

# Trajectory and Foothold Optimization using Low-Dimensional Models for Rough Terrain Locomotion

Carlos Mastalli<sup>1</sup>, Michele Focchi<sup>1</sup>, Ioannis Havoutis<sup>2,4</sup>, Andreea Radulescu<sup>1</sup>,  
Sylvain Calinon<sup>2</sup>, Jonas Buchli<sup>3</sup>, Darwin G. Caldwell<sup>1</sup>, Claudio Semini<sup>1</sup>

**Abstract**—We present a trajectory optimization framework for legged locomotion on rough terrain. We jointly optimize the center of mass motion and the foothold locations, while considering terrain conditions. We use a terrain costmap to quantify the desirability of a foothold location. We increase the gait’s adaptability to the terrain by optimizing the step phase duration and modulating the trunk attitude, resulting in motions with guaranteed stability. We show that the combination of parametric models, stochastic-based exploration and receding horizon planning allows us to handle the many local minima associated with different terrain conditions and walking patterns. This combination delivers robust motion plans without the need for warm-starting. Moreover, we use soft-constraints to allow for increased flexibility when searching in the cost landscape of our problem. We showcase the performance of our trajectory optimization framework on multiple terrain conditions and validate our method in realistic simulation scenarios and experimental trials on a hydraulic, torque controlled quadruped robot.

## I. INTRODUCTION

Legged locomotion can deliver substantial advantages in unstructured real-world environments as it can offer mobility unmatched by traditional vehicles. Such environments are common in disaster relief, search and rescue, forestry and construction site scenarios. Despite significant efforts in the field, motion planning and control over rough terrain remains an open problem. Moving across challenging environments requires planning motions while taking into account future terrain conditions. In these situations, the Center of Mass (CoM) motion and foothold selection must be jointly planned, while handling high-level user commands and selecting appropriate behaviors given the terrain topology.

Recently, trajectory optimization with contacts gained a lot of attention in the legged robotics community [1][2][3]. These optimization problems are often hard to solve, and the automatic synthesis of behaviors may be limited by the non-convexity of such domains, e.g. due to local minima. However, adaptation and automatic gait discovery can be solved using low-dimensional parametric models that capture the most relevant dynamics [4]. In fact, a combination

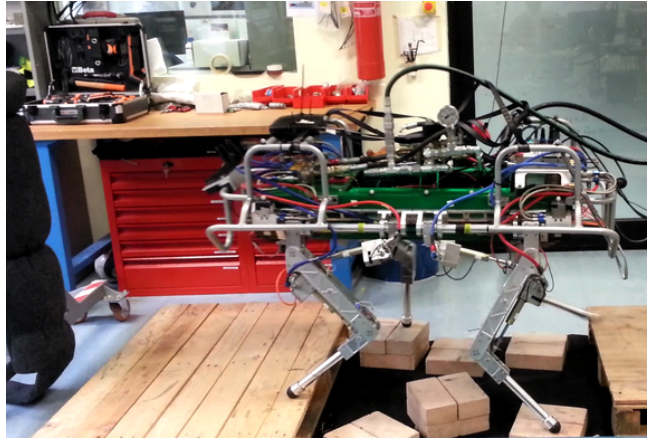


Fig. 1. The hydraulically actuated and fully torque controlled quadruped robot HyQ [5]. Here crossing over a set of sparse stepping stones of varying height.

of low-dimensional parametric models and stochastic-based exploration may be able to generate effective behaviors even without warm-starting the exploration.

This paper presents a trajectory optimization method for quadrupedal robots. We address the locomotion as a coupled planning problem of CoM motions and footholds, where the foothold locations are selected using a terrain costmap while the trunk height and attitude are adapted for coping with different terrain elevations. First, we optimize a sequence of control parameters (the Center of Pressure (CoP) displacement, the phase duration and the foothold locations) given the terrain costmap. Then, we jointly generate the CoM trajectory and the swing-leg trajectory using a sequence of parametric preview models and the terrain elevation map. To realize the low-dimensional plan, the controller selects appropriate torque commands, which are computed by the combination of a trunk controller with a joint-space torque controller. The proposed trajectory optimization method increases the locomotion capabilities of our legged robot, compared to our previous framework [6][7]. As shown in Fig. 1, our trajectory optimization framework generates motions that allow the Hydraulically actuated Quadruped (HyQ) robot to cross a set of sparse stepping stones of varying height.

The main contribution of this paper is a novel trajectory optimization approach for locomotion on rough terrain. In contrast to [4], we consider terrain topologies (in the form of a terrain costmap) for foothold selection in our trajectory optimization. Our method is capable of producing a wide range

<sup>1</sup>Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego, 30, 16163 Genova, Italy. *email*: {carlos.mastalli, michele.focchi, andreea.radulescu, darwin.caldwell, claudio.semini}@iit.it.

<sup>2</sup>Robot Learning and Interaction Group, Idiap Research Institute, Martigny, Switzerland. *email*: {ioannis.havoutis, sylvain.calinon}@idiap.ch

<sup>3</sup>Agile and Dexterous Robotics Lab, ETH Zurich, Zurich, Switzerland. *email*: buchli@ethz.ch

<sup>4</sup>Oxford Robotics Institute, Department of Engineering Science, University of Oxford, United Kingdom. *email*: ihavoutis@robots.ox.ac.uk

This work was in part supported by the DexROV project through the EC Horizon 2020 programme (Grant #635491).

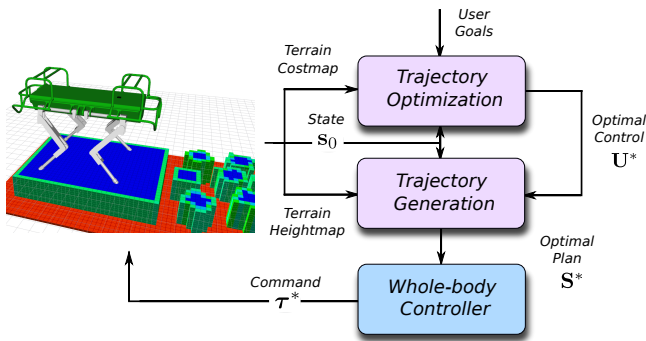


Fig. 2. Overview of the trajectory optimization framework for locomotion on rough terrain. We compute offline an optimal control sequence  $\mathbf{U}^*$  given the user’s goals, the actual state  $\mathbf{s}_0$  and the terrain costmap. Given this optimal control sequence, we generate the optimal plan  $\mathbf{S}^*$  that copes with the changes in the terrain elevation through trunk attitude planning. Lastly, the whole-body controller calculates the joint torques  $\boldsymbol{\tau}^*$  that satisfy friction-cone constraints.

of different locomotion behaviors using low-dimensional parametric models. The combination of these models with stochastic-based exploration and receding horizon planning helps us to automatically synthesize desired behaviors, that are critical for rough terrain locomotion. Moreover, we tackle the different terrain elevations by modulating the trunk height and attitude, and planning in the horizontal frame<sup>1</sup>. Additional contributions include trajectory optimization with different terrain costmaps, a suitable terrain description through a cost function, that makes the optimization more efficient, and a method for guaranteeing the dynamic stability when the robot adjusts the attitude of its trunk (Fig. 2). To the best of our knowledge, our approach is the first that jointly optimizes the CoM motion, the phase duration and the foothold selection *while considering terrain topology*.

The paper is structured as follows: after discussing previous research in the field of legged locomotion and trajectory optimization (Section II), we describe how to generate the CoM trajectories from a sequence of parametric preview models (Section III). Next, we describe our trajectory optimization framework for legged locomotion on rough terrain (Section IV). Section V shows how these desired motions are accurately and compliantly executed. In Section VI we evaluate the performance of our trajectory optimization approach in experimental trials on the HyQ robot [5]. Finally Section VII draws the conclusions and presents ideas for future work.

## II. RELATED WORK

Extensive research has been conducted in the field of quadrupedal locomotion on challenging terrain. A number of successful control architectures [6][8][9][10][11] that plan and execute footsteps for traversing such terrain have been proposed. Some avoid global footstep planning by simply choosing the next best reachable footholds [11], while others

<sup>1</sup>A horizontal frame is a reference frame whose  $xy$  plane is orthogonal to the gravity vector  $\mathbf{g}$  with the same origin as that of the base frame. Thus is a frame that is moving with the robot.

plan the complete footstep sequence from start to goal [10][12]. In most of the aforementioned approaches a sequence of footholds is selected using only kinematic criteria (i.e. decoupled from CoM planning). Furthermore, those approaches consider a fixed step duration which limits the richness of possible behaviors in rough terrain locomotion.

Terrain adaptation and automatic gait discovery can be approached using general trajectory optimization methods, similar to [1][2][3][13]. Nonetheless, these optimization methods tend to be plagued by local minima, limiting their applicability to rough terrain locomotion. Often, challenging terrain conditions may increase the non-convexity of the problem, and defining a good enough warm-start point might not be possible. Moreover, such approaches are typically computationally expensive, making them harder to integrate within an online locomotion framework. We believe that automatic walking pattern generation can be solved using low-dimensional models that better handle the problems related to local minima (i.e., by reducing the problem dimensionality). We use stochastic-based optimization to solve such non-convex problems. A similar approach has been recently proposed for bipedal locomotion on an animated character in [4]. The authors defined a simple preview schedule that allows the character to generate three behaviors: standing, walking and running on even terrain. On the other hand, a solution using hierarchical combination of optimization steps, that can be used to deal with local minima problems and guarantees joint constraints, was proposed in [14].

Preview models have been extensively used for legged locomotion on flat terrain, e.g. [15]. These approaches often decouple the foothold selection from the CoM planning, assuming a fixed step duration, and do not consider attitude planning [6][8][9]. However, in field applications, the terrain conditions are often rough and irregular. Below we describe our trajectory optimization framework, which does not suffer from the aforementioned drawbacks of the current approaches for rough terrain locomotion.

## III. TRAJECTORY GENERATION

This section describes the low-dimensional trajectory generation from an optimized sequence of control parameters and a given terrain heightmap. We generate the horizontal CoM trajectory and the 2D foothold locations using a sequence of low-dimensional preview models. In order to cope with the terrain elevation, we modulate the trunk attitude and height using an estimate of the support plane, and the maximum allowed angular accelerations of the trunk (for more details see Section III-A.2). We describe the sequence of control parameters w.r.t. the horizontal frame, which allows us to decouple the CoM and trunk attitude planning.

### A. Preview model

Preview models are low-dimensional representations that describe and capture different locomotion behaviors, such as walking and trotting, and provide an overview of the motion [4]. By reducing the dimensionality of the optimization problem we can generate complex locomotion behaviors and

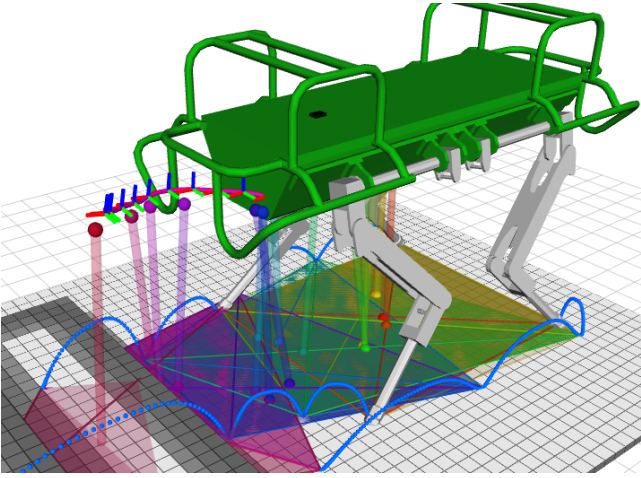


Fig. 3. A trajectory obtained from a low-dimensional model given a sequence of optimized control parameters and the terrain heightmap. The colored spheres represent the CoM and CoP positions of the terminal states of each motion phase. The CoP spheres lie inside the support polygon (same color is used). Note that color indicates the phase (from yellow to red). The trunk adaptation is based on the estimated support planes in each phase. Since the control parameters are expressed in the horizontal frame, the horizontal CoM trajectories and the trunk attitude are decoupled.

their transitions. This is more suitable for rough terrain as it simplifies the problem landscape. In the literature, different models that capture the legged locomotion dynamics have been studied [16][17] such as point-mass, inverted pendulum, cart-table, or contact wrench.

Our preview model decouples the CoM motion from the trunk attitude<sup>2</sup> (Fig. 3). For the CoM motion, we use the cart-table template [15]. The cart-table model (linear inverted pendulum) encompasses a point mass assumption which has no angular momentum. However, to control the attitude we need to apply a torque to the CoM. High centroidal moments (e.g. due to high trunk angular acceleration) can hamper the postural stability condition (e.g. causing shifts on the CoP that can move it out of the support polygon [18]) making the robot loses its capability to balance. Consequently, for the attitude planning, we limit the maximum moments applied to the CoM by limiting the maximum angular acceleration and setting a correspondent margin for the CoP on the support polygon (Section III-A.2).

1) *CoM motion*: In our previous work [6], we showed that for fixed step durations, the CoP movement is approximately linear, i.e.:

$$\mathbf{p}^H(t) = \mathbf{p}_0^H + \frac{\delta\mathbf{p}^H}{T}t. \quad (1)$$

Note that  $\mathbf{p}^H \in \mathbb{R}^2$  is the horizontal CoP position,  $\delta\mathbf{p}^H \in \mathbb{R}^2$  the horizontal CoP displacement and  $T$  is the phase duration.

Applying this linear control law in the cart-table model, we derive an analytic solution for the horizontal dynamics [4]:

$$\mathbf{x}^H(t) = \beta_1 e^{\omega t} + \beta_2 e^{-\omega t} + \mathbf{p}_0^H + \frac{\delta\mathbf{p}^H}{T}t, \quad (2)$$

<sup>2</sup>In this work, with “trunk attitude” we refer to roll and pitch only.

where the model coefficients  $\beta_{1,2} \in \mathbb{R}^2$  depend on the actual state  $\mathbf{s}_0$  (horizontal CoM position  $\mathbf{x}_0^H \in \mathbb{R}^2$  and velocity  $\dot{\mathbf{x}}_0^H \in \mathbb{R}^2$ , and CoP position), the trunk height  $h$ , the phase duration, and the horizontal CoP displacement:

$$\beta_1 = (\mathbf{x}_0^H - \mathbf{p}_0^H)/2 + (\dot{\mathbf{x}}_0^H T - \delta\mathbf{p}^H)/(2\omega T),$$

$$\beta_2 = (\mathbf{x}_0^H - \mathbf{p}_0^H)/2 - (\dot{\mathbf{x}}_0^H T - \delta\mathbf{p}^H)/(2\omega T),$$

where  $\omega = \sqrt{g/h}$  and  $g$  is the gravity acceleration.

2) *Trunk attitude*: A trunk attitude modulation is required when the terrain elevation varies. A simple approach consists of aligning the trunk with respect to the estimated support plane, avoiding that the robot reaches its kinematic limits. On the other hand, adjusting the trunk attitude requires applying a moment at the CoM, and as consequence, the CoP  $\mathbf{p} \in \mathbb{R}^3$  will be shifted by a proportional amount  $\Delta\mathbf{p}$  (for more details see (5) in [18]):

$$\Delta p_x = -\tau_{\text{com}_y}/mg, \quad (3)$$

$$\Delta p_y = \tau_{\text{com}_x}/mg,$$

where  $\tau_{\text{com}_y}$ ,  $\tau_{\text{com}_x}$  are the horizontal components of the moment about the CoM. By exploiting a simplified flywheel model for the inertia of the robot we can link these moments to the CoP displacement  $\Delta\mathbf{p}$  (rewritten in vectorial form) and to the angular acceleration  $\dot{\omega}$ :

$$\tau_{\text{com}} = \mathcal{I}\dot{\omega}, \quad (4)$$

$$\Delta\mathbf{p} = \tau_{\text{com}} \times mg. \quad (5)$$

where  $\mathcal{I} \in \mathbb{R}^{3 \times 3}$  is the time-invariant inertial tensor approximation of the centroidal inertia matrix of the robot. Therefore, we can guarantee the CoP condition by limiting the angular accelerations  $\dot{\omega}_{\text{max}}$  (i.e. the allowed applied moments) and setting a correspondent safety margin  $r$  on the support polygon in our optimization (Section IV-C) as:

$$r = \|(\mathcal{I}\dot{\omega}_{\text{max}}) \times mg\|. \quad (6)$$

We adapt the trunk attitude in such a way that it does not affect the CoP condition (i.e. by using the maximum allowed angular acceleration  $\dot{\omega}_{\text{max}}$ ). Note that we compute  $\dot{\omega}_{\text{max}}$  given the stability margin  $r$  (i.e., the support polygon margin).

We employ cubic polynomial splines to describe the trunk attitude motion (pitch and roll). The attitude adaptation can be done in different phases. For instance, we can compute the required angular accelerations given the phase duration and guarantee that it does not exceed the allowed angular accelerations. The trunk height is computed given the estimated support plane and we keep it constant along one phase.

## B. Preview schedule

Describing quadrupedal locomotion can be achieved through a sequence of different preview models — a preview schedule. Using this, the robot can automatically discover different foothold sequences by enabling or disabling different phases in our optimization process.

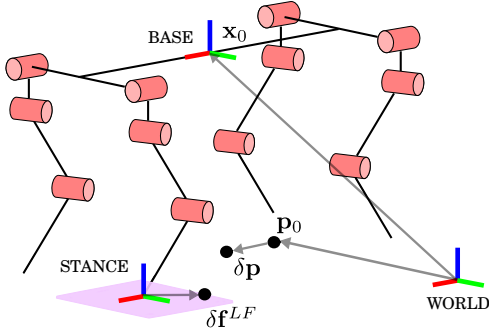


Fig. 4. Robot kinematics showing different variables and frames used in our optimization. The footshift  $\delta \mathbf{f}^{LF}$  is described w.r.t. the stance frame. The stance frame is calculated from the default posture and expressed w.r.t. the base frame.

In the preview schedule, we build up a sequence of control parameters that describes the locomotion action of the  $n$  phases:

$$\mathbf{U} = [\mathbf{u}_1^{s/f} \quad \dots \quad \mathbf{u}_n^{s/f}], \quad (7)$$

where  $\mathbf{u}_i^s = [T \quad \delta \mathbf{p}^{H^T}]^T$  and  $\mathbf{u}_i^f = [T \quad \delta \mathbf{p}^{H^T} \quad \delta \mathbf{f}^{l^T}]^T$  are the preview control parameters for the stance and step phases, respectively. Additionally, the footshift  $\delta \mathbf{f}^l$  is described with respect to the stance frame (Fig. 4), which is calculated from the default posture of the robot. Note that  $n$  is the number of phases, and  $l$  is the foot index.

We describe a dynamic walking gait as a combination of 6 different preview phases or timeslots (i.e.  $n = 6$ ) where 4 of them are step phases. Our combination of phases is stance, LH swing phase, LF swing phase, stance, RH swing phase and RF swing phase<sup>3</sup>. With this fixed preview schedule, we can describe different walking patterns by assigning a zero duration to a specific phase ( $T_i = 0$ ).

#### IV. TRAJECTORY OPTIMIZATION

The trajectory optimization computes a sequence of control parameters  $\mathbf{U}^*$  used for the generation of the low-dimensional trajectories (Section III). We formulate this as a receding horizon trajectory optimization problem, where the current timeslot is optimized while taking future timeslots into account. The horizon is described by a predefined number of preview schedules  $N$  with  $n$  timeslots or phases (e.g. our locomotion cycle has 6 timeslots). Considering future phases presents several advantages for rough terrain locomotion. It enables us to generate desired behaviors that anticipate future terrain conditions, and it results in smoother transitions between phases.

In our approach, the optimal solution at the current phase  $i$  comprises of a set of control parameters  $\mathbf{u}_i^*$  describing the duration of phase  $T_i^*$ , the CoP displacement  $\delta \mathbf{p}^{H_i^*}$ , and the footshift  $\delta \mathbf{f}_i^*$  of the corresponding phase. We define the footshift in the nominal stance frame which corresponds to

<sup>3</sup>The robot is in stance phase when all the feet are on the ground. LH, LF, RH and RF are Left-Hind, Left-Front, Right-Hind and Right-Front legs, respectively.

the default posture. Note that there are phases without foot swing.

#### A. Receding horizon planning

Given an initial state  $\mathbf{s}_0$ , we optimize a sequence of control parameters inside a predefined horizon, and apply the optimal control of the current phase. We find the sequence of control parameters  $\mathbf{U}^*$ , through an unconstrained optimization problem, given the desired user commands (trunk velocities):

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmin}} \sum_j \omega_j g_j(\mathbf{S}(\mathbf{U})), \quad (8)$$

where  $\mathbf{S} = [\mathbf{s}_1 \quad \dots \quad \mathbf{s}_{Nn}]$  is the sequence of preview states. The preview state is defined by the CoM position and velocity  $(\mathbf{x}, \dot{\mathbf{x}})$ , CoP position  $\mathbf{p}$  and the stance support region  $\mathbf{F}$ , i.e.  $\mathbf{s} = [\mathbf{x} \quad \dot{\mathbf{x}} \quad \mathbf{p} \quad \mathbf{F}]$ . Where  $\mathbf{F} = [\mathbf{f}_1 \quad \dots \quad \mathbf{f}_j]$  is defined by the position of the stance feet  $\mathbf{f}_j$ . We solve the trajectory optimization using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [19]. CMA-ES is capable of handling optimization problems that have multiple local minima, such as those introduced by the costmap and the phase duration. In the description of our optimization problem, we use soft-constraints as these provide the required freedom to search in the landscape of our optimization problem. The cost functions and soft-constraints  $g_i(\mathbf{S})$  describe: 1) the user command tracking with step duration and length, and travel direction, 2) the CoM energy, 3) the terrain cost, 4) stability soft-constraint, the i.e. CoP condition, and 5) the preview model soft-constraint, i.e. the linear inverted pendulum.

#### B. Cost functions

We encode the desired body velocity from the user by mapping it into a ‘default’ step duration and length. Additionally, the CoM trajectory should accelerate as little as possible during the phases. Note that this implicitly reduces the required joint torques. We evaluate the step duration and length in every locomotion phase  $i$  as follows:

$$g_{step-duration} = \left( \sum_{i=1}^{Nn} (t_i - i T_{step}) \right)^2, \quad (9)$$

$$g_{step-length} = \left( \sum_{i=1}^{Nn} (d_i - i d_{step}) \right)^2, \quad (10)$$

where  $t_i$  is the sum of individual durations until phase  $i$ ,  $T_{step}$  the desired step duration,  $d_i = \mathbf{d}^T(\mathbf{x}_i - \mathbf{x}_0)$  is the displacement of the CoM along the desired travel direction expressed in the horizontal frame (defined by the desired yaw angle), and  $d_{step}$  is the desired step length.

To encourage movements in the desired travel direction, we penalize lateral drift just in the 4-feet stance phase:

$$g_{step-drift} = \left( \sum_{i=1}^{Ns} d_i^\perp \right)^2, \quad (11)$$

where  $s$  is the number of 4-feet stance phase per locomotion cycle,  $d_i^\perp$  is the orthogonal vector of the desired travel direction. Note that step duration and length define the

desired linear velocity and the lateral drift defines the desired yaw angular velocity of the trunk. This choice of cost terms encourages equal trunk velocities between all the locomotion phases.

Minimizing the changes in the CoM accelerations reduces the required joint torques. We achieve this by applying:

$$g_{com-energy} = \sum_{i=1}^{Nn} \int_0^{T_i} \|\ddot{\mathbf{x}}(t)\|_2 dt. \quad (12)$$

To cope with different terrain difficulties, we compute a costmap from an onboard sensor as proposed in [12]. The costmap quantifies how desirable it is to place a foot at a specific location using geometric features such as *height standard deviation*, *slope* and *curvature*. This allows the robot to negotiate different terrain conditions (Fig. 5). Thus, given a footshift and CoM position, we compute the foothold location cost as:

$$g_{terrain} = \mathbf{w}^\top \mathbf{T}(x, y), \quad (13)$$

where  $\mathbf{w}$  and  $\mathbf{T}(x, y)$  are the weights and feature values, respectively. We use a cell grid resolution of 4 cm, approximately equal to the robot's foot size, and the terrain features are computed from a voxel resolution of 2 cm. As in [12], we demonstrated that this coarse map is a good trade-off in terms of computation time and information resolution for foothold selection. We cannot guarantee convexity in the terrain costmap, which has to be considered in our optimization process.

### C. Soft constraints

As we mentioned in Section III-A.2, the CoP trajectory must be kept inside the support polygon which is shrunk by a margin  $r$ . This margin guarantees dynamic stability when a maximum moment is applied to the CoM (Section III-A.2). We use a set of nonlinear inequality constraints to describe the support region:

$$\mathbf{l}(\mathbf{F})^\top \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} > \mathbf{0}, \quad (14)$$

where  $\mathbf{l}(\cdot) \in \mathbb{R}^{l \times 3}$  are the coefficients of the  $l$  lines,  $\mathbf{F}$  the support region defined from the selected foothold locations, and  $\mathbf{p}$  the CoP position. Note that the stability constraints are nonlinear as a consequence of adding the foothold positions as decision variables.

Due to the decoupling of the horizontal and vertical motions, we implement a preview model soft-constraint that ensures the cart-table height is approximately equal to:

$$h = \|\mathbf{x} - \mathbf{p}\| \quad (15)$$

where  $\mathbf{x}$  and  $\mathbf{p}$  are the CoM and CoP positions, respectively. Note that when the cart-table is falling down, the CoM trajectory increases exponentially in (2). This effect arises from the fact that we decouple the horizontal and vertical dynamics, hence adding this soft-constraint guarantees the validity of the model.

To reduce the computation time, we impose both soft-constraints only in the initial and terminal time of each

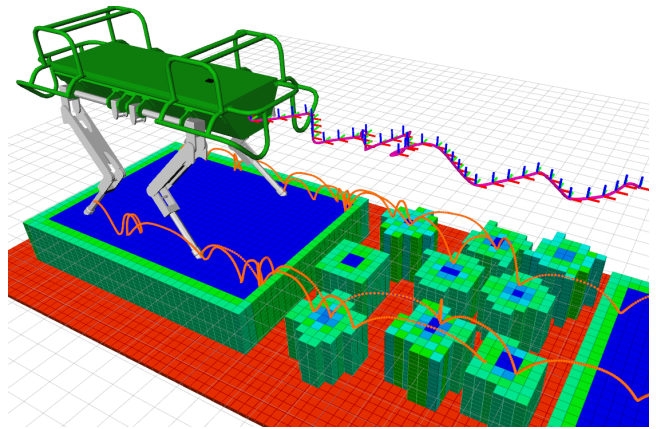


Fig. 5. We used a costmap which allows the robot to negotiate different terrain conditions while following the desired user commands. The costmap is computed from onboard sensors as described in [12]. The cost values are continuous and represented in color scale, where blue is the minimum and red is the maximum cost.

phase, as they will be guaranteed in the entire phase. In fact, the linear CoP trajectory will belong to the convex support polygon if the initial and terminal positions are inside this region. We ensure this by limiting the foothold search region, i.e. by bounding the footshift (see Fig. 4). These soft-constraints are described as quadratic cost terms.

## V. WHOLE-BODY CONTROLLER

The motion of the robot body (CoM and trunk orientation) is controlled by a trunk controller [20] that computes the joint torques necessary to achieve the desired motions without violating friction constraints.

At the joint-space level, an impedance controller is acting in parallel to address unpredictable events, such as a foot slippage on an unknown surface. This controller receives a set-point which is consistent with the body motion in order to prevent a conflicting target with the trunk controller. In nominal operations the biggest part of the torques is generated by the trunk controller.

Our aim for balancing is to control the position of the robot's CoM, and the orientation of the trunk (base link). We compute a desired linear acceleration for the CoM ( $\ddot{\mathbf{x}}_{com}^d \in \mathbb{R}^3$ ) and the trunk angular acceleration ( $\dot{\omega}_b^d \in \mathbb{R}^3$ ) using a PD control law written in the operational space, i.e. a *virtual model* of the form:

$$\begin{aligned} \ddot{\mathbf{x}}_{com}^d &= \mathbf{P}_x(\mathbf{x}_{com}^d - \mathbf{x}_{com}) + \mathbf{D}_x(\dot{\mathbf{x}}_{com}^d - \dot{\mathbf{x}}_{com}), \\ \dot{\omega}_b^d &= \mathbf{P}_\theta e(\mathbf{R}_b^d \mathbf{R}_b^\top) + \mathbf{D}_\theta(\omega_b^d - \omega_b), \end{aligned} \quad (16)$$

where  $\mathbf{x}_{com}^d \in \mathbb{R}^3$  is the desired CoM position, and  $\mathbf{R}_b, \mathbf{R}_b^d \in \mathbb{R}^{3 \times 3}$  are the rotation matrices representing the actual and desired orientation of the trunk respectively,  $e(\cdot) : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$  is a mapping from a rotation matrix to the associated rotation vector,  $\omega_b \in \mathbb{R}^3$  is the angular velocity of the base.

As shown in [21], if the CoM velocity is used as a generalized velocity instead of the base velocity, the robot's dynamic equations get simplified. In this case, we can write

the centroidal robot dynamics as in [17]:

$$m(\ddot{\mathbf{x}}_{\text{com}} + \mathbf{g}) = \sum_{i=1}^c \mathbf{f}_i = \mathbf{F}_{\text{com}}, \quad (17)$$

$$\mathcal{I}_G \dot{\boldsymbol{\omega}}_b + \dot{\mathcal{I}}_G \boldsymbol{\omega}_b = \sum_{i=1}^c (\mathbf{p}_{\text{com},i} \times \mathbf{f}_i) = \boldsymbol{\Gamma}, \quad (18)$$

where  $\mathcal{I}_G$  is the instantaneous centroidal composite rigid body inertia that represents the aggregate rigid body inertia of the entire robot computed at its CoM,  $\mathbf{p}_{\text{com},i} \in \mathbb{R}^3$  is a vector going from the CoM to the position of the  $i^{\text{th}}$  foot defined in an inertial world frame,  $c$  is the number of contact points and  $\mathbf{f}_1, \dots, \mathbf{f}_c \in \mathbb{R}^3$  are the Ground Reaction Forces (GRFs). Since our platform has nearly point-like feet, we assume that it cannot generate moments at the contacts, but only pure forces. Furthermore, we neglect the term  $\dot{\mathcal{I}}_G \boldsymbol{\omega}_b$  since we can assume the legs to be massless<sup>4</sup>.

Then, the desired wrench  $\mathbf{W}^d = [\mathbf{F}_{\text{com}}^d{}^\top, \boldsymbol{\Gamma}^d{}^\top]^\top$  can be computed from the desired CoM linear and trunk angular accelerations and by rewriting (18) in matrix form, we can then map  $\mathbf{W}^d$  into GRFs:

$$\underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & \cdots & \mathbf{I}_{3 \times 3} \\ [\mathbf{p}_{\text{com},1} \times] & \cdots & [\mathbf{p}_{\text{com},c} \times] \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_c \end{bmatrix}}_{\mathbf{f}} = \underbrace{\begin{bmatrix} m(\ddot{\mathbf{x}}_{\text{com}}^d + \mathbf{g}) \\ \mathcal{I}_G \dot{\boldsymbol{\omega}}_b^d \end{bmatrix}}_{\mathbf{b}}. \quad (19)$$

The redundancy in the mapping yields 6 equations with up to 12 unknowns as we can have 4 feet on the ground. Hence, we can form a quadratic optimization problem aiming to satisfy additional optimality criteria, such as ensuring that the GRFs lie inside the friction cones and fulfilling the unilaterality of the GRFs [20]. We approximate the friction cones with square pyramids to express them as linear inequality constraints:

$$\begin{aligned} \mathbf{f}^d &= \underset{\mathbf{f} \in \mathbb{R}^3}{\text{argmin}} (\mathbf{A}\mathbf{f} - \mathbf{b})^\top (\mathbf{A}\mathbf{f} - \mathbf{b}) + \mathbf{f}^\top \mathbf{W}\mathbf{f} \\ \text{s. t. } \quad \underline{\mathbf{d}} &< \mathbf{C}\mathbf{f} < \bar{\mathbf{d}}, \end{aligned} \quad (20)$$

where  $\mathbf{f}^\top \mathbf{W}\mathbf{f}$  is a regularization term to keep the solution bounded. We solve the optimization in real-time with an off-the-shelf Quadratic Programming (QP) solver.

In a second step we map the optimal solution  $\mathbf{f}^d$  into desired joint torques  $\boldsymbol{\tau}^d \in \mathbb{R}^n$  (where  $n$  is the number of joints) considering the gravitational/Coriolis contribution  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ :

$$\boldsymbol{\tau}_{\text{ff}} = \mathbf{h} - \mathbf{S}\mathbf{J}_c^\top(\mathbf{f}^d), \quad (21)$$

where  $\mathbf{J}_c \in \mathbb{R}^{k \times n+6}$  is the stacked Jacobian of the contact points ( $k = 3$  is the number of kinematically constrained DoFs) and  $\mathbf{S} = [\mathbf{0}_{n \times 6} \quad \mathbf{I}_{n \times n}]$  is a matrix that selects the actuated degrees of freedom.

Finally, the trunk controller torques  $\boldsymbol{\tau}_{\text{ff}}$  are summed with the joint PD torques to form the desired torque command that is sent to the low-level joint-torque controllers.

## VI. EXPERIMENTAL RESULTS

To evaluate our approach, we first validate the trunk attitude modulation (pitch and roll) for dynamic walking on flat terrain. Subsequently, we quantify the capabilities of our framework through a set of different terrain conditions: crossing a gap and a set of sparse stepping stones. For that, we plan and execute dynamic walking behaviors which enable the robot to adapt to different terrain conditions given the high-level user commands (desired trunk velocity: step duration and length, and travel direction). All the experiments are conducted with HyQ [5], a 85 kg hydraulically actuated quadruped robot. The HyQ robot is fully-torque controlled and equipped with precision joint encoders, a depth camera (Asus Xtion) and an Inertial Measurement Unit (MicroStrain). HyQ roughly has the dimensions of a goat, i.e. 1.0 m  $\times$  0.5 m  $\times$  0.98 m (length  $\times$  width  $\times$  height). The leg

<sup>4</sup>In the HyQ robot, the leg masses represent 8% of the robot weight.

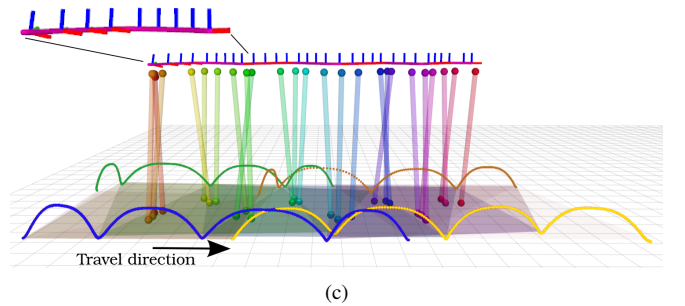
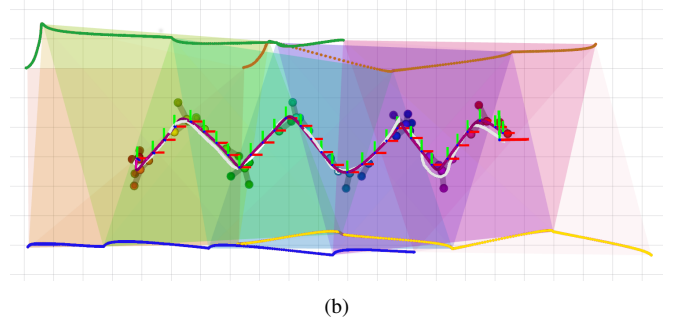
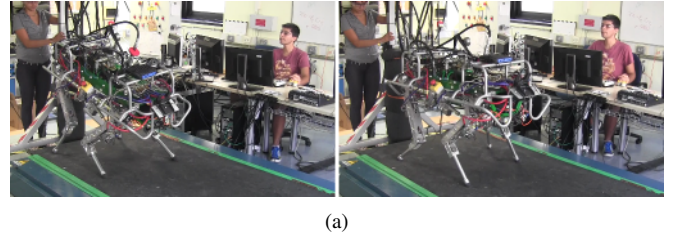


Fig. 6. (a) Dynamic attitude modulation. The initial trunk attitude (left image) is 0.17 and 0.22 radians in roll and pitch, respectively. (b) Body tracking when walking and dynamically modulating the trunk attitude. The planned CoM (magenta) and the executed trajectory (white) are shown together with the sequence of support polygons, CoP and CoM positions. Note that each phase is identified with a specific color. (c) A lateral view of the same motion shows the attitude correction (sequence of frames), and the cart-table displacement. Note that we use the RGB color convention for drawing the different frames. In (b)-(c) the brown, yellow, green and blue trajectories represent the LF, RF, LH and RH foot trajectories, respectively.

length ranges from 0.339-0.789 m and the front/hind hip-to-hip distance is 0.75 m.

### A. Dynamic attitude modulation

First, we showcase the automatic adjustment of the trunk attitude, during a dynamic walk, as illustrated in Fig. 6(a). To evaluate the attitude modulation feature, we plan a *fast*<sup>5</sup> dynamic walk with an average body velocity of 0.18 m/s. We do not use the costmap for generating the corresponding footholds, thus the resulting locations come from the dynamics of walking itself, while maximizing the stability of the gait. We define a stability margin of  $r = 0.1$  m for all our optimizations which is a good trade-off between modeling error and allowed trunk attitude adjustment in HyQ. The maximum allowed angular acceleration is computed using the trunk inertia matrix of HyQ, which results in  $0.11 \text{ rad/s}^2$ . Note that the trunk attitude planner uses the maximum allowed angular velocity as explained in Section III-A.2.

The resulting behavior shows HyQ successfully walking while changing its trunk roll and pitch angles. Note that the trunk attitude planner adjusts the roll and pitch angles given the estimated support region at each phase. Fig. 6(b) shows the tracking performance for initial trunk attitude of 0.17 and 0.22 rad in roll and pitch, respectively. In addition, Fig. 6(c) shows the attitude modulation, which is accomplished in the first 6 phases (i.e. one cycle of locomotion).

### B. Locomotion on challenging terrain

We tested our approach on various challenging terrains: gap and stepping stones with different terrain heights. For all these scenarios, we computed the costmap using the *standard deviation of the height values*, which is estimated through a regression in a  $4 \text{ cm} \times 4 \text{ cm}$  window around the cell of interest. Our costmap is built using a resolution of  $(4 \text{ cm} \times 4 \text{ cm} \times 2 \text{ cm})$  in  $(x, y, z)$ , respectively. The higher resolution value in  $z$  reduces the difference between the expected time of foot touch-down and the detected one. Reducing the foothold error improves the tracking performance of the controller since the desired base and joint positions and velocities are consistent with each other. We weigh equally and manually the desired user command and terrain costs, with a small weight for the CoM energy cost (around 5%). Both soft-constraints have higher weights, which ensures that their targets are met provided with enough exploration steps to the CMA-ES solver. Note that we do not need to define an initial guess, and moreover this might not even help the search due to changes in the terrain topology. We used the same stability margin and allowed angular acceleration (as in Section VI-A) for the trunk attitude planner, and our horizon is  $N = 1$ , i.e. 1 cycle of locomotion or 4 steps.

Crossing a gap and/or trunk attitude adaptation tends to overextend the legs, since large motions are required (Fig. 7(a)). To avoid kinematic limits, we defined a foot search region that ensures leg kinematic feasibility up to 12 cm of terrain height difference, as is illustrated in

Fig. 7(b). For instance, we generated a trajectory with two stepping stones 6 cm higher than other ones. These terrain irregularities produce a trunk modulation in roll and pitch as can be observed in the second sequence. The execution performance on stepping stones with and without changes in terrain elevation is shown in Fig. 7(c)-(d). Compared with our previous work [6], we increased the walking velocity by approximately 80%, while also modulating the trunk attitude. Furthermore, the foothold error is on average less than 2 cm, which increases the success rate of the stepping stones trials to 90%; an increment of 30% when compared with our previous work [6]. Despite these improvements, the stochastic-based optimization tends to increase the computation time due to the non-convex nature of the problem. For our gap and stepping stones experiments, it takes around 10 min to compute the optimal trajectory. For the full sequence of the experiments, please see the [accompanying video](#)<sup>6</sup>.

Our trajectory optimization framework uses as input the high-level desired trunk velocities. We believe that this improves the operability of the system in real-environment applications. Moreover, optimizing a sequence of control parameters allows us to integrate reactive behaviors, which are important for increasing the robustness of the locomotion.

## VII. CONCLUSION

In this paper, we presented a trajectory optimization approach for locomotion on rough terrain that directly uses terrain information. The approach delivers an optimal CoM motion and corresponding optimal foothold locations. Moreover, the solution takes into consideration the trunk attitude modulation required for dynamic walking. We employ a combination of parametric preview models, stochastic-based exploration and receding horizon planning for successfully crossing over various challenging terrains.

We demonstrated how the combination of an impedance controller—which prevents friction cone violations—alongside a trunk controller can compliantly, yet accurately, track the desired whole-body motion. Real world experimental trials on the HyQ robot crossing over challenging terrain demonstrated the capabilities of our framework. Compared with our previous results [6][12], we improved the locomotion without any loss of performance. HyQ walked faster while making trunk attitude adjustments. Moreover, the accuracy of execution was improved, as the error between desired and achieved footholds was reduced from 8 cm to approximately 2 cm. This increased the success rate in the stepping stones by around 30%.

We showed that linear displacements of the CoP in every phase produce similar results to [6]. This assumption allows us to describe a movement as a sequence of parameters. Our experimental results suggest that combining CoM trajectories and foothold selection produces better solutions in terms of avoiding joint limits (both in position and torques). In fact, the foothold locations help to minimize the CoM energy, thus it reduces the applied joint torques.

<sup>5</sup>*fast* for common walking gait velocities of the HyQ robot.

<sup>6</sup><https://youtu.be/79bb2KTULrw>

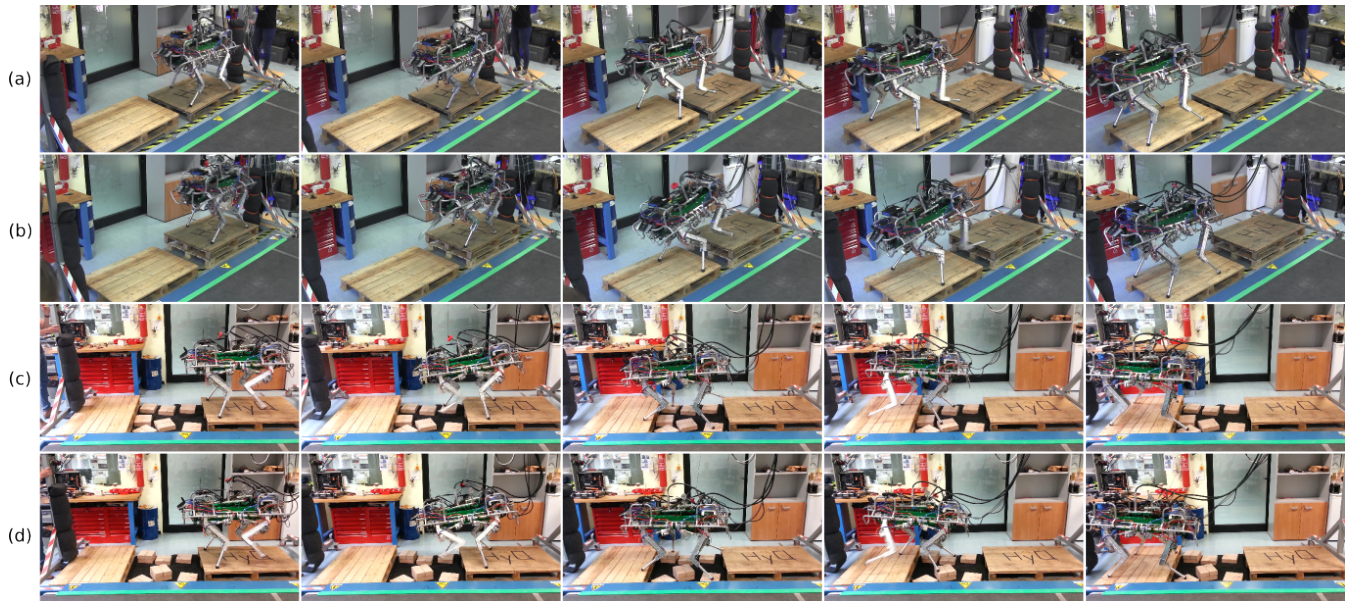


Fig. 7. Snapshots of experimental trials used to evaluate the performance of our trajectory optimization framework. (a) Crossing a gap of 25 cm while climbing up 6 cm. (b) Crossing a gap of 25 cm while climbing down 12 cm. (c) Crossing a set of 7 stepping stones. (d) Crossing a sparse set of stepping stones while dealing with different stone elevations (6 cm).

Future work includes integrating reactive behaviors, such as haptic triggering of stance and step reflex. Our aim is to increase the robustness of the system, for coping with errors in the terrain perception and state estimation. Another extension can be the automatic gait discovery, i.e. transitions from walking to trotting, and vice-versa, while crossing a rough terrain. Finally, working towards online planning is a crucial feature for real applications.

#### REFERENCES

- [1] I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, 2012.
- [2] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body Motion Planning with Simple Dynamics and Full Kinematics,” in *IEEE/RAS International Conference on Humanoid Robots*, 2014.
- [3] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, “Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds,” *ArXiv preprint arXiv:1607.04537*, 2016.
- [4] I. Mordatch, M. de Lasa, and A. Hertzmann, “Robust physics-based locomotion using low-dimensional planning,” *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1, 2010.
- [5] 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, pp. 831–849, 2011.
- [6] A. 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [7] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, D. G. Caldwell, and C. Semini, “Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [8] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Learning, planning, and control for quadruped locomotion over challenging terrain,” *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 2, pp. 236–258, 2010.
- [9] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, “A control architecture for quadruped locomotion over rough terrain,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 811–818.
- [10] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, “Optimization and learning for rough terrain legged locomotion,” *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 2, pp. 175–191, 2011.
- [11] J. R. Reubla, P. D. Neuhaus, B. V. Bonnländer, M. J. Johnson, and J. E. Pratt, “A controller for the littledog quadruped walking on rough terrain,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1467–1473.
- [12] C. Mastalli, A. Winkler, I. Havoutis, D. G. Caldwell, and C. Semini, “On-line and On-board Planning and Perception for Quadrupedal Locomotion,” in *IEEE International Conference on Technologies for Practical Robot Applications (TEPRA)*, 2015.
- [13] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research (IJRR)*, 2013.
- [14] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, “Hierarchical Planning of Dynamic Movements without Scheduled Contact Sequences,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 1620–1626.
- [16] R. Full and D. Koditschek, “Templates and anchors: neuromechanical hypotheses of legged locomotion on land,” *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.
- [17] D. E. Orin, A. Goswami, and S. H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, vol. 35, pp. 161–176, 2013.
- [18] M. B. Popovic, A. Goswami, and H. Herr, “Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications,” *The International Journal of Robotics Research (IJRR)*, vol. 24, pp. 1013–1032, 2005.
- [19] N. Hansen, “CMA-ES: A Function Value Free Second Order Optimization Method,” in *PGMO COPI 2014*, Paris, France, 2014.
- [20] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, “High-slope terrain locomotion for torque-controlled quadruped robots,” *Autonomous Robots*, pp. 1–14, 2017.
- [21] C. Ott, M. A. Roa, and G. Hirzinger, “Posture and balance control for biped robots based on contact force optimization,” in *IEEE/RAS International Conference on Humanoid Robots*, 2011, pp. 26–33.