Feasibility-Based Convexification of MPC Footstep Constraints for Humanoid Gait Generation

Andrew S. Habib, *Graduate Student Member, IEEE*, Nicola Scianca, Leonardo Lanari, *Member, IEEE*, and Giuseppe Oriolo, *Fellow, IEEE*

***Abstract*—Model Predictive Control is an effective tool for robust humanoid gait generation in complex 3D environments. This challenging operating condition typically results in non-convex footstep constraints which, if directly included in the optimization problem, would introduce significant computation overhead. We present a feasibility-driven method to convexify such constraints so as to retain an efficient QP formulation. At each control cycle, the non-convex steppable region is decomposed into convex subregions, one of which is selected to set up the MPC problem, based on a criterion which balances QP feasibility and conformity to the original plan. Dynamic simulations on the HRP-4 humanoid demonstrate robust push recovery with leg crossing and reactive navigation in cluttered 3D environments, including stepping onto obstacles when beneficial. A comparison against a mixed-integer variant of the method confirms that the proposed selection strategy preserves feasibility and responsiveness with a limited computational load.**

***Index Terms*—Body balancing, humanoid and bipedal locomotion, optimization and optimal control.**

I. Introduction

HUMANOID robots are rapidly gaining traction thanks to recent advances in hardware and intelligence. Several labs have showcased impressive agile motions [1], often obtained using imitation learning techniques that replicate human movements collected via motion capture [2]. However, moving in complex and cluttered environments still poses significant challenges which are best addressed using model-based techniques. Among these, Model Predictive Control (MPC) is well suited for locomotion thanks to its explicit handling of constraints [3], [4], that can be used to define steppable regions in the area surrounding the robot. MPC can also serve as a baseline for residual learning techniques, providing inductive bias and improving sample efficiency.

Established approaches for footstep placement use search algorithms for scanning a tree of candidate sequences. These methods, which can either be deterministic [5], [6], [7] or randomized [8], are not aware of the robot dynamics and therefore cannot be used for on-line replanning.

Received 3 April 2026; accepted 24 July 2026. Date of publication 12 August 2026; date of current version 28 August 2026. This article was recommended for publication by Associate Editor G. Xin and Editor L. Pallottino upon evaluation of the reviewers' comments. *(Corresponding author: Giuseppe Oriolo.)*

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, 00185 Roma, Italy (e-mail: habib@diag.uniroma1.it; scianca@diag.uniroma1.it; lanari@diag.uniroma1.it; oriolo@diag.uniroma1.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2026.3723320, provided by the authors.

Digital Object Identifier 10.1109/LRA.2026.3723320

On-line adaptation can be realized by directly including the footsteps as decision variables of the MPC [9]. A key difficulty is that many MPC formulations rely on convex constraints (or behave much better when constraints are convex), while in practice the steppable regions are typically non-convex: an obstacle can require placing footsteps around it, multi-level terrain can create disconnected patches at different heights, and the humanoid kinematics themselves result in non-convex regions, if leg crossing is allowed.

A popular strategy is to decompose the environment into convex regions and use Mixed-Integer Quadratic Programming (MI-QP) to select the best one. This technique was used for footstep planning [10], [11], but also directly inside the MPC [12], [13], [14]. However, MI-QP adds significant computational overhead. A convex decomposition is also used in [15], which relies on closed-form trajectory computation rather than on-line optimization.

It is well-known that MPC constraints can lead to infeasibility when no solutions are available, which is often considered a failure mode. In this work, we adopt a different perspective: when feasibility conditions can be characterized, infeasibility becomes a meaningful signal that the robot can proactively exploit to guide the convexification of footstep constraints. MPC feasibility was already leveraged in [16] to handle non-convex steppable regions on flat ground in two hand-crafted scenarios, one involving replanning in reaction to a push, and the other focusing on obstacle avoidance.

In this letter, we propose an MPC framework that allows the robot to walk through a complex 3D environment constituted by an arbitrary number of horizontal patches at different elevations. As an input, our method only requires a basic footstep plan, which may even ignore completely the geometry of the environment. At each iteration, the steppable region is first determined based on the local map and then convexified via (1) a decomposition into convex steppable subregions and (2) the selection of one subregion based on feasibility-aware criteria. The key idea is in fact to exploit an explicit characterization of MPC feasibility to drive the convexification process itself, thereby transforming feasibility from a passive property of the optimization problem into an active tool for deciding where the robot should step. No explicit labeling of the obstacles is needed: regions where the robot cannot step will automatically be treated as obstacles, whereas an object with sufficient surface area will generate a steppable subregion. In the end, the MPC is formulated as a simple and efficient QP, with the region selection process adding negligible overhead. Dynamic simulations on the HRP-4 humanoid validate our approach, showing successful motion generation in real time.

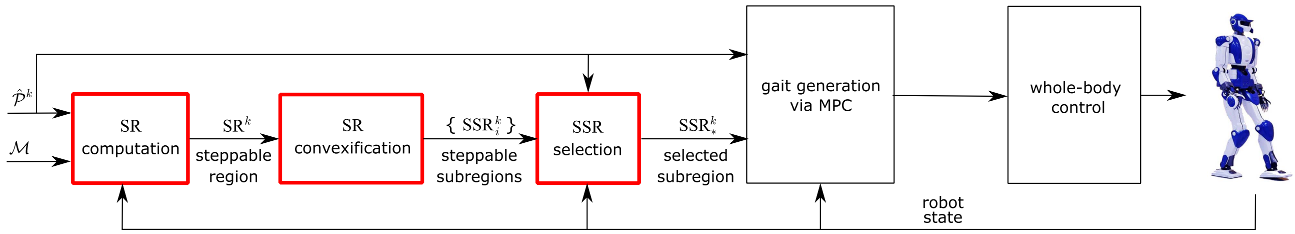

Fig. 1. Workflow of an iteration of the proposed method at each sampling time t_k.

The letter is organized as follows. Section II describes the problem and gives an overview of our approach. In Section III, we discuss in detail the computation and convexification of the steppable region. Section IV describes a possible algorithm for gait generation and shows how to characterize its feasibility for our purposes. Dynamic simulations are finally presented in Section V.

II. Problem and Approach

We consider the following problem. A humanoid robot must move in a *world of stairs*, i.e., an environment consisting of flat horizontal patches at different heights; depending on their size and height, some of these patches represent steps or steppable surfaces, while others constitute obstacles. The robot is given a candidate footstep plan, which may ignore the characteristics of the environment (for example, all footsteps may be at ground zero). The objective is to generate a gait which follows the given plan as closely as possible, while adapting them to comply with the actual geometry of the environment and to guarantee a stable behavior, even in the presence of perturbations on the robot.

The environment is described as an *elevation map* $\mathcal{M}$, i.e., a function mapping each point of the horizontal plane to the corresponding elevation of the terrain [17]. Consistently with the world-of-stairs hypothesis, $\mathcal{M}$ is composed by horizontal patches, with elevation changes occurring at the *edges*. The elevation map may be completely known in advance, or it can be built incrementally from perceptual information as the robot moves; in fact, only a local map of the area where the robot is about to step is required.

The proposed real-time method hinges upon an iteration to be repeated at each sampling time t_k, whose workflow is shown in Fig. 1. The inputs at t_k are $\mathcal{M}$ and the upcoming part $\hat{\mathcal{P}}^k = \left\{\hat{\boldsymbol{f}}^{j+1}, \hat{\boldsymbol{f}}^{j+2}, \dots\right\}$ of the candidate footstep plan, with each footstep $\boldsymbol{f}$ described by the 3D position (x_f, y_f, z_f) of the foot center. Note the following:

- $\boldsymbol{f}^j$ denotes the current location of the support foot.
- The length of $\hat{\mathcal{P}}^k$ (i.e., the number of footsteps in it) depends on the horizon used for gait generation.
- All footsteps have the same orientation (this does not necessarily mean that the robot is walking straight). See the conclusions for the case of arbitrary orientations.

The first step is to compute the intersection between the *kinematically admissible region* (KAR) associated to the current support foot (i.e., the 3D subset of points where the foot center can be placed, irrespective of the actual geometry of the environment) and the elevation map $\mathcal{M}$, see Fig. 2. The result is the *steppable region* SR^k at t_k, a collection of planar patches where the next footstep can be placed.

The patches composing SR^k are in general non-convex, as explained in detail in the next section. The next step consists therefore in convexifying SR^k, i.e., decomposing it in a collection of convex planar patches $\{\mathrm{SSR}_i^k\}$, called *steppable subregions*. The algorithm then selects one of these, denoted by SSR_*^k, for placing the next footstep. SSR_*^k is chosen based on a combination of criteria, two of which are related to the feasibility of the resulting MPC optimization problem at t^k.

Next in the workflow is the generation of a stable gait over a given control horizon, with the first footstep on SSR_*^k and successive footsteps which reproduce the rest of $\hat{\mathcal{P}}^k$ as closely as possible. For this step, we can adopt any MPC-based method, provided that we are able to characterize its feasibility quantitatively, so as to use it for selecting SSR_*^k.

The generated CoM trajectory, together with other tasks such as the swinging foot trajectory, are finally sent as reference to the whole-body controller, which computes the robot commands (either at the position or at the torque level, depending on the robot) needed for accurate tracking.

The entire procedure is repeated at every t_k in a receding-horizon fashion. Hence, neither SSR_*^k nor the resulting footstep sequence is irrevocably committed to, as both are continuously re-evaluated based on robot state and environment information. Therefore, our method should be interpreted as a receding-horizon approximation of the corresponding MI formulation rather than as a greedy selection strategy.

III. Choosing Where to Step

As discussed in the previous section, at each sampling time t_k the current KAR is intersected with the elevation map $\mathcal{M}$ to obtain the steppable region SR^k. This is then convexified, i.e., decomposed in a collection of convex steppable subregions $\{\mathrm{SSR}_1^k, \mathrm{SSR}_2^k, \dots\}$, from which the optimal SSR_*^k is selected. Below, we discuss each of these steps in detail.

A. Computation of the Steppable Region SR^k

Given the robot, one can compute off-line all the possible placements of one robot foot (swinging) when the other (support) is in contact with the ground; this is a purely kinematic computation which does not consider the specific morphology of the ground. Clearly, some of the placements must be discarded because they bring the swing leg in collision with the support leg. The resulting KAR will therefore be a non-convex subset (see Fig. 2, left).

Based on the pose of the support foot at the current time, the KAR can be placed in the environment and intersected with the elevation map $\mathcal{M}$ (in practice, its locally relevant portion), in which each horizontal patch has been first eroded by the footprint. The intersection will retain a subset of the eroded

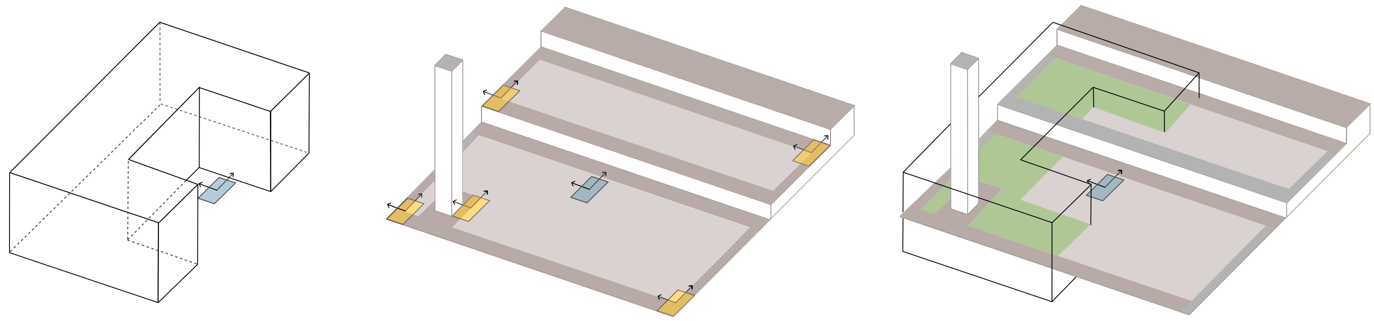

Fig. 2. Left: The KAR for the current placement of the support foot (a polyhedral KAR is shown here, but our method works unchanged with non-polyhedral KARs). Center: The relevant part of the elevation map $\mathcal{M}$ (gray) for the current placement of the support foot (blue), and its planar patches after footprint erosion (light gray). Right: The steppable region SR^k (green) resulting from the intersection between the KAR and the eroded map.

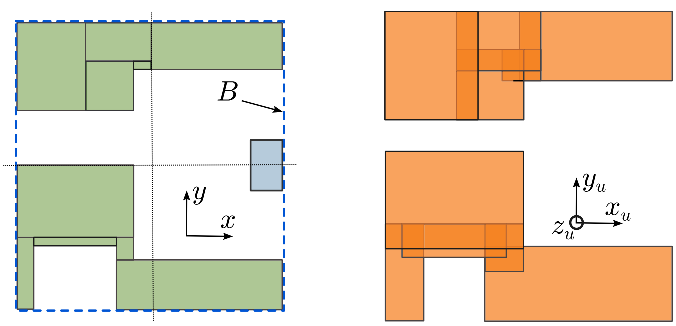

Fig. 3. Left: The convex decomposition of the steppable region SR^k of Fig. 2 in steppable subregions (green). Right: Top view of the feasibility regions (orange) associated to each SSR_i^k; note the partial overlap.

patches of $\mathcal{M}$, clipping them at the lateral boundary of the KAR; moreover, it will contain holes in correspondence to patches that are located above or below the KAR. The result is the steppable region SR^k (see Fig. 2, right): this is a collection of planar patches where the swinging foot center can be placed with the whole foot lying on the non-eroded patch.

Fig. 2, right, illustrates the geometry of the steppable region. SR^k is disconnected at the edges of $\mathcal{M}$ (elevation changes) due to the erosion, which leaves a gap around each edge. Moreover, each patch may be non-convex, because *(i)* the KAR itself is non-convex due to the support leg, *(ii)* the horizontal patches of $\mathcal{M}$ follow the environment geometry.

B. Convexification of the Steppable Region SR^k

The second stage in the workflow of Fig. 1 is the decomposition of the steppable region in convex subregions.

Define a local Cartesian frame whose x axis is aligned with the support foot, and denote by B the axis-aligned bounding box of the projection of SR^k on the ground, locally identified with the plane of the support foot (Fig. 3, left). The algorithm performs a recursive bisection of B: at the first iteration, it bisects B along its longest side, and checks each of the two resulting 'cells' to verify whether all its points belonging to the projection of SR^k have the same elevation. If this is true, the cell does not need any further subdivision; otherwise, the cell is bisected, and the procedure continues. Once the maximum resolution (level of bisection) is reached, the algorithm has produced a collection of rectangular cells SSR_i^k, $i = 1, \ldots, n$, that are aligned with the local frame and serve as convex building blocks for 'covering' SR^k; each cell is convex and consists of points having the same elevation (Fig. 3, left). Note that the geometry of the cells does not impose any restriction on the geometry of SR^k, which may remain arbitrarily non-convex and disconnected.

In the final step, adjacent cells with the same elevation and a (full) boundary segment in common are merged together, in order to reduce the final number of steppable subregions.

Note that in Fig. 2 the edges of $\mathcal{M}$ are assumed to be aligned with the local frame for the sake of clarity. However, the above procedure for computing and decomposing SR^k is completely general and works for arbitrary orientations of the edges of $\mathcal{M}$. In particular, the steppable subregions will always be rectangles aligned with the local frame, because the bisection is performed in accordance with it.

C. Selection of the Steppable Subregion SSR_*^k

The third stage in the workflow of Fig. 1 is the selection of the 'best' steppable subregion SSR_*^k among those available.

In principle, any SSR_i^k can be chosen as the planar surface where the next footstep must be placed, and therefore used to set up the footstep constraint for MPC-based gait generation, provided that the current state of the robot belongs to the corresponding feasibility region; if this is not the case, that steppable subregion must be discarded because the corresponding MPC would not be solvable.

However, as will be shown later (Section IV-B), MPC feasibility regions associated to different SSR_i^k may overlap, and therefore *multiple steppable subregions are in general admissible* for any robot state. To identify the most convenient choice, each candidate SSR_i^k is assigned a score based on the combination of three criteria.

1) Feasibility Margin: The first criterion for evaluating a steppable subregion looks at the feasibility of the associated MPC for gait generation. To this end, denote by FR_i^k the *feasibility region* of the MPC at t_k that uses SSR_i^k for the footstep constraint (i.e, the set of robot states for which a solution exists). We define the *feasibility margin* of SSR_i^k as

$$\gamma_1\left(\mathrm{SSR}_i^k\right) = \mu\left(\boldsymbol{x}_k, \mathrm{FR}_i^k\right), \tag{1}$$

where $\boldsymbol{x}_k$ is a generic variable indicating the robot state at the current time instant t_k and μ denotes the distance of $\boldsymbol{x}_k$ from the boundary of FR_i^k. The nature of $\boldsymbol{x}$ (and correspondingly, the space over which the feasibility regions are defined) will depend on the prediction model employed by the adopted MPC algorithm; this will also identify the proper distance μ to be used in (1). In Section IV-B, we shall instantiate these choices for a particular MPC-based gait generation. ■

2) Feasibility Volume: The second criterion for evaluating SSR_i^k is also related to the feasibility of the MPC problem. In particular, we define the *feasibility volume* of SSR_i^k as

$$\gamma_2\left(\mathrm{SSR}_i^k\right) = v\left(\mathrm{FR}_i^k\right), \tag{2}$$

where $v(\cdot)$ is a volume measure in the space of feasibility regions. Differently from γ_1, γ_2 does not depend on the current robot state; the idea is to privilege 'larger' feasibility regions, which provide more space for motion adaptation and are thus expected to result in increased robustness. ■

3) Plan Conformity: The last criterion quantifies how close the steppable subregion SSR_i^k is to the candidate footstep plan $\hat{\mathcal{P}}^k$. In particular, we extract from $\hat{\mathcal{P}}^k$ the subsequence $\left\{\hat{\boldsymbol{f}}^{j+1}, \hat{\boldsymbol{f}}^{j+3}, \ldots, \hat{\boldsymbol{f}}^{j+1+2m}\right\}$, where $\hat{\boldsymbol{f}}^{j+1}$ is the candidate position of the *next* footstep and m determines the length of the subsequence.[1] The *plan conformity* of SSR_i is computed as[2]

$$\gamma_3\left(\mathrm{SSR}_i^k\right) = -\sum_{l=0}^{m} d\left(\hat{\boldsymbol{f}}^{j+1+2l}, \mathrm{SSR}_i^k\right),$$

where $d\left(\cdot, \mathrm{SSR}_i^k\right)$ denotes the distance between a footstep position and the steppable region SSR_i^k, conventionally set to zero if the footstep belongs to SSR_i^k. ■

Each steppable subregion SSR_i^k such that $\boldsymbol{x}_k \in \mathrm{FR}_i^k$ is assigned a score given by the weighted sum of the criteria:

$$\gamma\left(\mathrm{SSR}_i^k\right) = \sum_{l=1}^{3} \alpha_l\, \gamma_l\left(\mathrm{SSR}_i^k\right), \tag{3}$$

whereas steppable subregions whose feasibility regions do not contain $\boldsymbol{x}_k$ are given a $-\infty$ score. Finally, we let

$$\mathrm{SSR}_*^k = \arg\max_{i=1,\ldots,n} \gamma\left(\mathrm{SSR}_i^k\right). \tag{4}$$

The accompanying video (https://youtu.be/ZmPM9ZPVofQ) shows the complete pipeline, from the computation of the steppable region to the selection of the steppable subregion.

We will express SSR_*^k in the support foot frame as follows:

$$\mathrm{SSR}_*^k = \left\{(x, y, z) \in I\!R^3 : \underline{b}_x^k \le x \le \bar{b}_x^k, \underline{b}_y^k \le y \le \bar{b}_y^k, z = b_z^k\right\},$$

where $\underline{b}_x^k, \bar{b}_x^k$ ($\underline{b}_y^k$ and $\bar{b}_y^k$) denote the lower, upper bound of SSR_*^k for x (for y), and b_z^k is the elevation of SSR_*^k.

At this point, SSR_*^k is sent to the gait generation stage to be used in setting up the footstep constraint.

IV. Gait Generation

Once the steppable subregion SSR_*^k has been chosen, the proposed method proceeds to set up an optimization problem for gait generation. Its solution should provide the trajectory of the CoM and adapted footstep positions over the control horizon, subject to the relevant constraints.

In principle, any MPC-based technique can be used, as long as we can characterize its feasibility region and use it to compute the scores of candidate steppable subregions. One such technique is IS-MPC [8], [18], whose formulation is summarized below before analyzing its feasibility.

A. MPC Formulation

This section presents the main MPC components, namely, the prediction model, the constraints and the quadratic programming formulation.

The scheme runs in discrete time, with the current time denoted by $t_k = k\delta$, where δ is the sampling interval. At each instant, an optimization problem is set up over a control horizon $[t_k, t_{k+C})$, whose solution provides the adapted positions $\left\{\boldsymbol{f}^{j+1}, \ldots, \boldsymbol{f}^{j+N}\right\}$ of all footsteps relevant for the control horizon, as well as the trajectories of the CoM and of the Zero Moment Point (ZMP) in the same interval.

1) Prediction Model: The prediction model is the 3D Linear Inverted Pendulum (LIP):

$$\ddot{\boldsymbol{p}}_c = \eta^2(\boldsymbol{p}_c - \boldsymbol{p}_z) + \boldsymbol{g}, \tag{5}$$

where $\boldsymbol{p}_c = (x_c, y_c, z_c)$ and $\boldsymbol{p}_z = (x_z, y_z, z_z)$ are respectively the position of the CoM and of the ZMP, $\boldsymbol{g} = (0, 0, -g)$, and $\eta^2 = g/h_c$, with h_c a design parameter corresponding to the vertical CoM-ZMP displacement at the equilibrium and $g = 9.81\,\mathrm{m/s}^2$. To generate smoother trajectories, the ZMP velocity $\dot{\boldsymbol{p}}_z$ is used as input to model (5).

2) Footstep Constraints: The first footstep $\boldsymbol{f}^{j+1}$ is constrained to be inside the selected steppable subregion SSR_*^k. The corresponding footstep constraint is

$$\underline{b}_x^k \le x_f^{j+1} \le \bar{b}_x^k \qquad \underline{b}_y^k \le y_f^{j+1} \le \bar{b}_y^k, \tag{6}$$

while z_f^{j+1} is directly set to the elevation b_z^k of SSR_*^k.

To keep the problem tractable, a simpler strategy is used for the placement of footsteps farther away in the horizon. In particular, the x, y position of the generic successive footstep must belong to a predefined rectangular subset of the (appropriately placed) KAR having dimensions $2d_x, 2d_y$:

$$\left|x_f^l - x_f^{l-1}\right| \le d_x \qquad \left|y_f^l - y_f^{l-1} \pm \ell\right| \le d_y, \tag{7}$$

for $l = j+2, \ldots, j+N$; the term $\pm\ell$ introduces an alternating displacement between left and right footsteps. As for the height z_f^l of $\boldsymbol{f}^l$, it is set to the elevation associated in $\mathcal{M}$ to its planned position $(\hat{x}_f^l, \hat{y}_f^l)$.

3) Balance Constraints: In 3D walking, one way to express the balance condition for a humanoid [19] is to require that the ZMP is inside a polyhedral cone having its apex at the CoM and edges going through (some of) the vertexes of the convex hull of the contact surfaces. Unlike the equivalent for flat ground (ZMP inside the support polygon), this condition would be nonlinear in the decision variables.

To maintain linearity, one can conservatively approximate the polyhedral cone as a moving box whose centroid $\boldsymbol{p}_m = (x_m, y_m, z_m)$ coincides with the position of the support foot during single support, and translates from one foot to the other during double support (no rotation is necessary thanks to the constant orientation assumption).

Denote by T_s the step duration, divided into T_{ds} for double support and T_{ss} for single support, and by t_s^l the time instant

[1] The footstep index is incremented by two so as to extract only the future candidate positions of the foot (left or right) which is currently swinging.

[2] By incorporating a sequence of future footsteps, γ_3 introduces a form of look-ahead in the steppable-subregion selection process, favoring choices that remain compatible with the subsequent evolution of the nominal plan.

at which the foot lands at $\boldsymbol{f}^l$, i.e., the start of the double support phase from $\boldsymbol{f}^{l-1}$ to $\boldsymbol{f}^l$. For $t \in \left[t_s^l, t_s^{l+1}\right)$, the center of the moving box is in $\boldsymbol{p}_m(t) = \boldsymbol{p}_f^{l-1} + (\boldsymbol{p}_f^l - \boldsymbol{p}_f^{l-1})\sigma^l(t)$, where $\sigma^l(t)$ is a ramp going from 0 to 1 in $[t_s^l, t_s^l + T_{ds})$ and is 1 in $\left[t_s^l + T_{ds}, t_s^{l+1}\right)$. Letting (w_x, w_y, w_z) be the half-dimensions of the moving box and denoting with v^h the value of the generic variable v at the sampling time t_h, the balance constraints for $h = k+1, \ldots, k+C$ are

$$\left|x_z^h - x_m^h\right| \leq w_x \quad \left|y_z^h - y_m^h\right| \leq w_y \quad \left|z_z^h - z_m^h\right| \leq w_z. \quad (8)$$

4) Stability Constraint: Model (5) is unstable, due to the presence of a positive eigenvalue. Hence, balance satisfaction alone is insufficient, as the CoM may diverge from the ZMP. IS-MPC avoids this by imposing the following *stability constraint* involving the unstable component $\boldsymbol{p}_u = \boldsymbol{p}_c + \dot{\boldsymbol{p}}_c/\eta$

$$\eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} \boldsymbol{p}_z(\tau) d\tau = \boldsymbol{p}_u^k - \tilde{\boldsymbol{c}}^k + \frac{\boldsymbol{g}}{\eta^2}. \quad (9)$$

In $\tilde{\boldsymbol{c}}^k = \eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_k)} \tilde{\boldsymbol{p}}_z(\tau) d\tau$, the unknown future $\boldsymbol{p}_z(\tau)$ is replaced with the *anticipative tail* $\tilde{\boldsymbol{p}}_z(\tau)$, i.e., a ZMP trajectory conjectured on the basis of the footstep plan.

Constraint (9) requires $\boldsymbol{p}_u^k$ to match a specific value which depends on the evolution of $\boldsymbol{p}_z$ in the control horizon. In the implementation, one uses a version of the constraint where the corresponding control inputs $\dot{\boldsymbol{p}}_z$ appear explicitly.

5) QP Formulation: At each instant t_k, the MPC solves a QP problem whose cost function penalizes over the control horizon the difference between the actual and the planned footsteps, the deviation of the ZMP from the center of the moving box, and the norm of the ZMP velocity. Collecting the decision variables in $\boldsymbol{F}^k = \left(\boldsymbol{f}^{j+1}, \ldots, \boldsymbol{f}^{j+N}\right)$ and $\dot{\boldsymbol{P}}_z^k = \left(\dot{\boldsymbol{p}}_z^k, \ldots, \dot{\boldsymbol{p}}_z^{k+C-1}\right)$, and correspondingly defining $\hat{\boldsymbol{F}}^k = (\hat{\boldsymbol{f}}^{j+1}, \ldots, \hat{\boldsymbol{f}}^{j+N})$, $\boldsymbol{P}_z^{k+1} = \left(\boldsymbol{p}_z^{k+1}, \ldots, \boldsymbol{p}_z^{k+C}\right)$ and $\boldsymbol{P}_m^{k+1} = \left(\boldsymbol{p}_m^{k+1}, \ldots, \boldsymbol{p}_m^{k+C}\right)$, the QP is formulated as follows:

$$\begin{cases} \displaystyle\min_{\boldsymbol{F}^k, \dot{\boldsymbol{P}}_z^k} \beta_1 \left\|\boldsymbol{F}^k - \hat{\boldsymbol{F}}^k\right\|^2 + \beta_2 \left\|\boldsymbol{P}_z^{k+1} - \boldsymbol{P}_m^{k+1}\right\|^2 + \beta_3 \left\|\dot{\boldsymbol{P}}_z^k\right\|^2 \\ \text{subject to:} \\ \bullet \text{ footstep constraints (6) and (7), for } l = j+2, \ldots, j+N; \\ \bullet \text{ balance constraints (8), for } h = k+1, \ldots, k+C; \\ \bullet \text{ stability constraint (9).} \end{cases}$$

In a standard MPC fashion, the LIP model (5) is integrated using the first control sample $\dot{\boldsymbol{p}}_z^k$ to compute the next reference value of the CoM position, while the first footstep $\boldsymbol{f}^{j+1}$ provides the target for generating the next reference position of the swing foot. Both reference values are then sent to the whole-body controller.

B. Feasibility Region

The steppable subregion SSR_*^k to be used in (6) must be identified by the selection procedure of Section III-C. To this end, we build upon previously developed feasibility results for IS-MPC to explicit characterize the feasibility region of our QP problem, in which the bounds of the next footstep constraint (6) are replaced by generic values $\underline{b}_x, \bar{b}_x$ and $\underline{b}_y, \bar{b}_y$ (and b_z^k is replaced by b_z). Denote this new problem by QP°.

Proposition 1: QP° is feasible at t_k if and only if $\boldsymbol{p}_u^k$ belongs to the feasibility region (vector inequalities must be intended component-wise)

$$\text{FR}^k = \left\{\boldsymbol{p}_u \middle| \, \underline{\boldsymbol{p}}_u^k \leq \boldsymbol{p}_u \leq \bar{\boldsymbol{p}}_u^k\right\}, \quad (10)$$

with the components of the lower bound $\underline{\boldsymbol{p}}_u^k = \left(\underline{x}_u^k, \underline{y}_u^k, \underline{z}_u^k\right)$

$$\begin{aligned} \underline{x}_u^k &= \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} \underline{x}_z(\tau) d\tau + \tilde{c}_x^k \\ \underline{y}_u^k &= \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} \underline{y}_z(\tau) d\tau + \tilde{c}_y^k \\ \underline{z}_u^k &= \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} \underline{z}_z(\tau) d\tau + \tilde{c}_z^k + \frac{g}{\eta^2}, \end{aligned} \quad (11)$$

where (omitting time dependency)

$$\begin{aligned} \underline{x}_z &= x_f^{j-1} + \left(x_f^j - x_f^{j-1}\right)\sigma^j + \left(\underline{b}_x - x_f^j\right)\sigma^{j+1} \\ &\quad - \sum_{i=j+2}^{j+N} d_x \sigma^i - w_x \\ \underline{y}_z &= y_f^{j-1} + \left(y_f^j - y_f^{j-1}\right)\sigma^j + \left(\underline{b}_y - y_f^j\right)\sigma^{j+1} \\ &\quad - \sum_{i=j+2}^{j+N} (d_y \mp \ell)\sigma^i - w_y \\ \underline{z}_z &= z_f^{j-1} + \left(z_f^j - z_f^{j-1}\right)\sigma^j + \left(b_z - z_f^j\right)\sigma^{j+1} \\ &\quad + \sum_{i=j+2}^{j+N} \left(z_f^i - z_f^{i-1}\right)\sigma^i - w_z, \end{aligned}$$

and the components of the upper bound $\bar{\boldsymbol{p}}_u^k = \left(\bar{x}_u^k, \bar{y}_u^k, \bar{z}_u^k\right)$ given by analogous expressions.

Proof: Let us consider the x component first. To prove necessity, we must show that all feasible ZMP trajectories satisfy $\underline{x}_u^k \leq x_u^k \leq \bar{x}_u^k$. Any feasible trajectory must comply with (8), i.e., $x_m(t) - w_x \leq x_z(t) \leq x_m(t) + w_x$ (note that we are using the continuous-time version of the original discrete-time constraint). Focusing on the larger-than inequality and writing the expression of $x_m(t)$ leads to

$$x_z(t) \geq x_f^{j-1} + \sum_{i=j}^{j+N} \left(x_f^i - x_f^{i-1}\right)\sigma^i(t) - w_x,$$

from which, using (6) and (7), we obtain $x_z(t) \geq \underline{x}_z(t)$. Multiplying the left- and the right-hand side of this inequality by $\eta e^{-\eta(t - t_k)}$, integrating over the interval $[t_k, t_{k+C}]$ and adding $\tilde{c}_x^k$ to both sides, we obtain $x_u^k \geq \underline{x}_u^k$. A similar reasoning can be used to show $x_u^k \leq \bar{x}_u^k$, thus completing the proof of necessity for the x component.

To prove sufficiency, assume $\boldsymbol{p}_u^k \in \text{FR}^k$, so that x_u^k can be written as a convex combination of the region bounds:

$$x_u^k = a\, \underline{x}_u^k + (1-a)\bar{x}_u^k \quad \text{with } a \in [0, 1].$$

Consider the ZMP x-trajectory $\breve{x}_z(t) = a\, \underline{x}_z(t) + (1-a)\bar{x}_z(t)$ with the footstep x-sequence $\{x_f^{j+1} = a\, \underline{b}_x + (1-a)\bar{b}_x$, $x_f^i = x_f^{i-1} + d_x(1-2a)$ for $i = j+2, \ldots, j+N\}$. It is easy to verify that $\breve{x}_z(t)$ satisfies the balance constraints and the associated x-footsteps satisfy the footstep constraints. The stability constraint (9) is also satisfied, since

$$x_u^k = a\,\underline{x}_u^k + (1-a)\,\bar{x}_u^k = \eta\int_{t_k}^{t_{k+C}} e^{-\eta(\tau-t_k)}\breve{x}_z(\tau)\,d\tau + \tilde{c}_x^k.$$

This completes the proof of sufficiency for the x component.

The proof for y and z follows the same arguments, the only difference being in the footstep sequence used for bounding the ZMP trajectory and proving sufficiency. For y, one incorporates the lateral displacement ℓ in the sequence ($y_f^{j+1} = a\,\underline{b}_y + (1-a)\bar{b}_y$, $y_f^i = y_f^{i-1} \pm \ell + (1-2a)d_y$); for z, the footstep heights are fixed (see end of Section IV-A2). ■

Note the following points.

- While a steppable subregion is a rectangle in x, y, the corresponding feasibility region (10) is a rectangular parallelepiped in a space whose coordinates are x_u, y_u, z_u. This 3-dimensional space is actually a subspace of the 6-dimensional state space of the LIP (5).
- The bounds (11) of the feasibility region are expressed in integral form for compactness, but they can be easily written in closed form. This is important when computing in real time the feasibility margin (1) and the feasibility measure (2) of a steppable subregion. In particular, γ_1 is the shortest Euclidean distance from $\boldsymbol{p}_u^k$ to the faces of FR^k, i.e., the minimum clearance of $\boldsymbol{p}_u^k$ along any dimension; while γ_2 is the volume of FR^k itself, given by the product of its three dimensions.
- Whenever two steppable subregions SSR_A^k and SSR_B^k are adjacent, the corresponding feasibility regions FR_A^k and FR_B^k will overlap. Indeed, when $\underline{b}_{x,A}^k = \bar{b}_{x,B}^k$, the distance between the lower x-bound of FR_A^k and the upper x-bound of FR_B^k is

$$\underline{x}_{u,A}^k - \bar{x}_{u,B}^k = -\left(1-e^{-\eta T_c}\right)w_x - \sum_{i=j+2}^{j+N} 2\,d_x\phi_i,$$

  where $\phi_i = \eta\int_{t_k}^{t_{k+C}} e^{-\eta(\tau-t_k)}\sigma^i(\tau)\,d\tau$. Since this distance is strictly negative, FR_A^k and FR_B^k do overlap along x, and the same is true along y. See Fig. 3, right, for an illustration of the overlap.

The last point is particularly important, as it confirms that, given a robot state, multiple steppable subregions are in general admissible for setting up the MPC problem (see the discussion at the beginning of Section III-C). Ultimately, this provides a theoretical foundation for the approach presented in this letter, i.e., using at each time instant the best possible steppable subregion based on a feasibility analysis.

V. Simulations

We validated the proposed approach via dynamic simulations in DART. The robotic platform is the HRP-4 humanoid by Kawada Robotics. The following parameters were used.

- Robot parameters: $\eta = 3.5\ \mathrm{s}^{-1}$, corresponding to a reference CoM-ZMP displacement of 0.8 m.
- Step parameters: step duration $T_s = 1$ s, single and double support duration $T_{ss} = 0.7$ s and $T_{ds} = 0.3$ s.
- SSR selection parameters: weights in (3) are $\alpha_1 = 0.6$, $\alpha_2 = 0.1$ and $\alpha_3 = 0.3$.
- MPC parameters: sampling time $\delta = 0.01$ s, control horizon $T_c = 1$ s (thus, $N = 2$); half-dimensions of the moving box (6) are $w_x = w_y = w_z = 0.04$ m, half-dimensions of the KAR (7) are $d_x = 0.30$ and $d_y = 0.09$ m, cost function weights are $\beta_1 = 1000$, $\beta_2 = 100$ and $\beta_3 = 1$.

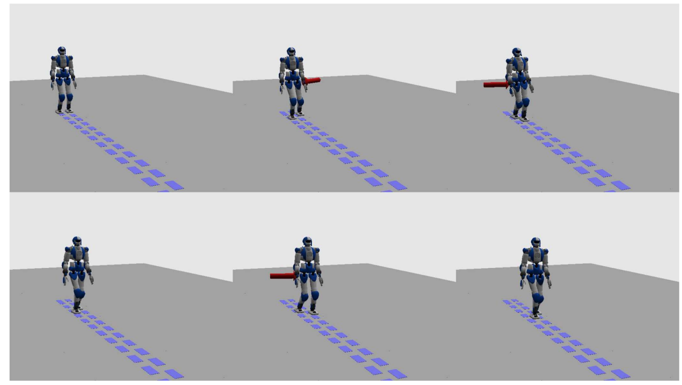

Fig. 4. Push recovery: In the presence of external pushes, the robot adapts footsteps, occasionally crossing its feet to maintain balance (see video).

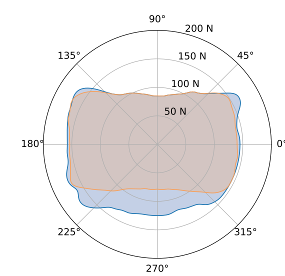

Fig. 5. Push recovery: Graphical representation of the maximum push that the robot can withstand in the various directions: using a non-convex KAR (blue) vs. a rectangular KAR (gray). HRP-4 is walking in the 0° direction.

Our implementation, which uses HPIPM [20] as QP solver, comfortably achieved real-time performance (see Table I).

A. Dynamic Simulations in DART

Three scenarios are considered in the dynamic simulations: push recovery, stepping stones and staircase climbing. See the accompanying video (https://youtu.be/ZmPM9ZPVofQ) for animated clips of the simulations (including two additional scenarios, with the robot walking on a narrow walkway and through an automatic door).

Push Recovery This scenario is aimed at showing how the proposed method is able to withstand external perturbations.

In the simulation of Fig. 4, the robot is subject to five consecutive horizontal pushes along various directions while walking. Each push lasts 0.1 s and is applied at the beginning of the single support phase of its respective footstep. The magnitude of each push varies between 90 and 145 N. While all footsteps are actually adapted in order to optimize the composite cost function of the QP problem, a more drastic adjustment is required to withstand impulsive forces. Since all pushes come from the side of the swinging foot, the capture point $\boldsymbol{p}_u$ gets displaced toward the support foot. To maintain balance, the robot then selects the SSR that allows to displace the next footstep accordingly, leading to the swinging foot (partially) crossing the support foot.

Clearly, keeping balance via foot crossing is made possible by the specific non-convex shape of the KAR. To emphasize this fact, we performed a simulation campaign where the robot

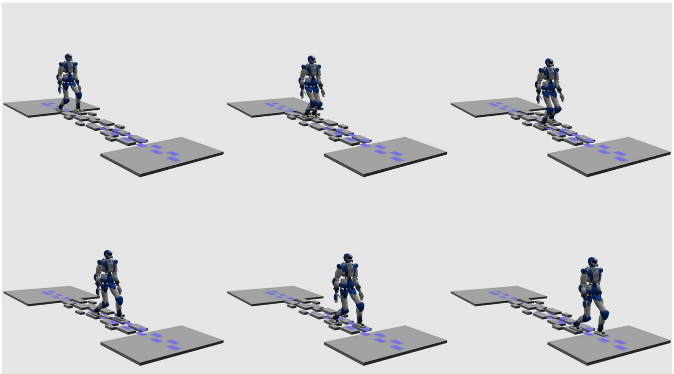

Fig. 6. Stepping stones: The robot walks on properly sized stones and avoids the others (see video).

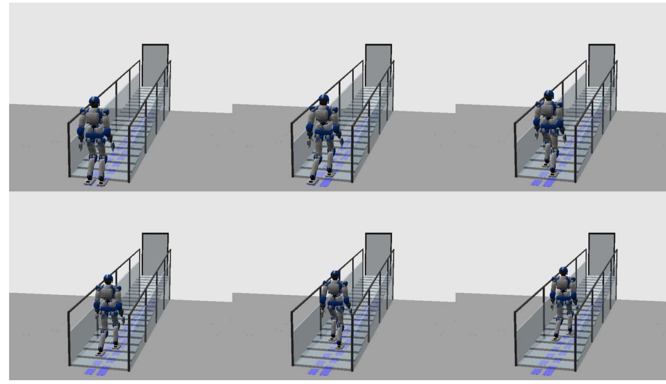

Fig. 7. Staircase climbing: The robot adapts a ground-level footstep plan to ascend an unexpected staircase in 'one foot per step' fashion (see video).

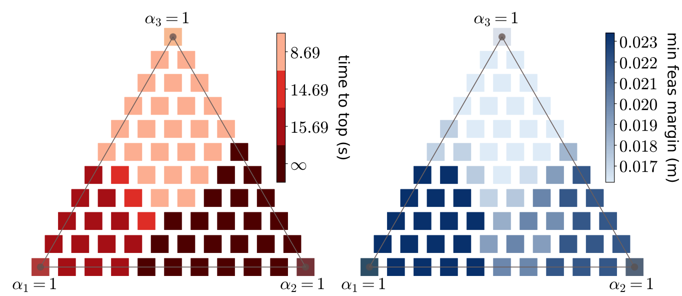

Fig. 8. Staircase climbing: time to the top (left) and minimum overall feasibility margin (right) for different weight combinations such that $\sum_{i=1}^{3} \alpha_i = 1$. An infinite time to top means that the robot maintained balance at all times but was not able to complete the ascent.

is pushed along 120 uniformly distributed radial directions. The results of Fig. 5 show that robustness to pushes directed toward the support foot is increased by 50% when using a non-convex KAR in place of a rectangular KAR.

The video also shows a situation where the robot is pushed from the front while climbing a step; to maintain balance, the robot takes a step back before resuming its ascent.

Stepping Stones In this scenario, HRP-4 should walk in a straight line but the ground is replaced by a number of small flat objects (stones) of various shapes and heights, which are separated by gaps (Fig. 6). The robot has no prior knowledge about the stones, which are therefore ignored in the footstep plan. To test the reactivity of the proposed framework, we make the rather extreme assumption that each stone is detected only when the next KAR (the one associated with the first footstep in the control horizon) intersects it.

When the next KAR intersects a stone, the robot can in principle decide to step on it or avoid it altogether. The SR computation of Section III-A automatically retains a stone as steppable only if its surface lies inside the KAR and the whole foot fits on it, treating it as an obstacle otherwise. The decomposition of Section III-B then yields subregions at different elevations, among which (4) selects the best. As a result, HRP-4 is able to navigate this complex environment efficiently, either stepping on or avoiding stones as deemed convenient by the algorithm.

Also shown in the video is a simulation where the stones are scattered on to ground and the robot adapts the original plan by avoiding or stepping on them.

Staircase Climbing In this scenario, the robot is faced with a staircase that was not considered in the footstep plan. With the previous parameters, the proposed method leads to HRP-4 slowly climbing the stairs one step at a time (see video). A quicker gait in which the robot ascends with 'one foot per step' (Fig. 7) was obtained by decreasing from 0.6 to 0.2 the weight α_1 associated to the feasibility margin in the SSR selection (3), and correspondingly increasing from 0.3 to 0.7 the weight α_3 associated to plan conformity.

To assess the sensitivity of the proposed method to the choice of the weights in (3), we repeated the staircase-climbing experiment for combinations of weights obtained by discretizing the simplex $\sum_{i=1}^{3} \alpha_i = 1$. Fig. 8 reports two performance indicators: the time needed to reach the top of the staircase and the minimum feasibility margin attained during the ascent. The results confirm the expected role of the weights: larger values of α_3 (plan conformity) lead to faster climbs, as the robot tends to follow closely the original footstep plan and therefore adopts the 'one foot per step' behavior. Conversely, larger values of α_1 (feasibility margin) result in more conservative motions, characterized by a larger minimum γ_1 but also by a slower ascent. Most importantly, the performance varies smoothly across the simplex, with no evidence of critical dependence on a particular parameter choice. This supports the interpretation of the weights as high-level behavioral preferences governing the trade-off between conservativeness and plan adherence, rather than as scenario-specific tuning parameters.

Finally, the video includes a successful simulation on an irregular staircase with steps of variable depth and height.

B. Comparison With Mixed-Integer-MPC

A different version of our method can be envisaged, in which rather than selecting the 'best' SSR via (4) for convexifying the next footstep constraint (6), the optimization problem considers all the steppable subregions simultaneously. This leads to a Mixed-Integer MPC formulation, where the choice of the subregions is encoded by binary decision variables and the next footstep constraints are written using the big-M technique [13]. An appropriate solver must be used; in our implementation, we chose Gurobi.

We have compared the performance of our convexified MPC with that of the Mixed-Integer MPC in 50 simulated stepping stones scenarios. Each scenario was randomly generated, with up to 10 stones having length and depth in the range $0.05 \div 0.1$ m, and height between 0.01 and 0.05 m. The results, summarized

TABLE I

method	iteration time, avg [ms]	standard deviation [ms]	iteration time, max [ms]	success rate
Convexified MPC	6	2	14	100%
Mixed-Integer MPC	242	98	880	98%

in Table I in terms of the time needed to solve the optimization problem within a single iteration, confirm that pre-selecting the best SSR is essential for achieving real-time efficiency. At the same time, this does not lead to a degradation in performance: the success rate is actually better, consistently with the fact that our selection strategy rewards feasibility.

VI. Conclusion

This letter presented a feasibility-driven method for robust gait generation in complex 3D environments. At each control cycle, the non-convex steppable region — arising from the intersection of the kinematically admissible region with the elevation map — is decomposed into convex subregions. Among these, the one maximizing a weighted combination of feasibility margin, feasibility volume, and plan conformity is selected to define a standard QP footstep constraint. A key ingredient of the proposed method is the closed-form characterization of the MPC feasibility region as a box in capture-point space. This enables real-time evaluation of feasibility margin and feasibility volume, which form the basis of the proposed steppable-subregion selection strategy.

Dynamic simulations on the HRP-4 humanoid validated the approach across three challenging scenarios, demonstrating robust and reactive behavior in all cases. A direct comparison with the corresponding Mixed-Integer MPC formulation confirmed that pre-selecting the steppable subregion before optimization yields a $40\times$ reduction in average computation time while maintaining a higher success rate

The proposed approach can be used with any MPC-based gait generation technique (possibly using a template model other than LIP) as long as one can characterize its feasibility.

We are currently working on the extension of the proposed method to footsteps with arbitrary orientations, which requires generalizing the characterization of feasibility regions to account for the rotation of the KAR; this can be achieved by relying on the notion of support function of convex sets. A byproduct of this will be the direct applicability of the method to the case of environments consisting of non-horizontal patches (*world of ramps*).

Future work will also include the integration with on-line map building from on-board perception, and the incorporation of residual learning techniques to further improve adaptability in unstructured environments.

References

[1] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," in *Proc. 8th Conf. Robot Learn.*, 2025, pp. 1975–1991.
[2] J. P. Sleiman et al., "ZEST: Zero-shot embodied skill transfer for athletic robot control," 2026, *arXiv:2602.00401*.
[3] P. B. Wieber, "Viability and predictive control for safe locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 1103–1108.
[4] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robot. Automat. Lett.*, vol. 2, no. 1, pp. 10–17, Jan. 2017.
[5] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda ASIMO humanoid," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 629–634.
[6] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots*, 2019, pp. 9–16.
[7] M. Lee et al., "Dynamic humanoid locomotion over rough terrain with streamlined perception-control pipeline," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 4111–4117.
[8] M. Cipriano, P. Ferrari, N. Scianca, L. Lanari, and G. Oriolo, "Humanoid motion generation in a world of stairs," *Robot. Auton. Syst.*, vol. 168, 2023, Art. no. 104495.
[9] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Robot.*, vol. 24, no. 5/6, pp. 719–737, 2010.
[10] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 279–286.
[11] B. Aceituno-Cabezas et al., "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2531–2538, Jul. 2018.
[12] A. Ibanez, P. Bidaud, and V. Padois, "Emergence of humanoid walking behaviors from mixed-integer model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 4014–4021.
[13] M. R. Maximo and R. J. Afonso, "Mixed-integer quadratic programming for automatic walking footstep placement, duration, and rotation," *Optimal Control Appl. Methods*, vol. 41, no. 6, pp. 1928–1963, 2020.
[14] B. Acosta and M. Posa, "Perceptive mixed-integer footstep control for underactuated bipedal walking on rough terrain," *IEEE Trans. Robot.*, vol. 41, pp. 4518–4537, 2025.
[15] Y. Kojio et al., "Footstep modification including step time and angular momentum under disturbances on sparse footholds," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4907–4914, Jul. 2020.
[16] A. S. Habib, F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Handling non-convex constraints in MPC-based humanoid gait generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13167–13173.
[17] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, no. 34, pp. 189–206, 2013.
[18] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1171–1188, Aug. 2020.
[19] T. Sugihara, K. Imanishi, T. Yamamoto, and S. Caron, "3D biped locomotion control including seamless transition between walking and running via 3D ZMP manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6258–6263.
[20] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

Open Access funding provided by 'Sapienza Universita di Roma' within the CRUI CARE Agreement