig. 11: Simulation of HyQ forced near its kinematic limits while traversing a difficult non-coplanar terrain (Template 2). The robot configurations shown are at the end of a move-body phase. Realizing orientations based on (a) the TP strategy and (b) based on the FP strategy results in different leg configurations (top). The resulting regions shown in the bottom plots are: friction regions (dashed), feasible regions (grey), and the scaled feasible regions (black). Large difference in the resulting feasible regions can be seen, in turn affecting the feasibility of the CoM trajectory (blue cube and red ball represent the projections of the CoM target and the actual CoM, respectively).

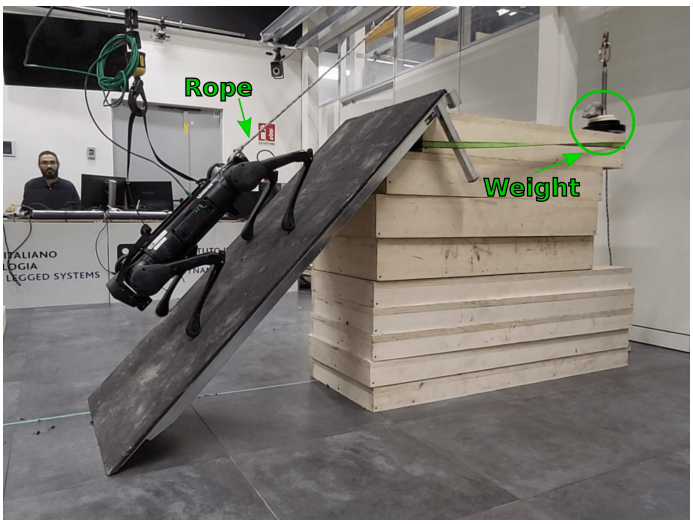


Fig. 12: Aliengo descending a 45° ramp with a reduced robot height of 20 cm (similar to the scenario shown in simulation). A rope is attached to the back of the robot's trunk to compensate for gravity through a counterweight.

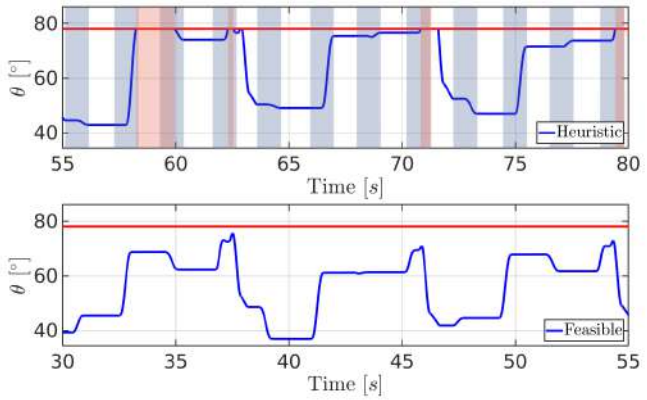


Fig. 13: Experimental results showing the Right-Hind Hip Flexion-Extension (HFE) joint trajectory of Aliengo during the ramp descent. SP strategy (above): the knee starts to hit the virtual kinematic limit (red line) during the *move body* phases (shaded blue). The violations are in shaded red. FP strategy (bottom): no kinematic limit violations are observed.

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

B. Regions intersection

The *improved feasible region* can simply be obtained by intersecting the *feasible region* with the *reachable region*. This is in contrast with the case of attempting to obtain the *feasible region* $\mathcal{Y}_{fa}$ by the simple intersection of the friction region $\mathcal{Y}_f$ and the actuation region (centroidal mapping of joint-torques) $\mathcal{Y}_a$ as explained in [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 joint-torque constraints) must be carried out first before projecting the resulting set. The converse is inaccurate since the intersection and projection

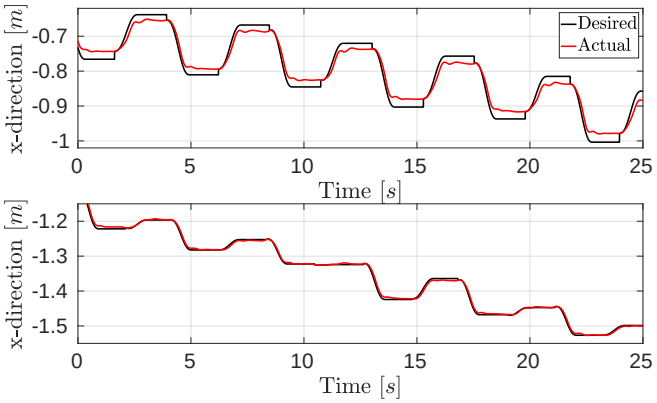


Fig. 14: Experimental results showing the CoM position tracking of HyQ in the x direction. A deterioration can be seen with the SP strategy (upper plot) due to the joint kinematic limit violations while good tracking is observed with the FP strategy (bottom plot).

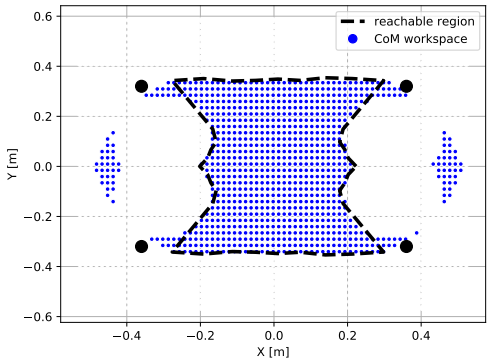


Fig. 15: Comparison between the CoM workspace (ground-truth blue) and *reachable region* (black dashed) for HyQ at 0.49 m CoM height.

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.

C. Non-convex scaling

Scaling a non-convex polygon through an *affine* transformation with respect to a reference point (e.g., the Chebyshev center or the centroid) could result in a scaled region with parts outside the original one. On the other hand, inward *polygon offsetting* algorithms guarantee that the scaled polygon always remains inside the original one. One downside of this algorithm is that the scaled polygon can suffer from topological changes (e.g., some edges might contract until they vanish [36]). Furthermore, the scaling during the offsetting procedure is defined by a distance. The centroid based scaling, on the other hand, characterizes the scaling in terms of a percentage. This allows the algorithm to be directly scalable and consistent with robots of different dimensions. Although offsetting non-convex polygons is still a hard problem in itself, [37] proposed a solution for non-convex polygons.

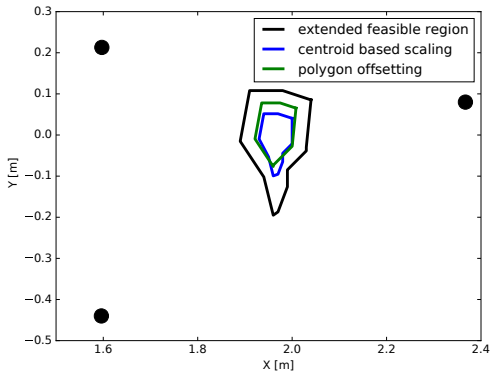


Fig. 16: Comparison between different methods of scaling the (non-convex) improved feasible region: scaling based on the centroid and scaling using polygon offsetting.

In Fig. 16 we compare the result achieved using the polygon offsetting algorithm to the output of the affine scaling with respect to the centroid. An improved feasible region (shown in black) is computed for a three contacts phase with a height of 0.37 m for the HyQ robot. We set a scaling factor $s = 0.5$ for the affine scaling (black) and an offset of $r = 0.03$ m for the polygon offsetting (green). A small difference in the area of the scaled polygons and a slight shift can be observed, with the offsetted polygon changing its topology due to the already small size of the original polygon. We chose to utilize the affine scaling for the direct scalability and its efficiency (an average computation time of 0.005 ms as compared to 5 ms for the polygon offsetting). To guarantee that the scaled region would strictly be a member of the original one, an additional efficient step of using the intersection of the original and the scaled region can be performed.

ACKNOWLEDGEMENTS

We would like to thank Victor Barasuol for the valuable advice during the experiments and review process of this paper. We would also like to thank Shamel Fahmi, Chundri Boelens, and all the members of the DLS lab for the help provided in the development of this work.

REFERENCES

[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] F. M. Smaldone, N. Scianca, V. Modugno, L. Lanari, and G. Oriolo, "Gait generation using intrinsically stable mpc in the presence of persistent disturbances," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 651–656.

[16] S. Mason, N. Rotella, S. Schaal, and L. Righetti, "An mpc walking framework with external contact forces," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1785–1790.

[17] H. Audren and A. Kheddar, "3-d robust stability polyhedron in multicontact," *IEEE Transactions on Robotics*, vol. PP, 02 2018.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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*, pp. 5761–5768, 2018.

[23] 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.

[24] 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.

[25] "Unitree robotics. aliengo," https://m.unitree.com/products/aliengo, accessed: 15 October 2022.

[26] J. C. Trinkle, J. S. Pang, S. Sudarsky, and G. Lo, "On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction," *Zeitschrift Angewandte Mathematik und Mechanik*, vol. 77, no. 4, pp. 267–279, Jan. 1997.

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

[28] I. Kao, K. Lynch, and J. W. Burdick, *Contact Modeling and Manipulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 647–669. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_28

[29] 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.

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

[31] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, "Passive Whole-body Control for Quadruped Robots: Experimental Validation over Challenging Terrain," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2553–2560, Jul. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02086071

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

[33] 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.

[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.

[35] 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.

[36] S.-W. Cheng and A. Vigneron, "Motorcycle graphs and straight skeletons," *Algorithmica*, vol. 47, no. 2, pp. 159–182, Feb 2007. [Online]. Available: https://doi.org/10.1007/s00453-006-1229-7

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

Abdelrahman Abdalla was born in Cairo, Egypt. He received the B.Sc. degree in mechatronics from the German University in Cairo, Cairo, Egypt, in 2017, the M.Sc. degree in control engineering from the Sapienza Università, Rome, Italy, in 2020, and the Ph.D. degree in advanced and humanoid robotics from the Italian Institute of Technology, Genoa, Italy, in 2021. In December 2021, he joined the Dynamic Legged Systems Lab, Istituto Italiano di Tecnologia for the Ph.D. degree. His Ph.D. focuses on wrench-based locomotion planning and the development of efficient locomotion feasibility criteria.

Michele Focchi was born in Rimini, Italy, in 1980. He received both the B.Sc. and the M.Sc. degree in control systems engineering from Politecnico di Milano, Italy in 2007. He received the Ph.D. degree in robotics from Istituto Italiano di Tecnologia (IIT), Genova, Italy, in 2013.

From 2014 to 2021 he worked as a researcher at IIT. His research has been concerned with the software development of planning and control strategies for quadruped robots. On 2022 he joined University of Trento, Italy where he works as a lecturer. Currently his research interests are focused on pushing the performances of quadrupeds in traversing unstructured environments, by using optimization-based techniques and devising novel robotic platforms.

Romeo Orsolino completed his Ph.D. in Bioengineering and Robotics from Istituto Italiano di Tecnologia (IIT) in 2019, where he worked at the DLS lab focusing on motion planning for legged locomotion in rough terrains. He then joined the DRS group at the Oxford Robotics Institute, University of Oxford, as a postdoctoral researcher where he pursued his research on multi-contact motion planning, optimal control, dynamics and perception.

Since December 2020, Romeo has moved to industry where he works as a robotics research engineer on R&D projects to increase the use of advanced robotics in the automotive and energy sectors.

Claudio Semini (Member, IEEE) received the M.Sc. degree in electrical engineering and information technology from ETH, Zurich, Switzerland, in 2005, and the Ph.D. degree in humanoid technologies from the Istituto Italiano di Tecnologia (IIT), Genoa, Italy, in 2010. During his doctorate, he developed the hydraulic quadruped robot HyQ and worked on its control.

Since 2012, he has been leading the Dynamic Legged Systems Lab, IIT. He is/was the coordinator/partner of several European Union, National, and Industrial projects (including HyQ-REAL, INAIL Teleop, Moog@IIT joint lab, ESA-ANT, RAISE, FAIR, etc.). He is the author and co-author of over 100 journal and conference publications. His research interests include the design and control of legged robots for real-world operations, locomotion, and hydraulics.

Dr. Semini is a Co-Founder of the Technical Committee on Mechanisms and Design of the Robotics and Automation Society.