2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

Handling Non-Convex Constraints in MPC-Based Humanoid Gait Generation

Andrew S. Habib, Filippo M. Smaldone, Nicola Scianca, Leonardo Lanari, Giuseppe Oriolo

***Abstract*— In most MPC-based schemes used for humanoid gait generation, simple Quadratic Programming (QP) problems are considered for real-time implementation. Since these only allow for convex constraints, the generated gait may be conservative. In this paper we focus on the non-convex reachable region of the swinging foot, also known as Kinematic Admissible Region (KAR), and the corresponding constraint. We represent an approximation of such non-convex region as the union of multiple non-overlapping convex sub-regions. By leveraging the concept of feasibility region, i.e., the subset of the state space for which a QP problem is feasible, and introducing a proper selection criterion, we are able to maintain linearity of the constraints and thus use our Intrinsically Stable Model Predictive Control (IS-MPC) scheme with a negligible additional computational load. This approach allows for a wider range of possible generated motions and is very effective when reacting to a push or avoiding an obstacle, as illustrated in dynamically simulated scenarios.**

I. Introduction

Humanoid robots aim at achieving the flexibility of biped locomotion by mimicking human morphology. Designing control schemes capable of emulating human behavior, however, is a challenging task. In order to achieve real-time control, the complex dynamics of humanoids are often approximated using the Linear Inverted Pendulum (LIP) [1] model, which captures the essential relation between the Center of Mass (CoM) and the Zero Moment Point (ZMP, the point with respect to which horizontal momenta are zero). Dynamic balance can be guaranteed by keeping the ZMP inside the convex hull of the contact surfaces, i.e., the support polygon. The linearity of the LIP allows for efficient Model Predictive Control (MPC) [2] formulations, where the ZMP is kept within the support polygon by means of constraints.

Among the beneficial aspects of MPC, there is the possibility to include the footstep positions as decision variables of the optimization problem [3], [4], thus allowing the robot to not only plan CoM and ZMP trajectories in real-time, but also to perform reactive stepping, e.g., when subject to a disturbance. To make sure that the resulting footsteps are kinematically realizable by the robot, they must be confined to an admissible region. In general, this region is non-convex, and its boundaries depend on both the robot configuration and the environment, thus it is not realistic to compute it exactly in a real-time implementation. By far, the most common approach is to restrict this region to a convex polytope with fixed shape and dimensions (e.g., a rectangle).

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy. E-mail: {lastname}@diag.uniroma1.it.

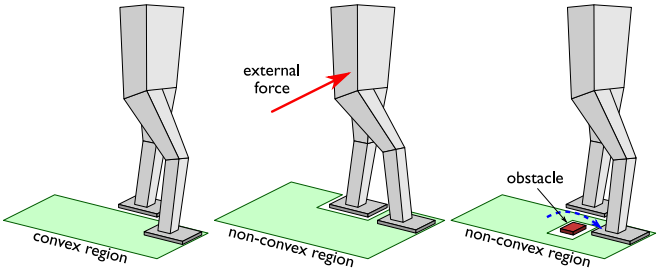

Fig. 1. MPC-based gait generation schemes commonly prescribe to place footsteps within convex regions (left). By allowing these regions to be non-convex it is possible to let the robot place one foot in front of the other (middle), hence enhancing its reactive stepping capabilities, or delimit an area where stepping is not allowed (right), e.g., in order to avoid a small obstacle.

Restricting the allowed region for the footsteps to a convex polytope allows to adopt a linear-quadratic formulation for the optimization problem, which can be very efficiently solved using the techniques of Quadratic Programming (QP) [3], [5].

On the other hand, placing footsteps in front of each other can be quite effective against perturbations, as shown in [6] where the authors use pre-computed trajectories. Obtaining this behavior in MPC-based gait generation is not straightforward as the kinematic admissible region becomes non-convex (see Fig. 1). Mixed-Integer Quadratic Programming (MIQP) permits to reformulate a non-convex constraint as a set of mutually exclusive convex constraints by introducing additional integer variables [7], [8], and is quite effective for footstep planning applications, but the introduced extra computational load does not allow for real-time implementations.

Non-convex regions arise in problems of obstacle avoidance, as the allowed states lie outside a finite region around the obstacle. Planning algorithms can usually deal efficiently with non-convex constraints [9], [10], [11], but real-time approaches often require convex approximations of the admissible regions [4]. Real-Time Iteration (RTI) (e.g., see [12]) can work with non-convex constraints by determining time-varying convex approximations around a trajectory, but it often lacks convergence guarantees, and the result might heavily rely on the quality of the initial guess. It is possible to decompose the non-convex region into multiple convex sub-regions, and select a single one to formulate a convex constraint. This is done in [13], [14], which however use a heuristic selection of the sub-region that does not involve feasibility considerations.

In [15], we presented an Intrinsically Stable MPC (IS-

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981419

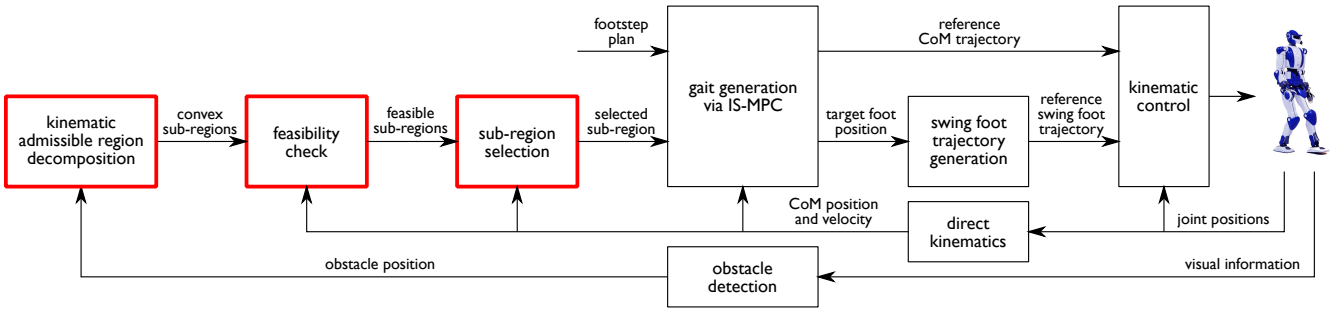

Fig. 2. A block scheme of the proposed approach. Red blocks highlight additions to the standard IS-MPC framework.

MPC) for humanoid gait generation, which features an explicit stability constraint that guarantees a bounded CoM trajectory with respect to the ZMP. IS-MPC allows automatic footstep placement, and enforces convex allowed regions for the footsteps in order to comply with the robot kinematic limits. We introduced the concept of *feasibility region*, i.e., the subset of the state space where the constrained optimization problem admits a solution, and in [16] we showed how this region can be employed in the design of a step timing adaptation module, that would otherwise require nonlinear optimization.

In this paper, we show how the feasibility region can be used to efficiently treat non-convex allowed regions for the footsteps. The idea is to divide a non-convex polytope into a set of convex sub-regions, and determine which one can be used in a QP formulation according to a specific criterion. The result of our algorithm is a unique convex sub-region, allowing for a very efficient solution since the underlying formulation remains linear-quadratic. The criterion according to which the sub-region is selected involves evaluating the feasibility of the resulting QP problem, and thus ensures that a solution will be found if it exists. Differently from related approaches (e.g., [17]) the proposed method does not require solving multiple optimization problems at a given time-step, thanks to the fact that the feasibility region can be evaluated in closed-form.

This general idea can be used in different practical applications. We consider two: extending the allowed region for the footsteps in order to improve reactive stepping, and performing obstacle avoidance (see Fig. 1). We validate the proposed method by means of dynamic simulations on an HRP-4 humanoid robot.

The paper is organized as follows. Section II provides an overview of the approach, while Sect. III briefly recalls standard IS-MPC. Section IV describes the main contribution of this paper enabling handling non-convex regions, and Sect. V show dynamic simulations on a HRP-4 humanoid robot. Concluding remarks are provided in Sect. VI.

II. The Proposed Approach

Standard IS-MPC, introduced in [15], solves at each time instant a QP problem, which admits a solution if the current state lies inside a subset of the state space called feasibility region. The bounds of this region can be easily computed given the enforced constraints. In particular, one of these constraints aims at keeping the footsteps within a convex region.

We extend this scheme in order to allow a non-convex reachable region of the swinging foot – the non-convex Kinematic Admissible Region (KAR) – and thus increase the reactivity of the humanoid, e.g., to an external push. However, since we want to keep solving QP problems for real-time implementation, we cannot include a non-convex constraint. In our extension, we first approximate the KAR with a non-convex polytope and then decompose this approximation into a set of non-overlapping convex sub-regions. Finally we choose one of the convex sub-regions in such a way to ensure feasibility of the corresponding QP problem in IS-MPC. We perform this procedure only for the first footstep inside the control horizon, although in principle it can be extended to multiple predicted footsteps.

The general architecture of the modified IS-MPC scheme is shown in Fig. 2. A footstep plan, consisting of a sequence of candidate footstep positions, orientations, and their associated timings, is given as input to an IS-MPC block which computes CoM/ZMP trajectories and the actual footstep positions in order to match the original plan as close as possible. A set of linear constraints are enforced at each time instant in the underlying QP problem.

One of these constraints requires the footsteps to be within a specific convex area. In the reported scheme, this area, a convex sub-region of the KAR, is determined by a set of basic steps:

- in the *kinematic admissible region decomposition* block the KAR region is approximated as a non-convex polytope and then divided into a set of non-overlapping convex sub-regions;
- the *feasibility check* block evaluates the feasibility of each sub-region, i.e., determines if using any of the convex sub-regions as an allowed area for the footsteps results in a feasible QP problem. This block outputs a set of feasible convex sub-regions;
- the *sub-region selection* block evaluates all these sub-regions and, according to a specific criterion, chooses a unique feasible sub-region. This region will activate the corresponding kinematic constraint in the IS-MPC QP problem.

The IS-MPC block generates CoM/ZMP trajectories and

footstep positions, along with the swing foot trajectory, which are sent to a kinematic control block where joint commands are finally computed.

We propose two possible applications of this extended IS-MPC scheme: i) the area where the swinging foot is allowed to step is enlarged with respect to the usual conservative rectangle, thus allowing to place a foot in front of the other; this extends the capabilities of reactive stepping w.r.t. external perturbations; ii) if one or more small obstacles are present, we consider the non-convex obstacle free region as the KAR. It is assumed that the obstacle is detected.

This technique can be applied to a wide range of scenarios, and it adds no significant computational cost, as the the formulation of the optimization problem at the core of MPC remains quadratic with linear convex constraints. We apply the procedure only to the first predicted footstep in order to simplifies the algorithm. Nevertheless, a more general formulation could easily be implemented and still have negligible impact on the computational load. The first footstep, however, has the most immediate effect on the resulting gait, and extension to footsteps farther into the future has diminishing returns.

III. Intrinsically Stable MPC

This section will describe the standard IS-MPC algorithm. The scheme works in a discrete fashion over sampling intervals of duration δ. Its goal is to determine CoM and ZMP trajectories, as well as footstep positions, such that the latter follow as closely as possible a given footstep plan, which is made available over a preview horizon $T_p = P\delta$, across P sampling intervals. A prediction model is used to forecast the evolution of the system along a control horizon of duration $T_c = C\delta$, across C sampling intervals, with $C < P$. The current time is denoted as $t_k = k\delta$, and the generic time along the control horizon is indicated as $t_{k+i} = (k+i)\delta$, with $i = 0, \dots, C-1$. Sampled variables are denoted with a superscript, e.g., $x(t_{k+i}) = x^{k+i}$.

We assume that all footsteps have the same orientation, which simplifies the exposition by decoupling the x and the y component in every equation[1]. Whenever possible, we will only give equations for the x component, with the understanding that there will be an identical counterpart for the y component.

The footstep plan specifies the beginning of each step, denoted with t_s^j, taken to be at the beginning of the corresponding double support phase. The j-th step will have a duration T_{s}^j, split into a double support phase of duration T_{ds}^j and a single support phase of duration T_{ss}^j.

A. Prediction Model

The prediction model for the CoM/ZMP dynamics is the LIP, which is derived by considering the moment balance around the ZMP and assuming a flat horizontal ground, constant CoM height and negligible angular momentum variation around the CoM. The dynamics along the x axis (sagittal) and y axis (coronal) can be represented by two decoupled identical linear differential equations, e.g., for x

$$\ddot{x}_c = \eta^2 (x_c - x_z), \tag{1}$$

where x_c and x_z denote the CoM and ZMP respectively, while $\eta = \sqrt{g/\bar{z}_c}$ is the pendulum natural frequency, with g the gravity acceleration, and $\bar{z}_c$ the constant height of the CoM. In order to achieve smoother trajectories, the model is dynamically extended to have the ZMP velocity $\dot{x}_z$ as input.

B. ZMP Constraints

To keep the ZMP within the support polygon, we adopt a *moving constraint* formulation [18], where the allowed area for the ZMP is given by a region with fixed shape, having the same dimensions of the footprint. The center of this region moves in such a way that the moving shape is always contained within the support polygon.

The ZMP constraint can be written as

$$\left| x_z - x_f^{j-1} - (x_f^j - x_f^{j-1}) \sigma(t, t_s^j, t_s^j + T_{ds}^j) \right| \leq \frac{1}{2} d_{z,x},$$

for $t \in \left[t_s^j, t_s^{j+1}\right)$, where $d_{z,x}$ is the square footprint width. Here σ is a piecewise linear sigmoidal function, defined as

$$\sigma\left(t, t_i, t_f\right) = \frac{1}{t_f - t_i} \left(\rho\left(t - t_i\right) - \rho\left(t - t_f\right)\right),$$

where $\rho(t) = t\delta_{-1}(t)$ is the unit ramp. In a more compact form, it is possible to write the ZMP constraints as

$$x_z^m(t, x_f^1, \dots, x_f^F) \leq x_z \leq x_z^M(t, x_f^1, \dots, x_f^F), \tag{2}$$

for $t \in \left[t_s^j, t_s^{j+1}\right)$, where the ZMP bounds x_z^m and x_z^M appear as functions of time as well as of the predicted footstep positions.

C. Stability Constraint

Due to the unstable nature of humanoid dynamics, it is not sufficient to ensure that the ZMP is inside the support polygon at all times, as the associated CoM trajectory might be divergent w.r.t. the ZMP, leading to an unrealizable motion. In order to avoid this, IS-MPC includes a stability constraint that guarantees that the CoM trajectory is bounded with respect to the ZMP.

In the LIP, the unstable behavior is highlighted by defining the coordinate x_u, often referred to as Divergent Component of Motion (DCM) [19] or capture point [20]: $x_u = x_c + \dot{x}_c/\eta$, which results in the unstable dynamics

$$\dot{x}_u = \eta \left(x_u - x_z\right).$$

x_c will remain bounded with respect to x_z provided that

$$x_u^k = \eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} x_z(\tau) d\tau, \tag{3}$$

and that $\dot{x}_z$ is bounded. The integral in the right-hand side can be split as the integral from t_k to $t_k + T_c$, that can be readily expressed in terms of the ZMP velocity inputs $\dot{x}_z$ within the MPC control horizon, plus the integral from $t_k + T_c$ to $+\infty$, which depends on the inputs outside the control

[1]This assumption is adopted to simplify the exposition of the IS-MPC scheme itself. The proposed extension is unaffected, and would work identically even with variable orientations.

horizon and therefore makes condition (3) non-causal. To obtain a causal constraint, we compute an approximate value for the second integral by using an anticipative tail, i.e., a ZMP trajectory $\tilde{x}_z$ from $t_k + T_c$ to $+\infty$ conjectured on the basis of short-term information given by the footstep plan. This leads to the following stability constraint

$$\eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} x_z(\tau) d\tau = x_u^k - \tilde{c}_x^k, \tag{4}$$

where

$$\tilde{c}_x^k = \eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_k)} \tilde{x}_z(\tau) d\tau.$$

Note that the integral in the left-hand side of (4) can be written in closed form as a linear function of the ZMP velocity inputs $\dot{x}_z^k, \ldots, \dot{x}_z^{k+C-1}$.

D. Kinematic Constraints

The kinematic constraint is enforced to guarantee that all footsteps are positioned in such a way to comply with the kinematic limits of the robot. In the standard version of IS-MPC, the j-th predicted footstep must lie within a convex region $\bar{\mathcal{K}}^j$, chosen as a rectangular approximation of the KAR:

$$(x_f^j,\, y_f^j) \in \bar{\mathcal{K}}^j, \quad j = 1, \ldots, F, \tag{5}$$

with $\bar{\mathcal{K}}^j$ given by

$$\bar{\mathcal{K}}^j = \left\{ (x, y) : \left| x - x_f^{j-1} \right| \leq \frac{d_{a,x}}{2}, \left| y - y_f^{j-1} \pm \ell \right| \leq \frac{d_{a,y}}{2} \right\}.$$

Here, $d_{a,x}$ and $d_{a,y}$ are the dimensions of this rectangular approximation, ℓ is a lateral displacement, and the $\pm$ sign is a shorthand notation to express the fact that the displacement occurs alternatively towards the positive and negative direction of the y axis, depending on whether the support foot is the right or left one.

E. QP Formulation

At each iteration the IS-MPC solves a QP problem whose decision variables are given by

$$\dot{X}_z^k = \begin{pmatrix} \dot{x}_z^k & \ldots & \dot{x}_z^{k+C-1} \end{pmatrix}^T, \quad \dot{Y}_z^k = \begin{pmatrix} \dot{y}_z^k & \ldots & \dot{y}_z^{k+C-1} \end{pmatrix}^T,$$
$$X_f^k = \begin{pmatrix} x_f^1 & \ldots & x_f^F \end{pmatrix}^T, \qquad Y_f^k = \begin{pmatrix} y_f^1 & \ldots & y_f^F \end{pmatrix}^T.$$

The QP problem is formulated as

$$\begin{cases} \min\limits_{\substack{\dot{X}_z^k, \dot{Y}_z^k, \\ X_f^k, Y_f^k}} \left\| \dot{X}_z^k \right\|^2 + \left\| \dot{Y}_z^k \right\|^2 + \beta \left(\left\| X_f - \hat{X}_f \right\|^2 + \left\| Y_f - \hat{Y}_f \right\|^2 \right) \\ \quad \text{subject to:} \\ \text{- ZMP constraints (2) for } x \text{ and } y \\ \text{- stability constraints (4) for } x \text{ and } y \\ \text{- kinematic constraints (5).} \end{cases} \tag{6}$$

This QP problem is solved at each iteration, computing the full predicted trajectory over the control horizon. In the typical MPC fashion, the first control sample from the solution $(\dot{x}_z^k, \dot{y}_z^k)$ is used to integrate (1), obtaining a desired CoM position. The first predicted footstep (x_f^1, y_f^1) is used as a target position to generate a swing foot trajectory. These CoM and swing foot trajectories are treated as references to be tracked by the kinematic controller.

F. Feasibility Region

As shown in [16], in order for IS-MPC to be feasible, i.e., the QP to be solvable, the state at t_k must be within a feasibility region. The bounds of this region can be determined based on the constraints imposed on the QP problem constructed at time t_k. On the converse, if the state is not inside the feasibility region, the QP problem (6) does not admit a solution.

For example, when the robot is subject to a strong perturbation, which almost instantaneously modifies the state, we may have two possible outcomes: i) the modified state might still be inside the feasibility region, and so the MPC will be perfectly capable of recovering from the perturbation; ii) the modified state might exit the feasibility region, in which case the constrained optimization has no solution and the algorithm fails.

Let us recall a result that will be used in the following section.

Proposition 1: IS-MPC is feasible at t_k if and only if $(x_u^k, y_u^k) \in \mathcal{F}^k$, with

$$\mathcal{F}^k = \{(x_u, y_u) : x_u^{k,m} \leq x_u \leq x_u^{k,M}, y_u^{k,m} \leq y_u \leq y_u^{k,M}\}$$

with the following compact notation for $x_u^{k,m}$ and $x_u^{k,M}$ (identical quantities hold for the y component)

$$\begin{aligned} x_u^{k,m} &= \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} x_z^{m,\min}(\tau) d\tau + \tilde{c}_x^k \\ x_u^{k,M} &= \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} x_z^{M,\max}(\tau) d\tau + \tilde{c}_x^k \end{aligned}$$

in which

$$x_z^{m,\min}(t) = \min_{X_f^k} x_z^m(t, x_f^1, \ldots, x_f^F) \tag{7}$$

$$x_z^{M,\max}(t) = \max_{X_f^k} x_z^M(t, x_f^1, \ldots, x_f^F) \tag{8}$$

with X_f^k subject to the kinematic constraints (5).

Proof. See [16]. ∎

Equations (7) and (8) represent respectively the minimum and maximum allowed trajectories for the ZMP, which are made independent of the footstep position by performing a $\min$ and $\max$ operation with respect to X_f^k, subject to the kinematic constraint. Despite here being defined implicitly, these terms can be expressed in closed form, and so can the bounds of the feasibility region.

In the following, we will make use of the fact that the bounds of $\mathcal{F}^k$ depend on the allowed area for the footsteps, due to the necessity to satisfy the kinematic constraint. In particular, it will be important to express its dependency on the allowed area for the first footstep in the control horizon. In order to highlight this dependency, let us adopt the notation $\mathcal{F}^k = \mathcal{F}^k(\bar{\mathcal{K}}^1)$.

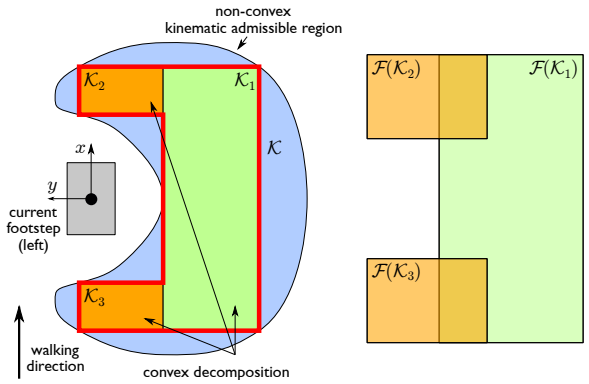

Fig. 3. Left: in standard IS-MPC the approximation of the KAR is limited to $\mathcal{K}_1$ (see Sect. III-D). In the proposed method we also consider the orange regions (see Sect. IV-D) in order to better approximate the actual KAR (blue region). The resulting non-convex polytope $\mathcal{K}$ is outlined in red. Right: a possible configuration of the corresponding feasibility regions for each sub-region (not-to-scale for improved clarity).

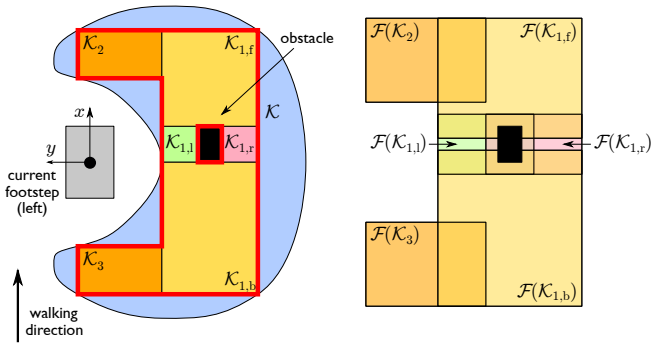

Fig. 4. Left: non-convex approximation $\mathcal{K}$ of the KAR (red outline) when an obstacle is present (black), and the corresponding decomposition in sub-regions. Right: a possible configuration of the corresponding feasibility regions for each sub-region (not-to-scale for improved clarity).

IV. Handling the Non-Convex Constraint

In this section we expand the three red blocks of Fig. 2 which constitute the novelty of the proposed approach.

A. Kinematic Admissible Region Decomposition

Let us approximate the KAR for the first step inside the control horizon as a non-convex polytope $\mathcal{K}$. This polytope is an extension of the region $\bar{\mathcal{K}}^1$, defined in Sect. III-D (see Fig. 3). Note that we drop the superscript to lighten the notation, as the following procedure is only applied to the first predicted step.

The *kinematic admissible region decomposition* block performs a decomposition of $\mathcal{K}$ into a set of N non-overlapping convex sub-regions, i.e.,

$$\mathcal{K} = \bigcup_{\nu=1}^{N} \mathcal{K}_\nu \quad \text{s.t.} \quad \bigcap_{\nu=1}^{N} \mathcal{K}_\nu = \emptyset.$$

In general, there is not a unique way to decompose a given non-convex set. We will describe our decomposition procedure for the the two considered scenarios in Sects. IV-D and IV-E. Other choices are of course possible.

B. Feasibility Check

Any sub-region $\mathcal{K}_\nu$, if used in the kinematic constraint (5), will produce a QP problem that is structurally identical to (6), and whose feasibility can be evaluated by checking whether

$$(x_u^k,\ y_u^k) \in \mathcal{F}^k(\mathcal{K}_\nu). \tag{9}$$

i.e., if $(x_u^k,\ y_u^k)$ belongs to the feasibility region $\mathcal{F}^k(\mathcal{K}_\nu)$. Therefore, the *feasibility check* block checks (9) for each sub-region $\mathcal{K}_\nu$, with $\nu = 1, \ldots, N$, and determines the subset of all feasible sub-regions.

Note that, while the sub-regions $\mathcal{K}_\nu$ are chosen to be non-overlapping, the corresponding feasibility regions $\mathcal{F}^k(\mathcal{K}_\nu)$ can overlap. This means that in general there may be more than one feasible sub-region for $(x_u^k,\ y_u^k)$ at any given time.

C. Sub-Region Selection

The *sub-region selection* block chooses a unique sub-region among all the ones that have been found to be feasible. The criterion for this selection will be articulated in two stages, which will lead to a unique result.

At first, a priority $p(\mathcal{K}_\nu)$ is assigned at each sub-region, where higher priority indicates that it is preferable to step inside the corresponding sub-region $\mathcal{K}_\nu$. Only the highest priority regions are kept – since the feasibility regions can overlap, we may have more than one sub-region with highest priority – thus discarding all the lower priority ones.

Then, among all feasible sub-regions with maximum priority, we choose the sub-region $\mathcal{K}_\nu$ that maximizes the minimum distance between the boundary $\mathcal{F}^k(\mathcal{K}_\nu)$ and $(x_u^k,\ y_u^k)$. This aims at maximizing robustness with respect to perturbations. Intuitively, it will also reduce the probability to pick a different region in subsequent iterations.

D. Decomposition Procedure: Improving Reactive Stepping

We want to endow the robot with the capability of placing one foot in front of the other. This becomes particularly useful in case of external perturbations directed laterally from the swinging foot towards the support foot, because the next footstep can be placed further along the direction of the push, thus granting a higher chance of recovery. We make this possible by considering the KAR and its non-convex polytope approximation as shown in Fig. 3. This polytope includes the rectangular KAR approximation used in standard IS-MPC (see Sect. III-D), and augments it by further extending it in front and behind the stance foot. The convex decomposition of this polytope is operated by simply having the region $\mathcal{K}_1 \equiv \bar{\mathcal{K}}^1$ extended as much as possible along the x direction, and two smaller rectangular regions $\mathcal{K}_2$ and $\mathcal{K}_3$ in front and behind the current support foot. The larger region $\mathcal{K}_1$ is given priority 1, while, the smaller regions $\mathcal{K}_2$ and $\mathcal{K}_3$ are given priority 0. In this way the robot will place footsteps in front of each other only when this is strictly necessary in order to recover from a perturbation, and will prefer a normal gait under any other circumstance.

E. Decomposition Procedure: Obstacle Avoidance

Obstacle avoidance involves non-convex constraints, as the admissible region is outside of a finite set (a region surrounding the obstacle). We describe a procedure for dealing with a small obstacle, i.e., such that the robot can step past it but not directly on it.

We propose to approach this case by dividing the non-convex polytope approximation of the KAR, as shown in Fig. 4. The obstacle is enclosed in a rectangular region, and the surrounding space is divided in four sub-regions. For consistency, we keep the notation as close as possible to the naming used in Sect. IV-D. As such, the regions in front and behind the support foot $\mathcal{K}_2$ and $\mathcal{K}_3$ are identical as in the previous subsection, while $\mathcal{K}_1$ is further divided into $\mathcal{K}_{1,\mathrm{f}}$, $\mathcal{K}_{1,\mathrm{b}}$, $\mathcal{K}_{1,\mathrm{l}}$ and $\mathcal{K}_{1,\mathrm{r}}$ (respectively front, back, left and right with respect to the obstacle). The decomposition is operated[2] by having $\mathcal{K}_{1,\mathrm{f}}$ and $\mathcal{K}_{1,\mathrm{b}}$ maximally extended in the y direction, while the remaining space is assigned to $\mathcal{K}_{1,\mathrm{l}}$ and $\mathcal{K}_{1,\mathrm{r}}$, as depicted in Fig. 4. We assign priority 1 to the sub-regions $\mathcal{K}_{1,\mathrm{f}}$, $\mathcal{K}_{1,\mathrm{b}}$, $\mathcal{K}_{1,\mathrm{l}}$ and $\mathcal{K}_{1,\mathrm{r}}$, while sub-regions $\mathcal{K}_2$ and $\mathcal{K}_3$ have priority 0.

The decomposition procedure just described can always be carried out whenever the obstacle is fully contained inside $\mathcal{K}_1$. If this is not the case, a different treatment will be required. For example, the obstacle might be partially contained in $\mathcal{K}_1$, or on the boundary between $\mathcal{K}_1$ and $\mathcal{K}_2$. These cases poses no particular challenge, as a decomposition procedure can be found just as easily. We omit the details here for compactness.

V. Dynamic Simulations

We performed dynamic simulations on an HRP-4 humanoid robot in the DART environment, using the following parameters: $z_c = 0.75$ m, $d_{z,x} = 0.08$ m, $d_{z,y} = 0.08$ m, $T_c = 1$ s, $T_p = 2.5$ s, $\delta = 0.01$ s. All footsteps have the same duration $T^j_{\mathrm{s}} = 1$ s, with $T^j_{\mathrm{ds}} = 0.3$ s and $T^j_{\mathrm{ss}} = 0.7$ s. The size of the sub-regions $\mathcal{K}_1$ is 0.5 m along x and 0.16 m along y, with a lateral displacement $\ell = 0.2$ m. The size of the sub-region $\mathcal{K}_2$ and $\mathcal{K}_3$ is 0.1 m along x and 0.2 m along y. The QP problems are solved using *hpipm* [21]. A full iteration takes less than 10 ms on a standard PC, and can thus run in real-time. Videos and comparisons with standard IS-MPC are included in the accompanying video.

A. Push Recovery

In the first simulation, HRP-4 is walking and is subject to a diagonal push with magnitude $(100,\ 135,\ 0)$ N, occurring at $t = 4$ s (the beginning of the single support phase of the fourth step) and lasting for 0.1 s.

As shown in Fig. 5, the push causes a displacement of (x^k_u, y^k_u) which exits the feasibility region $\mathcal{F}^k(\mathcal{K}_1)$. However, it is now contained in $\mathcal{F}^k(\mathcal{K}_2)$, which means that the robot can recover by placing the right foot in $\mathcal{K}_2$, i.e., in front of the left foot. See also the accompanying video.

[2]There are multiple ways of subdividing this polytope into convex sub-regions, and the proposed choice is one of the most natural. However, other choices might be just as effective.

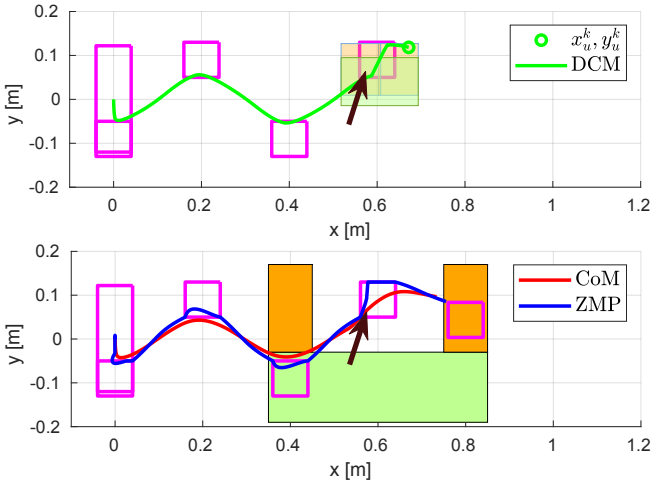

Fig. 5. Push recovery simulation. The top plot ($t = 4.45$ s) shows the state right after the push (denoted as an arrow). In particular, $(x^k_u,\ y^k_u)$ is inside $\mathcal{F}^k(\mathcal{K}_2)$, so the push is recovered by putting the right foot in front of the left, as shown in the bottom plot ($t = 4.99$ s).

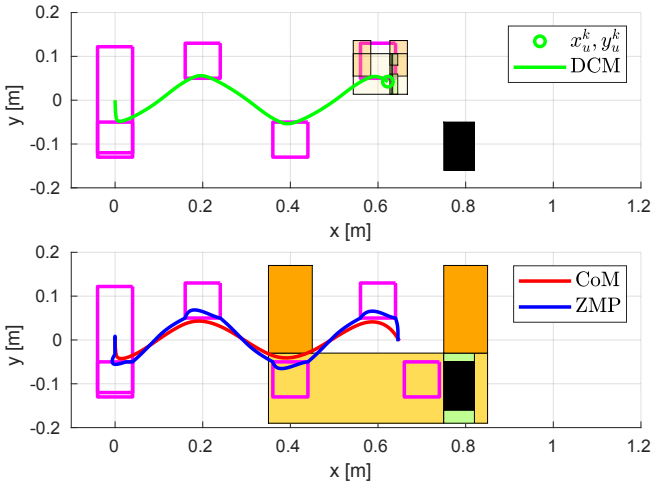

Fig. 6. Obstacle avoidance simulation. The top plot ($t = 4.30$ s) shows the decomposition of the non-convex polytope when the obstacle is encountered, as well as the position of $(x^k_u,\ y^k_u)$. The bottom plot ($t = 4.85$ s) shows the resulting placement of the next footstep.

B. Obstacle Avoidance

In this second simulation, a small obstacle with dimensions 0.04 m×0.07 m×0.01 m is placed along the robot path.

Following the footstep plan would lead the robot to step onto the black obstacle, but since this is accounted for in the decomposing procedure, the footstep lands behind the obstacle. Stepping over the obstacle will be achieved at the subsequent step. Figure 6 shows the gait and footsteps, and the accompanying video reports the full animation, as well as a more challenging scenario involving multiple obstacles.

C. Mixed Scenario

In the last simulation, we show how the robot reacts to multiple pushes as well as to the presence of an obstacle. The size of the obstacle is the same as in the previous simulation.

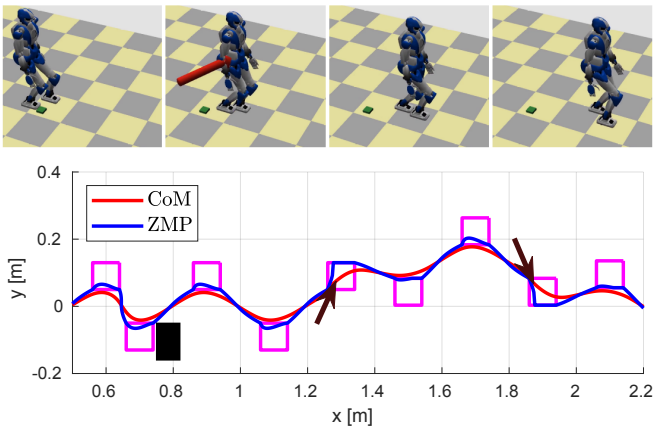

Fig. 7. Push recovery and obstacle avoidance simulation: snapshots (top) and CoM/ZMP plots (bottom). The robot successfully recovers from multiple pushes and avoids stepping on an obstacle.

The robot encounters the obstacle during the fourth step, and is then subject to two consecutive pushes during the sixth and seventh step, with magnitude $(80,\ -105,\ 0)$ N and $(80,\ 120,\ 0)$ N respectively. As shown in Fig. 7 (bottom), thanks to the proposed approach, the robot is able to both avoid the obstacle and suppress the effect of the perturbations by placing one foot in front of the other. Figure 7 (top) shows snapshots of the simulation which is included in the accompanying video.

VI. Conclusions

The proposed method enhances the original IS-MPC framework allowing non-convex areas for the reachable region of the swinging foot, while adding a negligible computational load. DART simulations showed that this method increases the reactive capabilities of the original scheme, and can be employed to perform obstacle avoidance.

Future work will address the following points:

- the proposed scheme can be integrated within a larger framework, including feasibility-driven step timing adaptation [16];
- by using an appropriate 3D model in place of the LIP it can be extended to non-flat ground, where non-convex allowed regions for the footsteps arise very naturally in the presence of stair steps, and similar features;
- in order to work in cluttered environments, where several obstacles can be present at the same time, one could include a state machine in order to discern between the possible topological arrangements of the obstacles within the kinematic admissible region;
- in order to account for tall obstacles, that do not allow the robot to step past them, one should extend the region in which the foot cannot be placed, in order to include all areas that would lead to a foot collision during a swinging motion.

References

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *2001 IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2001, pp. 239–246.

[2] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *6th IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 137–142.

[3] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[4] D. De Simone, N. Scianca, P. Ferrari, L. Lanari, and G. Oriolo, "MPC-based humanoid pursuit-evasion in the presence of obstacles," in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 5245–5250.

[5] A. Bratta, R. Orsolino, M. Focchi, V. Barasuol, G. G. Muscolo, and C. Semini, "On the hardware feasibility of nonlinear trajectory optimization for legged locomotion based on a simplified dynamics," in *2020 IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 1417–1423.

[6] Z. Gu, N. Boyd, and Y. Zhao, "Reactive locomotion decision-making and robust motion planning for real-time perturbation recovery," in *2022 IEEE Int. Conf. on Robotics and Automation*, 2022, pp. 1896–1902.

[7] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 279–286.

[8] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018.

[9] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime search-based footstep planning with suboptimality bounds," in *12th IEEE-RAS Int. Conf. on Humanoid Robots*, 2012, pp. 674–679.

[10] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *2019 IEEE-RAS Int. Conf. on Humanoid Robots*, 2019, pp. 9–16.

[11] P. Ferrari, N. Scianca, L. Lanari, and G. Oriolo, "An integrated motion planner/controller for humanoid robots on uneven ground," in *18th European Control Conf.*, 2019, pp. 1598–1603.

[12] 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 Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2016.

[13] Y. Kojio, Y. Omori, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Footstep modification including step time and angular momentum under disturbances on sparse footholds," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4907–4914, 2020.

[14] H. Jeong, J.-H. Kim, O. Sim, and J.-H. Oh, "Avoiding obstacles during push recovery using real-time vision feedback," in *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 483–490.

[15] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1171–1188, 2020.

[16] F. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Feasibility-driven step timing adaptation for robust MPC-based gait generation in humanoids," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1582–1589, 2021.

[17] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712–2733, 2018.

[18] A. Aboudonia, N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "Humanoid gait generation for walk-to locomotion using single-stage MPC," in *17th IEEE-RAS Int. Conf. on Humanoid Robots*, 2017, pp. 178–183.

[19] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot - 1st report: Walking gait pattern generation," in *2009 Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[20] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *6th IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 200–207.

[21] 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.