MPC for Humanoid Gait Generation: Stability and Feasibility

Nicola Scianca, Daniele De Simone, Leonardo Lanari, and Giuseppe Oriolo, *Fellow, IEEE*

***Abstract*—In this article, we present an intrinsically stable Model Predictive Control (IS-MPC) framework for humanoid gait generation that incorporates a stability constraint in the formulation. The method uses as prediction model a dynamically extended Linear Inverted Pendulum with Zero Moment Point (ZMP) velocities as control inputs, producing in real time a gait (including footsteps with timing) that realizes omnidirectional motion commands coming from an external source. The stability constraint links future ZMP velocities to the current state so as to guarantee that the generated Center of Mass (CoM) trajectory is bounded with respect to the ZMP trajectory. Being the MPC control horizon finite, only part of the future ZMP velocities are decision variables; the remaining part, called *tail*, must be either conjectured or anticipated using preview information on the reference motion. Several options for the tail are discussed, each corresponding to a specific terminal constraint. A feasibility analysis of the generic MPC iteration is developed and used to obtain sufficient conditions for recursive feasibility. Finally, we prove that recursive feasibility guarantees stability of the CoM/ZMP dynamics. Simulation and experimental results on NAO and HRP-4 are presented to highlight the performance of IS-MPC.**

***Index Terms*—Gait generation, humanoid robots, internal stability, legged locomotion, predictive control, recursive feasibility.**

I. Introduction

MANY gait generation approaches for humanoids guarantee that balance is maintained during locomotion by enforcing the condition that the Zero Moment Point (ZMP, the point where the horizontal component of the moment of the ground reaction forces becomes zero) remains at all times within the support polygon of the robot. Correspondingly, these approaches identify the ZMP as the fundamental variable to be controlled.

Due to the complexity of full humanoid dynamics, however, direct control of the ZMP is very difficult to achieve. In view of this, simplified models are generally used to relate the evolution of the ZMP to that of the Center of Mass (CoM) of the robot, which can be instead effectively controlled. Widely adopted linear models are the Linear Inverted Pendulum (LIP), in which the ZMP represents an input, and the Cart-Table (CT), where the ZMP appears as the output [1]. The first is appropriate for inversion-based control approaches: given a sequence of footsteps, and thus a ZMP trajectory interpolating them, the LIP is used to compute a CoM trajectory which corresponds to the ZMP trajectory (see, e.g., [2]–[4]). The CT model lends itself more naturally to the design of feedback laws for tracking ZMP trajectories, the most successful example in this context being the LQ preview controller of [5].

Regardless of the adopted model, there is a potential instability issue at the heart of the problem. In particular, a certain ZMP trajectory may be realized by an infinity of CoM trajectories, which, due to the nature of the CoM/ZMP dynamics, will in general be *divergent* with respect to the ZMP trajectory itself. In this situation, dynamic balance can be in principle achieved by properly choosing the ZMP trajectory, but *internal instability* indicates that such motion will not be feasible in practice for the humanoid.

The seminal paper [6] reformulates the gait generation problem in a Model Predictive Control (MPC) setting. This is convenient because it allows to generate simultaneously the ZMP and the CoM trajectories while satisfying constraints, such as the ZMP balance condition as well as kinematic constraints on the maximum step length and foot rotation [7]. Moreover, the MPC approach guarantees a certain robustness against perturbations. It is, therefore, not surprising that it has been adopted in many methods for gait generation; e.g., see [8]–[11] for linear MPC and [12] and [13] for nonlinear MPC.

As for all control schemes, a fundamental issue in MPC approaches is the stability of the obtained closed-loop system, especially in view of the previous remark about the instability of the CoM/ZMP dynamics. As discussed in [14], two main approaches have emerged for achieving stability when MPC is used for humanoid gait generation. The first is heuristic in nature and consists in using a sufficiently long control horizon [15], so that the optimization process can discriminate against diverging behaviors, as done, for example, in [7]. The second approach has been to enforce a terminal state constraint (i.e., a constraint

Manuscript received June 6, 2019; accepted November 26, 2019. This work was supported by the European Commission through the H2020 Project 645097 COMANOID. This article was recommended for publication by Associate Editor K. Mombaur and Editor E. Yoshida 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 Rome, Italy (e-mail: scianca@diag.uniroma1.it; desimone@diag.uniroma1.it; lanari@diag.uniroma1.it; oriolo@diag.uniroma1.it).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material consists of a video containing MATLAB simulations on the linear inverted pendulum showing the effectiveness of intrinsically stable model-predictive control in guaranteeing both stability and feasibility during gait generation. It also includes dynamic simulations on the humanoid robot HRP-4 and experiments on two different humanoid platforms: HRP-4 and NAO. Contact Giuseppe Oriolo (e-mail: oriolo@diag.uniroma1.it) for further questions about this article.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2019.2958483

on the state at the end of the control horizon), based on the fact that the MPC literature highlights the beneficial role of such constraints for closed-loop stability in set-point control problems [16].

In particular, terminal constraints were used for humanoid balancing in [17] and for gait generation in [18]. The latter makes use of an LIP model, requiring its unstable component to stop at the end of the control horizon, a kind of terminal constraint referred to as *capturability constraint* (from the concept of capture point [19]). This constraint has also been used in [20], where it is imposed only at the foot landing instant, and in [21], which addresses locomotion in a multicontact setting.

Another approach focusing on the instability issue relies on the concept of Divergent Component of Motion (DCM), used in [22] to identify an initial condition for stable execution of regular gaits, and in [23] to realize transitions between bipedal and quadrupedal gaits. The DCM concept has also been extended to the 3D context in [24] and [25]. More relevant to our review is [26], which presents an MPC scheme for gait generation that enforces a terminal constraint (actually converted to a terminal cost for the sake of feasibility) on the DCM component.

In this article, we move from the fundamental observation that the control problem addressed in MPC-based gait generation is neither a set-point nor a tracking problem. In fact, since the ZMP control objective is encoded via time-varying state constraints, there is no error to be regulated to (or close to) zero. The only significant stability issue in this context is *internal stability*, i.e., the boundedness of the CoM trajectory with respect to the ZMP trajectory. Therefore, one cannot simply claim that the use of a terminal constraint will automatically entail internal stability. In fact, to the best of our knowledge, no MPC-based gait generation method exists in the literature for which a rigorous analysis of the stability issue has been performed in connection with the use and the choice of a terminal constraint.

Another tightly related aspect to be considered is that terminal constraints may have a detrimental effect on *feasibility*, i.e., the existence of solutions for the optimization problem, which is at the core of any MPC scheme [27]. A particularly desirable property is *recursive feasibility*, which entails that if the optimization problem is feasible at a certain iteration, it will remain such in future iterations. It appears that this also crucial issue has seldom been explored for MPC-based gait generation, with the notable exceptions of [28] and [29].

In [30], we have introduced a novel MPC approach for humanoid gait generation, which relies on the inclusion of an explicit stability constraint in the formulation of the problem. In particular, the idea was to enforce a condition on the future ZMP velocities (representing the control inputs) so as to guarantee that the generated CoM trajectory remains bounded with respect to the ZMP trajectory. Since the control horizon of the MPC algorithm is finite, only part of the future ZMP velocities are decision variables and can, therefore, be subject to a constraint; the remaining part, called *tail*, must be conjectured.

Here, we fully develop our approach into a complete *Intrinsically Stable* MPC (IS-MPC) framework for gait generation. In particular, this article adds the following contributions with respect to [30].

1) We describe a footstep generation module that can be used in conjunction with our MPC scheme in order to modify step timing and length in real time in response to omnidirectional motion commands coming from a higher-level module.
2) Depending on the available preview information on the commanded motion, we discuss several versions of the tail (truncated, periodic, and anticipative) to be used in the stability constraint and show that each of them corresponds to a specific terminal constraint.
3) We analyze in detail the impact of the new constraint on feasibility and show analytically how, under certain assumptions, it is possible to guarantee recursive feasibility of the IS-MPC scheme.
4) We prove that recursive feasibility of IS-MPC implies the desired internal stability of the CoM/ZMP dynamics.
5) We validate our findings by providing dynamic simulations and actual experiments on two different humanoid robots: an HRP-4 and a NAO.

The results on tails, recursive feasibility, and internal stability are the main contributions of this article. We consider them particularly important because they indicate that, contrarily to what is often claimed in the literature, simply adding a terminal constraint (e.g., the capturability constraint) does not *per se* guarantee stability of MPC-based gait generation schemes. Indeed, the appropriate tail to be used in the stability constraint—equivalently, the appropriate terminal constraint—depends upon the future characteristics of the commanded motion. In this sense, to guarantee recursive feasibility, one should always choose the anticipative tail, which makes the most use of the available preview information on such motion. Once recursive feasibility is achieved, CoM/ZMP stability is automatically ensured in IS-MPC.

Another potential benefit of the theoretical analysis of feasibility is that it paves the road for a formal study of the robustness of IS-MPC. Although this is out of the scope of this article, by relying on this analysis, it is possible to devise modifications of the basic scheme, which will preserve recursive feasibility in the presence of quantified bounded uncertainties and/or disturbances.

The rest of this article is organized as follows. In the next section, we formulate the considered gait generation problem and discuss the structure of the proposed approach. Section III describes the algorithm, which generates timing and locations of the candidate footsteps. In Section IV, we introduce the prediction model and the constraints used in the IS-MPC scheme, with the exception of the stability constraint, which is given in a thorough discussion in the dedicated Section V. The IS-MPC algorithm is described in detail in Section VI. Section VII addresses the central issues of stability and feasibility of the proposed method; in particular, a theoretical analysis of the feasibility of the generic IS-MPC iteration is presented and used to obtain sufficient conditions for recursive feasibility, whose role in guaranteeing stability is rigorously established. Simulations on the HRP-4 humanoid are presented in Section VIII, while

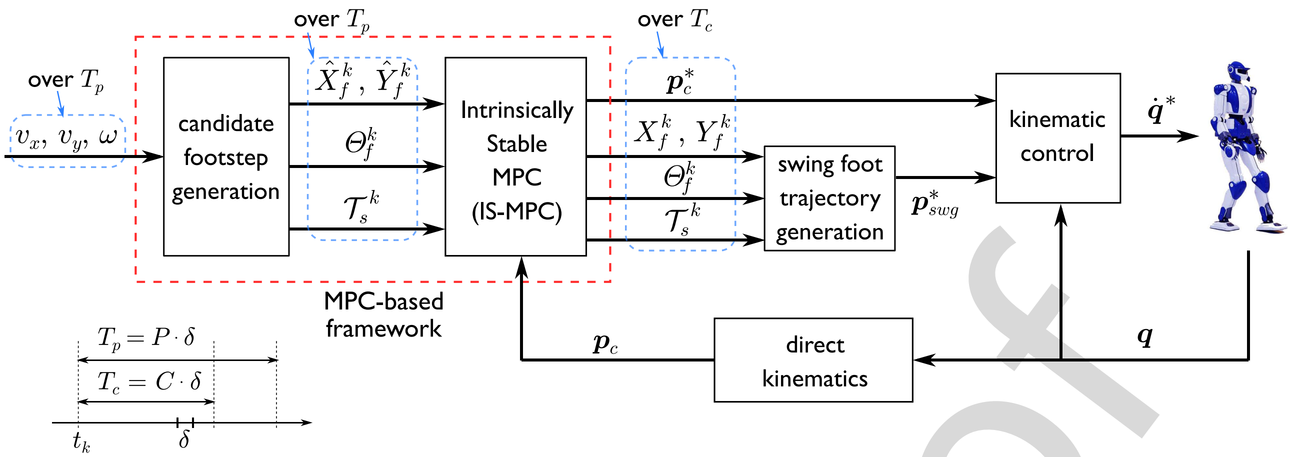


Fig. 1. Block scheme of the proposed MPC-based framework for gait generation.

experimental results on both the NAO and the HRP-4 humanoids are shown in Section IX. Section X concludes this article.

II. Problem and Approach

Consider the problem of generating a walking gait for a humanoid in response to high-level reference velocities, which are given as the driving (v_x, v_y) and steering (ω) velocities of an omnidirectional single-body mobile robot chosen as a template model for motion generation. These velocities, which may encode a persistent trajectory or converge to a stationary point, are produced by an external source; this could be a human operator in a shared control context, or another module of the control architecture working in open loop (planning) or in closed loop (feedback control).

The proposed MPC-based framework, whose block scheme is shown in Fig. 1, works in a digital fashion over sampling intervals of duration δ. Throughout this article, it is assumed that the reference velocities v_x, v_y, and ω are made available for gait generation with a *preview horizon* $T_p = P \cdot \delta$, with P being the number of intervals within the preview horizon. At the generic instant $t_k = k \cdot \delta$, the high-level reference velocities over $[t_k, t_k + T_p]$ are then sent to the footstep generation module, which uses quadratic programming (QP) to generate candidate footsteps over the same interval. In particular, vectors $\hat{X}_f^k$ and $\hat{Y}_f^k$ collect the Cartesian positions of the footsteps, with the "hat" indicating that these are candidates which can be modified by the MPC module, whereas vector Θ_f^k collects the footstep orientations, which will not be modified. The footstep generation module also generates the timing $\mathcal{T}_s^k$ of the sequence.

The output of the footstep generation module is sent to the IS-MPC module, which solves another QP problem to produce in real time the actual footstep positions X_f^k and Y_f^k and the trajectory $\boldsymbol{p}_c^*$ of the humanoid CoM over the *control horizon* $T_c = C \cdot \delta$, with C being the number of intervals within the control horizon. It is assumed that $T_c \leq T_p$, i.e., $C \leq P$. The inclusion of a stability constraint in the formulation guarantees that the CoM trajectory will be bounded, in a sense to be made precise later.

The pose (position and orientation) of the footsteps with the associated timing is used to generate—still in real time—the swing foot trajectory $\boldsymbol{p}_{\text{swg}}^*$ over the control horizon. Together with the CoM trajectory, this is sent to the kinematic control block, which generates velocity inputs at the joint level in order to achieve output tracking (we are assuming that the humanoid robot is velocity- or position- controlled).

In the next sections, we will discuss the proposed control scheme in detail. We will first describe the footstep generation scheme and then turn our attention to the IS-MPC algorithm, which is our core contribution. The kinematic control block can use any standard pseudoinverse-based feedback law and therefore will not be discussed further.

III. Candidate Footstep Generation

The proposed footstep generation module runs synchronously with the IS-MPC scheme and chooses both the timing and the candidate location of the next footsteps in response to the high-level reference velocities. Timing is determined first by a simple rule expressing the fact that a change in the reference velocity should affect both the step duration and length. The candidate footstep locations are then chosen through quadratic optimization.

Note that generating the timing and the orientation of the candidate footsteps outside the IS-MPC is essential to retain the linear structure of the latter. The IS-MPC scheme will still be able to adapt the position of the footsteps to guarantee reactivity to disturbances.

At each sampling instant t_k, the candidate footstep generation module receives in input the high-level reference velocities over the preview horizon, i.e., from t_k to $t_k + T_p = t_{k+P}$ (see Fig. 1). In output, it provides the candidate footstep sequence $(\hat{X}_f^k, \hat{Y}_f^k, \Theta_f^k)$ over the same interval with the associated timing $\mathcal{T}_s^k$. In particular, these quantities are defined[1] as

$$\hat{X}_f^k = (x_f^1 \; \dots \; x_f^F)^T$$

[1] To keep a light notation, the k symbol identifying the current sampling instant is used for the sequence vectors but not for their individual elements.

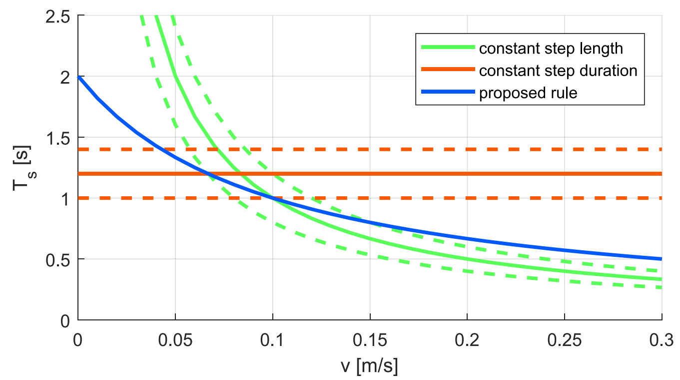


Fig. 2. Proposed rule for determining the step duration T_s as a function of the magnitude v of the reference Cartesian velocity. For comparison, the rules yielding constant step duration and constant step length are also shown.

$$\hat{Y}_f^k = (y_f^1 \ \dots \ y_f^F)^T$$
$$\Theta_f^k = (\theta_f^1 \ \dots \ \theta_f^F)^T$$

and

$$\mathcal{T}_s^k = \{T_s^1, \dots, T_s^F\}$$

where $(x_f^j, y_f^j, \theta_f^j)$ is the pose of the jth footstep in the preview horizon and T_s^j is the duration of the step between the $(j-1)$th and the jth footstep, taken from the start of the single support phase to the next. Since the duration of steps is variable, the number F of footsteps falling within the preview horizon T_p may change at each t_k.

In the following, we first discuss how timing is determined and then describe the procedure for generating the candidate footsteps.

A. Candidate Footstep Timing

In our method, the duration T_s of each step is related to the magnitude $v = (v_x^2 + v_y^2)^{1/2}$ of the reference Cartesian velocity at the beginning of that step.

Assume that a triplet of *cruise parameters* $(\bar{v}, \overline{T}_s, \bar{L}_s)$ has been chosen, where $\bar{v}$ is a central value of v and $\overline{T}_s$ and $\bar{L}_s$ are the corresponding values of the step duration and length, respectively, with $\bar{v} = \bar{L}_s/\overline{T}_s$. The choice of these parameters will depend on the specific kinematic and dynamic capabilities of the humanoid robot under consideration.

The idea is that a deviation from $\bar{v}$ should reflect on a change in *both* T_s and L_s. In formulas, we have

$$v = \bar{v} + \Delta v = \frac{\bar{L}_s + \Delta L_s}{\overline{T}_s - \Delta T_s}$$

with $\Delta L_s = \alpha \Delta T_s$. One easily obtains

$$T_s = \overline{T}_s \, \frac{\alpha + \bar{v}}{\alpha + v}. \tag{1}$$

Figure 2 shows the resulting rule for determining T_s as a function of v in comparison to other possible rules. For illustration, we have set $\bar{v} = 0.15$ m/s, $\overline{T}_s = 0.8$ s, $\bar{L}_s = 0.12$ m, and $\alpha = 0.1$ m/s. It is confirmed that an increase of v, for example, corresponds to both a decrease of T_s and an increase in L_s.

Note that the reference angular velocity ω does not enter into rule (1). The rationale is that the step duration and length along curved and rectilinear paths do not differ significantly if the Cartesian velocity v is the same. For a purely rotational motion ($v = 0$), where the humanoid is only required to rotate on the spot, the above rule would yield the maximum value of T_s.

In practice, equation (1) is iterated along the preview horizon $[t_k, t_k + T_p]$ in order to obtain the footstep timestamps:

$$t_s^j = t_s^{j-1} + \overline{T}_s \, \frac{\alpha + \bar{v}}{\alpha + v(t_s^{j-1})}$$

with t_s^0 equal to the timestamp of the last footstep before t_k. Iterations must be stopped as soon as $t_s^j > t_{k+P}$, discarding the last generated timestamp, since it will be outside the preview horizon. The resulting step timing will be $\mathcal{T}_s^k = \{T_s^1, \dots, T_s^F\}$, with $T_s^j = t_s^{j+1} - t_s^j$.

B. Candidate Footstep Placement

Once the timing of the steps in the preview horizon $[t_k, t_k + T_p]$ has been chosen, the poses of candidate footsteps are generated. To this end, we use a reference trajectory obtained by integrating the following template model under the action of the high-level reference velocities over T_p:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix}. \tag{2}$$

This is an omnidirectional motion model which allows the template robot to move along any Cartesian path with any orientation, so as to perform, e.g., lateral walks, diagonal walks, and so on.

The idea is to distribute the candidate footsteps around the reference trajectory in accordance to the timing $\mathcal{T}_s^k$ while taking into account the kinematic constraints of the robot. These constraints will also be used in the IS-MPC stage, and therefore we will provide their description directly in Section IV-C (see also Fig. 7).

A sequence of two QP problems is solved. The first is

$$\begin{cases} \min\limits_{\Theta_f^k} \ \sum_{j=1}^{F} \left(\theta_f^j - \theta_f^{j-1} - \int_{t_s^{j-1}}^{t_s^j} \omega(\tau) d\tau\right)^2 \\ \text{subject to} \quad |\theta_f^j - \theta_f^{j-1}| \le \theta_{\max}. \end{cases}$$

Here, $\theta_{\max}$ is the maximum allowed rotation between two consecutive footsteps. The second QP problem is

$$\begin{cases} \min\limits_{\hat{X}_f^k, \hat{Y}_f^k} \ \sum_{j=1}^{F} (\hat{x}_f^j - \hat{x}_f^{j-1} - \Delta x^j)^2 + (\hat{y}_f^j - \hat{y}_f^{j-1} - \Delta y^j)^2 \\ \text{subject to kinematic constraints (7).} \end{cases}$$

Here, $(\hat{x}_f^0, \hat{y}_f^0)$ is the known position of the support foot at t_k, and Δx^j and Δy^j are given by

$$\begin{pmatrix} \Delta x^j \\ \Delta y^j \end{pmatrix} = \int_{t_s^{j-1}}^{t_s^j} R_\theta \begin{pmatrix} v_x(\tau) \\ v_y(\tau) \end{pmatrix} d\tau \pm R_j \begin{pmatrix} 0 \\ \ell/2 \end{pmatrix}$$

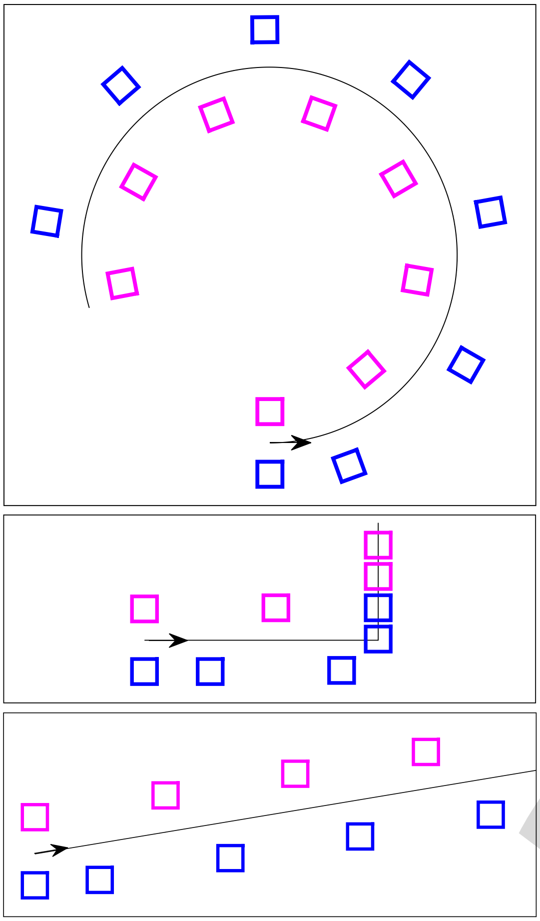


Fig. 3. Candidate footsteps generated by the proposed method for different high-level reference velocities corresponding to a circular walk (top), L-walk (center), and diagonal walk (bottom). The paths in black are obtained by integrating model (2) under the reference velocities. Footsteps in magenta and cyan refer, respectively, to the left and right feet.

where R_θ and R_j are the rotation matrices associated, respectively, with $\theta(\tau)$ (the orientation of the template robot at any given time τ) and the footstep orientation θ_j, and ℓ is the reference coronal distance between consecutive footsteps. The sign of the second term alternates for left/right footsteps.

At the end of this procedure, the candidate footstep sequence $(\hat{X}_f^k, \hat{Y}_f^k, \Theta_f^k)$ with the associated timing $\mathcal{T}_s^k$ is sent to the IS-MPC stage. The final footstep positions (X_f^k, Y_f^k) will be determined by the latter, while the footstep orientations Θ_f^k and timing $\mathcal{T}_s^k$ will not be modified.

Some examples of candidate footsteps generation are shown in Fig. 3. Note that the orientation of the humanoid robot is tangent to the path for the circular walk, but is kept constant ($\omega = 0$) for the other two walks, which represent then proper examples of omnidirectional motion.

IV. IS-MPC: Prediction Model and Constraints

The IS-MPC module uses the LIP as a prediction model. The constraints are of three kinds. The first concerns the position of the ZMP, which must be at all times within the support polygon defined by the footstep sequence and the associated timing. The second type of constraint ensures that the generated steps are compatible with the kinematic capabilities of the robot. The third is the new stability constraint guaranteeing that the CoM trajectory generated by our MPC scheme will be bounded with respect to the ZMP trajectory. The first two constraints must be verified throughout the control horizon, whereas the third is a single scalar condition on each coordinate.

In this section, we discuss in detail the prediction model and the constraints on ZMP and kinematic feasibility. The next section will be devoted to the stability constraint, which deserves a thorough discussion.

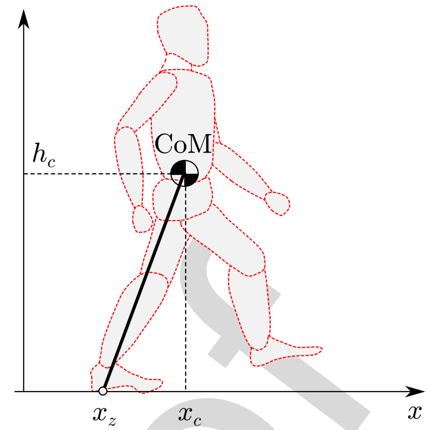


Fig. 4. LIP in the *x* direction.

A. *Prediction Model*

The LIP is a popular choice for describing the motion of the CoM of a biped walking on flat horizontal floor when its height is kept constant and no rotational effects are present. From now on, we express motions in the robot frame, which has its origin at the center of the current support foot, the x-axis *(sagittal)* aligned with the support foot, and the y-axis *(coronal)* orthogonal to the x-axis. In the LIP model, which applies to both point feet and finite-sized feet, the dynamics along the sagittal and coronal axes are governed by decoupled identical linear differential equations.

Consider the motion along the x-axis (see Fig. 4) for illustration, and let x_c and x_z be, respectively, the coordinate of the CoM and the ZMP. The LIP dynamics is

$$\ddot{x}_c = \eta^2 (x_c - x_z) \tag{3}$$

where $\eta = \sqrt{g/h_c}$, with g the gravity acceleration and h_c the constant height of the CoM. In this model, the ZMP position x_z represents the input, whereas the CoM position x_c is the output.

To obtain smoother trajectories, we take the ZMP velocity $\dot{x}_z$ as the actual control input. This leads to the following third-order prediction model (LIP + dynamic extension):

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \eta^2 & 0 & -\eta^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_z. \tag{4}$$

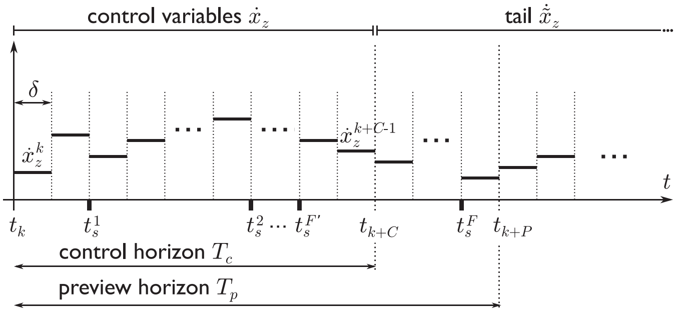


Fig. 5. At time t_k, the control variables determined by IS-MPC are the piecewise-constant ZMP velocities over the control horizon. The ZMP velocities after the control horizon are instead conjectured in order to build the tail (see Section V-B). Also shown are the F footstep timestamps placed by the footstep generation module in the preview horizon; F' of them fall in the control horizon.

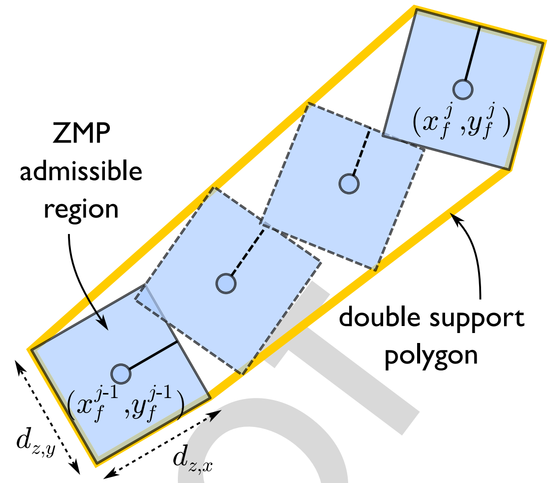


Fig. 6. ZMP moving constraint in double support.

Our MPC scheme uses piecewise-constant control over the sampling intervals (see Fig. 5)

$$\dot{x}_z(t) = \dot{x}_z^i, \quad t \in [t_i, t_{i+1}).$$

In particular, a bound of the form $|\dot{x}_z^i| \leq \gamma$, with γ a positive constant, will be satisfied for all i. In fact, the reference velocities v_x, v_y, and ω will be bounded in any realistic gait generation problem. As shown in Fig. 2, the footstep generation module will then produce a sequence of footstep along which the step duration is bounded below. This timing will be reflected in the associated ZMP constraints (see Section IV-B), which will, in turn, entail as solution a piecewise-continuous trajectory $x_z(t)$ with bounded derivative. Therefore, for $t \in [t_i, t_{i+1}]$ it will be

$$x_z(t) = x_z^i + (t - t_i)\,\dot{x}_z^i, \quad \text{with } |\dot{x}_z^i| \leq \gamma \tag{5}$$

where we have used the notation $x_z^i = x_z(t_i)$.

The generic iteration of IS-MPC plans over the control horizon, i.e., from t_k to $t_k + T_c = t_{k+C}$. Since $T_c \leq T_p$, a subset of the F candidate footsteps produced by the footstep generation module fall their inside the control horizon; denote their number by $F' < F$. The MPC iteration will then generate:

1) the control variables, i.e., the input values $\dot{x}_z^{k+i}$, $\dot{y}_z^{k+i}$, for $i = 0, \ldots, C-1$;
2) the other decision variables, i.e., the actual footstep positions (x_f^j, x_f^j), for $j = 1, \ldots, F'$;
3) as a byproduct, the output history $x_c(t)$, $y_c(t)$, for $t \in [t_k, t_{k+C}]$, which will be ultimately used to drive the actual humanoid.

As already mentioned, the orientations of the footsteps are instead inherited from the generated sequence (more on this in Section IV-B).

Note that the footsteps do not appear in the prediction model, but will show up in the constraints, as discussed in the rest of this section.

B. ZMP Constraints

The first constraint guarantees dynamic balance by imposing that the ZMP lies inside the current support polygon at all time instants within the control horizon.

When the robot is in single support on the jth footstep, the admissible region for the ZMP is the interior of the footstep, which can be approximated as a rectangle of dimensions $d_{z,x}$ and $d_{z,y}$, centered at (x_f^j, y_f^j), and oriented as θ^j. Using the fact that the ZMP profile is piecewise-linear, as entailed by (5), the constraint can be expressed as[2]

$$R_j^T \begin{pmatrix} \delta \sum_{l=0}^{i} \dot{x}_z^{k+l} - x_f^j \\ \delta \sum_{l=0}^{i} \dot{y}_z^{k+l} - y_f^j \end{pmatrix} \leq \frac{1}{2}\begin{pmatrix} d_{z,x} \\ d_{z,y} \end{pmatrix} - R_j^T \begin{pmatrix} x_z^k \\ y_z^k \end{pmatrix}. \tag{6}$$

If the above sampled-time ZMP constraint is satisfied, then the original continuous-time constraint is also satisfied thanks to the linearity of $x_z(t)$ within each sampling interval. Constraint (6), complete with the corresponding left-hand side, must be imposed throughout the control horizon ($i = 0, \ldots, C-1$) and for all the associated footsteps ($j = 0, \ldots, F'$).

Note that constraint (6) is nonlinear in the footstep orientation θ^j, which however is not a decision variable, being simply inherited from the footstep generation module. The constraint is instead linear in x_f^j and y_f^j, as well as in the ZMP velocity inputs.

During double support, the support polygon would be the convex hull of the two footsteps, whose boundary is a nonlinear function of their relative position. To preserve linearity, we adopt an approach based on *moving constraints* [31]. In particular, the admissible region for the ZMP in double support has exactly the same shape and dimensions it has in single support, and it roto-translates (i.e., simultaneously rotates and translates) from one footstep to the other in such a way to always remain in the support polygon (see Fig. 6). This results in a slightly conservative constraint, which is however linear in the decision variables.

[2]For compactness, we shall only write the right-hand side of bilateral inequality constraints. For example, constraint (6) should be completed by a left-hand side obtained by adding (rather than subtracting) the two terms that appear in the right-hand side.

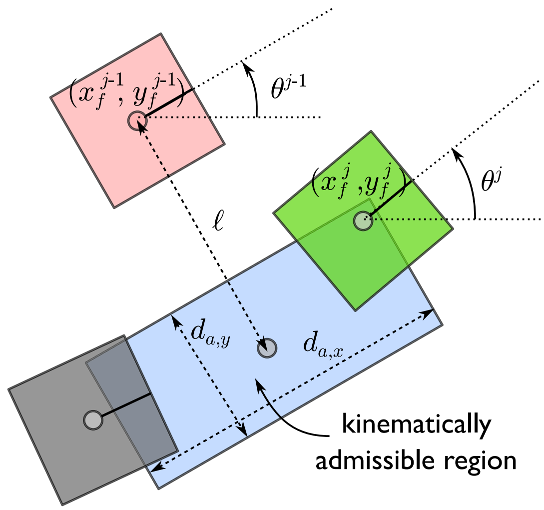


Fig. 7. Kinematic constraint on footstep placement.

C. Kinematic Constraints

The second type of constraint is introduced to ensure that all steps are compatible with the robot kinematic limits. Consider the jth step in T_c, with the support foot centered at (x_f^{j-1}, y_f^{j-1}) and oriented as θ^{j-1}. The admissible region for placing the footstep is defined as a rectangle having the same orientation θ^{j-1} and whose center is displaced from the support foot center by a distance ℓ in the coronal direction (see Fig. 7). Denoting by $d_{a,x}$ and $d_{a,y}$ the dimensions of the kinematically admissible region, the constraint can be written as

$$R_{j-1}^T \begin{pmatrix} x_f^j - x_f^{j-1} \\ y_f^j - y_f^{j-1} \end{pmatrix} \leq \pm \begin{pmatrix} 0 \\ \ell \end{pmatrix} + \frac{1}{2} \begin{pmatrix} d_{a,x} \\ d_{a,y} \end{pmatrix} \tag{7}$$

with the sign alternating for the two feet. The above constraint, complete with the corresponding left-hand side, must be imposed for all footsteps in the control horizon ($j = 1, \dots, F'$).

V. IS-MPC: Enforcing Stability

The LIP dynamics (3) is inherently unstable. As a consequence, even when the ZMP lies at all times within the support polygon *(gait balance)*, it may still happen that the CoM diverges exponentially with respect to the ZMP; in this case, the gait would obviously become unfeasible in practice, due to the kinematic limitations of the robot. The role of the stability constraint is then to guarantee that the CoM trajectory remains bounded with respect to the ZMP *(internal stability)*.

In this section, we first describe the structure of the stability constraint and then discuss the possible *tails* for its implementation.

A. Stability Constraint

Since we want to enforce boundedness of the CoM w.r.t. the ZMP, we can ignore the dynamic extension and focus directly on the LIP system.

By using the following change of coordinates:

$$x_s = x_c - \dot{x}_c/\eta \tag{8}$$

$$x_u = x_c + \dot{x}_c/\eta \tag{9}$$

the LIP part of system (3) is decomposed into a stable and an unstable subsystem

$$\dot{x}_s = -\eta\,(x_s - x_z) \tag{10}$$

$$\dot{x}_u = \eta\,(x_u - x_z). \tag{11}$$

The unstable component x_u is also known as *Divergent Component of Motion* (DCM) [22] or *capture point* [32].

In spite of the LIP instability, for any input ZMP trajectory $x_z(t)$ of the form (5) there exists a special initialization of x_u such that the resulting output CoM trajectory is bounded with respect to the input [33]. In particular, this is the (only) initial condition on x_u for which the free evolution of (11) exactly cancels the component of the forced evolution that would diverge with respect to $x_z(t)$. In the MPC context, where the initial condition at t_k is denoted by $x_u(t_k) = x_u^k$, the special initialization is expressed as

$$x_u^k = \eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} x_z(\tau) d\tau. \tag{12}$$

Note that this particular initialization depends on the future values of the LIP input, i.e., the ZMP coordinate x_z. In the following, we refer to (12) as the *stability condition*.

The stability condition, which involves x_u at the initial instant t_k of the control horizon, can be propagated to its final instant t_{k+C} by integrating (11) from x_u^k in (12):

$$x_u^{k+C} = \eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_{k+C})} x_z(\tau) d\tau. \tag{13}$$

Condition (12)—or equivalently, (13)—can be used to set up the corresponding constraint for the MPC problem. To this end, we use the piecewise-linear profile (5) of x_z to obtain explicit forms.

Proposition 1: For the piecewise-linear x_z in (5), condition (12) becomes

$$x_u^k = x_z^k + \frac{1 - e^{-\eta\delta}}{\eta} \sum_{i=0}^{\infty} e^{-i\eta\delta} \dot{x}_z^{k+i} \tag{14}$$

while (13) takes the form

$$x_u^{k+C} = x_z^{k+C} + \frac{1 - e^{-\eta\delta}}{\eta} e^{C\eta\delta} \sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{x}_z^{k+i}. \tag{15}$$

Proof: Rewrite (5) as

$$x_z(t) = x_z^k + \sum_{i=0}^{\infty} (\rho(t - t_{k+i}) - \rho(t - t_{k+i+1}))\dot{x}_z^{k+i} \tag{16}$$

where $\rho(t) = t\,\delta_{-1}(t)$ denotes the unit ramp and $\delta_{-1}(t)$ the unit step. Using Properties 1, 4, and 3 given in the Appendix, we get

$$\int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} (\rho(\tau - t_{k+i}) - \rho(\tau - t_{k+i+1})) d\tau$$

$$= \frac{1 - e^{-\eta\delta}}{\eta^2} e^{-i\eta\delta}.$$

Plugging this expression in condition (12) and using Property 2 of the Appendix, one obtains (14).

To prove (15), rewrite (16) as

$$x_z(t) = x_z^k + \sum_{i=0}^{C-1} (\rho(t - t_{k+i}) - \rho(t - t_{k+i+1}))\dot{x}_z^{k+i} + \sum_{i=C}^{\infty} (\rho(t - t_{k+i}) - \rho(t - t_{k+i+1}))\dot{x}_z^{k+i}.$$

The contribution of the first two terms of x_z to the integral in (13) is x_z^{k+C}. Using Properties 1, 3, and 4, one verifies that the contribution of the third term is exactly the second term on the right-hand side of (15). This completes the proof. ■

In (14), one should logically separate the values of $\dot{x}_z^i$ within the control horizon, i.e., the control variables $\dot{x}_z^i$ for $i = k, \ldots, k + C - 1$, from the remaining values, i.e., from $k + C$ on. The infinite summation is then split into two parts, and (14) can be rearranged as[3]

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = -\sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{x}_z^{k+i} + \frac{\eta}{1 - e^{-\eta\delta}} (x_u^k - x_z^k). \tag{17}$$

Observe the inversion between (14), which expresses the stable initialization at t_k for a given $x_z(t)$, and (17), which constrains the control variables so that the associated stable initialization matches the current state at t_k. In the following, we will refer to (17) as the *stability constraint*.

The control variables do not appear in condition (15), which involves only the value of the state variable x_u^{k+C} at the end of the control horizon. In other terms, this condition represents what is called a *terminal constraint* in the MPC literature.

Both the stability and the terminal constraint contain an infinite summation, which depends on $\dot{x}_z^{k+C}, \dot{x}_z^{k+C+1}, \ldots$, i.e., the ZMP velocities *after* the control horizon. These are obviously unknown, because they will be determined by future iterations of the MPC algorithm; as a consequence, including either of the constraints in the MPC formulation would lead to a noncausal (unrealizable) controller. However, by exploiting the preview information on v_x, v_y, and ω, we can make an *informed conjecture* at t_k about these ZMP velocities, which we will denote by $\dot{\tilde{x}}_z^{k+C}$, $\dot{\tilde{x}}_z^{k+C+1}, \ldots$ and refer to collectively as the *tail* in the following. Correspondingly, the stability constraint (17) assumes the form

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = -\sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+i} + \frac{\eta}{1 - e^{-\eta\delta}} (x_u^k - x_z^k) \tag{18}$$

while the terminal constraint (15) becomes

$$x_u^{k+C} = x_z^{k+C} + \frac{1 - e^{-\eta\delta}}{\eta} e^{C\eta\delta} \sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+i}. \tag{19}$$

[3]Constraint (17) can be written as a function of the actual state variables of our prediction model (x_c, $\dot{x}_c$, and x_z) using the coordinate transformation (9). The same is true for all subsequent forms of the stability constraint as well as of the terminal constraint.

Using either of these in the MPC formulation will lead to a causal (realizable) controller.

B. Tails

We now discuss three possible options for the structure of the tail depending on the assumed behavior of the ZMP velocities after the control horizon. Basically, they correspond to: 1) neglecting them; 2) assuming they are periodic; and 3) anticipating a more general profile based on preview information. For each option, we shall explicitly compute the corresponding form of both the stability and the terminal constraint.

1) Truncated Tail: The simplest option is to *truncate* the tail, by assuming that the corresponding ZMP velocities are all zero. This is a sensible choice if the preview information indicates that the robot is expected to stop at the end of the control horizon.

Proposition 2: Let *(truncated tail)*

$$\dot{\tilde{x}}_z^{k+i} = 0 \quad \text{for} \quad i \geq C.$$

The stability constraint becomes

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = \frac{\eta}{1 - e^{-\eta\delta}} (x_u^k - x_z^k) \tag{20}$$

while the terminal constraint becomes

$$x_u^{k+C} = x_z^{k+C}. \tag{21}$$

Proof: The above expressions are readily derived from the general constraints (18) and (19), respectively. ■

Interestingly, the terminal constraint (21) is equivalent to the *capturability constraint*, originally introduced in [18].

2) Periodic Tail: The second option is to use a *periodic tail* obtained by infinite replication of the ZMP velocities within the control horizon. This assumption is justified when the reference velocities are themselves periodic (in particular, constant) in T_c, which is typically chosen as the gait period (total duration of two consecutive steps) or a multiple of it. Formulas for a replication period different from the control horizon may be easily derived.

Proposition 3: Let *(periodic tail)*

$$\begin{aligned} \dot{\tilde{x}}_z^{k+i} &= \dot{x}_z^{k+i-C}, \quad &\text{for} \quad i = C, \ldots, 2C - 1 \\ \dot{\tilde{x}}_z^{k+i} &= \dot{\tilde{x}}_z^{k+i-C}, \quad &\text{for} \quad i \geq 2C. \end{aligned}$$

The stability constraint becomes

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = \eta \frac{1 - e^{-C\eta\delta}}{1 - e^{-\eta\delta}} (x_u^k - x_z^k) \tag{22}$$

while the terminal constraint becomes

$$x_u^{k+C} - x_z^{k+C} = x_u^k - x_z^k. \tag{23}$$

Proof: If the tail is periodic, the infinite summation in (18) can be rewritten as follows:

$$\begin{aligned} \sum_{i=C}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+i} &= e^{-C\eta\delta} \sum_{i=0}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+C+i} \\ &= e^{-C\eta\delta} \sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} \left(1 + e^{-C\eta\delta} + \cdots\right) \end{aligned}$$

$$= \frac{e^{-C\eta\delta}}{1 - e^{-C\eta\delta}} \sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i}$$

which can be plugged in (18) and (19), respectively, to obtain (22) and (23). ■

Note that, using (11), the terminal constraint (23) can be rewritten as

$$\dot{x}_u^{k+C} = \dot{x}_u^k.$$

3) Anticipative Tail: In the general case, one can use the candidate footsteps produced by the footstep generation module *beyond* the control horizon to conjecture a tail in $[T_c, T_p]$. This is done in two phases: in the first, we generate in $[T_c, T_p]$ a ZMP trajectory which belongs at all times to the admissible ZMP region defined by the footsteps $\{(\hat{x}_f^{F'}, \hat{y}_f^{F'}, \theta_f^{F'}), \ldots, (\hat{x}_f^F, \hat{y}_f^F, \theta_f^F)\}$. In the second phase, we sample the time derivative of this ZMP trajectory every δ seconds.

Denote the samples obtained by the above procedure by $\dot{x}_{z,\mathrm{ant}}^{k+i}$, for $i = C, \ldots, P-1$. The *anticipative tail* is then obtained by:

1) setting $\dot{\tilde{x}}_z^{k+i} = \dot{x}_{z,\mathrm{ant}}^{k+i}$ for $i = C, \ldots, P-1$;
2) using a truncated or periodic expression for the residual part of the tail located *after* the preview horizon, i.e., for $\dot{\tilde{x}}_z^{k+i}$, $i = P, P+1, \ldots$.

The stability constraint (18) then becomes

$$\sum_{i=0}^{C-1} e^{-i\eta\delta} \dot{x}_z^{k+i} = -\sum_{i=C}^{P-1} e^{-i\eta\delta} \dot{x}_{z,\mathrm{ant}}^{k+i} - \sum_{i=P}^{\infty} e^{-i\eta\delta} \dot{\tilde{x}}_z^{k+i} + \frac{\eta}{1 - e^{-\eta\delta}} (x_u^k - x_z^k).$$

Once a form is chosen for the residual part of the tail, this formula leads to a closed-form expression of the stability constraint which consists of a finite number of terms, and is, therefore, still amenable to real-time implementation. Similarly, one can use (19) to derive the corresponding expression of the terminal constraint.

In the following, and specifically in the feasibility analysis of Section VII-B2, we will use a particular form of anticipative tail such that 1) the ZMP trajectory in $[T_c, T_p]$ is always at the center of the ZMP admissible region, and 2) the residual part of the tail is truncated.

VI. IS-MPC: Algorithm

Each iteration of our IS-MPC algorithm solves a QP problem based on the prediction model and constraints described in Section IV, with the addition of the stability constraint discussed in the previous section.

A. Formulation of the QP Problem

Collect in vectors

$$\dot{X}_z^k = (\dot{x}_z^k \ \ldots \ \dot{x}_z^{k+C-1})^T$$
$$\dot{Y}_z^k = (\dot{y}_z^k \ \ldots \ \dot{y}_z^{k+C-1})^T$$
$$X_f^k = (x_f^1 \ \ldots \ x_f^{F'})^T$$
$$Y_f^k = (y_f^1 \ \ldots \ y_f^{F'})^T$$

all the MPC decision variables.

At this point, the QP problem can be formulated as

$$\begin{cases} \min\limits_{\substack{\dot{X}_z^k, \dot{Y}_z^k \\ X_f^k, Y_f^k}} \|\dot{X}_z^k\|^2 + \|\dot{Y}_z^k\|^2 + \beta\left(\|X_f - \hat{X}_f\|^2 + \|Y_f - \hat{Y}_f\|^2\right) \\ \qquad \text{subject to} \\ \bullet \text{ ZMP constraints (6)} \\ \bullet \text{ kinematic constraints (7)} \\ \bullet \text{ stability constraints (18) for } x \text{ and } y \end{cases}.$$

Note the following points.

1) While the ZMP and kinematic constraints involve simultaneously the x and y coordinates, the stability constraints must be enforced separately along the sagittal and coronal axes.
2) The actual expression of the stability constraint will depend on the chosen tail (truncated, periodic, anticipative).
3) The same expression of the stability constraint is obtained by imposing the corresponding terminal constraint for x and y.
4) The CoM coordinate x_c only appears through x_u in the stability (or terminal) constraints.

B. Generic Iteration

We now provide a sketch of the generic iteration of the IS-MPC algorithm. The input data are the sequence $(\hat{X}_f^k, \hat{Y}_f^k, \Theta_f^k)$ of candidate footsteps, with the associated timing $\mathcal{T}_s^k$, as well as the high-level reference velocities used for footstep generation (these are used explicitly in the MPC if the anticipative tail is chosen). As initialization, one needs x_c, $\dot{x}_c$, and x_z at the current sampling instant t_k. Depending on the available sensors, one may either use measured data (typically true for the CoM variables) or the current model prediction (often for the ZMP position).

The IS-MPC iteration at t_k goes as follows.

1) Solve the QP problem to obtain $\dot{X}_z^k, \dot{Y}_z^k, X_f^k,$ and Y_f^k.
2) From the solutions, extract $\dot{x}_z^k$, $\dot{y}_z^k$, the first control samples.
3) Set $\dot{x}_z = \dot{x}_z^k$ in (4) and integrate from $(x_c^k, \dot{x}_c^k, x_z^k)$ to obtain $x_c(t)$, $\dot{x}_c(t)$, and $x_z(t)$ for $t \in [t_k, t_{k+1}]$. Compute $y_c(t)$, $\dot{y}_c(t)$, and $y_z(t)$ similarly.
4) Define the 3D trajectory of the CoM as $\boldsymbol{p}_c^* = (x_c, y_c, h_c)$ in $[t_k, t_{k+1}]$ and return it.
5) Return also the actual footstep sequence $(X_f^k, Y_f^k, \Theta_f^k)$ with the (unmodified) timing $\mathcal{T}_s^k$.

We recall that the footstep sequence is used by the swing foot trajectory generation module for computing $\boldsymbol{p}_{\mathrm{swg}}^*$ in $[t_k, t_{k+1}]$ (actually, only the first footstep is needed for this computation). This is then sent to the kinematic controller together with $\boldsymbol{p}_c^*$ (see Fig. 1).

VII. IS-MPC: Feasibility and Stability

In this section, we address the crucial issues of feasibility and stability of the proposed IS-MPC controller in itself, i.e., independently from the footstep generation module. We start by reporting some simulations that show how the introduction of the stability constraint is beneficial in guaranteeing that the CoM trajectory is always bounded with respect to the ZMP trajectory. A theoretical analysis of the feasibility of the generic IS-MPC iteration is then presented and used to obtain explicit conditions for recursive feasibility; simulations are used again to confirm that the choice of an appropriate tail is essential for achieving such a property. Finally, we formally prove that internal stability of the CoM/ZMP dynamics is ensured, provided that IS-MPC is recursively feasible.

A. Effect of the Stability Constraint

We present here some MATLAB simulation results of IS-MPC for the dynamically extended LIP model, in which we have set $h_c = 0.78$ m (an appropriate value for the HRP-4 humanoid robot, see Section VIII). A sequence of evenly spaced footsteps is given with a constant step duration $T_s = 0.5$ s, split in $T_{ss} = 0.4$ s (single support) and $T_{ds} = 0.1$ s (double support). The dimensions of the ZMP admissible regions are $d_{z,x} = d_{z,y} = 0.04$ m, and the sampling time is $\delta = 0.01$ s. For simplicity, the footstep sequence given to the MPC is not modifiable (this corresponds to β going to infinity in the QP cost function of Section VI-A); correspondingly, the kinematic constraints (7) are not enforced. The QP problem is solved with the `quadprog` function, which uses an interior-point algorithm.

We compare the performance of the proposed IS-MPC scheme with a standard MPC. In IS-MPC, we have used (22) as the stability constraint, which corresponds to choosing a periodic tail. In the standard MPC, the stability constraint is removed, and the ZMP velocity norms in the cost function are replaced with the CoM jerk norms in order to bring the CoM into play. This corresponds to entrusting the boundedness of the CoM trajectory entirely to the cost function, in the hope that minimization of the CoM jerk will penalize diverging behaviors, as done in early MPC approaches for gait generation.

Figure 8 shows the performance of IS-MPC and standard MPC for $T_c = 1.5$ s, i.e., 1.5 times the gait period. Both gaits are stable, with the IS-MPC gait more aggressively using the ZMP constraints in view of its cost function that penalizes ZMP variations.

Figure 9 compares the two schemes when the control horizon is reduced to $T_c = 1$ s. The standard MPC loses stability: the resulting ZMP trajectory is always feasible, but the associated CoM trajectory diverges[4] with respect to it, because the control horizon is too short to allow sorting out the stable behavior via jerk minimization. With IS-MPC, instead, boundedness of the CoM trajectory with respect to the ZMP trajectory is preserved in spite of the shorter control horizon, thanks to the embedded stability constraint. The accompanying video shows an animation of the evolutions in Figs. 8 and 9.

[4] In particular, in this case the divergence occurs on the coronal coordinate y_c. However, it is also possible to find situations where divergence occurs on the sagittal coordinate x_c, or even on both coordinates.

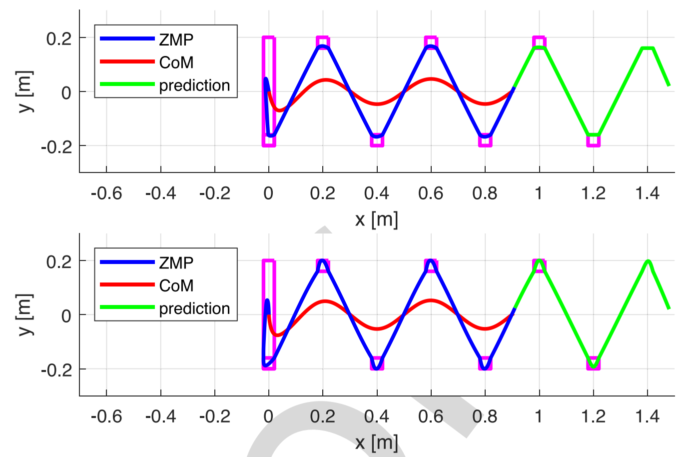


Fig. 8. Simulation 1: Gaits generated by IS-MPC (top) and standard MPC (bottom) for $T_c = 1.5$ s. The given footstep sequence is shown in magenta. Note the larger region corresponding to the initial double support.

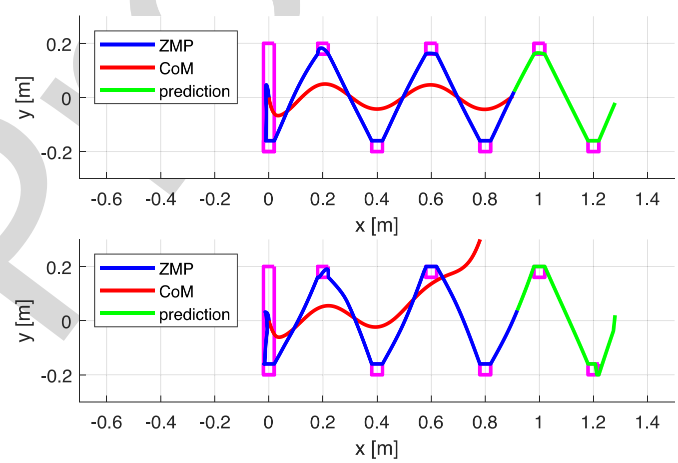


Fig. 9. Simulation 2: Gaits generated by IS-MPC (top) and standard MPC (bottom) for $T_c = 1.0$ s. Note the instability in the standard MPC solution.

Another interesting situation is that of Fig. 10, in which the CoM height is increased to $h_c = 1.6$ m while keeping the “long” control horizon $T_c = 1.5$ s of Simulation 1. Once again, standard MPC is unstable, while IS-MPC guarantees boundedness of the CoM with respect to the ZMP. Since it is $\eta^2 = g/h_c$, a similar situation can be met when g is decreased, as in gait generation for low-gravity environments (e.g., the moon).

We emphasize that the onset of instability in standard MPC cannot be avoided by adding to the cost function a term for keeping the ZMP close to the foot center. The result of this common expedient is shown in Fig. 11, in which the divergence occurs even earlier than in Fig. 9, because the additional cost term has actually the effect of depenalizing the norm of the CoM jerk. Instead, IS-MPC remains stable also with this modified cost function, with the ZMP pushed well inside the constraint region.

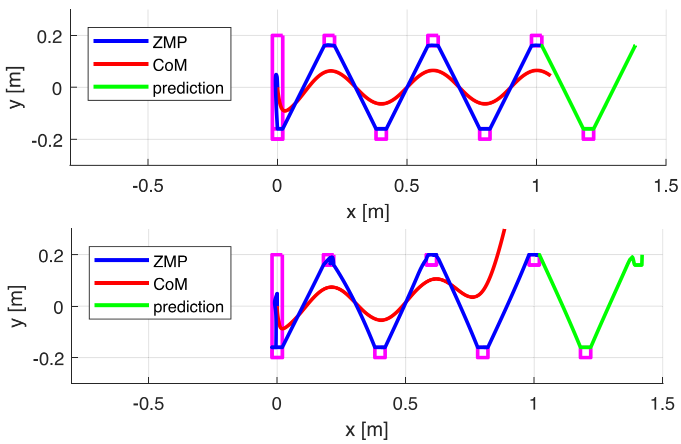


Fig. 10. Simulation 2 bis: Gaits generated by IS-MPC (top) and standard MPC (bottom) for $T_c = 1.5$ s and a higher CoM. Note the instability in the standard MPC solution.

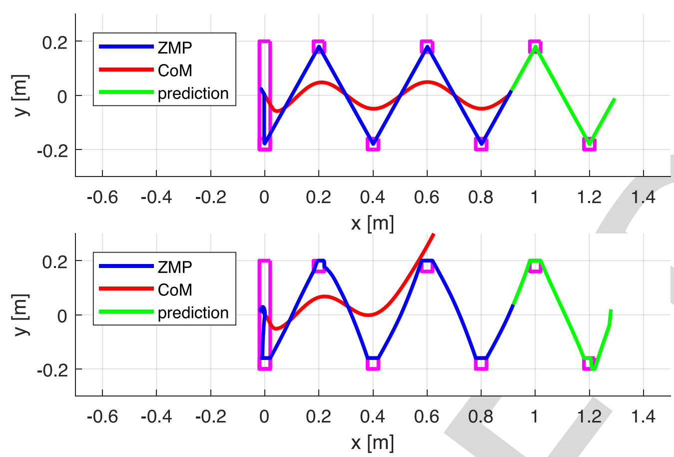


Fig. 11. Simulation 2 ter: Gaits generated by IS-MPC (top) and standard MPC (bottom) for $T_c = 1.0$ s, adding in the cost function a term for keeping the ZMP close to the foot center. The standard MPC solution is still unstable.

B. Feasibility Analysis

The introduction of the stability constraint (or the corresponding terminal constraint), although beneficial in guaranteeing boundedness of the CoM trajectory, has the effect of reducing the *feasibility region*, i.e., the subset of the state space for which the QP problem of Section VI-A admits a solution. In some situations, this might even lead to a loss of feasibility, i.e., the system may find itself in a state where it is impossible to find a solution satisfying all the constraints.

In the following, we show how to determine the feasibility region at a given time. Then, we address *recursive feasibility*: this property holds if, starting from a feasible state, the MPC scheme always brings the system to a state which is still feasible. In particular, we will prove that one can achieve recursive feasibility by using the preview information conveyed by the sequence of candidate footsteps.

1) Feasibility Regions: To focus on the feasibility issue, consider the case of given footsteps ($\beta \to \infty$ in the QP cost function) with fixed orientation. Thanks to the latter assumption, and to the use of a moving ZMP constraint in double support (see Fig. 6), the QP problem separates in two decoupled problems: one for the x and one for the y ZMP coordinate. Let us focus on the x coordinate henceforth, with the understanding that every development is also valid for the y coordinate. The general coupled case can be treated by using an appropriate coordinate change.

Consider the kth step of the IS-MPC algorithm. The QP problem is feasible at t_k if there exists a ZMP trajectory $x_z(t)$ that satisfies both the ZMP constraint for $t \in [t_k, t_{k+C}]$

$$x_z^m(t) \le x_z(t) \le x_z^M(t) \tag{24}$$

and the stability constraint

$$\eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} x_z(\tau) d\tau = x_u^k - \eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_k)} \tilde{x}_z(\tau) d\tau \tag{25}$$

where:

- $x_z^m(t)$ and $x_z^M(t)$ are, respectively, the lower and upper bounds of the ZMP admissible region at time t, as derived from (6);
- $\tilde{x}_z$ is the ZMP position[5] corresponding (through integration) to the chosen velocity tail;
- both the ZMP and the stability constraint have been expressed in continuous time for later convenience (in particular, (25) is obtained from (12) by splitting the integral in two and plugging the tail in the second integral);
- the kinematic constraints (7) are not enforced since footsteps are given.

Proposition 4: At time t_k, IS-MPC is feasible if and only if

$$x_u^{k,m} \le x_u^k \le x_u^{k,M} \tag{26}$$

where

$$x_u^{k,m} = \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} x_z^m d\tau + \eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_k)} \tilde{x}_z d\tau$$

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

Proof: To show the necessity of (26), multiply each side of the ZMP constraint (24) by $e^{-\eta(t-t_k)}$ and integrate over time from t_k to t_{k+C}. Adding to all sides the integral term in the right-hand side of (25), the middle side becomes exactly x_u^k, while the left- and right-hand sides become $x_u^{k,m}$ and $x_u^{k,M}$, as defined in the thesis.

The sufficiency can be proven by showing that if (26) holds, then the ZMP trajectory

$$x_z(t) = x_z^M(t) - \frac{x_u^{k,M} - x_u^k}{1 - e^{-\eta T_c}}$$

satisfies both the ZMP constraint (24) and the stability constraint (25). ■

The interpretation of (26) is the following: it is the admissible range for x_u at time t_k to guarantee solvability of the QP problem

[5] In the rest of this section, for simplicity, we will use the term "tail" for both the ZMP velocity and the corresponding position.

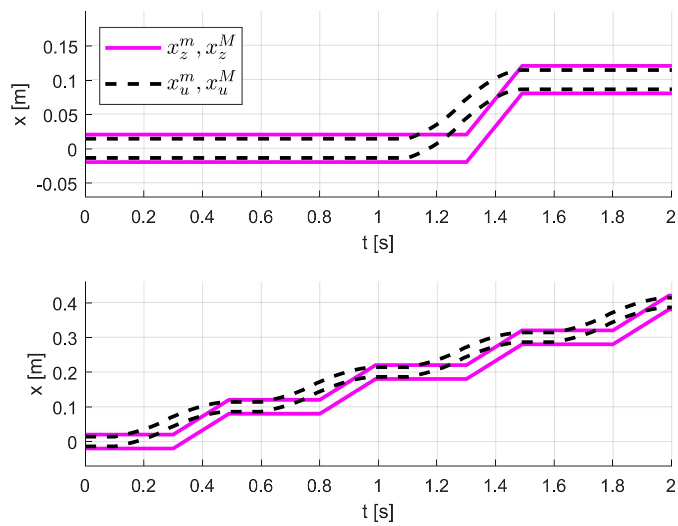


Fig. 12. Feasibility regions. (Top) The robot is taking a single step. (Bottom) The robot is taking a sequence of steps. The anticipative tail is used in both cases.

associated with the current iteration of IS-MPC. Since x_u is related to the state variables of the prediction model through (9), equation (26) actually identifies the feasibility region in state space.

Note that

$$x_u^{k,M} - x_u^{k,m} = \eta \int_{t_k}^{t_{k+C}} e^{-\eta(\tau - t_k)} (x_z^M - x_z^m) d\tau$$
$$= d_{z,x}(1 - e^{-\eta T_c}) \tag{27}$$

where we have used the fact that $x_z^M(t) - x_z^m(t) = d_{z,x}$ for all t, as implied by (6). This shows that the extension $x_u^{k,M} - x_u^{k,m}$ of the admissible range for x_u depends on the dimension $d_{z,x}$ of the ZMP admissible region and tends to become exactly $d_{z,x}$ as the control horizon T_c is increased. On the other hand, the midpoint of this range depends on the tail chosen for the stability constraint (25), because $\eta \int_{t_{k+C}}^{\infty} e^{-\eta(\tau - t_k)} \tilde{x}_z d\tau$ acts as an offset in both the left- and right-hand sides of (26).

Figure 12 illustrates how the admissible range for x_u moves over time, for the case of a single step and of a sequence of steps. These results were obtained with $h_c = 0.78$ m, $d_{z,x} = 0.04$ m, and $T_c = 0.5$ s. In both cases, an anticipative tail was used, with the residual part truncated; the preview horizon is $T_p = 1$ s. Note that, as expected, the extension of the range is constant and smaller than $d_{z,x}$, and that the range itself gradually shifts toward the next ZMP admissible region as a step is approached.

2) Recursive Feasibility: Next, we prove that the use of an anticipative tail provides recursive feasibility under a (sufficient) condition on the preview horizon T_p.

Proposition 5: Assume that the anticipative tail is used in the stability constraint (25). Then, IS-MPC is recursively feasible if the preview horizon T_p is sufficiently large.

Proof: To establish recursive feasibility, we must show that if the IS-MPC QP problem is feasible at t_k, it will be still feasible at time t_{k+1}.

Let us assume that (26) holds. This implies that the ZMP constraint (24) holds for $t \in [t_k, t_{k+C}]$, and that the stability constraint (25) is satisfied, i.e.,

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

with $\tilde{x}_z$ chosen as the anticipative tail at t_k.

Using (11), the value of x_u at t_{k+1} is written as

$$x_u^{k+1} = e^{\eta\delta} x_u^k - \eta \int_{t_k}^{t_{k+1}} e^{\eta(t_{k+1} - \tau)} x_z(\tau) d\tau.$$

Plugging the above expression for x_u^k in this equation, simplifying, and considering that $x_z(t) \le x_z^M(t)$ for $t \in [t_k, t_{k+C}]$, we obtain

$$x_u^{k+1} \le \eta \int_{t_{k+1}}^{t_{k+C}} e^{\eta(t_{k+1} - \tau)} x_z^M(\tau) d\tau$$
$$+ \eta \int_{t_{k+C}}^{\infty} e^{\eta(t_{k+1} - \tau)} \tilde{x}_z(\tau) d\tau.$$

According to Proposition 4, feasibility at t_{k+1} requires[6]

$$x_u^{k+1} \le \eta \int_{t_{k+1}}^{t_{k+C+1}} e^{\eta(t_{k+1} - \tau)} x_z^M(\tau) d\tau$$
$$+ \eta \int_{t_{k+C+1}}^{\infty} e^{\eta(t_{k+1} - \tau)} \tilde{x}'_z(\tau) d\tau$$

with $\tilde{x}'_z(\tau)$ in the second integral denoting the anticipative tail at t_{k+1}. Recursive feasibility is then guaranteed if the right-hand side of the last equation is not larger than that of the penultimate. This condition can be rewritten as

$$\int_{t_{k+C}}^{t_{k+C+1}} e^{\eta(t_{k+1} - \tau)} \tilde{x}_z(\tau) d\tau + \int_{t_{k+P}}^{\infty} e^{\eta(t_{k+1} - \tau)} \tilde{x}_z(\tau) d\tau$$
$$\le \int_{t_{k+C}}^{t_{k+C+1}} e^{\eta(t_{k+1} - \tau)} x_z^M(\tau) d\tau + \int_{t_{k+P}}^{\infty} e^{\eta(t_{k+1} - \tau)} \tilde{x}'_z(\tau) d\tau$$

where we have used the fact that the anticipative tails at t_k and t_{k+1} coincide over $[t_{k+C+1}, t_{k+P}]$. From this, we derive the equivalent inequality

$$\int_{t_{k+C}}^{t_{k+C+1}} e^{\eta(t_{k+1} - \tau)} (x_z^M(\tau) - \tilde{x}_z(\tau)) d\tau$$
$$+ \int_{t_{k+P}}^{\infty} e^{\eta(t_{k+1} - \tau)} (\tilde{x}'_z(\tau) - \tilde{x}_z(\tau)) d\tau \ge 0.$$

At this point, exploiting the fact (see the end of Section V-B3) that 1) $x_z^M(t) - \tilde{x}_z(t) = d_{z,x}/2$ in the preview horizon, and 2)

[6]From now on, we focus only on the right-hand side of the feasibility condition for compactness. In fact, imposing the left-hand side leads to the same condition (28).

the residual part of the anticipative tail is truncated, a lengthy but simple calculation leads to the condition

$$\frac{e^{-\eta(T_p-T_c)}}{\eta}(\dot{\tilde{x}}'_z)^{k+P}+\frac{d_{z,x}}{2}\geq 0$$

where $(\dot{\tilde{x}}'_z)^{k+P}$ is the last velocity sample in the preview horizon of the anticipative tail at t_{k+1}. Finally, if we denote by $v_{z,x}^{\max}$ the upper bound on the absolute value of $(\dot{\tilde{x}}'_z)^{k+P}$, we can claim that a sufficient condition for recursive feasibility is

$$T_p \geq T_c + \frac{1}{\eta}\log\frac{2\,v_{z,x}^{\max}}{\eta\, d_{z,x}} \tag{28}$$

thus concluding the proof. ■

Note the following points.

1) An upper bound $v_{z,x}^{\max}$ to be used in (28) can be derived (and enforced in the tail) based on the dynamic capabilities of the specific robot or, even more directly, using the information embedded in the footstep sequence and timing. This is the same kind of reasoning that led us to postulate the existence of an upper bound γ on $\dot{x}_z^i$ in (5).
2) Equation (28) shows that a longer preview horizon T_p is needed to guarantee recursive feasibility for taller and/or faster robots (larger η and/or $v_{z,x}^{\max}$, respectively), or for robots with more compact feet (smaller $d_{z,x}$).
3) Proposition 5 provides only a sufficient condition and, therefore, does not exclude that recursive feasibility of IS-MPC can be achieved with a smaller preview horizon, or even with a different tail. For example, in the next subsection we will describe a case (Simulation 3), in which the periodic tail represents a sufficiently accurate conjecture and therefore recursive feasibility is achieved.

3) Recursive Feasibility—Simulations: We now report some comparative MATLAB simulations aimed at showing how different choices for the tail lead to different results in terms of recursive feasibility. We use the same LIP model and parameters of Section VII-A. The MPC still operates under the assumption that the footstep sequence is given and not modifiable. The control horizon T_c is 0.8 s, while the preview horizon T_p is 1.6 s.

Figure 13 shows a comparison between IS-MPC using the truncated and periodic tail for a regular footstep sequence. When using the truncated tail, gait generation fails because the system reaches an unfeasible state, due to the significant mismatch between the truncated tail and the persistent ZMP velocities required by the gait. Recursive feasibility is instead achieved by using the periodic tail, which coincides with an anticipative tail for this case.

Figure 14 refers to a situation in which the assigned footstep sequence is irregular: two forward steps are followed by two backward steps on the same footsteps. Use of the periodic tail leads now to a loss of feasibility, as IS-MPC is wrongly conjecturing that the ZMP trajectory will keep on moving forward. The anticipative tail, which is the recommended choice for this scenario, correctly anticipates the irregularity, therefore achieving recursive feasibility.

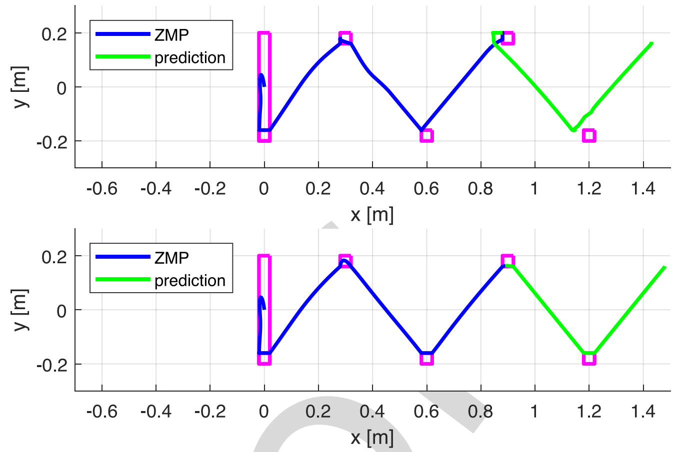


Fig. 13. Simulation 3: Gaits generated for a regular footstep sequence with different tails: truncated (top) and periodic (bottom). Note the loss of feasibility when using the truncated tail.

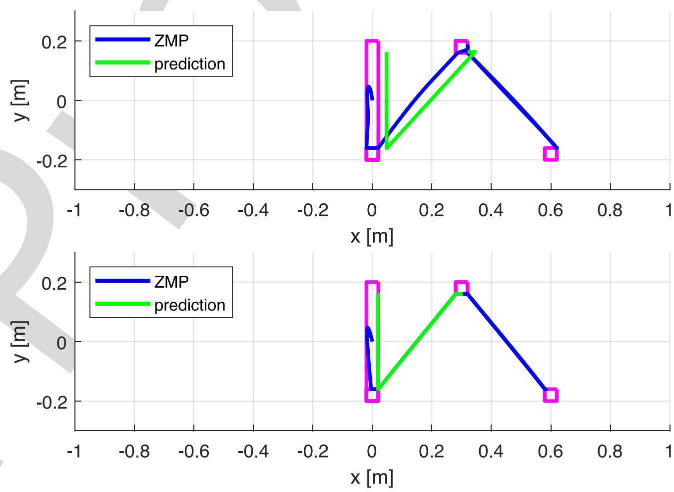


Fig. 14. Simulation 4: Gaits generated for an irregular footstep sequence with different tails: periodic (top) and anticipative (bottom). The footstep sequence consists of two forward steps followed by two backwards steps on the same footsteps. Note the loss of feasibility when using the periodic tail.

The accompanying video shows an animation of the evolutions in Figs. 13 and 14.

C. Recursive Feasibility Implies Stability

In Section VII-B2, it has been shown that recursive feasibility can be guaranteed by using the anticipative tail, provided that the preview horizon T_p is sufficiently large (see Proposition 5). Now, we prove that recursive feasibility, in turn, implies internal stability (i.e., boundedness of the CoM trajectory with respect to the ZMP).

We recall a definition first. A function $f(t)$ is said to be *of exponential order* α_0 if [34]

$$\lim_{t\to\infty} f(t)e^{-\alpha t}=0 \qquad \text{when} \quad \alpha>\alpha_0.$$

According to this definition, any bounded or polynomial function is of exponential order 0, whereas e^{at} is of exponential order

a. In particular, x_z is of exponential order 0 in IS-MPC, because it is piecewise linear with bounded derivative, see (5).

Proposition 6: If IS-MPC is recursively feasible, then internal stability is guaranteed.

Proof: We establish the result by contradiction, that is, we assume that internal stability is violated and show that this is inconsistent with IS-MPC being recursively feasible. We focus on the dynamics along the sagittal axis x; an identical reasoning can be done along the coronal axis y.

Assume that internal stability is violated, i.e., $x_c - x_z$ diverges. This implies that $x_u - x_z$ diverges, because 1) $x_c = (x_s + x_u)/2$ in view of (8) and (9), and 2) $x_s - x_z$ is bounded (in fact, its dynamics is BIBO-stable and forced by $\dot{x}_z$, which is bounded). Since the dynamics of $x_u - x_z$ has a single eigenvalue η and is also forced by $\dot{x}_z$, then $x_u - x_z$ will diverge with exponential order η. Finally, this implies that the feasibility condition (26) will be violated at a future instant of time, as the upper and lower bounds in the inequality are functions of the same exponential order as x_z. This contradicts the assumption that IS-MPC is recursively feasible. ■

D. Wrapping Up

As discussed at the end of Section V-A, a causal MPC can only contain an approximate version of the stability constraint, because the tail in (17) is unknown and, therefore, must be conjectured. Nevertheless, Proposition 6 states that the repeated enforcement of this constraint at each iteration of IS-MPC is effective, in the sense that *internal stability is achieved as long as the controller is recursively feasible*. In turn, the latter property is guaranteed if the anticipative tail is used with a T_p that extends beyond T_c enough to make the approximation sufficiently accurate [see Proposition 5, and in particular (28)].

At this point, the reader may wonder whether there is a requirement on the minimum control horizon T_c in order for IS-MPC to work. The answer is that T_c may indeed be arbitrarily small, with one caveat: as shown by (27), the feasibility region shrinks as T_c decreases. However, once the system is initialized in this reduced region, the recursive feasibility of IS-MPC will depend only on T_p through the sufficient condition (28).

The possibility of decreasing T_c without affecting stability is a distinct advantage of IS-MPC with respect to schemes which need sufficiently long T_c to work. In fact, a shorter T_c means less computation, which may be important for real-time onboard implementation on low-cost platforms, such as the NAO of our experiments. Moreover, since the MPC needs to know the (candidate) footstep locations in the control horizon, decreasing T_c means that footsteps are required over a smaller interval, making it possible to use short-term reactive planners.

VIII. Simulations

We now report some complete gait generation results (footstep generation + IS-MPC) obtained in the V-REP simulation environment. The humanoid platform is HRP-4, a 34-dof, 1.5 m tall humanoid robot. We enabled dynamic simulation using the Newton Dynamics engine.

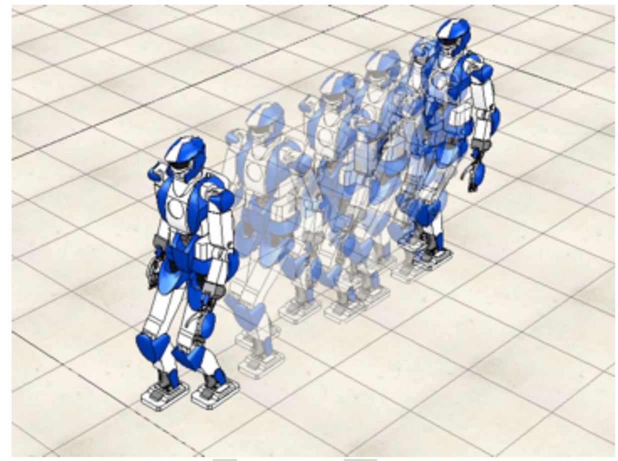


Fig. 15. Simulation 5. HRP-4 following a variable reference velocity.

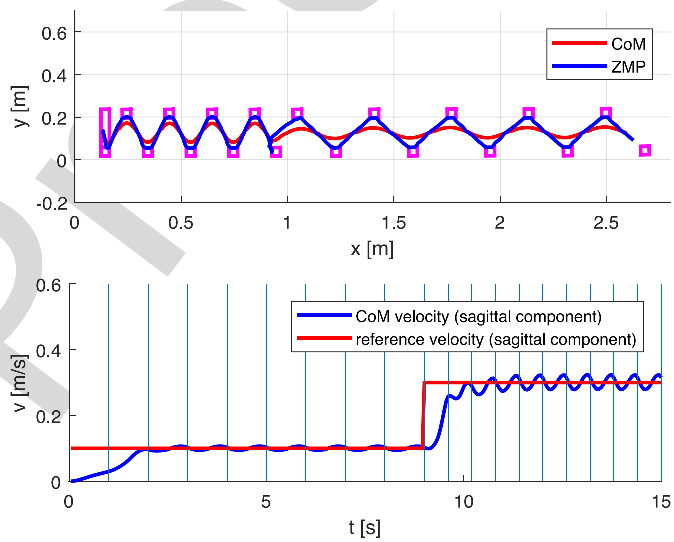


Fig. 16. Simulation 5: CoM and ZMP trajectories (top) and sagittal velocity (bottom).

The whole gait generation framework runs at 100 Hz ($\delta = 0.01$ s). Footstep timing is determined using rule (1) with $\bar{L}_s = 0.12$ m, $\bar{T}_s = 0.8$ s, and $\bar{v} = 0.15$ m/s as cruise parameters, and $\alpha = 0.1$ m/s (as in Fig. 2). Each generated T_s is split into T_{ss} (single support) and T_{ds} (double support) using a 60–40% distribution. Candidate footsteps are generated as explained in Section III-B, with $\theta_{\max} = \pi/8$ rad and $\ell = 0.18$ m. In the IS-MPC module, which uses a control horizon T_c of 1.6 s, we have set $h_c = 0.78$ m. The dimensions of the ZMP admissible region are $d_{z,x} = d_{z,y} = 0.04$ m, while those of the kinematically admissible region are $d_{a,x} = 0.3$ m and $d_{a,y} = 0.07$ m. The weight in the QP cost function is $\beta = 10^4$. The qpOASES library was used to solve the QP, here as well as in the experiments to be presented in the next section.

Figure 15 shows a stroboscopic view of the first simulation (see the accompanying video for a clip). The robot is commanded a sagittal reference velocity v_x of 0.1 m/s, which is then abruptly

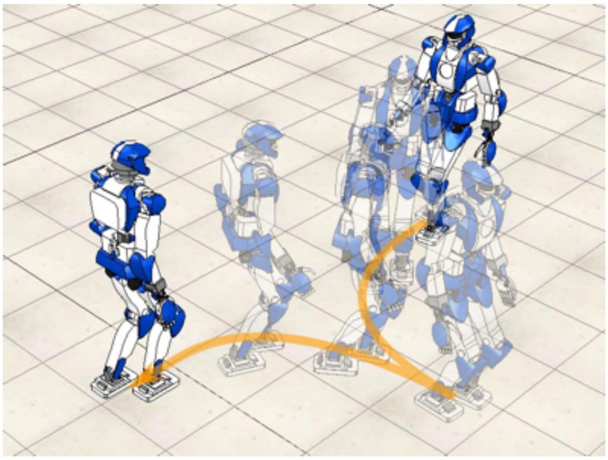


Fig. 17. Simulation 6: HRP-4 walking along a cusp.

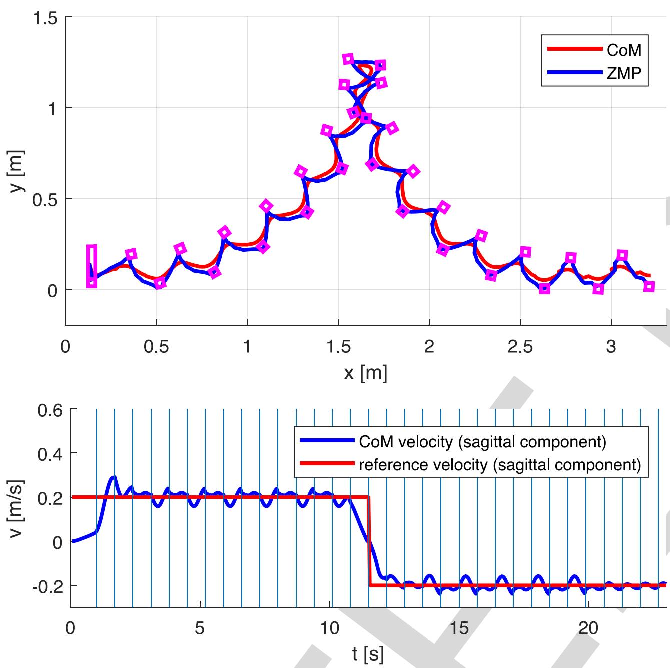


Fig. 18. Simulation 6: CoM and ZMP trajectories (top) and sagittal velocity (bottom).

increased to 0.3 m/s. The preview horizon is $T_p = 3.2$ s, and the anticipative tail is used. The generated CoM and ZMP trajectories together with the sagittal CoM velocity are shown in Fig. 16. As expected, the higher commanded velocity is realized by increasing both the step length and the frequency.

In the second simulation, shown in Fig. 17 and the accompanying video, the reference velocities are aimed at producing a cusp trajectory. In particular, initially, we have $v_x = 0.2$ m/s and $\omega = 0.2$ rad/s; after a quarter turn, we change v_x to -0.2 m/s; after another quarter turn, ω is zeroed. As before, T_p is 3.2 s, and the anticipative tail is used for the stability constraint. Figure 18 shows plots of the generated ZMP and CoM trajectories, together with the sagittal CoM velocity.

Video clips of the complete simulations are shown in the accompanying video.

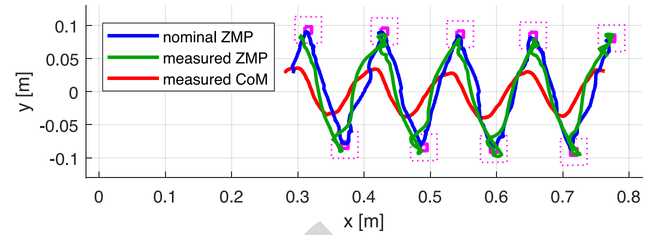


Fig. 19. Nominal ZMP, measured ZMP, and measured CoM along a forward gait of HRP-4. Note the restricted ZMP regions (magenta, solid) and the original ZMP regions used in the simulations (magenta, dotted).

IX. Experiments

Experimental validation of the proposed method for gait generation was performed on two platforms, i.e., the NAO and HRP-4 humanoid robots.

NAO is a 23-dof, 58 cm tall humanoid equipped with a single-core Intel Atom running at 1.6 GHz. Our method, implemented as a custom module in the B-Human RoboCup SPL team framework [35], runs in real time on the onboard CPU at a control frequency of 100 Hz ($\delta = 0.01$ s). Footstep timing is determined using rule (1) with $\bar{L}_s = 0.075$ m, $\bar{T}_s = 0.5$ s, and $\bar{v} = 0.15$ m/s as cruise parameters, and $\alpha = 0.1$ m/s (as in Fig. 2). Candidate footsteps are generated as explained in Section III-B, with $\theta_{\max} = \pi/8$ rad and $\ell = 0.1$ m. In the IS-MPC module, we have set $T_c = 1.0$ s and $h_c = 0.23$ m. The dimensions of the ZMP admissible region are $d_{z,x} = d_{z,y} = 0.03$ m, while those of the kinematically admissible region are $d_{a,x} = 0.1$ m and $d_{a,y} = 0.05$ m. The weight in the QP cost function is $\beta = 10^4$. The anticipative tail is used with a preview horizon $T_p = 2.0$ s.

The software architecture of HRP-4 requires control commands to be generated at a frequency of 200 Hz ($\delta = 0.005$ s). Gait generation runs on an external laptop PC, and joint motion commands are sent to the robot via Ethernet using TCP/IP. The parameters are the same of the V-REP simulations in the previous section, including $T_c = 1.6$ s, with the exception of $d_{z,x}$ and $d_{z,y}$ that are reduced to 0.01 m for increased safety. The anticipative tail is used in the stability constraint.

Before presenting complete locomotion experiments, we report in Fig. 19 some data from a typical forward gait of HRP-4. In particular, the plot shows the nominal ZMP trajectory, as generated by IS-MPC, together with the ZMP measurements reconstructed from the force-torque sensors at the robot ankles [36]. Note how the restriction of the ZMP admissible region is effective, in the sense that while the measured ZMP violates the constraints, it stays well within the original ZMP admissible region used in the simulation.

The accompanying video shows two successful experiments for each robot. In the first, the robots are required to perform a forward–backward motion, as shown in Fig. 20. The reference velocities are $v_x = \pm 0.15$ m/s for the NAO and $v_x = \pm 0.2$ m/s for the HRP-4.

In the second experiment, which is shown in Fig. 21, the robots are given reference velocities aimed at performing an L-shaped motion. In particular, we have $v_x = 0.15$ m/s followed by $v_y = 0.05$ m/s for the NAO, and $v_x = 0.2$ m/s followed by $v_y = 0.2$ m/s for the HRP-4.

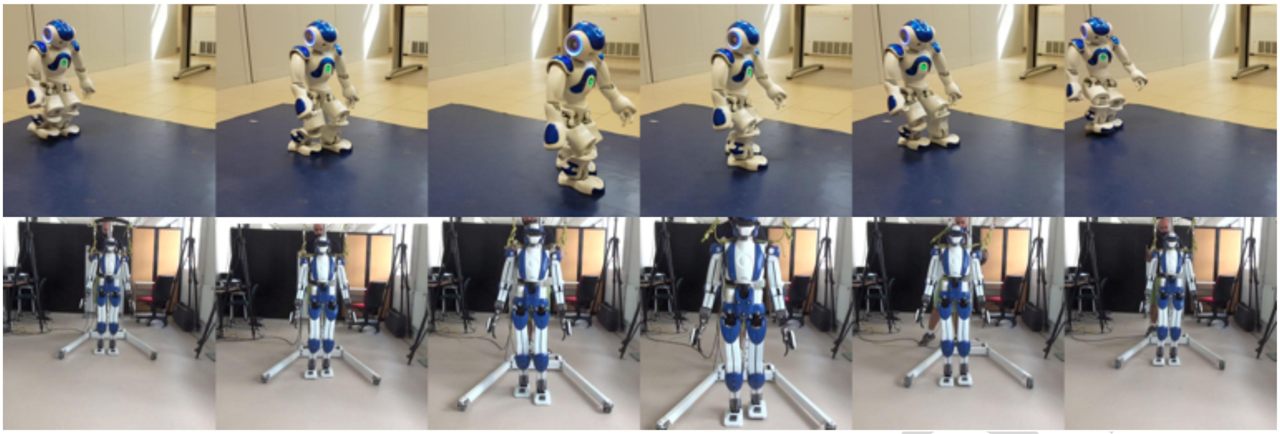


Fig. 20. Experiments 1 and 2: NAO and HRP-4 walking forward and backward. See the accompanying video.

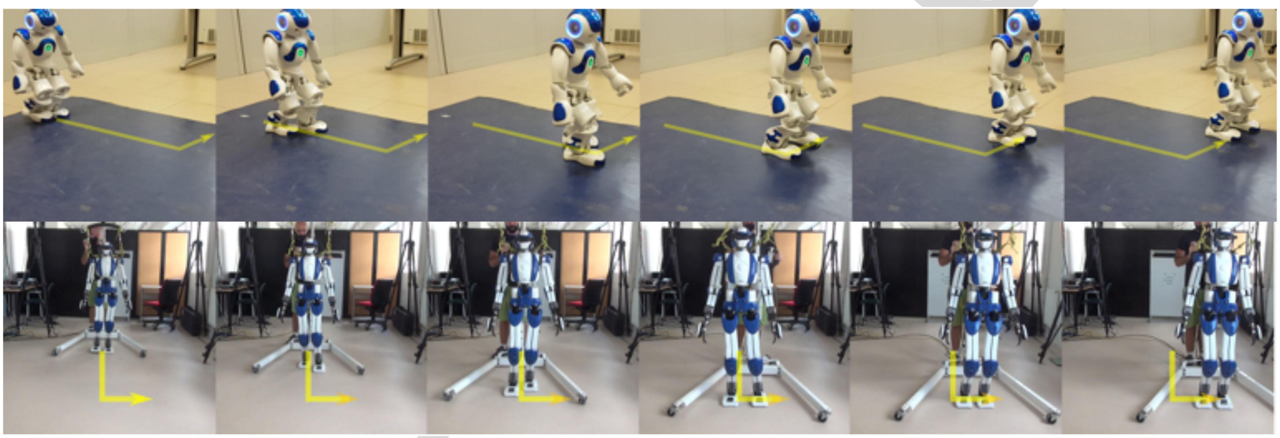


Fig. 21. Experiments 3 and 4: NAO and HRP-4 walking along an L. See the accompanying video.

X. Conclusion

In this article, we presented a complete MPC framework (IS-MPC) for generating intrinsically stable humanoid gaits that realize high-level cartesian velocity commands. We discussed various versions of the newly introduced stability constraint, which may be used depending on the available quantity of preview information on the reference motion. It was also shown how the different stability constraints can be interpreted as terminal constraints, some of which were new in the literature.

A detailed study of the feasibility of the generic MPC iteration was developed and used to derive conditions under which recursive feasibility can be guaranteed. Comparative simulations were presented to illustrate the effect of the different tails on the resulting gait and confirmed that incorporating preview information in the tail was essential to preserve feasibility. Finally, it was shown that recursive feasibility of IS-MPC implies internal stability of the CoM/ZMP dynamics.

Experimental results obtained with an onboard NAO implementation proved that the proposed algorithm is viable even in the presence of limited computing capabilities. Additional successful experiments were carried out on the full-sized humanoid HRP-4.

The advantages of IS-MPC can be summarized as follows.

1) It includes an explicit stability constraint, which, through the choice of the tail, can be declined on the basis of the preview information so as to accommodate different gaits to be executed.
2) It is guaranteed to be recursively feasible if the anticipative tail is used and the preview horizon is sufficiently long (see Proposition 5). This clarifies the role and the amount of the required preview information, in contrast with most literature where such an analysis is missing.
3) It is the first MPC-based gait generation with an explicit proof of internal stability, which is shown to be a direct consequence of recursive feasibility (see Proposition 6).
4) It is general enough to be applicable to different humanoids (such as NAO and HRP-4) without significant adaptation.

We are currently working on several extensions of the proposed approach, such as:

1) developing a robust version of the proposed IS-MPC scheme that can withstand unmodeled dynamics and disturbances [37];
2) extending our approach to the 2.5D case (piecewise-horizontal ground, such as stairs or flat step stones), for

which we have presented a preliminary version of IS-MPC in [38] and a footstep planner in [39];

3) investigating the use of learning techniques in conjunction with MPC in order to improve performance.

Appendix

We collect here some useful properties used in the proofs of the various propositions. For compactness, we use the following notation:

$$\eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} x_z(\tau) d\tau = x_u^*(t_k; x_z(t)). \tag{A.1}$$

Property 1. Linearity in $x_z(t)$:

$$x_u^*(t_k; a x_z^a(t) + b x_z^b(t)) = a x_u^*(t_k; x_z^a(t)) + b x_u^*(t_k; x_z^b(t)).$$

Property 2: If $x_z(t) = \delta_{-1}(t - t_k)$, we get

$$x_u^*(t_k; \delta_{-1}(t - t_k)) = 1.$$

Property 3: If $x_z(t) = \rho(t - t_k)$, we get

$$x_u^*(t_k; \rho(t - t_k)) = 1/\eta.$$

Properties 1–3 are easily derived by explicit computation of the integral in (A.1).

Property 4: If $x_z(t) = 0$ for $t < t_k$, we get

$$x_u^*(t_k; x_z(t - T)) = e^{-\eta T} x_u^*(t_k; x_z(t)), \quad T \geq 0.$$

Proof:

$$\begin{aligned} x_u^*(t_k; x_z(t - T)) &= \eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} x_z(\tau - T) d\tau \\ &= \eta \int_{t_k - T}^{\infty} e^{-\eta(\theta - t_k + T)} x_z(\theta) d\theta \\ &= \eta e^{-\eta T} \int_{t_k - T}^{\infty} e^{-\eta(\theta - t_k)} x_z(\theta) d\theta \\ &= e^{-\eta T} \eta \int_{t_k}^{\infty} e^{-\eta(\theta - t_k)} x_z(\theta) d\theta \\ &= e^{-\eta T} x_u^*(t_k; x_z(t)). \end{aligned}$$

■

Property 4 (*time shifting*) shows how the stability condition for the time-shifted function $x_z(t - T)$ can be written in terms of the stability condition for the original function $x_z(t)$.

Acknowledgment

The authors would like to thank Dr. A. Kheddar of CNRS for hosting D. De Simone at the LIRMM in Montpellier and allowing him to perform experiments on the HRP-4 humanoid robot.

References

[1] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. New York, NY, USA: Springer, 2014.

[2] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method for real-time gait planning for humanoid robots," *Int. J. Humanoid Robot.*, vol. 03, pp. 1–19, 2006.

[3] M. Morisawa *et al.*, "A biped pattern generation allowing immediate modification of foot placement in real-time," in *Proc. 6th IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 581–586.

[4] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in *Proc. 7th IEEE-RAS Int. Conf. Humanoid Robots*, 2007, pp. 1–6.

[5] S. Kajita *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1620–1626.

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

[7] 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, nos. 5/6, pp. 719–737, 2010.

[8] J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, "Robust feedback control of ZMP-based gait for the humanoid robot NAO," *Int. J. Robot. Res.*, vol. 32, nos. 9/10, pp. 1074–1088, 2013.

[9] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, "Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1943–1950.

[10] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1763–1768.

[11] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust dynamic walking using online foot step optimization," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2016, pp. 5373–5378.

[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 Robot. Autom. Lett.*, vol. 2, no. 1, pp. 10–17, Jan. 2017.

[13] S. Caron and A. Kheddar, "Dynamic walking over rough terrains by nonlinear predictive control of the floating-base inverted pendulum," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2017, pp. 5017–5024.

[14] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Handbook of Robotics*. New York, NY, USA: Springer, 2016, pp. 1203–1234.

[15] P. B. Wieber, "Viability and predictive control for safe locomotion," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2008, pp. 1103–1108.

[16] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[17] B. Henze, C. Ott, and M. A. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2014, pp. 3253–3258.

[18] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *Proc. 14th IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 444–450.

[19] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *Int. J. Robot. Res.*, vol. 31, no. 9, pp. 1094–1113, 2012.

[20] T. Sugihara and T. Yamamoto, "Foot-guided agile control of a biped robot through ZMP manipulation," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2017, pp. 4546–4551.

[21] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning feasibility constraints for multi-contact locomotion of legged robots," in *Proc. Robot.: Sci. Syst.*, 2017.

[22] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot—1st report: Walking gait pattern generation," in *Proc. Int. Conf. Intel. Robots Syst.*, 2009, pp. 1084–1091.

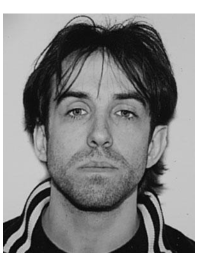
[23] T. Kamioka *et al.*, "Dynamic gait transition between walking, running and hopping for push recovery," in *Proc. 17th IEEE-RAS Int. Conf. Humanoid Robots*, 2017, pp. 1–8.

[24] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, Apr. 2015.

[25] S. Caron, A. Escande, L. Lanari, and B. Mallein, "Capturability-based pattern generation for walking with variable height," *IEEE Trans. Robot.*, to be published.

[26] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," in *Proc. 10th IFAC Symp. Robot Control*, 2012, pp. 165–171.

[27] E. C. Kerrigan, “Robust constraint satisfaction: Invariant sets and predictive control,” Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2001.
[28] M. Ciocca, P.-B. Wieber, and T. Fraichard, “Strong recursive feasibility in model predictive control of biped walking,” in *Proc. 17th IEEE-RAS Int. Conf. Humanoid Robots*, 2017, pp. 730–735.
[29] A. Sherikov, “Balance preservation and task prioritization in whole body motion control of humanoid robots,” Ph.D. dissertation, Univ. Grenoble Alpes, Grenoble, France, 2016.
[30] N. Scianca, M. Cognetti, D. De Simone, L. Lanari, and G. Oriolo, “Intrinsically stable MPC for humanoid gait generation,” in *Proc. 16th IEEE-RAS Int. Conf. Humanoid Robots*, 2016, pp. 101–108.
[31] A. Aboudonia, N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, “Humanoid gait generation for walk-to locomotion using single-stage MPC,” in *Proc. 17th IEEE-RAS Int. Conf. Humanoid Robots*, 2017, pp. 178–183.
[32] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *Proc. 6th IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 200–207.
[33] L. Lanari and S. Hutchinson, “Inversion-based gait generation for humanoid robots,” in *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, 2015, pp. 637–642.
[34] W. LePage, *Complex Variables and the Laplace Transform for Engineers*. New York, NY, USA: McGraw-Hill, 1961.
[35] T. Röfer *et al.*, “B-human team report and code release 2015,” 2015. [Online]. Available: http://www.b-human.de/downloads/publications/2015/CodeRelease2015.pdf
[36] A. Tanguy, D. De Simone, A. I. Comport, G. Oriolo, and A. Kheddar, “Closed-loop MPC with dense visual SLAM-stability through reactive stepping,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 1397–1403.
[37] F. M. Smaldone, N. Scianca, V. Modugno, L. Lanari, and G. Oriolo, “Gait generation using intrinsically stable MPC in the presence of persistent disturbances,” in *Proc. 19th IEEE-RAS Int. Conf. Humanoid Robots*, 2019, pp. 682–687.
[38] A. Zamparelli, N. Scianca, L. Lanari, and G. Oriolo, “Humanoid gait generation on uneven ground using intrinsically stable MPC,” *IFAC-PapersOnLine*, vol. 51, pp. 393–398, 2018.
[39] P. Ferrari, N. Scianca, L. Lanari, and G. Oriolo, “An integrated motion planner/controller for humanoid robots on uneven ground,” in *Proc. 18th Eur. Control Conf.*, 2019, pp. 1598–1603.



Nicola Scianca received the bachelor’s degree in mechanical engineering and the master’s degree in systems engineering, respectively in 2010 and 2014, from the Sapienza University of Rome, Rome, Italy, where he is currently working toward the Ph.D. degree in control engineering.

In 2019, he was a Visiting Student with the Model Predictive Control Laboratory, University of California at Berkeley, CA, USA. His main research interests include model-predictive control for humanoid robots.

Daniele De Simone received the master’s degree in artificial intelligence and robotics and the Ph.D. degree in control engineering from the Sapienza University of Rome, Rome, Italy, in 2015 and 2019, respectively.

In 2018, he was a Visiting Student with the Laboratoire d’Informatique, de Robotique et de Microélectronique de Montpellier, Montpellier, France. His research interests include motion planning techniques and reactive behaviors for collision avoidance for humanoid robots.

Leonardo Lanari received the Ph.D. degree in control engineering from the Sapienza University of Rome, Rome, Italy, in 1992.

He is currently an Associate Professor of Automatic Control with the Department of Computer, Control and Management Engineering, Sapienza University of Rome. His research interests include control of robotic systems, with an emphasis on humanoid control and robots with elastic joints and links.

Giuseppe Oriolo (S’89–M’92–SM’02–F’16) received the Ph.D. degree in control engineering from the Sapienza University of Rome, Rome, Italy, in 1992.

He is currently with the Department of Computer, Control and Management Engineering (DIAG), Sapienza University of Rome, where he is a Full Professor of Automatic Control and Robotics and the Director of the DIAG Robotics Laboratory. His research interests include planning and control of robotic systems.

Prof. Oriolo was an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 2001 to 2005 and an Editor for the IEEE TRANSACTIONS ON ROBOTICS from 2009 to 2013.