Walking motion generation with online foot position adaptation based on ℓ_1- and ℓ_∞-norm penalty formulations

Dimitar Dimitrov, Antonio Paolillo, Pierre-Brice Wieber

Abstract— **The article presents an improved formulation of an existing model predictive control scheme used to generate online "stable" walking motions for a humanoid robot. We introduce: (i) a change of variable that simplifies the optimization problem to be solved; (ii) a simply bounded formulation in the case when the positions of the feet are predetermined; (iii) a formulation allowing foot repositioning (when the system is perturbed) based on ℓ_1- and ℓ_∞-norm minimization; (iv) a formulation that accounts for (approximate) double support constraints when foot repositioning occurs.**

I. INTRODUCTION

The realization of "stable" walking motions for a humanoid robot is heavily limited by the *unilateral constraints* [1] between the feet and the ground. When walking on a flat ground, these constraints can be represented by the condition that the Zero Moment Point (ZMP) [2] can only lie within the support polygon. A variety of methods, accounting explicitly for this condition, based on knowledge of the full dynamic characteristics of the system are introduced [3], [4]. Such methods strongly rely on the model accuracy, and usually assume that the precomputed trajectories of the state variables can be executed in a straightforward way in an error-free environment. Another possible approach is for the walking motions to be generated online, based on the use of an approximate dynamical model, where the approximation is compensated by the application of a preview controller with (possibly) fast sampling time. The second alternative is rather attractive because perturbations due to uncertainty of the environment (or feedback errors) can be compensated for, leading to a more robust (to uncertainty) control/planing scheme.

A promising approach based on a linear approximation of the dynamics of a humanoid robot in combination with a Linear Quadratic Regulator (LQR) has been presented in [5]. An important modification (among many others presented since the publication of [5], e.g., [7], [8]) is to account explicitly for the constraints on the ZMP [6], essentially turning the LQR scheme into a more general Linear Model Predictive Control (LMPC) scheme, which led to significant improvement when dealing with perturbations [13]. Additional flexibility, allowing to alter the placement of the feet if the predefined profile could not be followed due to strong perturbations was introduced in [12], and extended in [10], [14] by replacing the requirement of specifying reference feet placement with a reference average speed for the Center of Mass (CoM).

One disadvantage of the LMPC scheme in [10] compared to the LQR scheme in [5], is the necessity to solve a Quadratic Program (QP) with general inequality constraints at each sampling time, hence its successful application to systems with short sampling times is very dependent on the efficiency of forming and solving the underlying QP. There has been a great deal of related research, in the context of humanoid walking [13], [11], [9], and in general [15], [16].

Even though the LMPC scheme discussed above has been successfully tested (to the authors' knowledge) on the HRP-2 and NAO platforms [17], there are still issues that need to be addressed. One of them is that when repositioning of the feet is allowed, the formulation (as presented in [12], [10]) assumes that no sampling time falls strictly in a double support phase, hence double support is not considered at all. This could impose difficulties depending on the walking pattern to be executed. A second limitation is that the variation of the feet (from a given reference position) is penalized only using a quadratic ℓ_2-norm, which leads to changing feet positions even if it is not necessary. As will be discussed in the sequel, for any finite value of this penalty, feet repositioning occurs.

The focus of this article is twofold:

- to introduce a new efficient formulation for the LMPC scheme (discussed above), in the case when the reference profile of the feet cannot change. Through a change of variable, and a sequence of Givens rotations, we are able to arrive at a simply bounded QP, which can be solved much faster compared to a QP with general inequality constraints;
- a modification of the formulation in [12] by computing the change of the positions of the feet (from the predefined reference positions) based on the minimization of quadratic ℓ_2-norm in combination with ℓ_1- or ℓ_∞-norm, which can lead to improved results. Furthermore, an approach for handling double support constraints in case of both fixed and variable feet is introduced.

The article is organized as follows: Section II, contains background and notation related to the LMPC scheme presented in [12], [10]. In Section III, single support and double support constraints for the ZMP (in the case when foot variation is allowed) are presented. Section IV introduces a simply bounded formulation. In Section V we present two formulations (based on the minimization of ℓ_1- and ℓ_∞-norm) for computing optimal foot repositioning. Finally, in Section VI simulation results are presented.

II. BACKGROUND

The ZMP preview control scheme proposed in [5] approximates the dynamics of a humanoid robot with that of a 3D linear inverted pendulum. Such approximation, and with the assumption that the CoM of the system is constrained to move on a horizontal plane with constant height, result in a decoupled set of equations governing the motion of the ZMP.

$$z^x = c^x - \frac{c^z}{g}\ddot{c}^x, \qquad z^y = c^y - \frac{c^z}{g}\ddot{c}^y, \tag{1}$$

where $\boldsymbol{z} = [z^x,\ z^y]^T$ are the coordinates of the ZMP on the flat floor, c^x, c^y and c^z are the coordinates of the CoM (note that the altitude c^z is assumed constant), g is the norm of the acceleration due to gravity (e.g. $g \approx 9.8\ m/s^2$), and a dot over a variable denotes a time derivative.

We will consider trajectories of the CoM with piecewise constant jerks $\dddot{c}^x$, $\dddot{c}^y$ over time intervals of constant length T. Let us denote $\hat{\boldsymbol{c}}_k = \left[\begin{array}{cccccc} c_k^x & \dot{c}_k^x & \ddot{c}_k^x & c_k^y & \dot{c}_k^y & \ddot{c}_k^y \end{array}\right]^T$, and $\boldsymbol{c}_k = \left[\begin{array}{cc} c_k^x & c_k^y \end{array}\right]^T$. Starting from $\hat{\boldsymbol{c}}_k$ and performing trivial integration leads to

$$\hat{\boldsymbol{c}}_{k+1} = \mathbb{A}\hat{\boldsymbol{c}}_k + \mathbb{B}\dddot{\boldsymbol{c}}_k, \tag{2}$$

while equations (1) leads to

$$\boldsymbol{z}_k = \mathbb{C}_z\hat{\boldsymbol{c}}_k, \tag{3}$$

where

$$\mathbb{A} = \left[\begin{array}{cccccc} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

$$\mathbb{B} = \left[\begin{array}{cc} T^3/6 & 0 \\ T^2/2 & 0 \\ T & 0 \\ 0 & T^3/6 \\ 0 & T^2/2 \\ 0 & T \end{array}\right],$$

$$\mathbb{C}_z = \left[\begin{array}{cccccc} 1 & 0 & -\frac{c^z}{g} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{c^z}{g} \end{array}\right].$$

$\hat{\boldsymbol{c}}_k \in \mathbb{R}^6$ represents the state of the system, $\dddot{\boldsymbol{c}}_k \in \mathbb{R}^2$ is the control input, $\boldsymbol{z}_k \in \mathbb{R}^2$ is a vector of measured (or estimated) outputs which are to be controlled (to satisfy given constraints and when possible to follow certain reference profile). In order to express the behavior of (2) and (3) for N discrete intervals in the future as a function of $\hat{\boldsymbol{c}}_k$ and $n = 2N$ control actions, we perform the following recursion

$$\boldsymbol{z}_{k+\tau} = \mathbb{C}_z\mathbb{A}^\tau\hat{\boldsymbol{c}}_k + \mathbb{C}_z\sum_{\rho=0}^{\tau-1}\mathbb{A}^{(\tau-\rho-1)}\mathbb{B}\dddot{\boldsymbol{c}}_{k+\rho}\ ,\ (\tau = 1, ..., N).$$

In a matrix form we have

$$\boldsymbol{Z} = \left[\begin{array}{c} \boldsymbol{z}_{k+1} \\ \vdots \\ \boldsymbol{z}_{k+N} \end{array}\right] = \boldsymbol{P}_{zs}\hat{\boldsymbol{c}}_k + \boldsymbol{P}_{zu}\left[\begin{array}{c} \dddot{\boldsymbol{c}}_k \\ \vdots \\ \dddot{\boldsymbol{c}}_{k+N-1} \end{array}\right] = \boldsymbol{P}_{zs}\hat{\boldsymbol{c}}_k + \boldsymbol{P}_{zu}\boldsymbol{U}. \tag{4}$$

If in the above recursion instead of $\mathbb{C}_z$ one uses

$$\mathbb{C}_p = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right], \text{or}$$

$$\mathbb{C}_v = \left[\begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

one obtains

$$\boldsymbol{C} = \left[\begin{array}{c} \boldsymbol{c}_{k+1} \\ \vdots \\ \boldsymbol{c}_{k+N} \end{array}\right] = \boldsymbol{P}_{ps}\hat{\boldsymbol{c}}_k + \boldsymbol{P}_{pu}\boldsymbol{U}, \tag{5}$$

$$\dot{\boldsymbol{C}} = \left[\begin{array}{c} \dot{\boldsymbol{c}}_{k+1} \\ \vdots \\ \dot{\boldsymbol{c}}_{k+N} \end{array}\right] = \boldsymbol{P}_{vs}\hat{\boldsymbol{c}}_k + \boldsymbol{P}_{vu}\boldsymbol{U}, \tag{6}$$

respectively, where $\boldsymbol{C}$ and $\dot{\boldsymbol{C}}$ represent the evolution of the position and velocity of the CoM in the preview window. The matrices $\boldsymbol{P}_{zs}$, $\boldsymbol{P}_{ps}$, $\boldsymbol{P}_{vs} \in \mathbb{R}^{n\times 6}$ and $\boldsymbol{P}_{zu}$, $\boldsymbol{P}_{pu}$, $\boldsymbol{P}_{vu} \in \mathbb{R}^{n\times n}$ are invertible see [10][1].

The application of a LMPC scheme is based on the minimization of a strictly convex quadratic objective function over a prediction horizon, subject to input and state constraints. We consider them next.

III. Constraints

In contrast to previous approaches, instead of using the jerk ($\boldsymbol{U}$), we consider directly the ZMP ($\boldsymbol{Z}$) as a decision variable. This results in simplified constraints (presented in this section), and is used to obtain a simply-bounded formulation presented in Section IV.

A. Single support constraints on the ZMP

Since the feet of the robot can only push on the ground, the ZMP can only lie within the support polygon (the convex hull of the contact points between the feet and the ground). Let there be m single support steps in the preview window. The condition that $\boldsymbol{z}$ is within the polygon defined by the i^{th} step, can be expressed as a set of linear inequalities of the form

$$\bar{\boldsymbol{D}}_z\boldsymbol{R}_i^T(\boldsymbol{z} - \boldsymbol{F}_i) + \bar{\boldsymbol{d}}_z \geq \boldsymbol{0}, \tag{7}$$

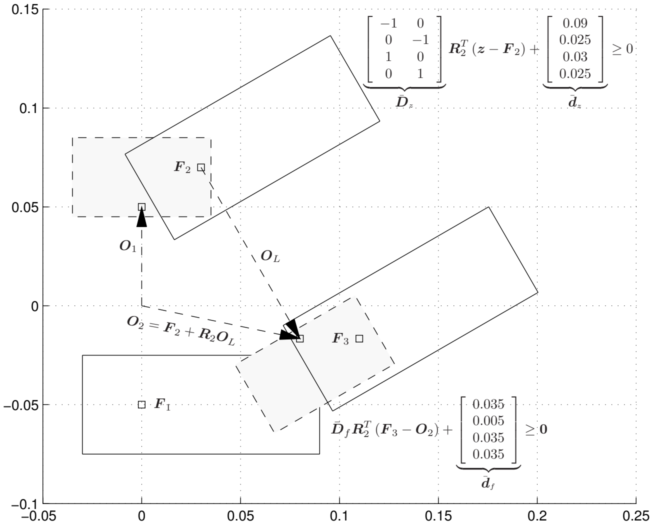
where $\bar{\boldsymbol{D}}_z$ and $\bar{\boldsymbol{d}}_z$ are constant, $\boldsymbol{F}_i \in \mathbb{R}^2$ is the position (of a point of interest, see Fig. 1) of foot i, and $\boldsymbol{R}_i \in \mathbb{R}^{2\times 2}$ is a rotation matrix defining the orientations of the normal vectors in $\bar{\boldsymbol{D}}_z$ (assumed to be normalized). Note that the i^{th} single support step can appear in multiple sampling times in the preview window.

Let us represent the position of the i^{th} footstep as

$$\boldsymbol{F}_i = \boldsymbol{F}_i^{\text{ref}} + \boldsymbol{S}\Delta\boldsymbol{V}_i, \tag{8}$$

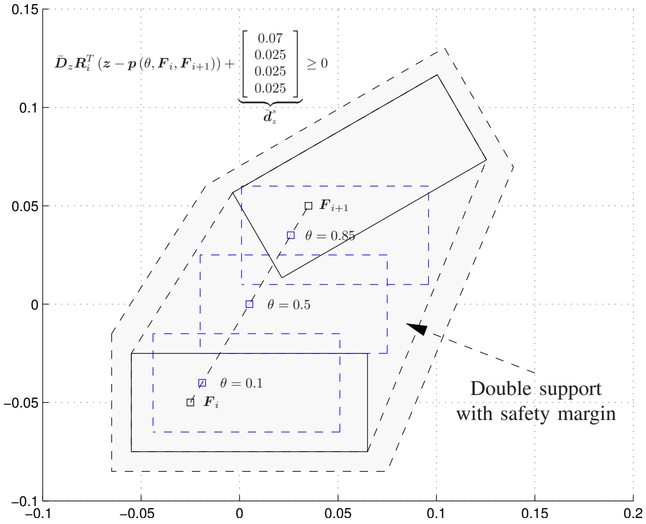
where $\boldsymbol{F}_i^{\text{ref}}$ is a reference position, $\Delta\boldsymbol{F}_i = \boldsymbol{S}\Delta\boldsymbol{V}_i$ is a variation from the reference position, and the matrix $\boldsymbol{S} \in$

[1] Note that in [10] the formulation is given by separating the x and y components of $\boldsymbol{z}$ and $\boldsymbol{c}$, which might be better for the actual implementation. Our notation is adopted for the sake of simplified presentation.



$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\bar{\boldsymbol{D}}_z} \boldsymbol{R}_2^T (\boldsymbol{z} - \boldsymbol{F}_2) + \underbrace{\begin{bmatrix} 0.09 \\ 0.025 \\ 0.03 \\ 0.025 \end{bmatrix}}_{\bar{\boldsymbol{d}}_z} \geq 0$
$\boldsymbol{F}_2$
$\boldsymbol{O}_1$
$\boldsymbol{O}_L$
$\boldsymbol{O}_2 = \boldsymbol{F}_2 + \boldsymbol{R}_2 \boldsymbol{O}_L$
$\boldsymbol{F}_3$
$\boldsymbol{F}_1$
$\bar{\boldsymbol{D}}_f \boldsymbol{R}_2^T (\boldsymbol{F}_3 - \boldsymbol{O}_2) + \underbrace{\begin{bmatrix} 0.035 \\ 0.005 \\ 0.035 \\ 0.035 \end{bmatrix}}_{\bar{\boldsymbol{d}}_f} \geq \boldsymbol{0}$
0.15
0.1
0.05
0
−0.05
−0.1
−0.05
0
0.05
0.1
0.15
0.2
0.25

Fig. 1. Constraints on the ZMP and position of the feet. $\boldsymbol{F}_i \in \mathbb{R}^2$ and $\boldsymbol{R}_i \in \mathbb{R}^{2\times 2}$ define the position (of a point of interest) and orientation of foot i relative to the world frame. $\boldsymbol{O}_L$ is an offset defining the position of (a point of interest of) the constraint for foot $i+1$ relative to foot i (expressed in the local frame fixed in foot i).



$\bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T (\boldsymbol{z} - \boldsymbol{p}(\theta, \boldsymbol{F}_i, \boldsymbol{F}_{i+1})) + \underbrace{\begin{bmatrix} 0.07 \\ 0.025 \\ 0.025 \\ 0.025 \end{bmatrix}}_{\bar{\boldsymbol{d}}_z^s} \geq 0$
$\boldsymbol{F}_{i+1}$
$\theta = 0.85$
$\theta = 0.5$
$\theta = 0.1$
$\boldsymbol{F}_i$
Double support
with safety margin
0.15
0.1
0.05
0
−0.05
−0.1
−0.1
−0.05
0
0.05
0.1
0.15
0.2

Fig. 2. Approximation of the constraints on the ZMP when in double support. The polygon depicted in gray represents the double support constraints without safety margin. The approximating constraints (in blue dashed line) have tighter safety margin compared to the normal single support constraints.

$\mathbb{R}^{2\times s}$ contains directions in which $\Delta \boldsymbol{V}_i \in \mathbb{R}^s$ is resolved ($s \geq 2$ is assumed). A "standard choice" for $\boldsymbol{S}$ would be the identity matrix $\boldsymbol{I}$, in which case $\Delta \boldsymbol{F}_i = \Delta \boldsymbol{V}_i$ (the reasoning for the case with $s > 2$ will become evident in Section V). Substituting (8) in (7) leads to

$$\bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \Delta \boldsymbol{V}_i \end{bmatrix} + \bar{\boldsymbol{d}}_z - \bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \boldsymbol{F}_i^{\text{ref}} \geq \boldsymbol{0}. \quad (9)$$

In what follows, footstep i will be referred to as *fixed* if $\Delta \boldsymbol{V}_i = \boldsymbol{0}$, and *variable*, if $\Delta \boldsymbol{V}_i$ could be different from zero (*i.e.*, there is a possibility for it to be repositioned). Let $\mathcal{V}$ be the set of *variable footsteps*, then determining an optimal (according to a given objective function) repositioning of the *variable footsteps*, will be done by adding $\Delta \boldsymbol{V}_i$, $(i \in \mathcal{V})$ to the decision variables of the optimization problem (see Section V).

B. Double support constraints

Let us consider Fig. 2, where two constraints corresponding to the i^{th} and $(i+1)^{\text{th}}$ single support step, and their convex hull are depicted. The number of facets of the double support constraint depends on the orientation and position of the two feet and ranges from four to seven. It is possible to approximate these constraints by using a number of rectangular constraints (depicted using blue dashed line in Fig. 2). Let a point $\boldsymbol{p}$ on the line segment between $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$ be defined as

$$\boldsymbol{p}(\theta, \boldsymbol{F}_i, \boldsymbol{F}_{i+1}) = \theta \boldsymbol{F}_{i+1} + (1-\theta)\boldsymbol{F}_i, \; (0 \leq \theta \leq 1). \quad (10)$$

We define the following constraint

$$\bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T (\boldsymbol{z} - \boldsymbol{p}(\theta, \boldsymbol{F}_i, \boldsymbol{F}_{i+1})) + \bar{\boldsymbol{d}}_z^s \geq \boldsymbol{0}, \quad (11)$$

where $\bar{\boldsymbol{d}}_z^s$ imposes a slightly tighter "safety margin" compared to $\bar{\boldsymbol{d}}_z$ (see Fig. 2). Note that the above constraint is defined to have the same orientation as foot i (alternatively the orientation of foot $i+1$ could be used).

1) Case **F-F***:* Both $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$ are fixed.

F-F double support implies that the preview window starts with a double support (and both feet in support are considered to be fixed). In this case, one could directly compute the real double support constraint, however, here we present two alternatives. From (11) we obtain

$$\bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \begin{bmatrix} \boldsymbol{I} & (\boldsymbol{F}_i - \boldsymbol{F}_{i+1}) \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \theta \end{bmatrix} + \bar{\boldsymbol{d}}_z^s - \bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \boldsymbol{F}_i \geq \boldsymbol{0}.$$

From Fig. 2 we see that if $\theta = 1$ is used, the approximation would leave the original double support constraint, hence we use the following heuristic constraint $0 \leq \theta \leq 0.85$, which seems to be "reasonable" when the relative orientation between two successive footsteps is not more than 30 deg. Note that the above equation implies that θ is considered as a variable of the QP.

Avoiding the heuristic constraint for θ and the tighter safety margin (from above), can be done by introducing, instead of one, two variables θ_1 and θ_2. Where, the former "moves" the i^{th} polygon, and the latter "moves" the $(i+1)^{\text{th}}$ polygon along the line segment connecting $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$. With this setting, the constraint for the ZMP can be defined as the intersection of the two polygons, and then both θ_1 and θ_2 could be swept safely between 0 and 1.

2) Case **V-V***:* Both $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$ are variable.

In this case, depending on the desired profile for the ZMP during a double support, a set of values for θ can be predefined to obtain

$$\bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \begin{bmatrix} \boldsymbol{I} & -(1-\theta)\boldsymbol{S} & -\theta \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \Delta \boldsymbol{V}_i \\ \Delta \boldsymbol{V}_{i+1} \end{bmatrix} + \bar{\boldsymbol{d}}_z^s \geq \boldsymbol{q}_z, \quad (12)$$

where $\boldsymbol{q}_z = \bar{\boldsymbol{D}}_z \boldsymbol{R}_i^T \left(\theta \boldsymbol{F}_{i+1}^{\text{ref}} - \theta \boldsymbol{F}_i^{\text{ref}} + \boldsymbol{F}_i^{\text{ref}}\right)$. For example, in Fig. 2 there are three sampling times strictly falling in double support, and three values for θ are defined.

The case when the i^{th} footstep is fixed and the $(i+1)^{\text{th}}$ step is variable (denoted by **F-V**), is handled by simply assuming that $\Delta \boldsymbol{V}_i = \boldsymbol{0}$ in (12).

C. Constraints on the foot placement

In order to assure that the positions of the footsteps are feasible with respect to joint limits, self-collision, and other similar geometric limitations, we use "safety zone" constraints for the positions of the feet. This approach is adopted from [14] (where a polygonal approximation of the "safety zone" constraints for HRP-2 are presented). In general they can be expressed as (see Fig. 1)

$$\bar{\boldsymbol{D}}_f \boldsymbol{R}_i^T \left[\begin{array}{cc} -\boldsymbol{S} & \boldsymbol{S} \end{array} \right] \left[\begin{array}{c} \Delta \boldsymbol{V}_i \\ \Delta \boldsymbol{V}_{i+1} \end{array} \right] + \bar{\boldsymbol{d}}_f - \boldsymbol{q}_f \geq \boldsymbol{0}, \qquad (13)$$

where $\boldsymbol{q}_f = \bar{\boldsymbol{D}}_f \boldsymbol{O}_L + \bar{\boldsymbol{D}}_f \boldsymbol{R}_i^T (\boldsymbol{F}_i^{\text{ref}} - \boldsymbol{F}_{i+1}^{\text{ref}})$. $\bar{\boldsymbol{D}}_f$, $\bar{\boldsymbol{d}}_f$ and $\boldsymbol{O}_L$ are constant and define the orientation, size and positioning of the polygonal approximation of the "safety-zone".

In certain cases, imposing additional constraints might be desirable, e.g., a heuristic constraint that accounts for maximum joint speed is presented in [12].

IV. Simply bounded formulation

In this section we make two assumptions: (i) all footsteps are fixed (*i.e.*, $\Delta \boldsymbol{V}_i = \boldsymbol{0}, \forall i$); (ii) all constraints for the ZMP in the preview window are rectangular (double support constraints are approximated using equation (12), with fixed $\boldsymbol{F}_i$, $\boldsymbol{F}_{i+1}$). With these two assumptions, it is possible to derive a simply bounded formulation.

We consider the minimization of the following objective function (in Section V it will be modified when variable foot position is discussed).

$$\underset{\boldsymbol{Z} \in \mathbb{R}^n}{\text{minimize}} \ \frac{\gamma}{2}\|\boldsymbol{U}\|^2 + \frac{\alpha}{2}\|\dot{\boldsymbol{C}}\|^2 + \frac{\beta}{2}\|\boldsymbol{Z} - \boldsymbol{Z}^{\text{ref}}\|^2 = \frac{1}{2}\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z} + \boldsymbol{Z}^T \boldsymbol{g}, \qquad (14)$$

where α, β, $\gamma > 0$ are gains, $\boldsymbol{Z}^{\text{ref}}$ is a reference profile for $\boldsymbol{Z}$, and

$$\begin{aligned} \boldsymbol{H} &= \gamma \boldsymbol{P}_{zu}^{-T} \boldsymbol{P}_{zu}^{-1} + \alpha \boldsymbol{P}_{zu}^{-T} \boldsymbol{P}_{vu}^T \boldsymbol{P}_{vu} \boldsymbol{P}_{zu}^{-1} + \beta \boldsymbol{I}, \\ \boldsymbol{g} &= \boldsymbol{P}_c \hat{\boldsymbol{c}}_k - \beta \boldsymbol{Z}^{\text{ref}}, \\ \boldsymbol{P}_c &= \alpha \boldsymbol{P}_{zu}^{-T} \boldsymbol{P}_{vu}^T \left(\boldsymbol{P}_{vs} - \boldsymbol{P}_{vu} \boldsymbol{P}_{zu}^{-1} \boldsymbol{P}_{zs} \right) - \gamma \boldsymbol{P}_{zu}^{-T} \boldsymbol{P}_{zu}^{-1} \boldsymbol{P}_{zs}. \end{aligned}$$

Let $\boldsymbol{D}_z \in \mathbb{R}^{4N \times n}$ be a constant block diagonal matrix, and let each of its 4×2 blocks be equal to $\bar{\boldsymbol{D}}_z$ (as given in Fig. 1). Let $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ be a block diagonal matrix (containing on its diagonal the rotation matrices $\boldsymbol{R}_i$) and $\boldsymbol{d} \in \mathbb{R}^{4N}$ be a vector, both varying from one preview window to the next. The k^{th} (2×2) block of $\boldsymbol{R}$, and 4×1 part of $\boldsymbol{d}$ are formed by using either equation (9) or (12), depending on whether the k^{th} sampling time of the preview window is a single or double support, respectively. Then, the constraints for the position of the ZMP in the whole preview window can be written as

$$\underbrace{\boldsymbol{D}_z \boldsymbol{R}^T}_{\boldsymbol{D}} \boldsymbol{Z} + \boldsymbol{d} \geq \boldsymbol{0}. \qquad (15)$$

Note that the matrix $\boldsymbol{D}$ contains at most two nonzero entries in each row and can be formed very efficiently. If instead of $\boldsymbol{Z}$, one uses $\boldsymbol{U}$ as an optimization variable (as done in [12], [10] for example), the leading matrix of the constraints is given by $\boldsymbol{D}\boldsymbol{P}_{zu}$ (which is dense).

By considering (14), (15) and a change of variable $\boldsymbol{Q} = \boldsymbol{R}^T \boldsymbol{Z}$ we obtain

$$\begin{aligned} &\underset{\boldsymbol{Q} \in \mathbb{R}^n}{\text{minimize}} \ \frac{1}{2}\boldsymbol{Q}^T \boldsymbol{R}^T \boldsymbol{H} \boldsymbol{R} \boldsymbol{Q} + \boldsymbol{Q}^T \boldsymbol{R}^T \boldsymbol{g} \qquad (16) \\ &\text{subject to} \ \ \boldsymbol{D}_z \boldsymbol{Q} + \boldsymbol{d} \geq \boldsymbol{0}. \end{aligned}$$

Since $\boldsymbol{D}_z$ is constant and contains on its main diagonal $\bar{\boldsymbol{D}}_z$, the above constraint can be rewritten simply as $\underline{\boldsymbol{d}} \leq \boldsymbol{Q} \leq \overline{\boldsymbol{d}}$, where $\underline{\boldsymbol{d}}$ and $\overline{\boldsymbol{d}}$ stand for lower and upper bound, respectively. The following example shows the relation between $\underline{\boldsymbol{d}}, \overline{\boldsymbol{d}}$ and $\boldsymbol{d}$. Let us consider only the first four constraints from above

$$\left[\begin{array}{cc} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{array} \right] \left[\begin{array}{c} Q_{x_1} \\ Q_{y_1} \end{array} \right] + \left[\begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \end{array} \right] \geq \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right]$$

which is equivalent to

$$\begin{aligned} -d_3 \leq Q_{x_1} \leq d_1 \\ -d_4 \leq Q_{y_1} \leq d_2. \end{aligned}$$

Note that only parts of $\boldsymbol{R}$ may vary from one preview window to the next (and due to the structure of both $\boldsymbol{H}$ and $\boldsymbol{R}$), the update of the Hessian matrix $\boldsymbol{R}^T \boldsymbol{H} \boldsymbol{R}$ (or directly its inverse, or its Cholesky factors) can be done very efficiently (with much less effort than one typical iteration of an active set method). By using a dedicated algorithm (e.g. see [18]), QP (16) can be solved much more efficiently than a QP with general constraints.

V. Optimal foot repositioning

A general limitation of the scheme presented in Section IV, is that the feet are assumed to be fixed. In the presence of strong perturbations, the ability to adapt their positions online adds additional flexibility.

Let us define a matrix $\boldsymbol{T} \in \mathbb{R}^{n \times (s m_v)}$ as

$$\boldsymbol{T} = \left[\begin{array}{ccc} \boldsymbol{T}_{1,1} & \dots & \boldsymbol{T}_{1,m_v} \\ \vdots & \ddots & \vdots \\ \boldsymbol{T}_{N,1} & \dots & \boldsymbol{T}_{N,m_v} \end{array} \right], \qquad (17)$$

where $\boldsymbol{T}_{k,i} \in \mathbb{R}^{2 \times s}$ shows the way the i^{th} footstep appear in the k^{th} sampling period (denoted by t_k), and $m_v = |\mathcal{V}|$ is the number of variable feet in a preview window.

$$\boldsymbol{T}_{k,i} = \begin{cases} \boldsymbol{0} & \text{if step } i \text{ does not appear during } t_k \\ \boldsymbol{S} & \text{if step } i \text{ is a single support during } t_k \\ (1-\theta)\boldsymbol{S} & \text{if step } i \text{ is a first step in a } \textbf{V-V} \text{ DS} \\ \theta \boldsymbol{S} & \text{or second step in a } \textbf{F-V} \text{ or } \textbf{V-V} \text{ DS} \end{cases}$$

Above we use DS as a shorthand for double support. We will assume that a **F-F** double support constraint is formed using a four-edge approximation (this assumption is made only to simplify the notation, however, in an actual implementation forming the real double support constraint is readily possible). Next, we outline the formulation given in [12], and then enhance it (for simplicity of presentation, we omit the constraints in (13)).

A. Quadratic ℓ_2-norm penalty

Let $\Delta\boldsymbol{V} \in \mathbb{R}^{sm_v}$ be a vector containing $\Delta\boldsymbol{V}_i$, $(i \in \mathcal{V})$. Using $\boldsymbol{S} = \boldsymbol{I}$ (hence $\Delta\boldsymbol{F} = \Delta\boldsymbol{V}$), the following QP can be used to perform foot adaptation online.

$$\begin{aligned} \underset{\boldsymbol{Z}\in\mathbb{R}^n,\, \Delta\boldsymbol{F}\in\mathbb{R}^{2m_v}}{\text{minimize}} \ & \frac{1}{2}\begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{F} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \mu\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{F} \end{bmatrix} + \\ & \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{F} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} \\ \text{subject to} \ & \boldsymbol{D} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{F} \end{bmatrix} + \boldsymbol{d} \geq \boldsymbol{0}, \end{aligned} \tag{18}$$

where $\mu > 0$ is a gain. The difference between the objective functions in (14) and (18) is the term $\frac{1}{2}\mu\Delta\boldsymbol{F}^T\Delta\boldsymbol{F}$, which is used to penalize the footstep variation. If the reference positions of the feet are considered as constraints, then in the limit $\mu \to \infty$ the *hard*-constrained problem is recovered (*i.e.* the footsteps are fixed). By using a finite gain μ, one can *soften* the constraints for the foot positions.

The drawback of the above formulation comes from the fact that for all finite values of μ the constraints will be violated to some extent, *i.e.*, foot variation will occur even if it is not necessary [20]. This is observed in [12], where even without a disturbance, the feet tend to move "towards the inside". One could argue that for very large values of μ, the foot variation would be negligible, however this might in turn affect in an undesirable way the response of the system when there is a disturbance.

B. Alternative ways for penalization of the variation

An important question is due: *how do we measure the magnitude of the foot variation*? The magnitude of a vector $\boldsymbol{v}$ is related to the choice of different ℓ_p-norm. In the field of predictive control it is common to use either ℓ_1-norm (defined as the sum of absolute values of the elements of $\boldsymbol{v}$), quadratic ℓ_2-norm (defined as $\boldsymbol{v}^T\boldsymbol{v}$), and ℓ_∞-norm (defined as the element of $\boldsymbol{v}$ with largest absolute value).

Penalizing different norms in the objective function, can result in solutions with completely different properties. For example (as discussed in [21], pp. 304), minimizing the ℓ_1-norm tends to produce solutions with large number of components equal to zero (sometimes referred to as *sparse solutions*). This property is heavily used in many applications, for example: compressed sensing; approximate solutions of cardinality problems; robust (to outliers, or noise) estimation in statistics; sparse design; sparse signal reconstruction, etc.. Some of the common formulations used in practice are that of: *robust estimator*, *basis pursuit*, *Chebyshev approximation problem* (or *minimax approximation problem*), the latter one being based on the minimization of ℓ_∞-norm (e.g., see [22]). In the context of nonlinear programming (and MPC), the minimization of ℓ_1- and ℓ_∞-norm are commonly used in order to produce *exact penalization* (see [23], Section 12.3), or impose *soft constraints* [20], pp. 97. Next, we discuss two alternatives for penalizing the foot variation.

C. Quadratic ℓ_2- plus ℓ_1-norm penalty

Here, we present a formulation based on the following penalty function

$$\begin{aligned} f(\boldsymbol{v}, \mu, \xi) &= \frac{\mu}{2}\boldsymbol{v}^T\boldsymbol{v} + \xi\boldsymbol{v}^T\boldsymbol{1} \\ \boldsymbol{v} &\geq 0, \end{aligned} \tag{19}$$

where $\boldsymbol{1}$ is a vector of ones (with appropriate dimensions), and $\xi \geq 0$ is a gain. Equation (19) can be viewed as a weighted sum of the quadratic ℓ_2- and ℓ_1-norm of $\boldsymbol{v}$. In formulation (18) the penalty function used is $f(\Delta\boldsymbol{F}, \mu, 0)$. Note that simply adding $\xi\Delta\boldsymbol{F}^T\boldsymbol{1}$ (with $\xi \neq 0$) to the objective function of (18) is not possible because in general the entries of $\Delta\boldsymbol{F}$ could be negative. In order to be able to use $\xi \neq 0$ we reformulate (18) so that the decision variables corresponding to the foot placement are nonnegative. One way of achieving this is by using four directions in $\boldsymbol{S}$ along the $\boldsymbol{x}$, $\boldsymbol{y}$, $-\boldsymbol{x}$ and $-\boldsymbol{y}$ axes (*i.e.*, $\boldsymbol{S} \in \mathbb{R}^{2\times 4}$) leading to

$$\begin{aligned} \underset{\boldsymbol{Z}\in\mathbb{R}^n,\, \Delta\boldsymbol{V}\in\mathbb{R}^{sm_v}}{\text{minimize}} \ & \frac{1}{2}\begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{V} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0} & \mu\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{V} \end{bmatrix} + \\ & \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{V} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{g} \\ \xi\boldsymbol{1} \end{bmatrix} \\ \text{subject to} \ & \boldsymbol{D} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z} \\ \Delta\boldsymbol{V} \end{bmatrix} + \boldsymbol{d} \geq \boldsymbol{0} \\ & \Delta\boldsymbol{V} \geq \boldsymbol{0}. \end{aligned} \tag{20}$$

Using $\mu = 0$ and $\xi = 1$ would result in the minimization of $\|\Delta\boldsymbol{V}\|_1$. The above formulation is very general, as by changing the number of directions in the matrix $\boldsymbol{S}$, one can "shape-up" alternative norms to be minimized, hence adding additional flexibility to the design and tuning of the LMPC scheme.

With this formulation $s \geq 3$ is required, as at least three vectors are needed in order to positively span the plane (follows from the *Caratheodory*' theorem).

D. Quadratic ℓ_2- plus ℓ_∞-norm penalty

Here, we present an equally general formulation as (20), based on the minimization of ℓ_∞-norm that uses only one additional variable. This comes at the expense of increasing the number of constraints. Below we use the observation that $|x|$ is the smallest number w that satisfies $x \leq w$ and $x \geq -w$, or in a more compact notation $-w \leq x \leq w$. Let $\boldsymbol{S} = \boldsymbol{I}$, then if $\mu = 0$, $\nu = 0$ and $\xi = 1$ are chosen, the following formulation minimizes $\|\Delta \boldsymbol{F}\|_\infty$.

$$\underset{\boldsymbol{Z}\in\mathbb{R}^n,\,\Delta\boldsymbol{V}\in\mathbb{R}^{sm_v},w}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix}\boldsymbol{Z}\\ \Delta\boldsymbol{V}\\ w\end{bmatrix}^T\begin{bmatrix}\boldsymbol{H} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \mu\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \nu\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}\\ \Delta\boldsymbol{V}\\ w\end{bmatrix} + \begin{bmatrix}\boldsymbol{Z}\\ \Delta\boldsymbol{V}\\ w\end{bmatrix}^T\begin{bmatrix}\boldsymbol{g}\\ \boldsymbol{0}\\ \xi\end{bmatrix} \qquad (21)$$

$$\text{subject to}\;\; \boldsymbol{D}\begin{bmatrix}\boldsymbol{I} & -\boldsymbol{T} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}\\ \Delta\boldsymbol{V}\\ w\end{bmatrix} + \boldsymbol{d} \geq \boldsymbol{0}$$

$$-w\boldsymbol{1} \leq \Delta\boldsymbol{V} \leq w\boldsymbol{1}.$$

The above constraints could be expressed as

$$\begin{bmatrix}\boldsymbol{D} & -\boldsymbol{DT} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{1}\\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{1}\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}\\ \Delta\boldsymbol{V}\\ w\end{bmatrix} + \begin{bmatrix}\boldsymbol{d}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} \geq \boldsymbol{0},$$

which explicitly shows the additional constraints. Note that formulation (21) requires only $s \geq 2$. Again (as in (20)), by increasing the number of directions in the matrix $\boldsymbol{S}$, one can "shape-up" alternative norms to be minimized.

E. Some remarks

Choosing "appropriate" values for the two gains μ and ξ in the formulations in Sections V-D and V-C, depends on the particular application and type of robot. An interesting theoretical result presented in [23], Section 12.3 is that, given a set of disturbances for which we do not want foot repositioning to occur, a lower bound on how "large" ξ should be (in order to achieve that), can be computed as the maximum dual norm of the Lagrange multipliers in the solution of the *hard*-constrained problem ($i.e.$, when the foot positions are fixed). The computation of this bound, however, is a very complex and time-consuming task (one possible solution in discussed in [19]), furthermore it depends on the magnitude (and direction) of the disturbances acting on the system. Hence, it is very common in practice to simply heuristically assign a constant positive value for ξ.

A remark, regarding double support constraints is due. One could argue that fixing values for θ, could be restrictive in the case of perturbations, however, this is performed only when a double support includes a variable step ($i.e.$ in the future). When the system is actually in double support one could either form the real double support constraint, or use the alternatives presented in Section III-B.1. Hence, by being slightly restrictive regarding the behavior of the system in the future, we can apply the LMPC scheme to many different walking patterns, and relax the assumption that the sampling time has to be equal to the double support time (see [12], [14], [10]).

VI. Simulation Results

Here, we present the results from a numerical simulation when using formulations (18), (20), and (21) to perform online walking motion generation for a humanoid robot. The envisioned target platform is the humanoid robot NAO. We use the following parameters: $g = 9.8$, $c_z = 0.26$, $N = 15$, $T = 0.1$, $\alpha = 150$, $\beta = 2000$, $\gamma = 1$. The number of iterations in single support and double support is 4 and 2, respectively. The "point of interest" for the single support constraints is chosen to be the projection of the ankle of the robot of the flat floor. For presentation purposes (due to the fact that the maximum distance between two successive footsteps of NAO is smaller than the size of the feet), we use $\bar{\boldsymbol{d}}_z = \begin{bmatrix} 0.04 & 0.015 & 0.02 & 0.015 \end{bmatrix}^T$.

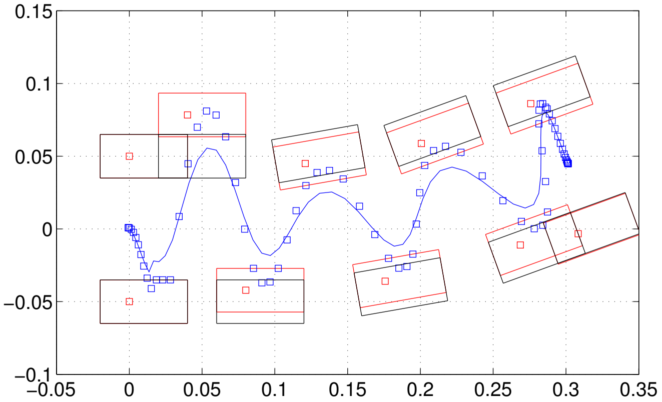


Fig. 3. Motion generation when using formulation (18) with $\mu = 2000000$.

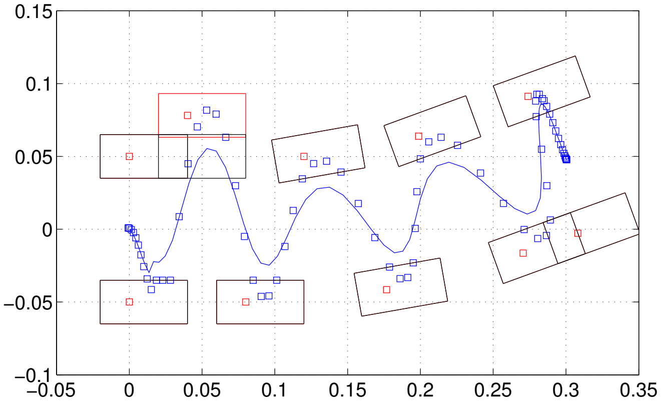


Fig. 4. Motion generation when using formulation (20) with $\mu = 1000$, $\xi = 4000$, $\boldsymbol{S}$ contains four vectors, namely $\boldsymbol{x}$, $\boldsymbol{y}$, $-\boldsymbol{x}$, $-\boldsymbol{y}$.

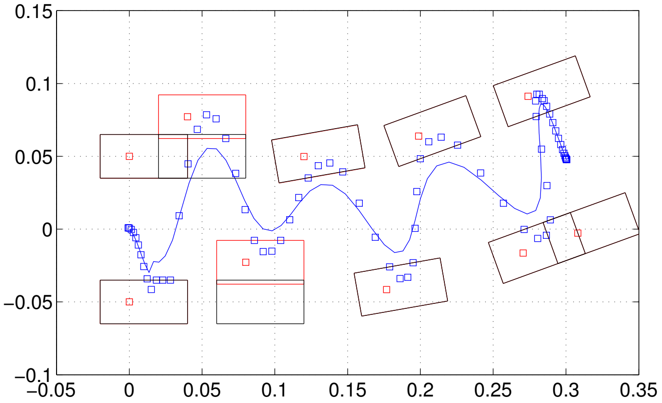


Fig. 5. Motion generation when using formulation (21) with $\mu = 1000$, $\xi = 4000$, $\nu = 0.001$, $\boldsymbol{S} = \boldsymbol{I}$.

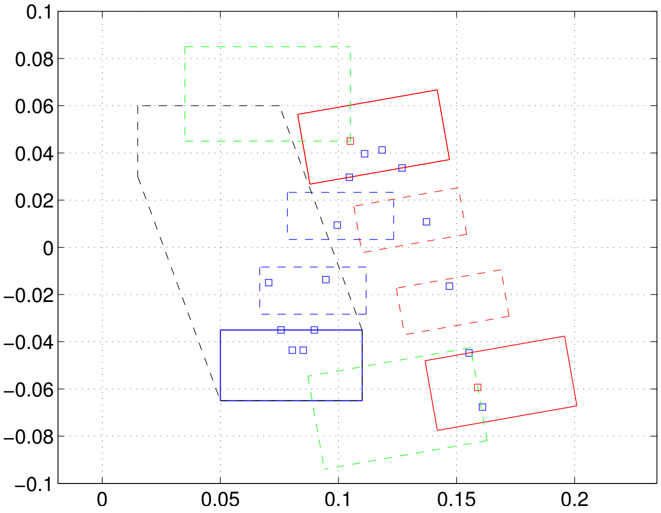


Fig. 6. Constraints appearing in a typical preview window.

Fig. 3, 4, 5 present the results when using formulations (18), (20), (21), respectively (double support constraints are not plotted for clarity). During the first (right) single support the system is disturbed (by a force acting in the direction of the positive y axis). Blue squares and blue line represent the profile of the ZMP and CoM, respectively. The constraints corresponding to the reference footsteps are depicted in black, while the constraints corresponding to the actual footsteps (generated online) are depicted in red rectangles (their positions are depicted with red squares). From Fig. 3 it is evident that the feet tend to move "towards the inside" even long after the disturbance. Even in the absence of a disturbance, similar result is obtained. In Fig. 4, 5 it can be observed that after the initial footstep repositioning (due to the disturbance), the system converges to the reference footsteps and follows them exactly.

Fig. 6 depicts the constraints appearing in a typical preview window. The blue rectangle is a fixed single support, while red rectangles correspond to variable feet. According to the cases in Section III, the rectangles depicted with a red dashed line correspond to a **V-V** double support, the rectangles depicted with a blue dashed line correspond to a **F-V** double support, while the polygon depicted in a dashed black like corresponds to a **F-F** double support, which is given as the convex hull of the feet currently in support (the two alternatives in III-B.1 could have been used instead). The rectangles depicted with a dashed green line correspond to foot constraints (13).

VII. Conclusions

The article presented a formulation for generating online "stable" walking motions for a humanoid robot. There are four contributions: (i) we work directly with the ZMP as a decision variable (instead of using the jerk, as done so far), which leads to very simple form of the constraint, hence resulting in a more efficient formulation; (ii) we introduced a simply bounded formulation in the case when the positions of the feet are predetermined; (iii) we used a more general penalty function in the LMPC formulation when foot adaptation is performed online, hence adding more flexibility to the design and tuning of the LMPC scheme (exact penalization for a given set of disturbances can be achieved); (iv) we presented a way to approximate the double support constraints, so that they could be considered even in the case when foot adaptation is performed online, leading to more flexibility when applying the LMPC scheme to different types of walking patters.

References

[1] B. Brogliato, "Nonsmooth Impact Mechanics," Springer-Verlag, 1999.

[2] M. Vukobratovic, and J. Stepanenko, "On the stability of anthromorphic systems," Mathematical Biosciences, 15:137, 1972.

[3] K. Hirai, M. Hirose, Y. Haikawa, and T. Takaneka, "The development of Honda humanoid robot," in *Proc. of the IEEE Int. Conf. on Robot. & Automat.,* pp. 1321-1326, 1998.

[4] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Trans. on Robotics and Automation,* Vol.17, No.3, June, pp. 280-289, 2001.

[5] 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 *Proc. of the IEEE Int. Conf. on Robot. & Automat.,* pp. 1620-1626, 2003.

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

[7] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," *International Journal of Humanoid Robotics,* 18(8) pp. 816-830, 2008.

[8] K. Nishiwaki, and S. Kagami, "Online walking control systems for humanoids with short cycle pattern generation," *International Journal on Robotics Research,* 28(6) pp. 729-742, 2009.

[9] D. Dimitrov, P-B.Wieber, O. Stasse, H.J. Ferreau, and H. Diedam, "An Optimized Linear Model Predictive Control Solver," in *Recent Trends in Optimization and its Applications in Engineering, Springer Verlag*.

[10] A. Herdt, H. Diedam, P-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," in *Advanced Robotics Vol. 24 No. 5-6,* April, 2010.

[11] D. Dimitrov, P-B. Wieber, O. Stasse, H. J. Ferreau, and H. Diedam, "An Optimized Linear Model Predictive Control Solver for Online Walking Motion Generation," in *Proc. of the IEEE Int. Conf. on Robot. & Automat.,* 2009.

[12] H. Diedam, D. Dimitrov, P.-B. Wieber, M. Katja, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *Proc. of the IEEE/RSJ (IROS),* pp. 1121-1126, 2008.

[13] D. Dimitrov, J. Ferreau, P.-B. Wieber, and M. Diehl, "On the implementation of model predictive control for on-line walking pattern generation," in *Proc. of the IEEE Int. Conf. on Robot. & Automat.,* pp. 2685-2690, 2008.

[14] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," *Proc. of the IEEE/RSJ (IROS),* 2010.

[15] S. Wright, "Applying new optimization algorithms to model predictive control," in *Proc. of CPC-V*, 1996.

[16] R. A. Bartlett, A. Wächter, and L. T. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proc. of the American Control Conference,* pp.4229-4233, June, 2000.

[17] D. Gouaillier, C. Collette, and C. Kilner, "Omni-directional Closed-loop Walk for NAO," *IEEE-RAS International Conference on Humanoid Robots,* pp. 448-454, 2010.

[18] J. Moré, and G. Toraldo, "Algorithms for Bound Constrained Quadratic Programming Problems," in *Numer. Math.,* 55, pp.377-400, 1989.

[19] E. Kerrigan, and J. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," *in Proc. UKACC International Conference,* 2000.

[20] J. Maciejowski, "Predictive Control with Constraints," 2002.

[21] S. Boyd, and L. Vandenberghe, "Convex Optimization," Cambridge, 2004.

[22] http://www.acm.caltech.edu/l1magic

[23] R. Fletcher, "Practical Methods of Optimization," Wiley, 1981.