Simultaneous Contact and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization

Bernardo Aceituno-Cabezas[1], Carlos Mastalli[2], Hongkai Dai[3], Michele Focchi[2], Andreea Radulescu[2], Darwin G. Caldwell[2], José Cappelletto[1], Juan C. Grieco[1], Gerardo Fernández-López[1], and Claudio Semini[2]

Abstract— Traditional motion planning approaches for multi-legged locomotion divide the problem into several stages, such as contact search and trajectory generation. However, reasoning about contacts and motions simultaneously is crucial for the generation of complex whole-body behaviors. Currently, coupling theses problems has required either the assumption of a fixed gait sequence and flat terrain condition, or non-convex optimization with intractable computation time. In this paper, we propose a mixed-integer convex formulation to plan simultaneously contact locations, gait transitions and motion, in a computationally efficient fashion. In contrast to previous works, our approach is not limited to flat terrain nor to a pre-specified gait sequence. Instead, we incorporate the friction cone stability margin, approximate the robot's torque limits, and plan the gait using mixed-integer convex constraints. We experimentally validated our approach on the HyQ robot by traversing different challenging terrains, where non-convexity and flat terrain assumptions might lead to sub-optimal or unstable plans. Our method increases the motion generality while keeping a low computation time.

I. INTRODUCTION

Planning motions for multi-legged robots can be a challenging task, as it involves both the discrete choice of the gait sequence, and continuous decisions on foot location and robot dynamics. Because of this, most traditional approaches [1], [2], [3], [4] simplify the planning problem by decoupling it into two stages: 1) find a sequence of contact locations; and 2) generate body trajectories based on a stability metric, like the Zero Moment Point (ZMP) [5] or the generalized Contact Wrench Cone (CWC) [6], [7] stability criterion. Ignoring the dynamics in the first stage might restrict the number of feasible motion on the second one, when dynamics are considered. Hence, in this paper we combine these two stages into one single problem, in order to efficiently generate complex behaviors.

Trajectory Optimization (TO) has emerged as a solution to the coupled motion planning problem [8], [9], [10], [11], [12]. Unfortunately, even state of the art approaches are often restricted to simple environments, a fixed gait, or require intractable computation time. Furthermore, none of them can guarantee global optimality, and can easily fall into local minima, since these are posed as non-convex optimization problems. On the other hand, Mixed-Integer Convex Programming (MICP) proved to be an efficient

The authors are with (1) the Mechatronics Research Group, Simón Bolívar University {12-10764, cappelletto, jcgrieco, gfernandez}@usb.ve, (2) are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia {firstname.surname}@iit.com, and (3) is with the Toyota Research Institute hongkai.dai@tri.global
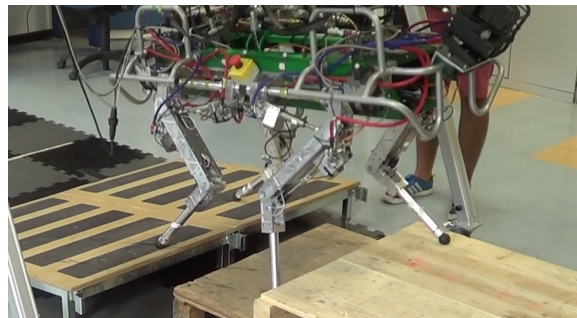
Fig. 1: The proposed approach generates robust plans in challenging terrain, simultaneously reasoning about contacts and motions. This figure shows HyQ executing an optimized motion with non-coplanar contacts.

tool to solve multi-contact motion planning problems. In [1], [3], the authors first find a set of safe surfaces, and then use integer variables to assign each contact to one surface. Further works incorporate a convex or mixed-integer convex dynamic model in the formulation [13], [14], which can efficiently optimize contacts and motions. Unfortunately, these approaches do not incorporate torque constraints nor maximize any generalized stability margin. Moreover, they do not reason about the sequence of contacts, being limited to a fixed gait sequence.

In this work, we introduce a novel MICP formulation for non-gaited multi-legged locomotion on challenging terrain. Here, we formulate a single mixed-integer convex optimization problem that simultaneously plans contacts and motions. In contrast to other approaches, our method guarantees dynamic feasibility and optimizes gaits sequences in challenging terrains. For this, we incorporate convex representations of friction cones, torque limits, and gait planning constraints. For the friction cone constraints, we exploit the segmentation of the terrain and pose the stability margin computation as a convex constraint. This optimization problem can then be solved efficiently to its *global optimum*, at least two orders of magnitude faster than similar non-convex approaches, using off-the-shelf optimization solvers. We validated our approach by traversing different challenging terrains with the Hydraulically actuated Quadruped (HyQ) robot, shown in Fig. 1. Our approach generates robust motions, even during gait transitions and non-coplanar contact scenarios.

The rest of this paper is organized as follows: Section II presents our simultaneous contact and motion planning approach. Section III presents our whole-body controller. Section IV presents a set of experiments on HyQ traversing challenging terrains, and Section V discusses and concludes on the contributions of this work.
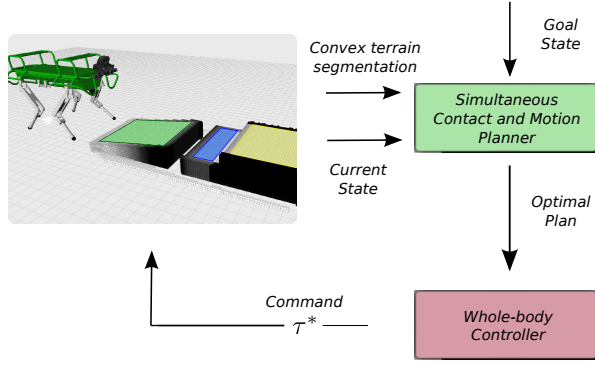
Fig. 2: Overview of our convex trajectory optimization approach. Here, we compute contact locations and motions within a single mixed-integer convex program, given a goal state and a convex segmentation of the safe terrain.

II. Simultaneous Contact and Motion Planning

In this section, we describe our formulation of the simultaneous contact and motion planning problem using Mixed-Integer Convex Optimization.

A. Approach Overview

Let us consider a robot with n_l legs and a locomotion plan, discretized on N time-knots. We formulate a MICP to find the optimal contact locations and motions, given a goal position $\mathbf{r}_G$ and a convex segmentation of the terrain. An illustration of this approach is shown in Fig. 2.

We describe the dynamic evolution of the system with a centroidal model (Section II-B). We include gait planning as part of the optimization (Section II-C), describing the motion through time-slots over which legs swing. This evolution is subject to robot and environmental constraints such as: approximated kinematic reachability (Section II-D), friction cone constraints (Section II-F), and approximated torque limits (Section II-H). describing the motion through time-slots over which we will generate leg swings. We adopt the formulation in [13] to further address the non-linearities of the angular dynamics by relying on a convex decomposition of bilinear terms, making our optimization problem convex (Section II-G).

B. Centroidal Dynamics

We describe the evolution of the system using a centroidal dynamics model [15], [16], as depicted in Fig. 3. In this case, the dynamics of the system are written as:

$$\begin{bmatrix} m\ddot{\mathbf{r}} \\ \dot{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} + \sum_{l=1}^{n_l} \boldsymbol{\lambda}_l \\ \sum_{l=1}^{n_l} (\mathbf{p}_l - \mathbf{r}) \times \boldsymbol{\lambda}_l \end{bmatrix}, \quad (1)$$

where variables are named as specified in Table I, m is the mass of the robot, and $\mathbf{g}$ is the gravity vector. This formulation of the dynamics is entirely *convex*, except for the cross product to compute the torque at center of mass.

Remark 1: Note that (1) neglects the effect of contact torques. This is done because most multi-legged platforms, like quadrupeds and hexapods, have approximate point contacts, without support surface to generate moments.
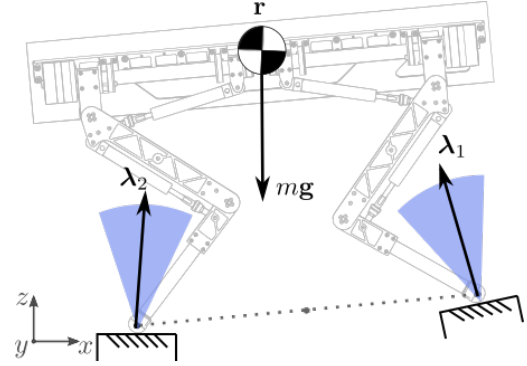

Fig. 3: Centroidal dynamic model of a quadruped.

TABLE I: Variables of the trajectory optimization

$\mathbf{r} = [r_x, r_y, r_z]^T$	Center of Mass (CoM) in the world frame
$\mathbf{k} = [k_x, k_y, k_z]^T$	Centroidal angular momentum
$\mathbf{p}_l = [p_{l_x}, p_{l_y}, p_{l_z}]^T$	End-effector positions in the world frame
$\boldsymbol{\lambda}_l = [\lambda_{l_x}, \lambda_{l_y}, \lambda_{l_z}]^T$	Contact force at the lth end-effector
N_t, N_k	Number of time-slots and knots per swing
N_f, N_r	Number of contacts and convex regions

Furthermore, the dynamics of the model are discretized in knots of Δt seconds, over which we will perform backward Euler integration.

C. Gait Sequence

In order to plan gait transitions, we represent the motion through N_f contact locations and N_t time-slots over which the legs will swing between adjacent contacts , and introduce a binary *transition matrix* $\mathbf{T} \in \{0, 1\}^{N_f \times N_t}$. Here, $\mathbf{T}_{ij} = 1$ means that the robot will move to the ith contact location at the jth time-slot, as shown in Fig. 4. Since each contact location in the plan is reached once, we enforce that:

$$\sum_{j=1}^{N_t} \mathbf{T}_{ij} = 1 , \ \forall i = 1, .., N_f. \quad (2)$$

Additionally, we enforce that each cycle of n_l contacts must be reached before the next transfer cycle starts. For this, we define a vector $\mathbf{t} = \mathbf{T} \begin{bmatrix} 1 & \cdots & N_t \end{bmatrix}^T$ which computes the corresponding slot assigned to each movement. Furthermore, since the gait follows a sequentially ordered contact plan, we enforce the following constraint:

$$\mathbf{t}_j > \mathbf{t}_{j-n_l} , \ \forall j = n_l + 1, \ldots, N_t. \quad (3)$$

Admissible heuristics can also be added in order to guide the gait through different terrains. These can be incorporated in the formulation as shown in [3].

D. Contact Location

In order to simplify the formulation, we only optimize the contact locations and contact timing, leaving the end-effector trajectory as an interpolation between adjacent contacts in the plan. Since swing phases do not contribute to the centroidal dynamics, as force in the leg becomes null, this simplification has no effect on the resulting plan.

As part of the decision variables, we describe contact locations for n_l end-effectors using an array of N_f vectors
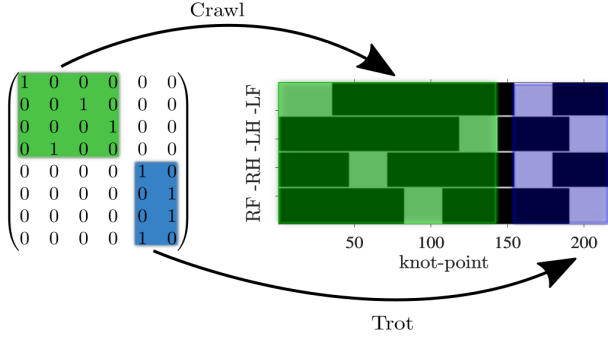
Fig. 4: Gait transition matrix and equivalent phase diagram. R, L, H and F represent right, left, hind and front supports.

in $\mathbb{R}^4$, ordered by end-effector number, of the form:

$$\mathbf{f} = (f_x, f_y, f_z, \theta),$$

representing the position of each contact in Cartesian space and the yaw orientation of the trunk when transitioning to that contact, neglecting roll and pitch positions.

1) Safe-region assignment: Here, we will invert the problem of avoiding obstacles by constraining the contacts to lie within one of N_r convex safe contact surfaces, shown colored in Fig. 5 (*left*). Each surface is represented as a polygon $\mathcal{R} = \{\mathbf{c} \in \mathbb{R}^3 | A_r\mathbf{c} \leq b_r\}$. The assignment of contacts to these surfaces is done through a binary decision matrix $\mathbf{H} \in \{0,1\}^{N_f \times N_r}$. The constraints, for the ith contact, are:

$$\sum_{r=1}^{N_r} \mathbf{H}_{ir} = 1, \tag{4}$$

$$\mathbf{H}_{ir} \Rightarrow A_r\mathbf{f}_i \leq b_r, \tag{5}$$

where the $\Rightarrow$ (implies) operator is represented with big-M formulation [17].

2) Kinematic constraints: In order to ensure kinematic reachability, we must account for the workspace of each independent leg. To do this, we constrain each contact location within the biggest square inscribed in the leg workspace, as shown in Fig. 5 (*right*). Algebraically:

$$\left| \mathbf{f}_i - \left[\mathbf{r}_{T(i)} + L_i \begin{pmatrix} \cos(\theta_i + \phi_i) \\ \sin(\theta_i + \phi_i) \end{pmatrix} \right] \right| \leq d_{lim}, \tag{6}$$

where $\mathbf{r}_{T(i)}$ is the CoM location after transitioning to the contact (given by the gait matrix $\mathbf{T}(i)$), d_{lim} is the square diagonal, L_i is the approximate distance from the trunk to the leg, and ϕ_i is a known offset for each foot. Here, the trigonometric functions are decomposed in terms piecewise linear approximations of the trunk orientation functions.

To have them expressed as mixed-integer convex, we will follow the approach of [1] and replace the trigonometric functions of (6) with piecewise linear approximations of N_s segments s and c. We define binary matrices $\mathbf{S}$ and $\mathbf{C}$ in $\{0,1\}^{N_f \times N_s}$ to assign linear segments, which is done with the following constraints:

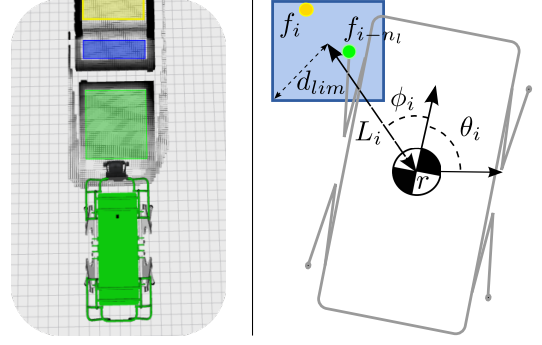$$\sum_{s=1}^{N_s} \mathbf{S}_{is} = 1 \qquad \sum_{s=1}^{N_s} \mathbf{C}_{is} = 1, \tag{7}$$

Fig. 5: *Left:* Convex segmentation of safe contact surfaces. *Right:* Approximate kinematic reachability constraints.

$$\mathbf{S}_{ik} \Rightarrow \begin{cases} \psi_{k-1} \leq \theta_i \leq \psi_k \\ s_i = m_k\theta_i + n_k \end{cases} \quad \mathbf{C}_{ik} \Rightarrow \begin{cases} \gamma_{k-1} \leq \theta_i \leq \gamma_k \\ c_i = m_k\theta_i + n_k \end{cases}, \tag{8}$$

where ψ and γ represent the boundaries between each linear segment, and m and n represent its slope and intersection.

E. End-effector Trajectories

As mentioned in Section II-D, the end-effector trajectories are defined by the gait transition matrix $\mathbf{T}$. For simplicity, we define the function $\gamma(j,t)$ to reference the knots over which an end-effector swings between two adjacent contacts in the plan, where j indicates the time-slot used for the swing and $t \in [1, \ldots, N_k]$ indicates the knot, where N_k is the number of knots allocated for each slot. Then, the end-effector motions are governed by the following constraint:

$$\mathbf{T}_{ij} \Rightarrow \mathbf{p}_{l(i)\gamma(j,N_k)} = \mathbf{f}_i, \tag{9}$$

where $l(i)$ is the leg number for the ith contact. This constraint enforces that the leg reaches the contact position $\mathbf{f}_i$ at the end of the jth slot. Also, it is important to constrain that the leg remains stationary when there is no transition. This is enforced using the following constraint over the lth leg:

$$\sum_{i \in C(l)} \mathbf{T}_{ij} = 0 \Rightarrow \mathbf{p}_{l\gamma(j,t)} = \mathbf{p}_{l\gamma(j,1)} \ \forall t \in [2, \ldots, N_k], \tag{10}$$

where $C(l)$ are the contact indexes assigned to the lth leg. Further constraints can be added to make the end-effector follow a specific swing trajectory. Moreover, we ensure kinematic feasibility by constraining the CoM position with respect to the end-effectors (Fig. 6). Here, we use the bounding box constraint:

$$\mathbf{d}_- < \mathbf{r}_j - \frac{\sum_{l=1}^{n_l} \mathbf{p}_{lj}}{n_l} < \mathbf{d}_+, \tag{11}$$

where $\mathbf{d}_-$ and $\mathbf{d}_+$ are the bounding box limits.

F. Contact Dynamics

We model the dynamic interaction of the forces through activation and friction cone constraints, which ensure stability in the motion. Here, we will describe how to pose these constraints as mixed-integer convex.
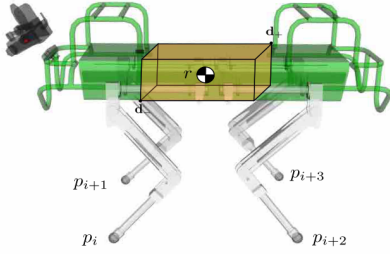
Fig. 6: We incorporate bounding box constraints, shown in orange, to ensure the CoM remains in its workspace.

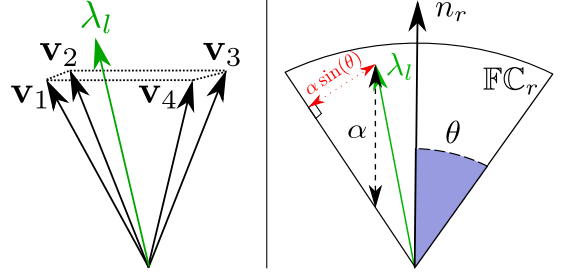

Fig. 7: *Left*: Friction polyhedron constraint. *Right*: Illustration of the α lower-bound of the friction cone stability margin.

1) Activation constraints: In order to ensure dynamic consistency, it is important that no force is present on a leg when it breaks contacts. This is constrained, for the lth leg, as:

$$\sum_{i\in C(l)} \mathbf{T}_{ij} = 1 \Rightarrow \boldsymbol{\lambda}_{l\gamma(j,t)} = 0 \ , \ \forall t \in NC(j), \quad (12)$$

where $NC(j)$ is the set of knots in the jth slot used for the swing. This relation can be seen as a complementarity constraint, which can be also used to activate contacts [11], [18], between $\sum_{i\in C(l)} \mathbf{T}_{ij}$ and $\boldsymbol{\lambda}_{l\gamma(j,i)}$.

2) Dynamic stability constraints: In order to maintain stability in non-coplanar contact conditions, the contact forces must remain within their friction cones [7]. We exploit the convex segmentation of the terrain in order to linearly approximate the friction cone constraints at each segment. This allows us to maintain the convexity in the problem, while also providing formal robustness guarantees.

Given that the rth surface has a normal unit vector $\hat{\mathbf{n}}_\mathrm{r}$, we approximate its friction cone as a polyhedron with N_e edges $\mathbf{v}_{r_1},\ldots,\mathbf{v}_{r_{N_e}}$. Therefore, each force remains in its respective contact cone $\mathbb{FC}_\mathrm{r}$ through the following constraint:

$$\boldsymbol{\lambda}_{lj} \in \mathbb{FC}_\mathrm{r} \Rightarrow \boldsymbol{\lambda}_{lj} = \sum_{e=1}^{N_e} \rho_e \mathbf{v}_{\mathrm{r}_e} \ , \ \rho_1,\ldots,\rho_{N_e} > 0,$$

where ρ_e are positive multipliers on each cone edge, as shown in Fig. 7 (*left*). Then, in order to add robustness to the motion, we maximize the distance between the nonlinear friction cone boundary and the force vector. This distance can be computed as $\alpha\sin\theta$, where θ is the half-angle of the cone, and α is defined as

$$\alpha = \arg\max_{\bar{\alpha}} \ \text{s.t} \ \boldsymbol{\lambda}_{lj} - \bar{\alpha}\hat{\mathbf{n}}_\mathrm{r} \in \mathbb{FC}_\mathrm{r},$$

namely, α is the distance from the contact force $\boldsymbol{\lambda}_{lj}$ to the boundary of the friction cone, along the normal force direction [19], as shown in Fig. 7 (*right*). Therefore, we seek to maximize the value of α at each knot, which increases the stability margin, and we introduce the following linear constraint over each safe surface:

$$\mathbf{T}_{ij} \ \text{and} \ \mathbf{H}_{ri} \Rightarrow \boldsymbol{\lambda}_{l(i)\gamma(j)} - \alpha_{l(i)\gamma(j)}\hat{\mathbf{n}}_\mathrm{r} \in \mathbb{FC}_\mathrm{r} \ , \ \alpha \geq 0. \ (13)$$

Since the contact cone must not change when it is in stance phase, we also add the constraint:

$$\sum_{l\in C(i)} \sum_{t\in NS(j)} \mathbf{T}_{lt} = 0,$$

$$\Rightarrow \boldsymbol{\lambda}_{l(i)\gamma(j)} - \alpha_{l(i)\gamma(j)}\mathbf{n}_\mathrm{r} \in \mathbb{FC}_\mathrm{r}, \quad (14)$$

where $NS(j)$ is the set of time-slots succeeding j. Note that this ensures that the cone constraint holds for all the succeeding knots in which the leg is stationary.

G. Convex Decomposition of Angular Dynamics

The angular dynamics of the centroidal model are non-convex due to the cross product $(\mathbf{p}_l - \mathbf{r}) \times \boldsymbol{\lambda}_l$. Other mixed-integer approaches have used McCormick envelopes to approximate these bilinear relations [14]. Nevertheless, this increases significantly the complexity of the problem. Here, we exploit the relation noted in [13], namely that this non-convexity can be addressed by reformulating bilinear constraints as a decomposition of quadratic terms:

$$\mathbf{ab} = \frac{\mathbf{u}^+ - \mathbf{u}^-}{4} \quad \mathbf{u}^+ \geq (\mathbf{a}+\mathbf{b})^2 \quad \mathbf{u}^- \geq (\mathbf{a}-\mathbf{b})^2. \quad (15)$$

This convex decomposition is represented via quadratic constraints, turning the problem into Mixed-Integer Quadratically Constrained Quadratic Programming (MIQCQP) and, thus, a convex optimization. We penalize the norm of $\mathbf{u}^+$ and $\mathbf{u}^-$ to bound the rate of the centroidal angular momentum. For more details, the readers can refer to [13].

H. Approximate Torque Limits

In order to improve the feasibility of the motion, we approximate torque limits by relying on a quasi-static motion assumption. This allows us to represent torques by projected the contact forces at each joint, compensating gravity terms, with respect to each leg frame:

$$\mathbf{J}_{l,j}^T \boldsymbol{\lambda}_{l,j} \leq \boldsymbol{\tau}_{max},$$

where $\mathbf{J}_{l,j} \in \mathbb{R}^{3\times3}$ is the operational space foot Jacobian for the lth leg at the jth knot, and $\boldsymbol{\tau}_{max}$ are the joint torque limits of the leg. Unfortunately, convex computation of the Jacobian requires prior knowledge of the joint positions. Thankfully, the invariance of the operational space allows us to approximate it around a nominal value $\mathbf{J}_{l,j}^*$. Therefore, we approximate joint limits with the following constraint:

$$\mathbf{J}_{l,j}^{*T} \boldsymbol{\lambda}_{l,j} \leq \boldsymbol{\tau}_{max}, \quad (16)$$

This constraint approximates the Actuation Wrench Polytope (AWP) to define an approximation of the Feasible Wrench Polytope (FWP), as defined in [20]. In practice, this approximation is useful for most motions, close to the nominal position and without aggressive speeds. For further precision, one could use robust optimization [21] to constrain over the variations of the Jacobian $\mathbf{J}^*_{l,j} \pm \delta\mathbf{J}_{l,j}$.

I. Trajectory Optimization

Given the constraints stated above, we formulate a convex trajectory optimization, as follows:

$$\min_{\mathbf{r},\mathbf{k}_o,\mathbf{p}_l,\boldsymbol{\lambda}_l} g_T + \sum_{k=1}^{N} g(k),$$

where T is the total number of knots, $g(k)$ is a running cost along the plan and g_T is a terminal cost. Our running cost $g(k)$ maximizes the stability of the motion, while seeking for the fastest and smoothest gait. For this, the objectives are:

1) Minimize the CoM acceleration $\ddot{\mathbf{r}}$.
2) Minimize the contact forces magnitude $\|\boldsymbol{\lambda}\|$.
3) Minimize the upper bound of quadratic terms $\mathbf{U} = (\mathbf{u}^-, \mathbf{u}^+)$ used in the convex angular dynamics model.
4) Maximize the stability margin α.
5) Minimize the execution time, by minimizing the sum of the elements in the time vector $\mathbf{t}$.

The running cost $g(k)$ is defined as:

$$g(k) = \|\ddot{\mathbf{r}}_k\|_{\mathbf{Q}_v} + \|\boldsymbol{\lambda}_{l,k}\|_{\mathbf{Q}_F} + q_u\mathbf{U}_k + q_t t_k - q_\alpha\alpha_k,$$

On the other hand, the terminal cost g_T biases the plan towards its goal, the terminal position $\mathbf{r}_G$, as:

$$g_T = \|\mathbf{r}_T - \mathbf{r}_G\|_{\mathbf{Q}_g}, \tag{17}$$

where q are positive weights, $\mathbf{Q}$ are positive-semidefinite weighting matrices, and $\|\mathbf{v}\|_{\mathbf{Q}}$ stands for the weighted squared-L_2 norm $\mathbf{v}^T\mathbf{Q}\mathbf{v}$. In practice, we add a small cost to $\|\dot{k}\|_{Q_k}$ in order to generate smoother motions.

III. WHOLE-BODY CONTROL

The CoM motion, body attitude and swing motions are controlled by a *trunk controller*. It computes the feed-forward joint torques $\boldsymbol{\tau}^*_{ff}$ necessary to achieve a desired motion without violating friction, torques or kinematic limits. To address unpredictable events (e.g. limit foot divergence in case of slippage on an unknown surface), an impedance controller computes in parallel the feedback joint torques $\boldsymbol{\tau}_{fb}$ from the desired joint motion $(\mathbf{q}^d_j, \dot{\mathbf{q}}^d_j)$. Note that the desired body and joint motions have to be consistent with each other in order to prevent conflicts with the trunk controller.

To achieve compliantly desired trunk motions, we compute a reference CoM acceleration ($\ddot{\mathbf{r}}^r \in \mathbb{R}^3$) and body angular acceleration ($\dot{\boldsymbol{\omega}}^r_b \in \mathbb{R}^3$) through a *virtual model*:

$$\ddot{\mathbf{r}}^r = \ddot{\mathbf{r}}^d + \mathbf{K_r}(\mathbf{r}^d - \mathbf{r}) + \mathbf{D_r}(\dot{\mathbf{r}}^d - \dot{\mathbf{r}}),$$
$$\dot{\boldsymbol{\omega}}^r_b = \dot{\boldsymbol{\omega}}^d_b + \mathbf{K}_\theta e(\mathbf{R}^d_b\mathbf{R}^T_b) + \mathbf{D}_\theta(\omega^d_b - \boldsymbol{\omega}_b), \tag{18}$$

where $(\mathbf{r}^d, \dot{\mathbf{r}}^d, \ddot{\mathbf{r}}^d) \in \mathbb{R}^3$ are the desired CoM position, velocity and acceleration respectively, $e(\cdot) : \mathbb{R}^{3\times3} \to \mathbb{R}^3$

is a mapping from the rotation matrix into the associated rotation vector, $\boldsymbol{\omega}_b \in \mathbb{R}^3$ is the angular velocity of the trunk. $\mathbf{K_r}, \mathbf{D_r}, \mathbf{K}_\theta, \mathbf{D}_\theta \in \mathbb{R}^{3\times3}$ are positive-definite diagonal matrices of proportional and derivative gains, respectively.

The target of our trunk controller is to minimize the error between the reference and actual accelerations while enforcing friction, torque and kinematic constraints. As mentioned above, the reference accelerations are computed from (18). We formulate the problem using Quadratic Programming (QP) with the generalized accelerations and the contact forces as decision variables, i.e. $\mathbf{x} = [\ddot{\mathbf{q}}^T, \boldsymbol{\lambda}^T]^T \in \mathbb{R}^{6+n+3n_l}$:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} g_{err}(\mathbf{x}) + \|\mathbf{x}\|_{\mathbf{W}}$$
$$\text{s.t.} \quad \mathbf{Ax} = \mathbf{b} \tag{19}$$
$$\underline{\mathbf{d}} < \mathbf{Cx} < \bar{\mathbf{d}}$$

where n represents the number of active Degree of Freedoms (DoFs). The first term of the cost function (19) penalizes the *tracking* error:

$$g_{err}(\mathbf{x}) = \left\| \begin{array}{c} \ddot{\mathbf{r}} - \ddot{\mathbf{r}}^r \\ \dot{\boldsymbol{\omega}}_b - \dot{\boldsymbol{\omega}}^r_b \end{array} \right\|_{\mathbf{S}}, \tag{20}$$

while the second one is a regularization factor to keep the solution bounded or to pursue additional criteria. Both costs are quadratic-weighted terms. As the CoM acceleration is not a decision variable, we compute them from the contact forces using the centroidal dynamic model. We then re-write the tracking cost (20) as $\|\mathbf{Gx} - \mathbf{g}_0\|$ where:

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n} & \frac{1}{m}\mathbf{I}_1 \cdots \frac{1}{m}\mathbf{I}_{n_l} \\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times n} & \mathbf{0}_{3\times 3n_l} \end{bmatrix}, \mathbf{g}_0 = \begin{bmatrix} \ddot{\mathbf{r}}^r + \mathbf{g} \\ \dot{\boldsymbol{\omega}}^r_b \end{bmatrix},$$

and $\mathbf{I}_k$ representing an identity matrix for the kth end-effector. The equality constraints $\mathbf{Ax} = \mathbf{b}$ encodes dynamic consistency, stance condition and swing task. On the other hand, the inequality constraints $\underline{\mathbf{d}} < \mathbf{Cx} < \bar{\mathbf{d}}$ encode friction, torque, and kinematic limits.

We map the optimal solution $\mathbf{x}^*$ into desired feed-forward joint torques $\boldsymbol{\tau}^*_{ff} \in \mathbb{R}^n$ using the actuated part of the full dynamics of the robot as:

$$\boldsymbol{\tau}^*_{ff} = \begin{bmatrix} \mathbf{M}^T_{bj} & \mathbf{M}_j \end{bmatrix} \ddot{\mathbf{q}}^* + \mathbf{h}_j - \mathbf{J}^T_{cj}\boldsymbol{\lambda}^* \tag{21}$$

where $\mathbf{M}_{bj} \in \mathbb{R}^{(6+n)\times n}$ represents the coupled inertia between the floating-base and joints, $\mathbf{M}_j \in \mathbb{R}^{n\times n}$ the joint contribution to the inertia matrix, $\mathbf{h}_j \in \mathbb{R}^n$ is the force vector that accounts for Coriolis, centrifugal, and gravitational forces to the joint torque, and $\mathbf{J}_{cj} \in \mathbb{R}^{3n_l\times n}$ is a stack of Jacobians of the n_l end-effectors.

Finally, the feed-forward torques $\boldsymbol{\tau}^*_{ff}$ are summed with the joint PD torques (i.e. feedback torques $\boldsymbol{\tau}_{fb}$) to form the desired torque command $\boldsymbol{\tau}^d$:

$$\boldsymbol{\tau}^d = \boldsymbol{\tau}^*_{ff} + PD(\mathbf{q}^d_j, \dot{\mathbf{q}}^d_j), \tag{22}$$

which is sent to a low-level joint-torque controller.

Fig. 8: Snapshots of experiments 1 through 3, used to evaluate the performance of our planner on challenging terrain. *Top*: Climbing a pallet and crossing a 20 cm gap, while climbing 3 cm. *Middle*: Climbing a 10° slope and crossing a 15 cm gap leading to a −10° slope. *Bottom*: Climbing a 10° slope and crossing a 20 cm gap, while climbing down 6 cm leading to an 11 cm stair.

IV. EXPERIMENTAL VALIDATION

We validated our approach on the HyQ robot [22], a 85 kg hydraulically actuated quadruped robot, traversing various challenging terrains while also optimizing the gait transition [1]. The HyQ robot is fully-torque controlled and equipped with high-precision joint encoders, a Multisense SL sensor (Carnegie Robotics) and an Inertial Measurement Unit (KVH). HyQ has approximately the dimensions of a goat: 1.0 m×0.5 m×0.98 m (length×width×height). The planned motions are computed off-board using the Drake Toolbox [23] in MATLAB 2015a, on a Dual-Core Laptop running Mac OS X Sierra. We use Gurobi 6.0.5 [24] as our MIQCQP solver. The whole-body controller and state estimation run fully on-board and in real-time. Our state estimation is based on a modular inertial-driven Extended Kalman Filter which incorporates a rugged probabilistic leg odometry component with additional inputs from stereo vision and LIDAR registration, as described in [25].

A. Locomotion on rough terrain

In our first experiments we showcase the different capabilities of our approach by navigating over various challenging terrains composed of gaps, ramps, and steps with different terrain heights (Fig. 8). Our method allows the robot to successfully cross those terrains because of the use of a) a convex model of robot's dynamics and terrain, b) an approximation of the torque limits, and c) stability maintenance capabilities in non-coplanar contact conditions.

1) Convex model of robot's dynamics and terrain: Traversing a gap while climbing requires to accommodate properly the body position because of the robot's kinematic limitations. In Fig. 8 (*top*), we show that our method exploits properly the leg reachability, requiring fewer contact locations to reach the goal compared with [8]. In fact, having a convex model of the robot's dynamics, and especially of the terrain, avoids to fall into a local minimum because of the global optimality guarantees. This convex terrain model
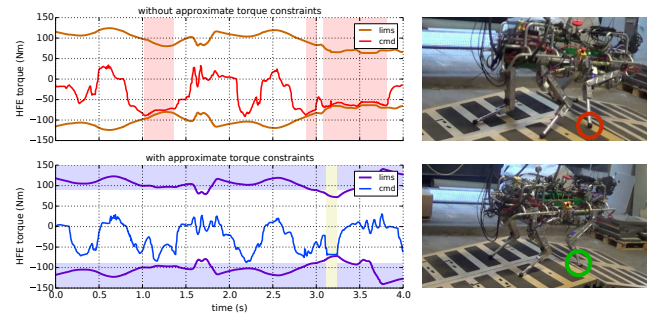

Fig. 9: Comparison of climbing up and down two slopes with (*bottom*) and without (*top*) approximate torque constraints. Figures show the real torque limits (*lims*) and the torque (*cmd*) for the RF hip pitch joint (HFE).

works well for most scenarios, however, it might lose validity when the terrain has significant non-linear curvature.

2) Approximate torque constraints: We compare two different planned motions, with and without approximate torque constraints in the optimization (see Fig. 9). While the robot climbed up successfully the slope in both cases, it was able to cross the gap only when the torque limits were considered. In fact, in these scenarios, the admissible set of contact forces is reduced due to the friction cones pointing outwards from the robot's CoM. Here, only the approximate torque constraints (highlighted in blue in Fig. 9) account for this reduction of the feasible wrench set, and as result, the contact forces are distributed accordingly, see Fig. 9 (*bottom*)[2]. However, if we do not consider the approximate torque constraints, the robot reaches torque limits three times, red shade in Fig. 9 (*top*), and it falls while climbing down, see Fig. 9 (*top-right*).

3) Stability in non-coplanar contact conditions: Crossing a gap and ascending up stairs is a challenging task, because the robot has to maintain stability while considering potential non-coplanar contacts, see Fig. 8 (*bottom*). For instance, ZMP-based approaches such as [8], [9] cannot ensure stability in such terrains. We experimentally found an increment on the robustness of the planned motion compared

---

[1]video with the experiments: https://youtu.be/s4PMXpbwUes

[2]Note that HyQ's torque limits depend on the joint position, resulting in limit differences in the yellow shade on Fig. 9 (*bottom*), which are not accessible using the centroidal dynamics and the quasi-static assumption.

Fig. 10: Snapshots of experiment 4, used to evaluate the performance of different gaits generated with our planning scheme in a roof-like terrain. *Top*: Fixed walking gait. *Middle*: Fixed trotting gait. *Bottom*: Automatically discovered gait.

to the aforementioned methods. This robustness is especially important because we rely on a reference for the robot attitude modulation as in [8]. This is done because we restrict our trajectories to small angular momentum changes, because of limitations in the convex model (see Section II-G).

B. Gait transitions

Afterwards, we tested the gait planning and transitioning capabilities of our approach. For this, we run trials over a challenging terrain with non-coplanar contacts to compare two different scenarios: a) crossing with a fixed gait that is predefined, walk and trot, and b) crossing while the gait sequence is optimized. Figure 10 shows snapshots of these experimental trials.

1) Fixed walking and trotting gaits: In these cases, as expected, the motions are executed robustly across the entire trial. For walking, we obtained motions of quality similar to those of the previous experiments. In the case of the trotting, we increased the execution speed compared to the walking case (from 18 cm/s to 31 cm/s). This fast motion is effectively tracked by our controller, despite relatively slow corrections, from the visual sources, in the state estimation.

2) Automatic gait discovery: In this trial, we showcase the trade-offs involved when optimizing the gait transitions Fig. 10 (*bottom*). First, in order to minimize acceleration, our motion planner selects a walking gait before smoothly transitioning to a dynamic trotting gait. As expected, the robot continues trotting until it faces a significant change of terrain. At that moment, the robot transitions to a slower gait (i.e. walking gait) in order to accommodate for the terrain conditions. For comparison, Fig. 11 shows the minimum α margin at each time-step, normalized by its contact force. Finally, it returns to a trotting gait as is detailed in Fig. 11. In contrast to [3], we do not include any terrain heuristics in the planning stage, resulting in an automatic gait discovery based on the centroidal dynamics of the robot.

C. Discussion

The experimental trials presented above showcase the main aspects of our approach. As expected, all of our plans

TABLE II: Computation time for a locomotion cycle for various terrain conditions and gaits

Experiment	convex surfaces	Gait	mean time (s)
Exp. 1	3	Walk	0.47
Exp. 2	3	Walk	0.64
Exp. 3	4	Walk	0.44
Exp. 4	3	Walk	0.48
Exp. 4	3	Trot	0.51
Exp. 4	3	Free	1.62

result in successful executions. Thanks to our incorporation of approximate torque limits, we are able to push the limits of our hardware, reaching faster walking speeds (around 15 cm/s) than previous approaches executed on the same robot [2], [8], [9]. Additionally, we have shown that our approach is able to handle gait transitions robustly even on challenging terrain. In our experiments, the convex approximation of the angular dynamics helps the planner to generate natural motions. Nonetheless, we found that this model does not extend well to more dynamical gaits, such as bounding or pace, since it cannot impose limits in the angular momentum, causing divergence.

Computation time is one of the main concerns in any optimization problem. In Table II, we report the required computation time (for a single locomotion cycle) of the above scenarios. We plan motion at least two orders of magnitude faster than previous work on similar terrains [8], [12] because of our convex formulation using MIQCQP. The kinematic constraints imposed help the solver to quickly discard infeasible surfaces. This captures the main advantage of our method, that it is able to efficiently plan motions through 3D environments with obstacles, while maintaining stability despite changes in the terrain.

Other factors might influence the computation time. For instance, leaving the gait as a decision variable increases computation time significantly. As in [1], we found that the incorporation of piecewise-affine trigonometric functions has a negative effect on the computation time. Likewise, quadratic constraints in (15) also influence the computation time, resulting in increases of around 500%. However, we believe that our method can potentially be extended to receding horizon fashion.
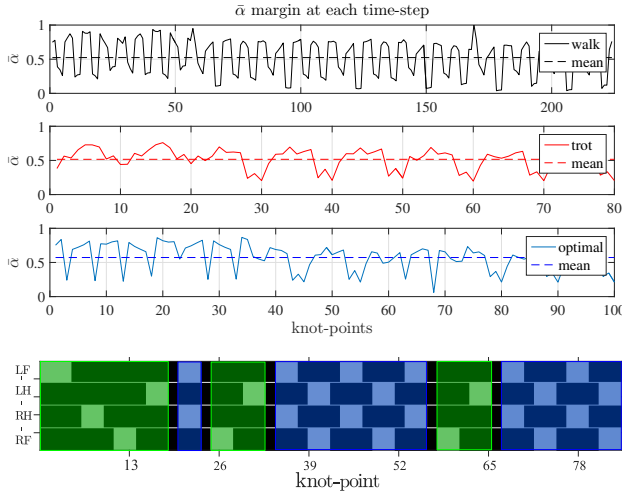
Fig. 11: *Top:* Normalized α margin for different gaits. *Bottom:* Resulting optimal gait sequence for navigating in roof-like terrain, blue and green segments represent trot and walk, respectively.

V. Conclusions

We have presented a novel approach for simultaneously planning contacts and motions on multi-legged robots based on MICP. Our approach is able to handle complex terrain, while also providing formal robustness guarantees on the plan. Moreover, we incorporate the gait sequences as a decision variable, which allows for automatic gait discovery. Our experimental trials use both a state-of-the-art whole-body controller and state estimation [8], [25]. We demonstrate the capabilities of our approach by traversing challenging terrains with the HyQ robot. To our knowledge, these hardware experiments constitute the first experimental validation of non-gaited uneven locomotion via friction cone stability margin. Moreover, our implementation is able to plan locomotion cycles in less than a second, even in complex scenarios, which is at least two orders of magnitude faster than previous non-convex approaches [8], [10].

In future work, we plan to expand on the capabilities of real-time planning that this approach might offer, possibly reaching similar computation time to [9]. To this end, friction cone constraints could be relaxed to vertex-based ZMP constraints [9], sacrificing stability guarantees but improving computation time. Here, we have relied on a convex decomposition of the angular dynamics. This approach works well for simple motions where angular momentum rates are often low. However, it can cause the angular momentum to diverge when the contact forces vary significantly. Because of this, we are interested in exploring other models of angular dynamics, like those proposed in [14], so as to generate more aggressive behaviors. In this work, we have approximated torque limits with a nominal operational space Jacobian that is updated iteratively. While this works well for the experiments performed, it loses validity for more aggressive motions. For this reason, we are interested in studying other approximations of torque limits. We are further interested in exploiting the piecewise affine structure of the multi-contact dynamics in the receding horizon.

References

[1] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Humanoids*. IEEE, 2014.

[2] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *ICRA*. IEEE, 2015.

[3] B. Aceituno-Cabezas, H. Dai, J. Cappelletto, J. C. Grieco, and G. Fernandez-Lopez, "A mixed-integer convex optimization framework for robust multilegged robot locomotion planning over challenging terrain," in *IROS*. IEEE, 2017.

[4] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," 2016.

[5] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE Transactions on Biomedical Engineering*, vol. BME-16, no. 1, 1969.

[6] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots-adios zmp," in *ICRA*. IEEE, 2006.

[7] P.-B. Wieber, "On the stability of walking systems," in *International workshop on humanoid and human friendly robotics*, 2002.

[8] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *ICRA*, 2017.

[9] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast trajectory optimization for legged robots using vertex-based zmp constraints," *RA-L*, 2017.

[10] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Humanoids*. IEEE, 2014.

[11] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, 2014.

[12] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, 2017.

[13] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Humanoids*. IEEE, 2016.

[14] A. K. Valenzuela, "Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[15] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 2013.

[16] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *Humanoids*. IEEE, 2016.

[17] A. Richards and J. How, "Mixed-integer programming for control," in *ACC*. IEEE, 2005.

[18] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Hierarchical planning of dynamic movements without scheduled contact sequences," in *ICRA*. IEEE, 2016.

[19] A. Del Prete, S. Tonneau, and N. Mansard, "Fast algorithms to test robust static equilibrium for legged robots," in *ICRA*. IEEE, 2016.

[20] R. Orsolino, M. Focchi, and C. Semini, "The Actuation-consistent Wrench Polytope (AWP) and the Feasible Wrench Polytope (FWP)," 2017.

[21] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, 2011.

[22] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, 2011.

[23] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: http://drake.mit.edu

[24] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[25] S. Nobili, M. Camurri, V. Barasuol, M. Focchi, D. Caldwell, C. Semini, and M. Fallon, "Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots," in *RSS*, 2017.