

# Intrinsically Stable MPC for Humanoid Gait Generation

Nicola Scianca, Marco Cognetti, Daniele De Simone, Leonardo Lanari, Giuseppe Oriolo

**Abstract**—We present a novel MPC method for humanoid gait generation that is guaranteed to produce stable CoM trajectories. This is obtained by using a dynamic extension of the LIP as motion model, with the ZMP velocity as a control variable, and embedding an explicit stability constraint in the formulation. Such constraint turns out to be linear in the control variables, leading to a standard QP problem with equality and inequality constraints. The intrinsically stable MPC framework is developed into a full-fledged gait generation scheme by including automatic footstep placement. Simulations show that the proposed method is very effective and performs robustly in the presence of changes in the prediction horizon.

## I. INTRODUCTION

One of the most relevant problems in the control of humanoids is the generation of locomotion gaits. The main requirement is obviously that the robot maintains dynamic balance while walking. A celebrated sufficient condition that guarantees this property entails that the Zero Moment Point (ZMP, the point where the horizontal component of the moment of the ground reaction forces becomes zero) should remain at all times within the support polygon of the robot.

Many gait generation schemes enforce this ZMP condition by computing a suitable trajectory for the robot Center of Mass (CoM). Due to the complexity of humanoid dynamics, simplified models are invariably used to relate the evolution of the CoM to that of the ZMP. A second-order linear system is often adopted, known as the Linear Inverted Pendulum (LIP) or the Cart-Table (CT) depending on whether the ZMP is seen as an input or as an output [1]. Once a CoM trajectory is generated, kinematic control provides joint commands that drive the robot along it.

The above approach was pioneered by the preview controller of [2], which is obtained by solving an LQ regulation problem for a dynamical extension of the CT. The control signal (the CoM jerk) is chosen so as to achieve asymptotic tracking of the desired ZMP trajectory. This would require in principle advance knowledge of the complete ZMP profile, but an effective solution is to use a finite preview window.

An important transition was made in [3] by recasting preview control into a Model Predictive Control (MPC) framework. The ZMP trajectory is generated on-line by minimizing the control effort with the balance condition as a constraint. Improvements were obtained in [4], [5] by incorporating footstep generation in the MPC problem. These papers triggered a new line of works, e.g., [6], [7], [8].

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. This work is supported by the EU FP7 project COMANOID.

A related approach to gait generation is based on the notion of capture point, which was first introduced in [9], [10] for investigating the push recovery problem and then extended to the 3D case in [11]. Its use in an MPC framework on the LIP model has been discussed in [12] and [13].

In spite of their diversity, all the above approaches face a well-known structural problem when connecting CoM to ZMP trajectories. This can be understood, e.g., by considering that the LIP is unstable, while the CT — which may be seen as its inverse system — is non-minimum phase. As a consequence, an instability problem will arise in both cases, meaning that the CoM motion associated to any ZMP trajectory will in general include a divergent component. To achieve effective gait generation, it is therefore essential to select a CoM trajectory which does not diverge, i.e., it is *stable*. Indeed, in the CT-based MPC framework there is no guarantee of stability; however, it is argued that jerk minimization produces stable CoM trajectories provided that the prediction horizon is sufficiently long [14].

A recent study [15] of the above LIP instability issue has identified a necessary and sufficient condition for the CoM trajectory to be stable in response to any ZMP profile. This is a constraint on the initial system state, whose expression involves the future history of the input. In view of the duality of the LIP and CT, this result can be recast in the framework of stable inversion for non-minimum phase systems [16].

Here, we propose a scheme that combines the flexibility of MPC with the robustness of guaranteed stability. To this end, we embed in the formulation of the MPC problem an explicit constraint that ensures that the generated CoM trajectory will be stable. We use as motion model a dynamic extension of the LIP, with the ZMP velocity as a control variable. The stability constraint turns out to be linear in this setting, leading to a standard constrained QP problem. The intrinsically stable MPC framework is developed into a complete gait generation scheme by including automatic footstep placement. The proposed method shows robust performance with respect to changes in the prediction horizon, whereas the CT-based MPC framework does not.

The paper is organized as follows. In Sect. II we recall the LIP model and the characterization of stable CoM trajectories. Intrinsically stable MPC is introduced in Sect. III for the case of assigned footsteps. To illustrate its benefits, comparative simulations with the standard approach are shown in Sect. IV. MPC with simultaneous footstep generation is discussed in Sect. V with the aid of simulations on the NAO humanoid. The concluding section hints at some future work.

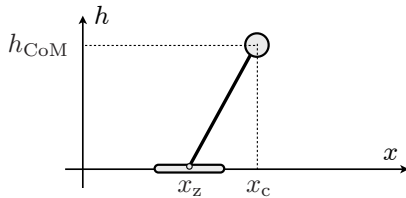


Fig. 1. The Linear Inverted Pendulum in the  $x$  direction.

## II. STABLE CoM TRAJECTORY GENERATION

In this section, we quickly recapitulate the LIP model and how it can be used to derive a characterization of stable CoM trajectories associated to a given ZMP trajectory. The reader should refer to [15], [16] for further details.

The LIP is a simplified model aimed at describing the motion of the CoM of a biped when its height is kept constant and no rotational effects are considered. In addition to being linear, the LIP has the advantage that the dynamics along the  $x$  and  $y$  directions are governed by decoupled, identical differential equations. Many studies have confirmed that use of LIP for gait generation is effective in spite of its simplicity.

Let us consider motion along the  $x$  axis for illustration (see Fig. 1). Denoting by  $x_c$  and  $x_z$  respectively the coordinate of the CoM and the ZMP, the LIP dynamics is expressed as<sup>1</sup>

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

where  $\omega = \sqrt{g/h_{\text{CoM}}}$  and  $h_{\text{CoM}}$  is the constant height of the CoM. Note that here  $x_z$  acts as an input signal.

System (1), whose dynamic matrix has eigenvalues  $\pm\omega$ , can be decomposed in two subsystems, one stable and the other unstable. To this end, one can use the following change of coordinates

$$\begin{aligned} x_s &= x_c - \dot{x}_c/\omega \\ x_u &= x_c + \dot{x}_c/\omega, \end{aligned}$$

that highlights the stable ( $x_s$ ) and unstable ( $x_u$ ) components.

Even if the system is unstable, for any input (i.e., for each ZMP trajectory  $x_z(t)$ ) there exists a special initialization of  $x_u$  such that the resulting CoM trajectory is *stable*, in the sense that it does not diverge w.r.t. the input itself. This initialization is expressed as

$$x_u(t_{\text{in}}) = x_c(t_{\text{in}}) + \dot{x}_c(t_{\text{in}})/\omega = x_u^*(t_{\text{in}}) \quad (2)$$

where  $t_{\text{in}}$  is the initial instant and

$$x_u^*(t_{\text{in}}) = \omega \int_{t_{\text{in}}}^{\infty} e^{-\omega(\tau-t_{\text{in}})} x_z(\tau) d\tau. \quad (3)$$

Note that  $x_u^*(t_{\text{in}})$  depends on the whole history of the control input, i.e., on the whole ZMP trajectory. The initialization of  $x_s$  is obviously irrelevant. In the following, we shall refer to (2–3) as the *stability constraint*.

<sup>1</sup>Note that (1) applies to both point feet and finite-sized feet [10].

## III. INTRINSICALLY STABLE MPC

To illustrate the proposed method, we assume first that the footstep sequence is assigned, together with the associated timing. This means in particular that for each step we know the duration of the single and the double support phase.

To simplify the presentation, we will suppose throughout the paper that all footsteps have the same orientation. The extension to the case of rotated footsteps is straightforward.

### A. Motion model and constraints

Based on the previously mentioned property of the LIP, we can consider the  $x$  and  $y$  components of the CoM motion as decoupled. In the following, let us first consider generation of motion in the  $x$  direction.

In our MPC scheme, the control variable is the ZMP velocity  $\dot{x}_z$ . Similarly to [17], we are then considering the following third-order system (LIP+dynamic extension)

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

We assume piecewise-constant control over time intervals of duration  $\delta$ . In particular, denote by  $t_j = j\delta$ ,  $j = 0, 1, \dots$ , the sampling instants, and by  $w^j = w(t_j)$  for the value of a generic signal  $w$  sampled at time  $t_j$ . We have then

$$\dot{x}_z(t) = \dot{x}_z^j, \quad t \in [t_j, t_{j+1}),$$

so that the ZMP position in the same time interval will be expressed as

$$x_z(t) = x_z^j + (t - t_j) \dot{x}_z^j, \quad t \in [t_j, t_{j+1}). \quad (5)$$

We consider a *prediction horizon*  $T_h = N \cdot \delta$  that extends from the current sampling instant  $t_k$  to  $t_{k+N} = t_k + T_h$ . Collect in vector

$$\dot{X}_z^k = (\dot{x}_z^k \dots \dot{x}_z^{k+N-1})^T$$

all the control variables within  $T_h$  (note that the ZMP velocity at  $t_{k+N}$  is not included).

For the problem at hand, the constraints to be verified over  $T_h$  are of two kinds. The first concerns the ZMP position, which must be at all times within the support polygon defined by the footstep sequence and the associated timing. The second is the stability constraint needed to guarantee that the CoM trajectory generated by our MPC scheme will be stable; this constraint must be satisfied at the start of the prediction horizon, i.e., at the current sampling instant  $t_k$ .

1) *ZMP constraint*: The ZMP constraint is expressed as

$$x_{z,\min}(t) \leq x_z(t) \leq x_{z,\max}(t), \quad t \in [t_k, t_{k+N}],$$

with  $x_{z,\min}(t)$  and  $x_{z,\max}(t)$  extracted from the footstep sequence. Since  $x_z$  is linear w.r.t.  $t$  within each interval (see (5)), the previous constraint is guaranteed to be satisfied whenever its sampled version is:

$$x_{z,\min}^j \leq x_z^j \leq x_{z,\max}^j, \quad j = k+1, \dots, k+N. \quad (6)$$

To rewrite this constraint in terms of our control variables, collect in  $X_z^{k+1} = (x_z^{k+1} \dots x_z^{k+N})^T$  all the ZMP position samples involved in (6). We can write

$$X_z^{k+1} = p x_z^k + P \dot{X}_z^k \quad (7)$$

having set

$$p = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad P = \begin{pmatrix} \delta & 0 & \dots & 0 \\ \delta & \delta & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \delta & \delta & \dots & \delta \end{pmatrix}.$$

Constraint (6) can be therefore expressed as

$$X_{z,\min}^{k+1} - p x_z^k \leq P \dot{X}_z^k \leq X_{z,\max}^{k+1} - p x_z^k, \quad (8)$$

with  $X_{z,\min}^{k+1} = (x_{z,\min}^{k+1} \dots x_{z,\min}^{k+N})^T$  and similarly for  $X_{z,\max}^{k+1}$ .

2) *Stability constraint*: One obtains the stability constraint for our MPC scheme by setting  $t_{\text{in}} = t_k$  in (2–3). We get

$$x_c(t_k) + \dot{x}_c(t_k)/\omega = x_u^*(t_k), \quad (9)$$

where

$$x_u^*(t_k) = \omega \int_{t_k}^{\infty} e^{-\omega(\tau-t_k)} x_z(\tau) d\tau.$$

To convert this to a constraint on the control variables we compute  $x_u^*(t_k)$  using the ZMP trajectory (5). We obtain

$$x_u^*(t_k) = \omega \sum_{j=k}^{\infty} \left( e^{\omega t_k} \int_{t_j}^{t_{j+1}} e^{-\omega\tau} (x_z^j + (\tau - t_j) \dot{x}_z^j) d\tau \right).$$

Letting  $s = \tau - t_j$ , this can be rewritten as

$$x_u^*(t_k) = \omega \sum_{j=k}^{\infty} \left( e^{-\omega(t_j-t_k)} \int_0^{\delta} e^{-\omega s} (x_z^j + s \dot{x}_z^j) ds \right),$$

or

$$x_u^*(t_k) = \sum_{j=k}^{\infty} e^{-\omega(t_j-t_k)} (\alpha x_z^j + \beta \dot{x}_z^j), \quad (10)$$

where

$$\begin{aligned} \alpha &= \omega \int_0^{\delta} e^{-\omega s} ds = 1 - e^{-\omega\delta} = 1 - \lambda \\ \beta &= \omega \int_0^{\delta} e^{-\omega s} s ds = \frac{1 - \lambda}{\omega} - \lambda\delta \end{aligned}$$

and we have set  $\lambda = e^{-\omega\delta}$ .

The summation in (10) can be split in two parts:

$$\begin{aligned} x_u^*(t_k) &= \sum_{j=k}^{k+N-1} \dots + \sum_{j=k+N}^{\infty} \dots \\ &= x_{u,1}^*(t_k) + x_{u,2}^*(t_k). \end{aligned} \quad (11)$$

The first depends on the ZMP velocities within the prediction horizon  $T_h$ , i.e., the control variables  $\dot{x}_z^k, \dots, \dot{x}_z^{k+N-1}$ ; whereas the second involves the ZMP velocities  $\dot{x}_z^{k+N}, \dot{x}_z^{k+N+1}, \dots$ , after the prediction horizon, which we will call the *tail* in the following.

Using (7), it is possible to write the the first term as

$$x_{u,1}^*(t_k) = (1 - \lambda^N) x_z^k + \left( \frac{1 - \lambda}{\omega} b^T - \lambda^N \delta p^T \right) \dot{X}_z^k, \quad (12)$$

where

$$b^T = (1 \quad \lambda \quad \dots \quad \lambda^{N-1}).$$

To compute  $x_{u,2}^*(t_k)$ , we must make a conjecture about the ZMP velocity tail. In particular, we will consider a tail obtained by *infinite replication*<sup>2</sup> of the control variables over the last gait period (i.e., the time interval associated to the last two steps). Assuming for simplicity that the prediction horizon  $T_h$  is equal to the gait period (formulas for the general case are easily obtained), we have

$$\left( \dot{x}_z^{k+iN} \dots \dot{x}_z^{k+(i+1)N-1} \right)^T = \dot{X}_z^k, \quad i = 1, 2, \dots$$

Using this, the second term in (11) can be manipulated so as to assume the following form:

$$x_{u,2}^*(t_k) = \lambda^N x_z^k + \lambda^N \left( \frac{1}{\omega} \frac{1 - \lambda}{1 - \lambda^N} b^T + \delta p^T \right) \dot{X}_z^k. \quad (13)$$

Plugging (12) and (13) in (11), and the latter in (9), gives the final form of the stability constraint:

$$\frac{1}{\omega} \frac{1 - \lambda}{1 - \lambda^N} b^T \dot{X}_z^k = x_c^k + \frac{\dot{x}_c^k}{\omega} - x_z^k. \quad (14)$$

Note the inversion between (9), which identifies the stable initialization at  $t_k$  for a certain  $x_z(t)$ , and (14), which constrains the control variables so that the associated stable initialization will match the current state at  $t_k$ . Correspondingly, the control variables are in the left-hand side of (14), whereas the right-hand side only contains known quantities.

### B. Formulation of the QP problem

In our MPC framework, constraints (8) and (14) act on the  $x_z$  coordinate of the ZMP. Constraints acting on the  $y_z$  coordinate have exactly the same expressions, with the obvious replacements. As a cost function, we use the sum of the squared norms of the control variables  $\dot{X}_z^k$  and  $\dot{Y}_z^k$ .

Therefore, the Quadratic Programming (QP) problem to be solved at each iteration is given by

$$\begin{aligned} &\min_{\dot{X}_z^k, \dot{Y}_z^k} \|\dot{X}_z^k\|^2 + \|\dot{Y}_z^k\|^2 \quad \text{subject to} \\ &\quad \% \text{ ZMP constraints} \\ &X_{z,\min}^{k+1} - p x_z^k \leq P \dot{X}_z^k \leq X_{z,\max}^{k+1} - p x_z^k \\ &Y_{z,\min}^{k+1} - p y_z^k \leq P \dot{Y}_z^k \leq Y_{z,\max}^{k+1} - p y_z^k \\ &\quad \% \text{ stability constraints} \\ &\frac{1}{\omega} \frac{1 - \lambda}{1 - \lambda^N} b^T \dot{X}_z^k = x_c^k + \frac{\dot{x}_c^k}{\omega} - x_z^k \\ &\frac{1}{\omega} \frac{1 - \lambda}{1 - \lambda^N} b^T \dot{Y}_z^k = y_c^k + \frac{\dot{y}_c^k}{\omega} - y_z^k \end{aligned}$$

<sup>2</sup>The rationale behind this choice is that we want to generate a persistent gait which can be expected to become periodic. Of course this is true provided that the footstep sequence — which is at this stage assumed to be given — is itself periodic. On the other hand, it is always possible to choose the tail so as to consider that, e.g., the robot is coming to a stop, or moving with an irregular gait.

The above formulation of the ZMP constraint is appropriate for the single support phase, when the ZMP is constrained to be within the support foot. During the double support phase, the two inequality constraints must be replaced by a single one, which is still linear in the control variables  $\dot{X}_z^k, \dot{Y}_z^k$ . We omit the details of this trivial modification.

### C. MPC algorithm

We can now provide a sketch of the MPC algorithm. Input data are the footsteps with the associated timing. This information is preliminarily used to compute the sampled ZMP bounds  $X_{z,\min}^{k+1}, X_{z,\max}^{k+1}, Y_{z,\min}^{k+1}, Y_{z,\max}^{k+1}$  over the entire time span. Alternatively, one may compute at each iteration the bounds within the current prediction horizon. As initialization, we need the starting position and velocity of the CoM, as well as the starting value of the ZMP.

The generic  $k$ -th iteration goes as follows.

- 1) Compute  $\dot{X}_z^k, \dot{Y}_z^k$  that solve the QP problem.
- 2) Extract the first control samples  $\dot{x}_z^k$  and  $\dot{y}_z^k$ , respectively from  $\dot{X}_z^k$  and  $\dot{Y}_z^k$ .
- 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  $x_c(t), \dot{x}_c(t), \dot{x}_z(t)$  for  $t \in [t_k, t_{k+1}]$ . Compute  $y_c(t), \dot{y}_c(t), \dot{y}_z(t)$  similarly.

## IV. COMPARATIVE SIMULATIONS

Before moving to proper gait synthesis by including footstep generation in our framework, we present some preliminary MATLAB simulations on the LIP aimed at highlighting the main features of the proposed approach. In particular, we shall present a comparison between the intrinsically stable MPC scheme proposed in the previous section and the standard MPC of [3]. Recall that the latter uses the CoM jerk as a control variable (and in the cost function) and does not include stability constraints, while the ZMP constraints are the same.

The simulated model is (4) with  $h_{CoM} = 0.26$  m (a value coming from NAO, which we will use as humanoid platform in the next section). A sequence of evenly spaced footsteps is assigned to be attained every 0.3 s, of which 0.1 s in double support and 0.2 s in single support. The foot is represented as a square with a side of 0.02 m, smaller than the actual foot to provide a safety margin. In our MATLAB implementation, we use  $\delta = 0.01$  s and solve QP problems with the *quadprog* function, which uses an interior-point algorithm.

In the first simulation (Fig 2) the prediction horizon is  $T_h = 0.9$  s, i.e., three steps ahead. For this reasonably long horizon, both MPC schemes produce stable (in fact, very similar) CoM trajectories. Results differ more under the viewpoint of ZMP, as the trajectory produced by intrinsically stable MPC is closer to the active constraint, due to the cost function that penalizes ZMP variations. A smoother ZMP can be easily obtained by adding a cost term penalizing the distance of the current ZMP from the center of the foot.

In a second simulation, the prediction horizon is reduced to  $T_h = 0.65$  s. As seen in Fig. 3, the standard MPC produces now asymmetric trajectories for both CoM and ZMP. This means that, in spite of the symmetry of the footstep sequence,

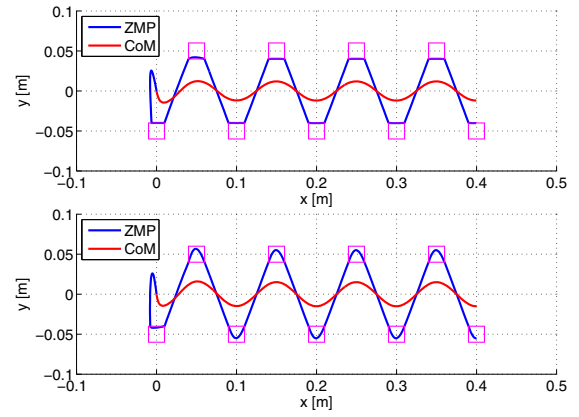


Fig. 2.  $T_h = 0.9$  s: Intrinsically stable (top) vs. standard MPC (bottom).

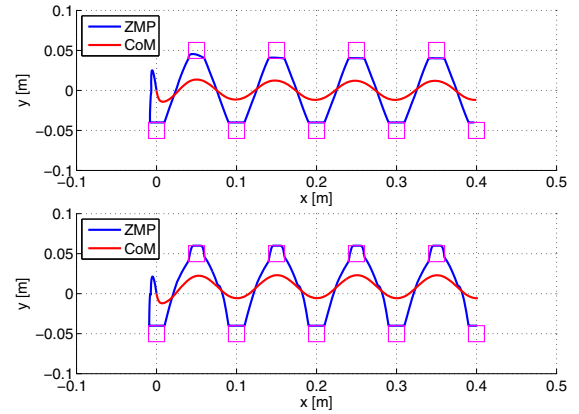


Fig. 3.  $T_h = 0.65$  s: Intrinsically stable (top) vs. standard MPC (bottom).

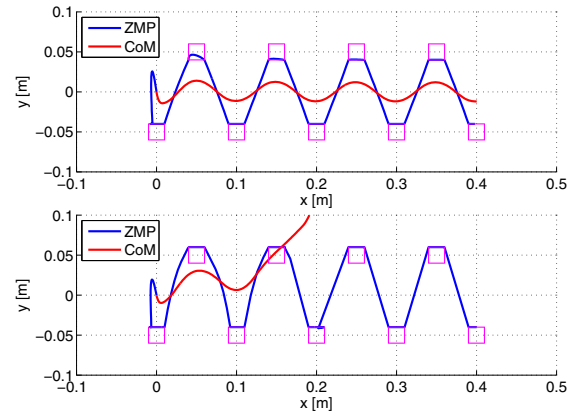


Fig. 4.  $T_h = 0.6$  s: Intrinsically stable (top) vs. standard MPC (bottom).

the LIP motion in the  $y$  direction is biased upwards: the pendulum will incline more when the support foot is on the right and less when it is on the left. By contrast, the intrinsically stable MPC solution is essentially unaffected.

A further reduction of the prediction horizon to 0.6 s (two steps ahead) leads to the results in Fig. 4. The CoM trajectory generated by standard MPC diverges, because  $T_h$  is too short to allow sorting out the stable behavior via jerk minimization. Once again, our MPC scheme produces a bounded, symmetric CoM trajectory, confirming the robustness gained by embedding the stability constraint in the QP formulation.

These comparative simulations are also shown in the video accompanying the paper.

## V. GAIT GENERATION

We now extend the basic framework presented in Sect. III to consider the case in which the footsteps are not assigned. This will lead to an intrinsically stable MPC scheme for gait generation which represents an alternative to the standard approach introduced in [4]. As before, we do not consider footstep rotation to keep the formulas reasonably compact.

### A. Formulation

We assume for simplicity that all steps have the same assigned duration  $T_{\text{step}}$ , and also that  $T_h = S \cdot T_{\text{step}}$ . The extension to more general cases is straightforward.

Under the previous assumptions, in each prediction horizon we need to choose the ZMP velocity samples as well as  $S$  footstep locations, identified by the position of the center of the foot. Denote by  $(x_{f,1}^k, y_{f,1}^k) \dots, (x_{f,S}^k, y_{f,S}^k)$  these locations. The control vector can be defined as

$$U^k = (U_x^k \ U_y^k)^T = (\dot{X}_z^k \ x_{f,1}^k \ \dots \ x_{f,S}^k \ \dot{Y}_z^k \ y_{f,1}^k \ \dots \ y_{f,S}^k)^T.$$

The cost function is modified by including a quadratic term penalizing deviations of the average CoM (horizontal) velocity w.r.t. a reference velocity  $v_{c,\text{ref}}$ :

$$\|\dot{X}_z^k\|^2 + \|\dot{Y}_z^k\|^2 + w \|v_{c,\text{ave}} - v_{c,\text{ref}}\|^2, \quad (15)$$

where  $w$  is a scalar weight,  $v_{c,\text{ave}} = (\dot{x}_{c,\text{ave}} \ \dot{y}_{c,\text{ave}})^T$  and

$$\dot{x}_{c,\text{ave}} = \frac{1}{\nu} \sum_{j=k+1}^{k+\nu} \dot{x}_c^j \quad \dot{y}_{c,\text{ave}} = \frac{1}{\nu} \sum_{j=k+1}^{k+\nu} \dot{y}_c^j$$

are computed by averaging CoM velocities over the first  $\nu$  samples within the prediction horizon.

By integration of the motion model (i.e., (4) and the corresponding equation along  $y$ ) one can easily write the CoM velocity as a function of the ZMP velocity. This leads to an expression of  $v_{c,\text{ave}}$  as a function of  $\dot{X}_z^k$  and  $\dot{Y}_z^k$ , which can in turn be used to rewrite cost function (15) as

$$(U^k)^T Q U^k + f_k^T U^k,$$

where  $Q$  is a square matrix and  $f_k$  is a vector. Both  $Q$  and  $f_k$  are constant over the prediction horizon, and  $f_k$  depends on the current state, i.e., position of the CoM, velocity of the CoM and position of the ZMP at  $t_k$ .

Coming to the constraints, note first that the stability constraints in the QP problem do not involve footsteps and are therefore unchanged. However, the ZMP constraints must be modified. First, we only consider the single support phase, because the double support would lead to nonlinear constraints involving products between control variables. Second, the constraints in single support now involve the footstep locations in addition to the ZMP trajectory. In particular, the ZMP constraints assume the form

$$\begin{aligned} m_{x,1}^k - p x_z^k &\leq M U_x^k \leq m_{x,2}^k - p x_z^k \\ m_{y,1}^k - p y_z^k &\leq M U_y^k \leq m_{y,2}^k - p y_z^k, \end{aligned}$$

where  $m_{x,1}^k, m_{x,2}^k, m_{y,1}^k, m_{y,2}^k$  are vectors and  $M$  is a square matrix. All these quantities are constant over  $T_h$ , and the vectors depend on the location of the support foot at  $t_k$ .

Other constraints can be easily added to this formulation. For example, one may impose that each footstep is contained in a bounding box whose location depends on the previous footstep: this is useful to enforce an upper bound on the step length as well as to avoid collisions between the feet. Such constraint has been included in our implementation but we omit its explicit expression for compactness.

### B. NAO simulations

We now present some V-REP simulations of the NAO humanoid to illustrate the performance of the intrinsically stable MPC scheme for gait generation described above. Due to lack of space, we do not present a comparison with the standard MPC scheme; however, we have confirmed that the performance of the latter is heavily dependent on the duration of the prediction horizon.

In the MPC implementation, we have set  $T_{\text{step}} = 0.3$  s and  $T_h = 0.6$  s, so that  $S = 2$  (see the start of Sect. V-A). There is a 0.05 m bound on the step length, while the useful area of the foot is again a central square with a 0.02 m side. Also, we used  $w = 10^4$  (for the cost function) and  $\nu = 5$  (for computing  $v_{c,\text{ave}}$ ).

Commands for the robot joints are computed by a simple kinematic control law based on pure pseudoinversion that tracks the CoM trajectory produced by the MPC scheme, in addition to the swing foot trajectory (which is exogenous).

In the presented plots, the ZMP trajectory (actual and predicted) is computed by the MPC for the LIP, whereas the CoM is that of the actual NAO humanoid (virtually coincident with the LIP CoM thanks to the kinematic controller).

In the first simulation, NAO is assigned a reference CoM velocity which is constant in modulus (0.08 m/s) but changes in direction halfway. The stable gait generated by our method is shown in Fig. 5. The asymmetry of the ZMP and CoM trajectories during the diagonal tract is due to the fact that the footsteps are not aligned with the path in that phase.

Figure 6 refers to a case in which the reference CoM velocity has constant direction but variable (coast-ramp-coast) modulus. In the generated stable gait the robot increases its speed by taking longer steps within the assigned bounds — consistently with the assumption that all steps have the same duration. The reference speed is tracked quite accurately.

In a final simulation, we test the robustness of the proposed method for gait generation by applying an impulsive lateral force to the robot while it is tracking a constant reference velocity. The push is simulated as an instantaneous jump in the CoM velocity that occurs when NAO is in single support. As shown in Fig. 7, the MPC scheme reacts to the push by automatically computing a stable gait (footsteps, ZMP and CoM) that is adapted to the perturbed state.

The above V-REP simulations are shown in detail in the video accompanying the paper.

## VI. CONCLUSIONS

We have presented a novel MPC method for humanoid gait generation that is guaranteed to produce stable CoM trajectories. This has been obtained by using a dynamic

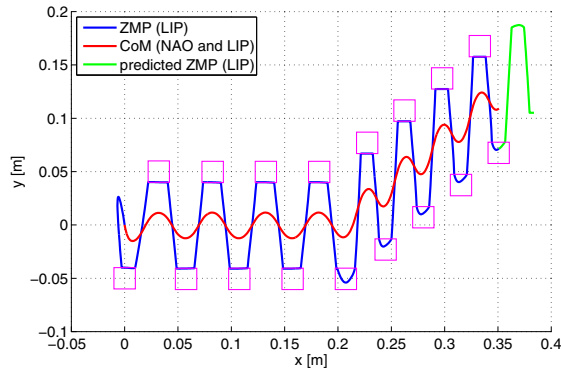


Fig. 5. Gait generation with change of direction: Intrinsically stable MPC.

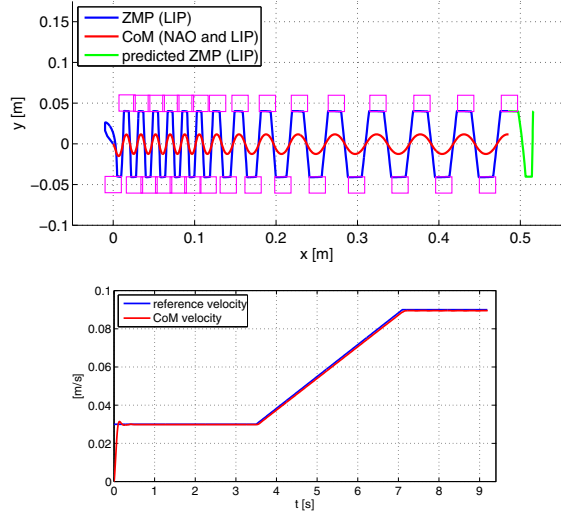


Fig. 6. Gait generation with variable speed: Intrinsically stable MPC (above) and associated CoM velocity (below).

extension of the LIP as motion model, with the ZMP velocity as a control variable, and embedding an explicit stability constraint in the formulation. Such constraint has been shown to be linear in the control variables, leading to a standard QP problem with equality and inequality constraints. The intrinsically stable MPC framework has been developed into a proper gait generation scheme by including automatic footstep generation. Simulations results on the LIP model and a NAO humanoid have shown that the proposed method is quite effective and performs robustly in the presence of changes in the prediction horizon.

Future work will address several points, such as:

- performing an experimental validation of the method;
- using a more complete approximate model which accounts for the swinging foot dynamics and removes the constant CoM height assumption.
- applying the proposed MPC approach to the generation of evasive motions [18].

## REFERENCES

[1] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Springer Publishing Company Inc., 2014.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *2003 IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1620–1626.

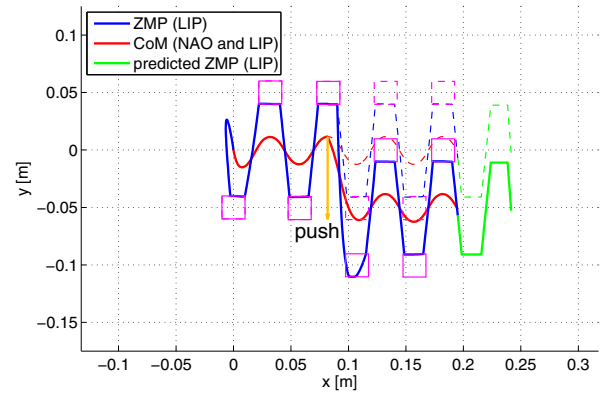


Fig. 7. Gait generation with push perturbation: Intrinsically stable MPC. Note how the prediction before the push (footsteps, CoM and ZMP: dashed) is adapted immediately afterwards to maintain balance.

[3] 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.

[4] 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.

[5] A. Herdt, N. Perrin, and P. B. Wieber, “Walking without thinking about it,” in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 190–195.

[6] A. Ibanez, P. Bidaud, and V. Padois, “Emergence of humanoid walking behaviors from mixed-integer model predictive control,” in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 4014–4021.

[7] 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 *2014 IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 1943–1950.

[8] 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, 2017.

[9] 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.

[10] 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. of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[11] J. Engelsberger, C. Ott, and A. Albu-Schäffer, “Three-dimensional bipedal walking control based on divergent component of motion,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

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

[13] R. J. Griffin and A. Leonessa, “Model predictive control for dynamic footstep adjustment using the divergent component of motion,” in *2016 IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 1763–1768.

[14] P. B. Wieber, “Viability and predictive control for safe locomotion,” in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 1103–1108.

[15] L. Lanari, S. Hutchinson, and L. Marchionni, “Boundedness issues in planning of locomotion trajectories for biped robots,” in *14th IEEE-RAS Int. Conf. on Humanoid Robots*, 2014, pp. 951–958.

[16] L. Lanari and S. Hutchinson, “Inversion-based gait generation for humanoid robots,” in *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 637–642.

[17] B. J. Stephens and C. G. Atkeson, “Push recovery by stepping for humanoid robots with force controlled joints,” in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 52–59.

[18] M. Cognetti, D. De Simone, L. Lanari, and G. Oriolo, “Real-time planning and execution of evasive motions for a humanoid robot,” in *2016 IEEE Int. Conf. on Robotics and Automation*, 2016, pp. 4200–4206.