Online Walking Motion Generation with Automatic Foot Step Placement

Andrei Herdt[1] Holger Diedam[2] Pierre-Brice Wieber[1] Dimitar Dimitrov[1]
Katja Mombaur[2] Moritz Diehl[3]

(1) INRIA & CNRS/AIST JRL, 305-8568 Tsukuba, Japan
(2) University of Heidelberg, 69120 Heidelberg, Germany
(3) Katholieke Universiteit Leuven, 3001 Leuven-Heverlee, Belgium

Abstract

The goal of this paper is to demonstrate the capacity of Model Predictive Control to generate stable walking motions without the use of predefined foot steps. Building up on well-known Model Predictive Control schemes for walking motion generation, we show that a minimal modification of these schemes allows designing an online walking motion generator which can track a given reference speed of the robot and decide automatically the foot step placement. Simulation results are proposed on the HRP-2 humanoid robot, showing a significant improvement over previous approaches.

Keywords: Walking Humanoid Robot, Linear Model Predictive Control

1 INTRODUCTION

The difficulty in generating a stable walking motion mostly lies in the fact that the displacement of the Center of Mass (CoM) entirely relies on the contact forces between the feet and the ground, with the constraint that feet can only push on the ground [18, 19]. This restricts the motions that a walking system can realize, strongly limiting its capacity to follow a predefined motion in the presence of perturbations [22]. There is a strong interest therefore in being able to generate walking motions online, continuously adapting them to the current dynamics of the system.

A promising approach making use of a Linear Quadratic Regulator (LQR) has been proposed in [7]. Based on a linear approximation of the dynamics of the system, this approach tries to keep the contact forces in the middle of the feasible set, stabilizing the motion of the CoM of the system by minimizing its jerk over a finite prediction horizon. This LQR based approach allows generating stable walking motions online, with the possibility to continuously take into account the current state of the system [10, 13]. Variants have also been proposed [12]. But ignoring the exact constraints on contact forces limits its capacity to deal with difficult cases such as fast changes in the desired motion or strong perturbations.

In order to overcome this limitation, it has been proposed to introduce these constraints explicitly into the regulator, turning the LQR scheme into a more general Linear Model Predictive Control (LMPC) scheme, what led to a significant improvement in its capacity to deal with these difficult cases [20]. But both of these propositions were designed to work with foot step positions decided and fixed beforehand by a foot step planner, and we know that being able to adapt step positions online can contribute significantly to dealing with these difficult cases. It has been proposed therefore in [1] to introduce new control variables corresponding to the positions of the foot steps, allowing their real time adaptation with only a minimal modification to the existing LMPC scheme. But this scheme still needed that foot steps were planned beforehand, modifying them only to ensure feasibility and deal with perturbations.

The goal of this paper is to show that this scheme can once again be slightly modified in order to obtain a fully automatic placement of the foot steps: a reference speed is given to the robot, which can be modified at any time, and according to this reference speed and the current state of the robot, a safe foot step placement is decided. An important feature of this new scheme is that even in case of an external perturbation or in case of a reference speed which is not realizable, the generated walking motion is always entirely safe and stable: the reference speed is tracked only as much as possible within the limits of stability.

Walking motion with automatic foot step placement has already been realized in [16] with a saturated PD regulator of the motion of the CoM, the saturation being used to satisfy the constraints on the Center of Pressure (CoP) and generate the foot step placements accordingly when required. However, feasibility and stability of these foot step placements is not ensured exactly. Moreover, the use of a fixed saturated PD controller for controlling at the same time the motion of the CoM and the foot step placements can only induce suboptimal reactions. In our case, feasibility and stability constraints are handled explicitly, and both the motion of the CoM and the foot step placements are entirely free to be used to satisfy these constraints optimally.

The paper is organized as follows: Section 2 will present the original LMPC scheme introduced in [7], while Section 3 will present how it can be simply modified in order to handle automatic foot step placement. Important details about how the constraints on the CoP need to be addressed will be given in Section 4 before giving simulation results and concluding.

2 THE ORIGINAL PREDICTIVE CONTROL SCHEME

The original Predictive Control scheme introduced in [7] to generate walking motions proposes to focus on the motion of the CoM of the walking robot. Even though Nonlinear MPC schemes are getting more and more accessible to fast systems requiring short computation times, thanks to state of the art mathematical methods [2], Linear MPC schemes still allow shorter computation times and therefore faster control loops. In order to obtain a Linear MPC scheme in our case, we assume that the robot walks on a constant horizontal plane, and that the motion of its CoM is also constrained to a horizontal plane at a distance h above the ground, so that its position in space can be defined using only two variables (x, y).

We consider trajectories of the CoM which have piecewise constant jerks $\dddot{x}$ and $\dddot{y}$ over time intervals of constant length T so that we can compute the corresponding dynamics at discrete times t_k:

$$\hat{x}_{k+1} = A\,\hat{x}_k + B\,\dddot{x}(t_k), \tag{1}$$

$$\hat{y}_{k+1} = A\,\hat{y}_k + B\,\dddot{y}(t_k) \tag{2}$$

with

$$\hat{x}_k = \begin{pmatrix} x(t_k) \\ \dot{x}(t_k) \\ \ddot{x}(t_k) \end{pmatrix}, \quad \hat{y}_k = \begin{pmatrix} y(t_k) \\ \dot{y}(t_k) \\ \ddot{y}(t_k) \end{pmatrix} \tag{3}$$

and

$$A = \begin{pmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} T^3/6 \\ T^2/2 \\ T \end{pmatrix}. \tag{4}$$

We consider furthermore an approximation of the position (z^x, z^y) of the CoP on the ground corresponding to this motion by neglecting the inertial effects due to the rotations of different parts of the robot:

$$z^x_k = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{x}_k, \tag{5}$$

$$z^y_k = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{y}_k \tag{6}$$

with h the constant height of the CoM above the ground and g the norm of the gravity force.

Using the dynamics (1) recursively, we can derive relationships between the jerk of the CoM, its position and velocity and the position of the CoP over longer time intervals, here of length NT:

$$X_{k+1} = \begin{pmatrix} x_{k+1} \\ \vdots \\ x_{k+N} \end{pmatrix} = P_{ps}\hat{x}_k + P_{pu}\dddot{X}_k, \tag{7}$$

$$\dot{X}_{k+1} = \begin{pmatrix} \dot{x}_{k+1} \\ \vdots \\ \dot{x}_{k+N} \end{pmatrix} = P_{vs}\hat{x}_k + P_{vu}\dddot{X}_k, \tag{8}$$

$$Z^x_{k+1} = \begin{pmatrix} z^x_{k+1} \\ \vdots \\ z^x_{k+N} \end{pmatrix} = P_{zs}\hat{x}_k + P_{zu}\dddot{X}_k \tag{9}$$

with

$$\dddot{X}_k = \begin{pmatrix} \dddot{x}_k \\ \vdots \\ \dddot{x}_{k+N-1} \end{pmatrix}. \tag{10}$$

The matrices P_{ps}, P_{vs}, $P_{zs} \in \mathbb{R}^{N\times 3}$ and P_{pu}, P_{vu}, $P_{zu} \in \mathbb{R}^{N\times N}$ introduced here follow directly from a recursive application of the dynamics (1):

$$P_{ps} = \begin{bmatrix} 1 & T & T^2/2 \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2T^2/2 \end{bmatrix}, \; P_{pu} = \begin{bmatrix} T^3/6 & 0 & 0 \\ \vdots & \ddots & 0 \\ (1{+}3N{+}3N^2)\,T^3/6 & \dots & T^3/6 \end{bmatrix},$$

$$P_{vs} = \begin{bmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{bmatrix}, \; P_{vu} = \begin{bmatrix} T^2/2 & 0 & 0 \\ \vdots & \ddots & 0 \\ (1{+}2N)\,T^2/2 & \dots & T^2/2 \end{bmatrix},$$

$$P_{zs} = \begin{bmatrix} 1 & T & T^2/2 - h/g \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2T^2/2 - h/g \end{bmatrix}, \; P_{zu} = \begin{bmatrix} T^3/6 - Th/g & 0 & 0 \\ \vdots & \ddots & 0 \\ (1{+}3N{+}3N^2)\,T^3/6 - Th/g & \dots & T^3/6 - Th/g \end{bmatrix}.$$

The same relationships apply to Y_{k+1}, $\dot{Y}_{k+1}$, Z^y_{k+1}, $\hat{y}_k$ and $\dddot{Y}_k$.

The MPC scheme introduced in [7] balances over a prediction horizon of length NT the minimization of the jerk $(\dddot{X}, \dddot{Y})$ of the CoM with the tracking of a reference position $(Z^{x\text{-}ref}, Z^{y\text{-}ref})$ of the CoP which is chosen to lie in the middle of the support polygon for an enhanced robustness against perturbations. This corresponds to the Quadratic Program (QP)

$$\min_{\dddot{X}_k, \dddot{Y}_k} \; \frac{\alpha}{2}\left\|\dddot{X}_k\right\|^2 + \frac{\gamma}{2}\left\|Z^x_{k+1} - Z^{x\text{-}ref}_{k+1}\right\|^2 + \frac{\alpha}{2}\|\dddot{Y}_k\|^2 + \frac{\gamma}{2}\left\|Z^y_{k+1} - Z^{y\text{-}ref}_{k+1}\right\|^2. \tag{11}$$

This QP can be expressed canonically as

$$\min_{u_k} \; \frac{1}{2} u_k^T Q u_k + p_k^T u_k \tag{12}$$

with

$$u_k = \begin{pmatrix} \dddot{X}_k \\ \dddot{Y}_k \end{pmatrix}, \tag{13}$$

$$Q = \begin{pmatrix} Q' & 0 \\ 0 & Q' \end{pmatrix}, \tag{14}$$

$$Q' = \alpha I + \gamma P_{zu}^T P_{zu} \tag{15}$$

and

$$p_k = \begin{pmatrix} \gamma P_{zu}^T (P_{zs}\hat{x}_k - Z^{x\text{-}ref}_{k+1}) \\ \gamma P_{zu}^T (P_{zs}\hat{y}_k - Z^{y\text{-}ref}_{k+1}) \end{pmatrix}. \tag{16}$$

Observe that the matrix Q here is constant with time, so it can be prefactorized to minimize online computation time.

3 AUTOMATIC FOOT STEP PLACEMENT

Since the feet of the robot can only push on the ground, the CoP can lie only within the support polygon; the convex hull of the contact points between the feet and the ground [18]. Any trajectory not satisfying this constraint can't be realized. It has been shown in [20] that the tracking of a reference position of the CoP in the QP (11) can be replaced by just enforcing these constraints on the position of the CoP. This allows to avoid specifying explicitly the shape of the walking motion, which then can be generated freely among all feasible motions.

A theoretical analysis of this Predictive Control scheme has been proposed in [21], showing that minimizing any derivative of the motion of the CoM of the robot while enforcing the constraints on the position of the CoP results in stable online walking motion generation. A nice detail there is that any control variable can be handled by this minimization process, including foot step placement. Noteworthy, in the original Predictive Control scheme introduced in [7] as well as in all similar works [10, 13, 17, 20], the foot step placement needs to be decided first of all.

Based on this observation and with a view on increasing the robustness of the walking motion with respect to external perturbations, an online adapation of the feet positions has been proposed in [1]. The Predictive Scheme (11) was globally kept unchanged except for the introduction of new variables (X^f_k, Y^f_k) corresponding to the positions of the m foot steps taking place in the prediction horizon. These variables were adapted then in the minimization process, leading to the adaptation of the foot step placements. But still, reference foot step placements need to be given to this scheme.

We can conclude from all these previous results that it should be possible to generate a stable walking motion by only regulating the speed of the CoM to a desired mean value $(\dot{x}^{ref}, \dot{y}^{ref})$ and letting the foot step placement adapt fully automatically. Note that the dynamics of walking presents unavoidable effects such as the lateral sway motion implying that only a mean desired speed of the CoM can be sought for, as we will see in Section 6. We can consider therefore a QP as simple as

$$\min_{u_k} \frac{\beta}{2} \left\| \dot{X}_{k+1} - \dot{X}^{ref}_{k+1} \right\|^2 + \frac{\beta}{2} \left\| \dot{Y}_{k+1} - \dot{Y}^{ref}_{k+1} \right\|^2 \tag{17}$$

with variables

$$u_k = \begin{pmatrix} \dddot{X}_k \\ X^f_k \\ \dddot{Y}_k \\ Y^f_k \end{pmatrix}, \tag{18}$$

but we will see that better results can be obtained when re-introducing the same terms as in the QP (11):

$$\begin{aligned} \min_{u_k} \frac{\alpha}{2} \left\| \dddot{X}_k \right\|^2 + \frac{\beta}{2} \left\| \dot{X}_{k+1} - \dot{X}^{ref}_{k+1} \right\|^2 + \frac{\gamma}{2} \left\| Z^x_{k+1} - Z^{x\_ref}_{k+1} \right\|^2 \\ + \frac{\alpha}{2} \left\| \dddot{Y}_k \right\|^2 + \frac{\beta}{2} \left\| \dot{Y}_{k+1} - \dot{Y}^{ref}_{k+1} \right\|^2 + \frac{\gamma}{2} \left\| Z^y_{k+1} - Z^{y\_ref}_{k+1} \right\|^2 . \end{aligned} \tag{19}$$

These terms have however a different meaning here than in the original QP (11). The minimization of the jerk $(\dddot{X}, \dddot{Y})$ was necessary in (11) to generate stable motions whereas this is obtained in the

QP (17) by just regulating the speed $(\dot{X}, \dot{Y})$. We will see however in the simulation results that a weakly weighted minimization of the jerk helps smoothing the contact forces and therefore the resulting motion. The tracking of a reference position of the CoP was necessary in (11) to generate a feasible motion whereas feasibility is obtained now by enforcing directly the constraints on the position of the CoP (which will be discussed in more length in the next section). We will see however that a weakly weighted centering the CoP under the feet allows faster and more robust reactions to changes in the state of the system or in the desired speed of the CoM.

More precisely, instead of having the CoP track a reference position $(Z^{x\text{-}ref}, Z^{y\text{-}ref})$ fixed in advance as in the original QP (11), we want it lie in the middle of the foot positions actually decided by our algorithm. With (X_k^{fc}, Y_k^{fc}) the current position of the foot on the ground (which can't be changed) and (X_k^f, Y_k^f) the positions of the following m steps which will be decided by our QP (17), we basically have

$$Z_{k+1}^{x\text{-}ref} = U_{k+1}^c X_k^{fc} + U_{k+1} X_k^f \tag{20}$$

$$Z_{k+1}^{y\text{-}ref} = U_{k+1}^c Y_k^{fc} + U_{k+1} Y_k^f \tag{21}$$

with

$$U_{k+1}^c := \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad U_{k+1} := \begin{pmatrix} 0 & 0 & \\ \vdots & \vdots & \\ 0 & 0 & \\ 1 & 0 & \\ \vdots & \vdots & \\ 1 & 0 & \\ 0 & 1 & \\ \vdots & \vdots & \\ 0 & 1 & \ddots \end{pmatrix}. \tag{22}$$

The ones in this vector $U_{k+1}^c \in \mathbb{R}^N$ and matrix $U_{k+1} \in \mathbb{R}^{N \times m}$ simply indicate which sampling times t_k fall into which step, where sampling times correspond to rows and steps to column, and therefore which foot position must be taken into account at what time.

We can express then this new QP in the same canonical form (12), but this time with a varying quadratic term because of the varying matrix U_{k+1}:

$$Q_k = \begin{pmatrix} Q'_k & 0 \\ 0 & Q'_k \end{pmatrix}, \tag{23}$$

$$Q'_k = \begin{pmatrix} \alpha I + \beta P_{vu}^T P_{vu} + \gamma P_{zu}^T P_{zu} & -\gamma P_{zu}^T U_{k+1} \\ -\gamma U_{k+1}^T P_{zu} & \gamma U_{k+1}^T U_{k+1} \end{pmatrix} \tag{24}$$

and

$$p_k = \begin{pmatrix} \beta P_{vu}^T(P_{vs}\hat{x}_k - \dot{X}_{k+1}^{ref}) + \gamma P_{zu}^T(P_{zs}\hat{x}_k - U_{k+1}^c X_k^{fc}) \\ -\gamma U_{k+1}^T(P_{zs}\hat{x}_k - U_{k+1}^c X_k^{fc}) \\ \beta P_{vu}^T(P_{vs}\hat{y}_k - \dot{Y}_{k+1}^{ref}) + \gamma P_{zu}^T(P_{zs}\hat{y}_k - U_{k+1}^c Y_k^{fc}) \\ -\gamma U_{k+1}^T(P_{zs}\hat{y}_k - U_{k+1}^c Y_k^{fc}) \end{pmatrix} \tag{25}$$

Hopefully, the matrix U_{k+1} varies cyclically with time so prefactorizing a whole cycle of matrices Q_k would still be possible to minimize online computation time.

4 CONSTRAINTS ON THE CENTER OF PRESSURE

The description of the QP (19) isn't complete without the constraints on the CoP, which must always lie within the support polygon [18]: any trajectory not satisfying this constraint can't be realized as is. During single support, considering a foot with a polygonal shape on the ground at a position (x^f, y^f) with an orientation θ, this constraint can be expressed as a set of linear constraints

$$\begin{pmatrix} d_x(\theta) & d_y(\theta) \end{pmatrix} \begin{pmatrix} z^x - x^f \\ z^y - y^f \end{pmatrix} \leq b(\theta) \tag{26}$$

on the position of the CoP. Here, the column vectors $d_x(\theta)$ and $d_y(\theta)$ gather the x and y coordinates of the normal vectors to the edges of the feet and the column vector $b(\theta)$ gathers their positioning, with a security margin if necessary (as presented later in Section 6).

We can observe that these inequalities are linear with respect to the position of the foot on the ground but nonlinear with respect to its orientation. In the case of double support, these inequalities won't be linear anymore with respect to the feet position but quadratic because of the cross-products hidden in the computation of the vectors d_x and d_y. Such nonlinearities are best avoided in order to keep the form of a QP with linear constraints which is very advantageous from a computational point of view. For this reason, we will consider here the orientations of the feet decided in advance and we will discuss now in more depth the question of double support.

It appears that satisfying the constraint on the position of the CoP only at discrete instants is largely enough for generating realizable motions, under the mandatory condition that it is satisfied at all transition times during single and double support phases. An important observation is that at these transition times, the constraints of both single and double support apply, but those of single support are the most restrictive and are therefore sufficient on their own. We choose here to satisfy the constraint on the position of the CoP at the sampling times t_k, with a period T in between chosen to be strictly equal to the length of the double support phases ($0.1\,s$ here, with single support periods of $0.7\,s$) so that no sampling time falls strictly inside them. This way, we end up having to consider only the single support constraint (26).

Expressing now this constraint at the instants t_k over the whole prediction horizon leads to

$$D_{k+1} \begin{pmatrix} Z^x_{k+1} - U^c_{k+1} X^{fc}_k - U_{k+1} X^f_k \\ Z^y_{k+1} - U^c_{k+1} Y^{fc}_k - U_{k+1} Y^f_k \end{pmatrix} \leq b_{k+1}, \tag{27}$$

or with respect to the vector u_k defined in (18),

$$D_{k+1} \begin{pmatrix} P_{zu} & -U_{k+1} & 0 & 0 \\ 0 & 0 & P_{zu} & -U_{k+1} \end{pmatrix} u_k \leq b_{k+1} + D_{k+1} \begin{pmatrix} U^c_{k+1} X^{fc}_k - P_{zs} \hat{x}_k \\ U^c_{k+1} Y^{fc}_k - P_{zs} \hat{y}_k \end{pmatrix}, \tag{28}$$

with D_{k+1} of the following simple double diagonal form,

$$D_{k+1} = \begin{pmatrix} d_x(\theta_1) & 0 & \dots & 0 & d_y(\theta_1) & \dots \\ 0 & d_x(\theta_1) & \ddots & \vdots & 0 & \ddots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots \\ 0 & \dots & 0 & d_x(\theta_m) & 0 & \dots \end{pmatrix}. \tag{29}$$

Due to the special structure of these matrices, no matrix multiplications are required for assembling the final inequality constraint (28), what can be done therefore very quickly. On top of that, their evolution in time is also highly structured, so this assembling need not even be realized at each time.

5 CONSTRAINTS ON THE FOOT STEP PLACEMENTS

We need to ensure that the foot step placements generated by our algorihtm are feasible with respect to maximum leg length, joint limits, self-collision avoidance, maximum joint speed and similar geometric and kinematic limitations. In order to keep the Linear MPC structure of our algorithm, what we need to do is to derive simple approximations of all these limitations that can be expressed in the form of linear constraints on the vector u_k defined in (18).

We can derive for example simple linear bounds on the positions of the feet one with respect to the other with minimum and maximum values preventing collision on one side and over-stretching of the legs on the other side. Collisions of thighs, knees, shins, ankles and feet are checked on a large quantity of simulations which are run offline with varying foot placements, the result of which is synthesized in a simple convex polygon of safe foot placements.

Concerning maximum joint speed, we have found that a very simple bound on the position of the next foot step depending on the current position of the foot in the air and a simple Cartesian maximum speed gives good results:

$$C \begin{pmatrix} \left(X^f_k\right)_1 - x^f(t) \\ \left(Y^f_k\right)_1 - y^f(t) \end{pmatrix} \leq (t_{touchdown} - t) v_{max} \tag{30}$$

with $\left(X^f_k, Y^f_k\right)_1$ the position of the next foot step, $\left(x^f(t), y^f(t)\right)$ the current position of the foot in the air, v_{max} a vector of approximate maximum Cartesian speed in the directions indicated by the matrix C and $t_{touchdown}$ the time when the foot in the air is planned to touch the ground.

6 SIMULATION RESULTS

We'll consider the following sample motion on a simulated HRP-2 humanoid robot [9]: the robot starts from rest in double support and will walk continuously for 20 s, making a step regularly every 0.8 s. The robot starts with a zero reference velocity which is switched to 0.3 ms^{-1} forward at the beginning of the first step. The robot is pushed to the left at the beginning of step 3, at time $t = 2.4$ s. Then, in the middle of step 7, at time $t = 6$ s, the reference velocity is switched to 0.2 ms^{-1} on the right. Then, in the beginning of step 15, at time $t = 12$ s, it is switched back to 0.3 ms^{-1} forward, and back to zero in the middle of step 22, at time $t = 18$ s.

With $N = 16$ time intervals of length $T = 0.1$ s, the prediction horizon is $NT = 1.6$ s, what corresponds to two steps. Assembling and solving the full QP (19) in this case takes less than 1 ms in average with state of the art solvers such as QL [15] or qpOASES [6, 5, 14], notwithstanding the possible optimizations presented in [3] or the optimized solver presented in [4] which can help reduce furthermore the computation time.

We can observe in Fig. 1 that the simple QP (17) manages to perfectly realize this desired motion and absorb the perturbation while always maintaining the CoP within the boundaries of the support polygon. More precisely, we have considered a safety margin, so the CoP always lie 3 cm inside the true boundaries of the support polygon. In fact, the position of the CoP plotted here corresponds to the approximate model (5)-(6), but the difference with the real CoP is usually less than 2 cm so this motion appears to be completely safe.

We can observe that when the push on the left occurs at the beginning of step 3, the robot is just beginning a single support on the left leg, which can not be moved therefore. And since it is forbidden for the robot to cross legs because of the risk of collisions between them, it is only at the end of step 4 that the left leg can be moved to the left in order to absorb the perturbation and recover a motion forward. In the mean time, the robot drifts to the left. This demonstrates one of the most valuable properties of this walking motion generation scheme: safety prevails, in the sense that the generated motion is always kept feasible, even if that means not realizing the desired motion. Here, the goal of the robot is to move forward, but this goal is fulfilled only when possible.

Still, this motion isn't completely satisfactory: the trajectory of the CoP looks messy sometimes, what can lead to difficulties on a real robot. This even has an effect on the speed of the robot, which can be seen to oscillate around its reference value in Fig. 2.

In the approximate model (5)-(6), the position of the CoP appears to be related to the position and acceleration of the CoM, so minimizing the derivative of this acceleration, the jerk $(\dddot{X}, \dddot{Y})$, should smoothen the trajectory of the CoP and the speed of the CoM. We can observe in Fig. 3 and Fig. 4 that it is indeed the case when introducing a gain $\alpha = 10^{-6}$ s^6 with $\beta = 1$ s^2 (and $\gamma = 0$) in the QP (19).

But once again, this motion isn't completely satisfactory yet: we can observe in Fig. 3 that during the lateral translation, the CoP is positionned at the front of the feet. Although perfectly correct from the point of view of the dynamics of the system, this position induces difficulties when a perturbation

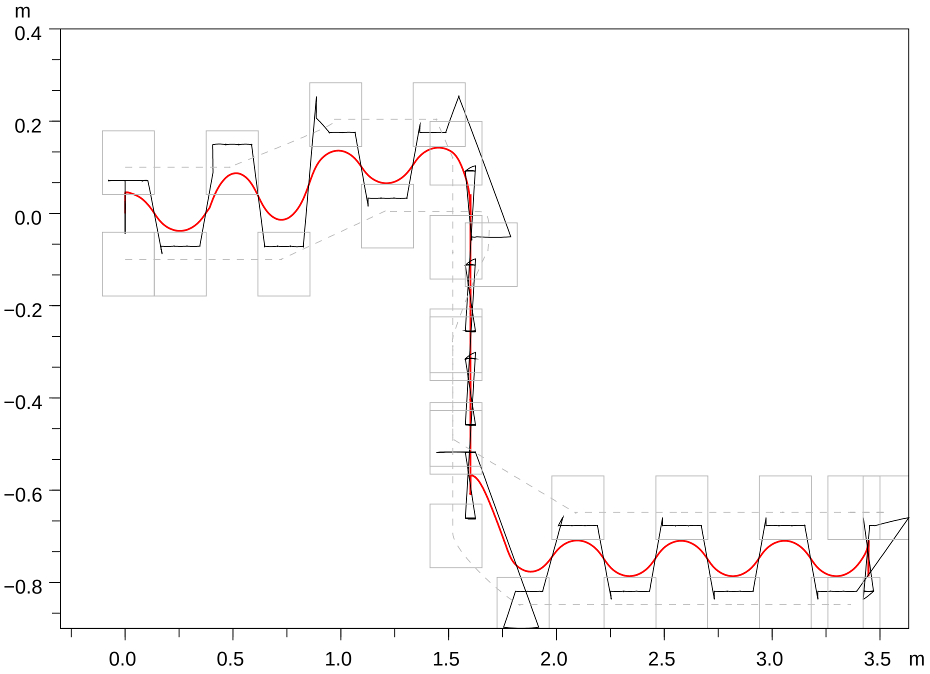

Figure 1: Foot step placement and ankle motion (grey), Center of Pressure (black) and Center of Mass (red) positions for the sample motion described in Section 6, obtained with the Predictive Control scheme (17).

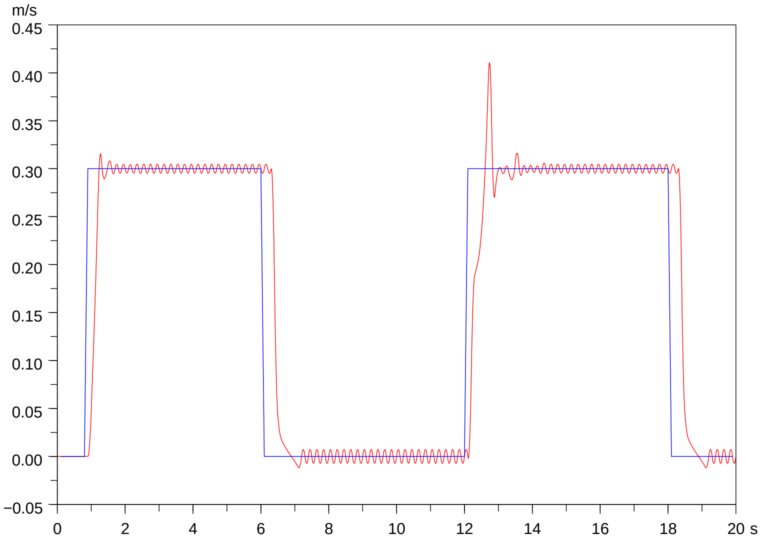

Figure 2: Forward speed of the CoM (red) and reference speed (blue) for the motion of Fig. 1.

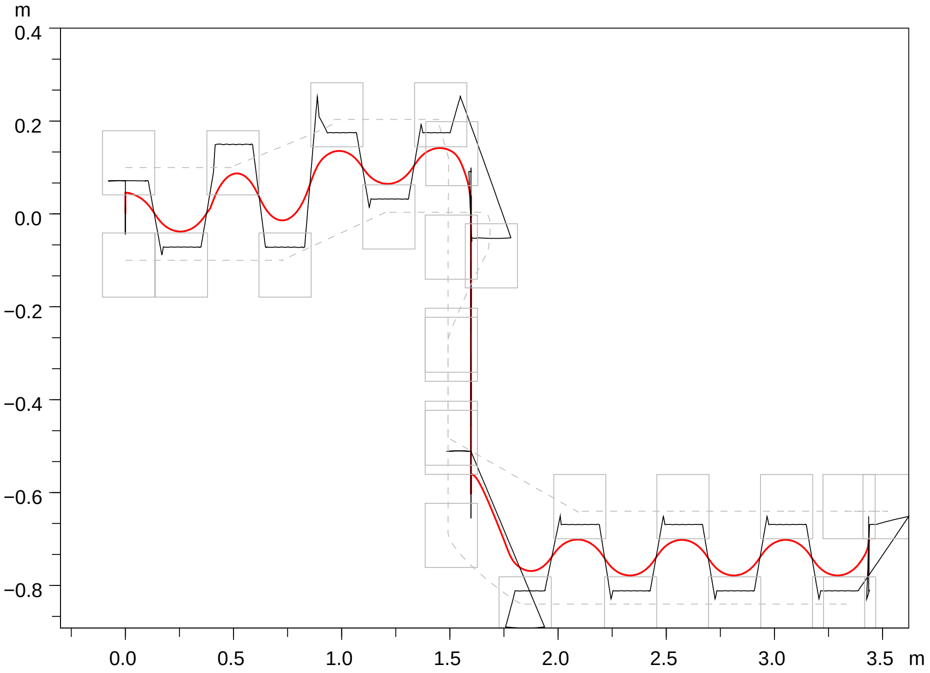

Figure 3: Foot step placement and ankle motion (grey), Center of Pressure (black) and Center of Mass (red) positions for the sample motion described in Section 6, obtained with the Predictive Control scheme (19), but with $\gamma = 0$.

or a change of desired velocity needs to be dealt with. This can be seen at the end of the lateral motion, at time $t = 12$ s: a delay and an overshoot can be observed in Fig. 4, much more than in the similar situation at time $t = 0.8$ s.

Adding a gain $\gamma = 10^{-6}$ in the QP (19) is enough to fix this problem, as can be seen in Fig. 5 and Fig. 6. A comparison between Fig. 3 and Fig. 5 shows that during the lateral motion, only the foot step placements have changed, not the trajectories of the CoM and CoP. Interestingly enough, in the original QP (11), the gain γ was used to drive the CoP in the middle of the feet positions, which were fixed. Here, it is the opposite: the position of the CoP is decided with respect to the desired motion of the CoM, and the foot step placement is decided then accordingly, centered around the CoP when possible, here during the lateral motion.

Let's focus in Fig. 7 on a comparison between the forward speed we obtained with our scheme and the speed that is obtained with the scheme proposed in [11]. This latter scheme allows immediate modification of foot place, but this foot place needs to be decided before-hand. The question of feasibility and stability of this foot place is not tackled therefore explicitly as in our scheme, and as we will see, the tracking of a reference velocity is neither as good nor as reactive. Let's consider the case when the

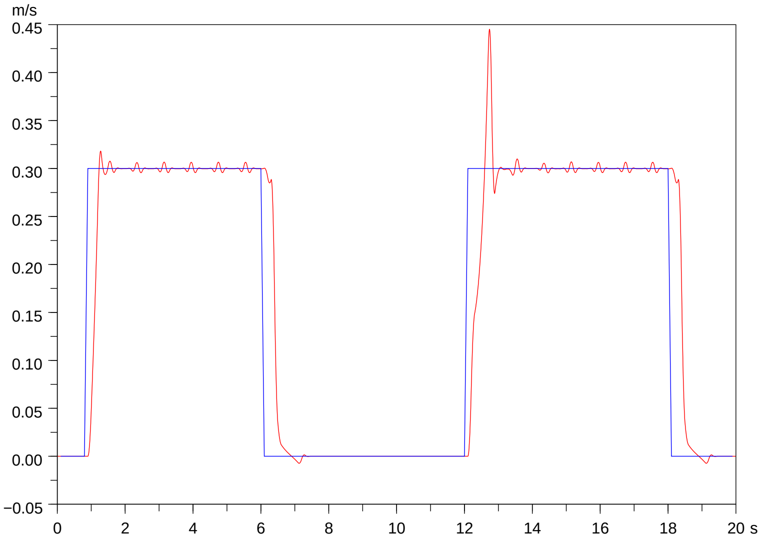

Figure 4: Forward speed of the CoM (red) and reference speed (blue) for the motion of Fig. 3.

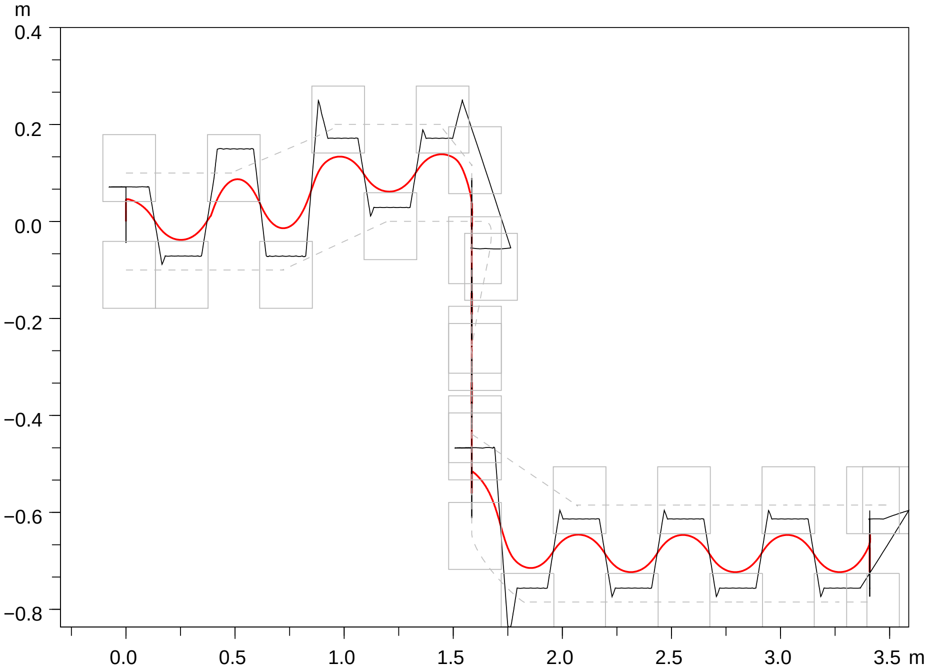

Figure 5: Foot step placement and ankle motion (grey), Center of Pressure (black) and Center of Mass (red) positions for the sample motion described in Section 6, obtained with the Predictive Control scheme (19).

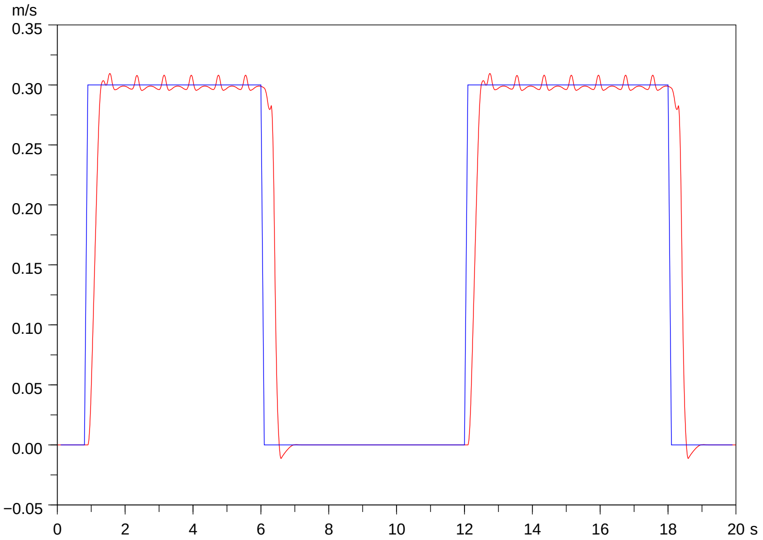

Figure 6: Forward speed of the CoM (red) and reference speed (blue) for the motion of Fig. 5.

desired speed is changed from 0 to 0.3 ms^{-1} (steps of length 24 cm every 0.8 s) at the beginning of a step. The first obvious observation is that with our scheme, the speed converges nearly perfectly to the desired value whereas with the scheme of [11], only the mean value of the speed converges to the desired value. However, this latter scheme was not designed for such a convergence, so this observation is not fair: the approximately quadratic shape of the speed is a classical result of continuously positioning the CoP in the middle of the feet [8], whereas in our scheme the CoP moves continuously forwards under the feet, as can be seen in Figures 1, 3 and 5, as a result of tracking at best a given speed for the CoM while not prescribing any position for the CoP. More interesting is the observation that the speed of the CoM rises nearly twice more quickly with our scheme. This is noteworthy since the scheme in [11] has been proposed for fast reaction of the robot. In this sense the motion of the robot appears to react almost twice faster with our scheme.

Having a look at the lateral speed of the CoM in Fig. 8, we can observe that because of the unavoidable lateral sway motion, only a mean desired speed of the CoM can be obtained. But having a more precise look at the mean speed over prediction horizons, which appears in black on this Figure, we can see that it is very different from the reference speed (in blue) during the lateral motion, between times $t = 6$ s and $t = 12$ s. This isn't surprising, since the Predictive Control scheme (19) tries to regulate the instantaneous speed $\dot{X}_{k+1}$ to the desired value, but during lateral motion, one step out of two must be realized in a direction opposite to the desired one since crossing legs isn't possible for this robot. So one step out of two, the instantaneous speed can be perfectly regulated to the desired value while one step of two it can reach only a far lower value, giving in the end a mean speed of about 2/3 of the desired

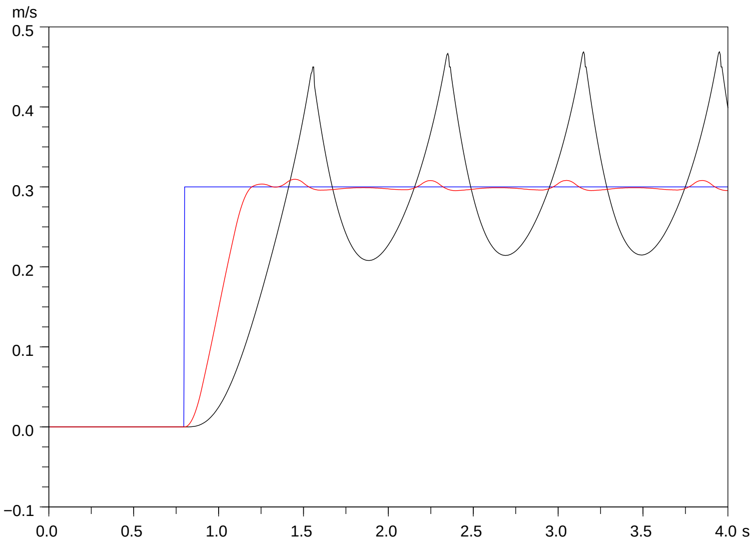

Figure 7: Comparison between the forward speed obtained with our method (red) and with the method proposed in [11] (black) when the reference speed (blue) is changed from 0 to 0.3 ms^{-1} at the beginning of a step.

value. An option could be to regulate instead the mean speed over 2 steps:

$$\begin{aligned}\min_{u_k} \frac{\alpha}{2}\left\|\dddot{X}_k\right\|^2 + \frac{\delta}{2}\left\|\frac{EX_{k+1}}{2T_{step}} - \dot{X}^{ref}_{k+1}\right\|^2 + \frac{\gamma}{2}\left\|Z^x_{k+1} - Z^{x\_ref}_{k+1}\right\|^2 \\ + \frac{\alpha}{2}\left\|\dddot{Y}_k\right\|^2 + \frac{\delta}{2}\left\|\frac{EY_{k+1}}{2T_{step}} - \dot{Y}^{ref}_{k+1}\right\|^2 + \frac{\gamma}{2}\left\|Z^y_{k+1} - Z^{y\_ref}_{k+1}\right\|^2 \qquad (31)\end{aligned}$$

with a selection matrix E of the form $\begin{bmatrix} -I & 0 & I \end{bmatrix}$, which computes the displacement of the CoM over periods of length $2T_{steps}$ out of the positions X_{k+1} and Y_{k+1}. We can see in Fig. 9 that the resulting mean speed corresponds to the reference speed, but this variant also brings two difficulties. The first one is that working with a mean speed over two steps instead of an instantaneous speed requires that the prediction horizon be doubled from two steps (1.6 s) with the scheme (19) to four steps (3.2 s), what implies much longer computation times. The second one is that regulating only the mean speed generates a stronger sway motion, that can be seen here when comparing the amplitude of the oscillations in Fig. 8 and Fig. 9. And this can have undesirable effects such as inducing a higher sensibility to perturbations.

7 Conclusion

The LQR scheme originally proposed in [7] opened the way to online walking motion generations which could adapt to the state of the robot. But this scheme was designed to work with foot step positions decided and fixed beforehand by a foot step planner. We have shown here that a minimal modification to

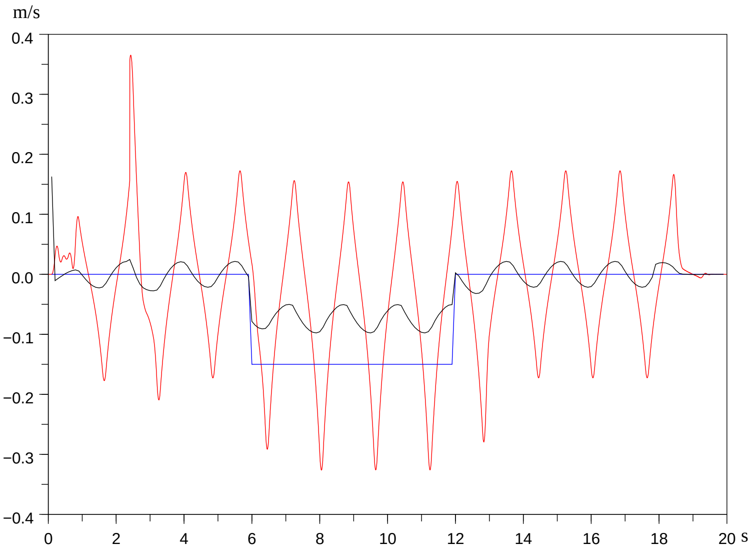

Figure 8: Lateral speed of the CoM (red), mean value of this speed over prediction horizons (grey) and reference speed (blue) for the motion of Fig. 5.

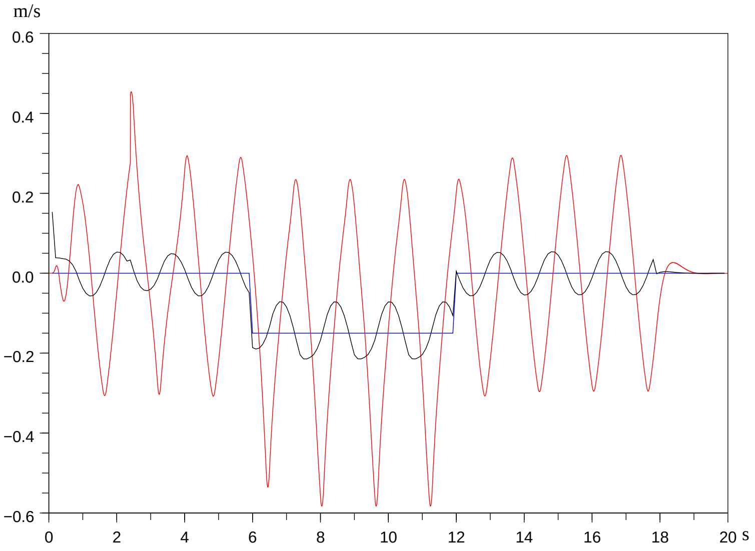

Figure 9: Lateral speed of the CoM (red), mean value of this speed over prediction horizons (grey) and reference speed (blue) obtained with the Predictive Control scheme (31).

this LQR scheme, introducing the exact constraints on the Center of Pressure and new control variables corresponding to the positions of the foot steps, allows a fully automatic placement of these foot steps. We obtain then an online walking motion generation that can track a given reference speed of the robot, which can be modified at any time, while keeping the safety of the system under control even in the case of strong perturbations. We are working now on demonstrating these results on the real HRP-2 humanoid robot.

8 Acknowledgments

The authors would like to thank Olivier Stasse without whom the comparison in Fig. 7 wouldn't have been possible. Andrei Herdt and Pierre-Brice Wieber would like to thank Kazuhito Yokoi and Aderrahmane Kheddar for hosting them in the CNRS/AIST Joint Robotics Lab in Tsukuba, Japan. This research was supported by the French Agence Nationale de la Recherche, grant reference ANR-08-JCJC-0075-01. This research was also supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering Center (OPTEC) and the German-French cooperation program Procope coordinated by the German Academic Exchange Service (DAAD) and the Programme Hubert Curien. The research of Pierre-Brice Wieber was supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme. The research of Katja Mombaur was supported by the Postdoc Program of the foundation Landesstiftung Baden-Württemberg.

REFERENCES

[1] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl. Online walking gait generation with adaptive foot positioning through linear model predictive control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 1121–1126, Nice, 2008.

[2] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, 2001.

[3] D. Dimitrov, J. Ferreau, P.-B. Wieber, and M. Diehl. On the implementation of model predictive control for on-line walking pattern generation. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 2685–2690, Pasadena, 2008.

[4] 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 *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1171–1176, Kobe, 2009.

[5] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[6] H.J. Ferreau, P. Ortner, P. Langthaler, L. del Re, and M. Diehl. Predictive control of a real-world diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31(2):293–301, 2007.

[7] 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 *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1620–1626, Taipei, 2003.

[8] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain - derivation and application of the linear inverted pendulum mode -. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1405–1411, Sacramento, 1991.

[9] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. Humanoid robot HRP-2. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1083–1090, New Orleans, 2004.

[10] S. Kanzaki, K. Okada, and M. Inaba. Bracing behavior in humanoid through preview control of impact disturbance. In *Proceedings of the International Conference on Humanoid Robotics*, pages 301–305, Tsukuba, 2005.

[11] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa. Experimentation of humanoid walking allowing immediate modication of foot place based on analytical solution. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 3989–3994, Roma, 2007.

[12] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi. Integrated motion control for walking, jumping and running on a small bipedal entertainment robot. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 3189–3194, New Orleans, 2004.

[13] K. Nishiwaki and S. Kagami. High frequency walking pattern generation based on preview control of zmp. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 2667–2672, Orlando, 2006.

[14] http://homes.esat.kuleuven.be/˜optec/software/qpoases/.

[15] K. Schittkowski. QL: A fortran code for convex quadratic programming - user's guide, version 2.11. Research Report, Department of Mathematics, University of Bayreuth, 2005.

[16] T. Sugihara. Simulated regulator to synthesize zmp manipulation and foot location for autonomous control of biped robots. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1264–1269, Pasadena, 2008.

[17] H. Takeuchi. Real time optimization for robot control using receding horizon control with equal constraint. *Journal of Robotic Systems*, 20(1):3–13, 2003.

[18] P.-B. Wieber. On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, 2002.

[19] P.-B. Wieber. Holonomy and nonholonomy in the dynamics of articulated motion. In *Proceedings of the Ruperto Carola Symposium on Fast Motion in Biomechanics and Robotics*, Heidelberg, 2005.

[20] P.-B. Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Proceedings of the International Conference on Humanoid Robotics*, pages 137–142, Genova, 2006.

[21] P.-B. Wieber. Viability and predictive control for safe locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 1103–1108, Nice, 2008.

[22] P.-B. Wieber and C. Chevallereau. Online adaptation of reference trajectories for the control of walking systems. *Robotics and Autonomous Systems*, 54(7):559–566, 2006.