Optimized Humanoid Walking with Soft Soles

Giovanni de Magistris, Adrien Pajon, Sylvain Miossec, Abderrahmane Kheddar

Optimized Humanoid Walking with Soft Soles

Giovanni De Magistris[a,b,*], Adrien Pajon[b,*], Sylvain Miossec[c], Abderrahmane Kheddar[a,b]

[a] *CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL, Japan*
[b] *CNRS-University of Montpellier, LIRMM, Interactive Digital Human, France*
[c] *PRISME Laboratory, University of Orléans, Bourges 18020, France*

Abstract

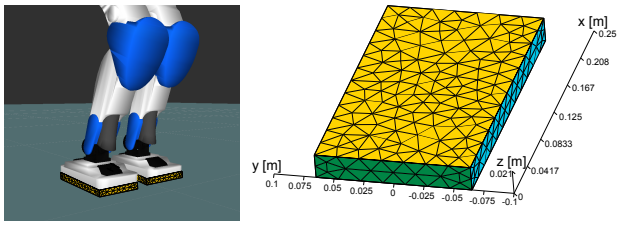
In order to control more efficiently the feet-ground interaction of humanoid robots during walking, we investigate adding outer soft (i.e. compliant) soles to the feet. The deformation subsequent to the contact of the soles with the ground is taken into account using a new walking pattern generator and deformation estimator. This novel humanoid walking approach ensures that the desired zero moment point for stability requirement is fulfilled. We validate our new controller using the HRP-4 humanoid robot performing walking experiments with and without the estimator. Also, to test the robustness of our approach and to obtain low-energy walking, we performed different walking motions.

Keywords: Compliant soles, Deformation Estimator, Walking Pattern Generator, Humanoid robots, Optimized walking gait

1. Introduction

Gaited or non-gaited walking is generated by alternating phases of contact creation and breaking with the environment [1]. With rigid links and without shock absorbing mechanism, impact forces with the ground must be thresholded through contact transitions with nearly zero speed. This considerably limits the walking dynamics. Therefore, compliant mechanisms are used in humanoid robotics to absorb shocks at impacts and prohibit their propagation along the entire structure that results in non-desirable vibrations and eventually unstable behaviors. One common solution is to add flexible mechanisms at the robot ankles [2, 3] that also protect the feet embedded force sensors. Unfortunately, such compliant mechanisms also act as passive joints whose deformations are hardly measurable [4]. In this way, the robot attitude is difficult to control, especially in complex maneuvers [5]. Another solution is to add the compliance between the foot and ground contact, see early work by [6]. In fact, humanoid robots have generally a thin rubber sole attached under each robot foot. Due to the thinness of this sole, the impact shocks are mainly absorbed by the ankle flexibility.

Alternatively, we favor removing the ankle flexibilities and investigate the use of thick soft soles under each foot of a humanoid robot (see Fig. 1). These soles not only absorb the impacts due to contact transitions, they also cast ground unevenness resulting in a relative increase of the contact surface. During walking, the compliance of these soles depends on the contact area variations. Also, since the compliance is put outside, it can be decoupled from the rigid dynamics. In order to generate balanced robot movements and directly control the contact with the environment, we developed a new Walking Pattern Generator (WPG) and deformation estimator. We validate our new methods for walking with compliant soles performing different experiments on HRP-4.



(a) (b)

Figure 1: (a): Rectangular parallelepiped soles mounted on HRP-4's feet; (b): meshed sole with 1494 tetrahedron elements

2. Problem formulation

In most simplified model-based planning and control, the ankle flexibility and the sole compliance are not modeled. They are left as 'perturbations' or 'uncertainties' to be tackled by the closed-loop controller. Therefore, controllers have to compensate for the errors in the attitude (i.e. free-floating orientation and position) due to the deformations of the flexible parts. In our control framework, we consider the model of the deformation that results from the contact of the soft sole with the ground using the Finite Element Model (FEM) and the mechanical laws of compliant contacts.

*Co-first authors. *Emails:* giova.dema85@gmail.com (G. De Magistris), adrien.pajon@gmail.com (A. Pajon)

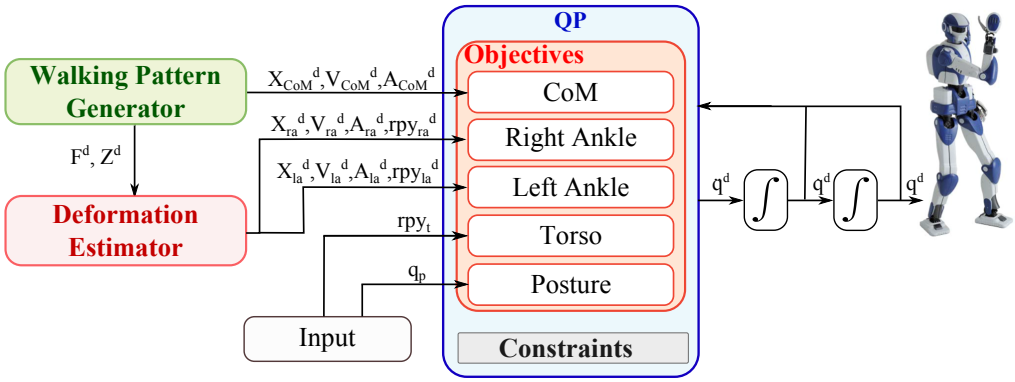


Figure 2: Control scheme. The ankle reference trajectories change according to sole deformations.

To control the humanoid robot attitude during walking, we use the multi-objective Quadratic Programming (QP) control scheme illustrated in Fig. 2. In particular, we developed a new WPG that is coupled with a sole deformation estimator to achieve good balance during walking. We also experimented the HRP-4's humanoid robot performances for different WPG.

In section 3, we detail our new WPG; in section 4 we illustrate how we can estimate the foot position and orientation taking into account the deformations; then in section 5 we explain the QP for the whole-body control of the robot, which tracks the previously determined trajectories; in section 6 we show how our WPG and the deformation estimator improve humanoid robot performances; finally, we conclude and explain the future works in section 7.

3. Walking pattern generator

The Zero Moment Point (ZMP) and the ankle positions are directly linked (see later Eq. (33)). To obtain smooth right ankle and left ankle trajectories in the control framework of Fig. 2, we define the ZMP trajectories using a 5th order polynomial function. As will be explained in section 3.3, the WPG developed in this paper calculates also smooth center of mass (COM) trajectories. This new WPG optimizes both single support phase (SSP) and double support phase (DSP) using a QP optimization.

Several works formulate the walking gait as an optimization problem [7][8][9]. They use an inverted pendulum model with a point mass to study feet convex-hull ZMP and not the whole ZMP trajectory of each foot. Therefore, the study of the DSP trajectory and force of each foot are not previously studied in the literature.

For the remainder of the paper, we denote with subscript "1" the foot that leaves the floor at the end of DSP, and with subscript "2" the foot that comes in contact at the beginning of DSP.

3.1. Optimization criteria

We want to generate walking gaits that minimize the energy consumption E:

$$\text{min. } (E) \tag{1}$$

From [9], the energy consumption that accounts for the motor and the gear models can be expressed as:

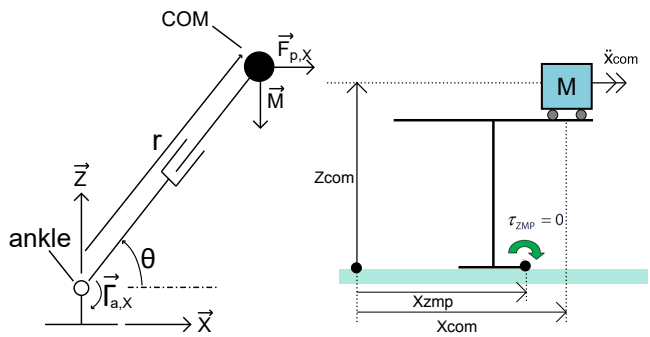
$$E = \sum_j \int_{t_i}^{t_f} (a_j \tau_j^2 + b_j \tau_j \dot{q}_j + c_j \dot{q}_j^2) dt \tag{2}$$

where τ_j is the force/torque, $\dot{q}_j$ is the joint j velocity, and a_j, b_j, c_j are the coefficients depending on joint j motor parameters and gear ratio.

Using the simplified robot model in Fig. 3a, Eq. (2) becomes:

$$E = \int_{t_i}^{t_f} (a_p F_p^2 + b_p F_p \dot{r} + c_p \dot{r}^2 + a_a \Gamma_a^2 + b_a \Gamma_a \dot{\theta} + c_a \dot{\theta}^2) dt \tag{3}$$

where subscript p denotes the prismatic joint and a the ankle joint, F_p is the prismatic force, $\dot{r}$ is the prismatic velocity, Γ_a is the ankle torque and $\dot{\theta}$ is the ankle velocity.



(a) Simplified robot model (b) Cart-table model

Figure 3: Simplified robot model in sagittal plane at left and cart-table model at right

To obtain a quadratic problem ($\dot{r}$ and $\dot{\theta}$ are nonlinear functions of the ZMP and the COM), we keep only the

force/torque terms (linear in terms of ZMP and COM parameters, see (13)). Therefore Eq. (3) becomes:

$$\begin{aligned} E &= \lambda \int_{t_i}^{t_f} \|\mathbf{F}_{\mathrm{COM}}(t)\|^2 \mathrm{dt} + (1-\lambda) \int_{\mathrm{SSP}} \|\mathbf{\Gamma}_{a_{\mathrm{SSP}}}(t)\|^2 \mathrm{dt} \\ &+ \frac{1}{2}(1-\lambda) \int_{\mathrm{DSP}} (\|\mathbf{\Gamma}_{a_1}(t)\|^2 + \|\mathbf{\Gamma}_{a_2}(t)\|^2) \mathrm{dt} \\ &+ \mu \int_{\mathrm{DSP}} (\|\ddot{\mathbf{P}}_{\mathrm{ZMP}_1}(t)\|^2 + \|\ddot{\mathbf{P}}_{\mathrm{ZMP}_2}(t)\|^2) \ ) \ \mathrm{dt} \end{aligned} \tag{4}$$

where λ and μ are the weights of each criterion; a_{SSP} denotes the ankle of the support foot in SSP; a_i refers to the ankle of the foot "1" or "2" in DSP; $\mathbf{P}_{\mathrm{ZMP}} = \begin{bmatrix} x_{\mathrm{ZMP}} & y_{\mathrm{ZMP}} \end{bmatrix}^T$ are the ZMP coordinates. The motor characteristics of the simplified model of Fig. 3a are not known and therefore the weights λ and μ are determined from experimental trials (see section 6.3).

Using the cart-table model in Fig. 3b, the COM force can be expressed as:

$$\mathbf{F}_{\mathrm{COM}}(t) = -\mathbf{F}_{\mathrm{ZMP}}(t) = M\left(\ddot{\mathbf{P}}_{\mathrm{COM}}(t) + g\vec{z}\right) \tag{5}$$

where $\ddot{\mathbf{P}}_{\mathrm{COM}}$ is the COM acceleration, M is the robot mass and $\mathbf{F}_{\mathrm{ZMP}}(t)$ is the floor reaction force (ZMP force).

Considering the foot in contact at a fixed position during the whole SSP, we obtain:

$$\mathbf{\Gamma}_a(t) = \mathbf{F}_{\mathrm{ZMP}}(t) \times (\mathbf{P}_a(t) - \mathbf{P}_{\mathrm{ZMP}}(t)), \tag{6}$$

where $\mathbf{\Gamma}_a$ is the torque at the ankle, $\mathbf{P}_a$ is the ankle position and $\mathbf{P}_{\mathrm{ZMP}}$ is the ZMP position. During the DSP the two feet are in contact and we obtain a similar equation to (6) for $\mathbf{\Gamma}_{a_1}(t)$ and $\mathbf{\Gamma}_{a_2}(t)$. The force distribution constraint between feet in DSP is detailed in AppendixB.

For $\lambda = 1$ in (4) we obtain a walking pattern close to [7][9], while for $\lambda = 0$ we obtain a ZMP trajectory similar to [8]. Intermediate values of λ give compromises between the two extreme behaviors. Our experimental trials reveal that the best obtained values for λ are not the extreme ones (i.e. $\lambda = 0$ or $\lambda = 1$).

In Eq. (4), $\mathbf{P}_{\mathrm{ZMP}}$ and $\mathbf{P}_{\mathrm{COM}}$ depend on $\mathbf{F}_{\mathrm{COM}}$ and $\mathbf{\Gamma}_a$ whereas $\mathbf{P}_{\mathrm{ZMP}_1}$ and $\mathbf{P}_{\mathrm{ZMP}_2}$ depend only on $\mathbf{\Gamma}_a$. For this reason, optimal $\mathbf{P}_{\mathrm{ZMP}_1}$ and $\mathbf{P}_{\mathrm{ZMP}_2}$ tend to be close to the ankles, resulting in a ZMP trajectory under each foot going from ankle to heel, then to toe and finally go back to the ankle. For a foot with flexible sole, obtaining an oscillating foot orientation is suboptimal from the point of view of energy consumption due to the ankle velocity in (3). To solve this problem, we added the last term to the optimization criterion (4) dependent on $\mathbf{P}_{\mathrm{ZMP}_1}$ and $\mathbf{P}_{\mathrm{ZMP}_2}$ accelerations.

3.2. ZMP trajectories

We divided the walking gait in three cycles:

1. *Start*: the robot begins with a DSP. This cycle starts on the point $\mathrm{ZMP}_{\mathrm{init}}$.
2. *Walking*: the robot does n steps alternating SSP and DSP.

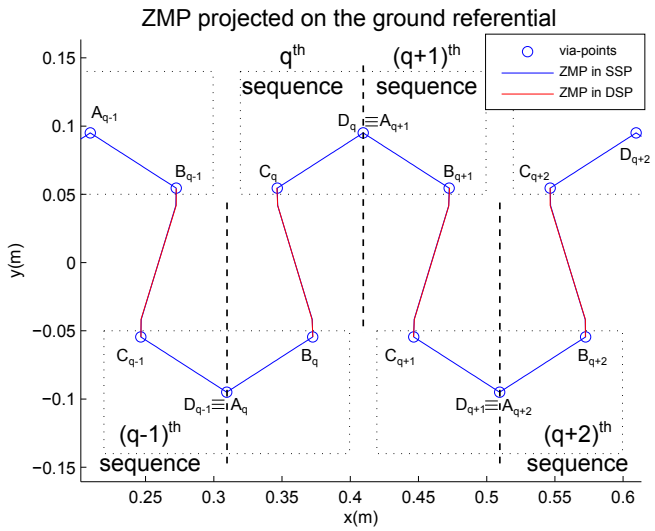


Figure 4: Walking sequences with via-points (o) and ZMP trajectories during SSP (blue) and DSP (red)

3. *Stop*: the robot ends with a DSP. This cycle finishes on the point $\mathrm{ZMP}_{\mathrm{end}}$.

Each SSP and start/stop DSP are defined by two C^2 polynomials and each remaining DSP is defined by one. Four via-points are defined to parametrize these phases (see Fig. 4). For a given sequence q, the SSP polynomials go from the via-point A_q to the via-point B_q and from the via-point C_q to the via-point D_q. The DSP polynomial goes from the via-point B_q to the via-point C_q. Therefore, we need $m = 3n+4$ polynomials to model the whole ZMP trajectory:

$$x_{\mathrm{ZMP}}^{(j)}(t) = \sum_{i=0}^{5} a_i^{(j)} (\Delta t_j)^i \tag{7}$$

where the superscript j denotes the jth phase, a_i is the ith polynomial coefficient and $\Delta t_j = t - t_{j-1}$.

To find the polynomial coefficients of (7) and to obtain smooth trajectories, we enforced the boundary conditions in position, speed and acceleration at each via-point. In particular, we interpolated the polynomial functions between $\mathrm{ZMP}_{\mathrm{init}} \rightarrow C_1$ during *Start*, $A_q \rightarrow B_q$, $B_q \rightarrow C_q$ and $C_q \rightarrow D_q$ during the sequence q and $B_{n+2} \rightarrow \mathrm{ZMP}_{\mathrm{end}}$ during *Stop*. $\mathrm{ZMP}_{\mathrm{init}}$ and ZMP_{end} are not necessarily equal to B_1 and C_{n+2} respectively. In addition, to reduce the number of via-points and using their boundary conditions, A_q and D_q are combined into one via-point A_q.

3.3. COM trajectories

Using the cart table model in Fig. 3b, we can write the COM acceleration in x-axis as:

$$\ddot{x}_{\mathrm{COM}}^{(j)} = \frac{g}{z_{\mathrm{COM}}} (x_{\mathrm{COM}}^{(j)} - x_{\mathrm{ZMP}}^{(j)}) \tag{8}$$

Substituting (7) into (8), the COM position can be expressed as following [10]:

$$x_{\text{COM}}^{(j)} = V^{(j)} c_j + W^{(j)} s_j + \sum_{i=0}^{5} A_i^{(j)} (\Delta t_j)^i \tag{9}$$

where:

$$\begin{aligned}
c_j &= \cosh(\omega_j \Delta t_j) \\
s_j &= \sinh(\omega_j \Delta t_j) \\
\omega_j &= \sqrt{\tfrac{g}{z_{\text{COM}}^{(j)}}} = \sqrt{g/z_{\text{COM}}} \\
A_i^{(j)} &= \begin{cases} a_i^{(j)} + \sum_{k=1}^{(5-i)/2} b_{i+2k}^{(j)} a_{i+2k}^{(j)} & \text{for } i = 0 \cdots 3 \\ a_i^{(j)} & \text{for } i = 4, 5 \end{cases} \\
b_{i+2k}^{(j)} &= \prod_{l=1}^{k} \frac{(i+2l)(i+2l-1)}{w_j^2}
\end{aligned}$$

$V^{(j)}$ and $W^{(j)}$ are the unknowns of (9) and $a_i^{(j)}$ coefficients are known from (7). This is the difference between (9) and the system of equations in [10].

Details of how to find $V^{(j)}$ and $W^{(j)}$ are given in the AppendixA.

3.4. ZMP of each foot in DSP

As for the ZMP, we define ZMP_1 and ZMP_2 trajectories as C^2 polynomials. We assign to them a new via-point B'_q and C'_q respectively. To obtain the coefficients of these polynomials, we interpolated them between $B_q \to B'_q$ and $C'_q \to C_q$.

Using the ZMP definition, we obtain:

$$\mathbf{\Gamma}_{\text{ZMP}}^{(j)}(\mathbf{F}_{\text{ZMP}_1}) + \mathbf{\Gamma}_{\text{ZMP}}^{(j)}(\mathbf{F}_{\text{ZMP}_2}) = 0 \tag{10}$$

where $\mathbf{\Gamma}_{\text{ZMP}}^{(j)}(\mathbf{F}_{\text{ZMP}_i})$ is the torque of the reaction force $\mathbf{F}_{\text{ZMP}_i}$ under foot $i \in \{1, 2\}$ applied on ZMP during the phase j.

Solving (10) in the direction $\vec{y}$ we can write:

$$\sum_{i=1}^{2} (x_{\text{ZMP}_i}^{(j)} - x_{\text{ZMP}}^{(j)}) F_{\text{ZMP}_i}^{z(j)} = 0 \tag{11}$$

where x_{ZMP} is the component x of the ZMP position and $F_{\text{ZMP}_i}^{z}$ is the component z of the reaction force under the foot $i \in \{1, 2\}$.

From (11), we obtain the following relationship between ZMP_1 and ZMP_2:

$$x_{\text{ZMP}_2}^{(j)} = x_{\text{ZMP}}^{(j)} - \frac{F_{\text{ZMP}_1}^{z(j)}}{F_{\text{ZMP}_2}^{z(j)}} (x_{\text{ZMP}_1}^{(j)} - x_{\text{ZMP}}^{(j)}) \tag{12}$$

From (12), if ZMP_1 trajectories and the distribution between $F_{\text{ZMP}_1}^{z}$ and $F_{\text{ZMP}_2}^{z}$ are known, we obtain ZMP_2 trajectories. Details of how to find the force distribution between $F_{\text{ZMP}_1}^{z}$ and $F_{\text{ZMP}_2}^{z}$, and the constraints that they must satisfy are given in AppendixB.

During the *start* and the *stop* sequences, to obtain a robust optimization algorithm with respect to the different walking parameters, we added new via-points to ZMP_1 and ZMP_2 trajectories, B''_q and C''_q respectively. Thus, during these sequences, ZMP_1 and ZMP_2 are defined by two consecutive polynomials interpolated respectively between $B_q \to B''_q \to B'_q$ and $C'_q \to C''_q \to C_q$. On these new via-points, ZMP_1 and ZMP_2 trajectories satisfy the boundary conditions in position, speed and acceleration.

3.5. Optimization parameters

The WPG is formulated as a QP optimization problem. In this section, we detailed the optimization variables.

$\mathbf{F}_{\text{COM}}$, $\mathbf{\Gamma}_a$ and $\ddot{\mathbf{P}}_{\text{ZMP}_i}$ in (4) are linear functions of optimization variables. They can be written as:

$$\begin{aligned}
\mathbf{F}_{\text{COM}} &= \mathbf{A}_{\mathbf{F}_{\text{COM}}} X + B_{\mathbf{F}_{\text{COM}}}, \\
\mathbf{\Gamma}_a &= \mathbf{A}_{\mathbf{\Gamma}_a} X + B_{\mathbf{\Gamma}_a}, \\
\ddot{\mathbf{P}}_{\text{ZMP}_i} &= \mathbf{A}_{\ddot{\mathbf{P}}_{\text{ZMP}_i}} X + B_{\ddot{\mathbf{P}}_{\text{ZMP}_i}}
\end{aligned} \tag{13}$$

where X is the vector of optimization variables:

$$X = \begin{bmatrix} \mathbf{x_0^{vp}} & \mathbf{x_0^{COM}} & \mathbf{x_0^{step}} & \mathbf{x_0^{vp'}} \end{bmatrix}^T \tag{14}$$

$\mathbf{x_0^{vp}}$ represents the boundary conditions in position, speed and acceleration at via-points of ZMP ($3m+3$ parameters), $\mathbf{x_0^{COM}}$ represents the initial and final boundary condition in position and speed of COM (4 parameters), $\mathbf{x_0^{step}}$ represents the foot step positions ($n-1$ parameters) and $\mathbf{x_0^{vp'}}$ represents the boundary conditions in position, speed and acceleration at via-points of ZMP_1 ($3m+6$ parameters).

$$\begin{aligned}
\mathbf{x_0^{vp}} &= \begin{bmatrix} x_0^{(1)} & \dot{x}_0^{(1)} & \ddot{x}_0^{(1)} & \cdots & x_0^{(m+1)} & \dot{x}_0^{(m+1)} & \ddot{x}_0^{(m+1)} \end{bmatrix}^T \\
\mathbf{x_0^{COM}} &= \begin{bmatrix} x_{\text{COM}}^{(1)}(t_0) & \dot{x}_{\text{COM}}^{(1)}(t_0) & x_{\text{COM}}^{(m+1)}(t_m) & \dot{x}_{\text{COM}}^{(m+1)}(t_m) \end{bmatrix}^T \\
\mathbf{x_0^{step}} &= \begin{bmatrix} x_1^{\text{step}} & x_2^{\text{step}} & \cdots & x_{n-1}^{\text{step}} \end{bmatrix}^T \\
\mathbf{x_0^{vp'}} &= \begin{bmatrix} x_1^{(1)} & \dot{x}_1^{(1)} & \ddot{x}_1^{(1)} & \cdots & x_1^{(m+2)} & \dot{x}_1^{(m+2)} & \ddot{x}_1^{(m+2)} \end{bmatrix}^T
\end{aligned}$$

We imposed:

- Initial and final feet positions
- Initial and final position, speed and acceleration of the ZMP
- SSP and DSP duration
- COM height
- Foot step directions. It can be changed to define the rotation during walking
- Foot step positions. It is *optional* and define a specific walking path

In order to generate feasible trajectories, the ZMP trajectories must be inside the support convex hull [11]. It is defined by the area of the support foot. To obtain this area, we added inequality constraints to the QP. During the SSP, this condition is linear and can be integrated as

inequality constraints into a QP. During the DSP, this condition is not linear. However, if ZMP_1 and ZMP_2 trajectories are in the support convex hull defined by the areas of each foot, the stability condition for the ZMP trajectories is fulfilled. Therefore, these conditions are linear and can also be integrated in the QP optimization.

Based on human walking results in [12] and for ensuring safe walking w.r.t the robot lack of precision [8], we defined a feet support areas within a security margin of $5-10\%$ of the foot length. To avoid self-collisions, we defined a minimal distance between the robot feet (3 cm). To avoid stretched legs' singularity, we chose a maximum step length (30 cm). To respect the boundary condition for the COM position and velocity, we added the equality constraints (A.2) and (A.6) to the optimization.

3.6. WPG summary

The problem that we solve is:

$$\begin{array}{l} \min\limits_{X} (E) \\ \text{subject to equality and inequality constraints} \end{array} \tag{15}$$

where the criteria E comes from (4) and it is based on a simplification of the energy consumption, and the optimization variables X include the ZMP limit conditions, the COM initial and final conditions and the feet positions (14).

The inequality constraints are:

1. ZMP $\in$ the convex hull during the SSP
2. ZMP_1 and ZMP_2 $\in$ the convex hull during the DSP
3. Maximum and minimum foot step stretching distance

The equality constraints are:

1. Initial robot state as condition on the first via-point of ZMP and ZMP_1 and on the initial condition of the COM.
2. Final robot state as condition on the last via-point of ZMP and ZMP_1 (*optional*)
3. Final condition on the COM as a goal to reach or condition on the COM over the whole walk (e.g. COM velocity...) (*optional*)
4. Foot step placement (*optional, e.g. used to define a specific path...*)

4. Deformation estimator

In this section, we develop an algorithm that calculates the ankle position/orientation and the deformation of the flexible sole [13] to match the interaction force $\mathbf{F}^{\mathrm{d}}_{\mathrm{ZMP}}(t)$ and the ZMP trajectory $\mathbf{P}^{\mathrm{d}}_{\mathrm{ZMP}}(t)$ given by the WPG (see the control scheme 2).

The Fig. 6 shows this algorithm. It is divided in two main blocks: frictional contact problem (see section 4.1) and update sole position and orientation (see section 4.2). For a more detailed presentation, see also [14].

To simulate the flexible behavior of the sole, a static linear FEM is used [15]. This model is valid for small deformations. Indeed, due to the dense shape of the soles, their deformations are relatively small compared to their size. In future works, we may consider using a nonlinear large-deformation model to improve precision. Static simulation is a good approximation since $\mathbf{F}^{\mathrm{d}}_{\mathrm{ZMP}}$ and $\mathbf{P}^{\mathrm{d}}_{\mathrm{ZMP}}$ variations are negligible w.r.t the sole dynamic response.

Ω is the sole shape meshed with 1494 tetrahedron elements (see meshed sole in Fig. 1), Ω_I is the interior volume of Ω, $\partial\Omega$ gather the surface S_D attached to the rigid robot foot (Dirichlet surface) and the surface S_S that could be in contact with the ground. S_D and S_S contain two disjoint set of nodes:

$$\partial\Omega = S_D \cup S_S, \quad \text{with} \quad S_D \cap S_S = \emptyset \tag{16}$$

The static linear FEM gives a relationship between the node displacements of Ω and the external forces:

$$\boldsymbol{K}\ \underline{{}^{l}\boldsymbol{U}} = \underline{{}^{l}\boldsymbol{F}} \tag{17}$$

where $\underline{{}^{l}\boldsymbol{F}} \in \mathbb{R}^N$ is the vector of nodal forces and $\underline{{}^{l}\boldsymbol{U}} \in \mathbb{R}^N$ is the vector of the node displacements. The superscript l before a letter denotes a vector expressed in the sole frame and no superscript means that the vector is expressed in the world frame.

From the definition, the Dirichlet nodes have a known null displacement. Thus, the system (17) can be rewritten:

$$\begin{bmatrix} \boldsymbol{K}_{DD} & \boldsymbol{K}_{DI} & \boldsymbol{K}_{DS} \\ \boldsymbol{K}_{ID} & \boldsymbol{K}_{II} & \boldsymbol{K}_{IS} \\ \boldsymbol{K}_{SD} & \boldsymbol{K}_{SI} & \boldsymbol{K}_{SS} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{0}} \\ \underline{{}^{l}\boldsymbol{U}_I} \\ \underline{{}^{l}\boldsymbol{U}_S} \end{bmatrix} = \begin{bmatrix} \underline{{}^{l}\boldsymbol{F}_D} \\ \underline{{}^{l}\boldsymbol{F}_I} \\ \underline{{}^{l}\boldsymbol{F}_S} \end{bmatrix} \tag{18}$$

where I, D and S denotes respectively nodes of Ω_I, S_D and S_S.

As in [16], we suppose that the internal interaction forces $\underline{{}^{l}\boldsymbol{F}_I}$ are zeros and therefore S_D is the only surface with nodal forces.

The system (18) can be rewritten:

$$\begin{bmatrix} \boldsymbol{K}_{II} & \boldsymbol{K}_{IS} \\ \boldsymbol{K}_{SI} & \boldsymbol{K}_{SS} \end{bmatrix} \begin{bmatrix} \underline{{}^{l}\boldsymbol{U}_I} \\ \underline{{}^{l}\boldsymbol{U}_S} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{0}} \\ \underline{{}^{l}\boldsymbol{F}_S} \end{bmatrix} \tag{19}$$

Based on [16] and (19), a condensed linear elasticity law is defined:

$$\begin{aligned} \underline{{}^{l}\boldsymbol{U}_S} &= \left(\boldsymbol{K}_{SS} - \boldsymbol{K}_{SI}\boldsymbol{K}_{II}^{-1}\boldsymbol{K}_{IS}\right)^{-1}\ \underline{{}^{l}\boldsymbol{F}_S} \\ &= \boldsymbol{K}_S^{-1}\ \underline{{}^{l}\boldsymbol{F}_S} = \boldsymbol{C}_S\ \underline{{}^{l}\boldsymbol{F}_S} \\ \underline{{}^{l}\boldsymbol{U}_I} &= -\boldsymbol{K}_{II}^{-1}\boldsymbol{K}_{IS}\ \underline{{}^{l}\boldsymbol{U}_S} \end{aligned} \tag{20}$$

where $\boldsymbol{K}_S$ is the stiffness surface matrix and $\boldsymbol{C}_S$ is the compliance surface matrix.

In order to obtain the shape deformation from (20), the contact with the ground directly induces $\underline{{}^{l}\boldsymbol{U}_S}$, which engender the computation of $\underline{{}^{l}\boldsymbol{U}_I}$.

4.1. Frictional contact problem

The frictional contact problem finds the contact space displacement vector $\underline{\boldsymbol{\delta}}$ and contact force vector $\underline{\boldsymbol{F}}$ at each time step t_s. This problem must be constrained to verify the Signorini's and Coulomb's laws.

The *inputs* of this algorithm are:

1. ${}^{l}\underline{\boldsymbol{P}}^{\text{free}}$: node positions
2. $\boldsymbol{C}_S$: surface compliance matrix from (20)
3. ${}^{l}\underline{\boldsymbol{P}}^{t_s-1}$: node positions at the previous time step
4. $\boldsymbol{O}_l$: foot position vector
5. $\boldsymbol{\Upsilon} = (\theta, \phi, \psi)$: Euler rotation angles of the foot

The *outputs* of this algorithm are:

1. $\underline{\boldsymbol{\delta}}$: relative position of the sole nodes to its ground projection (see Fig. 5). It is called node contact space displacement vector
2. $\underline{\boldsymbol{F}}$: node contact force vector

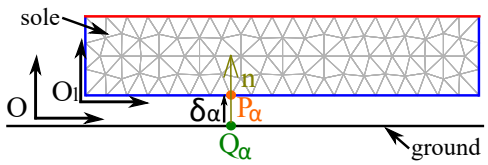


Figure 5: Contact between the sole and the ground

For a contact point α, $\boldsymbol{\delta}_\alpha$ is defined by:

$$\boldsymbol{\delta}_\alpha = \boldsymbol{P}_\alpha - \boldsymbol{Q}_\alpha \tag{21}$$

where $\boldsymbol{Q}_\alpha$ is the normal projection of α on the ground.

The normal component of a vector is denoted by the subscript n such as $\boldsymbol{v}_n = \mathbf{n}\boldsymbol{v}$ where $\mathbf{n} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$; the tangential components of a vector are denoted by the subscript t such as $\boldsymbol{v}_t = \mathbf{t}\boldsymbol{v}$, where $\mathbf{t} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$. The superscript T denotes the transpose operator.

The contact model used for calculating the deformation of the sole takes into account two very important mechanical laws: (i) the **Signorini's law** is used to enforce the non-interpenetration (in the computation) of the sole with the ground, and (ii) the **Coulomb's law** which defines the dry friction. Based on (21), we have:

1. **Signorini's law**:

$$\mathbf{0} \leq \boldsymbol{F}_{n,\alpha} \perp \boldsymbol{\delta}_{n,\alpha} \geq \mathbf{0} \tag{22}$$

   where $\boldsymbol{F}_\alpha$ is the contact force applied to a contact node α, subscript n is its normal component.

2. **Coulomb's law**. We can distinguish two state conditions: *stick* and *slip* motion.

   (a) **Stick condition**:

$$\|\boldsymbol{F}_{t,\alpha}\| < \mu|F_{n,\alpha}|, \;\; \boldsymbol{\delta}_{t,\alpha} = 0 \tag{23}$$

   (b) **Slip condition**:

$$\boldsymbol{F}_{t,\alpha} = -\mu F_{n,\alpha}\frac{\boldsymbol{\delta}_{t,\alpha}}{\|\boldsymbol{\delta}_{t,\alpha}\|} \tag{24}$$

   where μ is the friction coefficient.

In our simulation, we use a quasi-static contact model. Quasi-static means that at a given instant in time we can assume the problem is static. This assumption works well because the dynamics of the sole deformation are fast compared to the dynamics of the limit conditions applied to it (here Force and ZMP).

The foot velocity is not considered in the static case. In the quasi-static case taking into account the slippage, the foot velocity is considered by computing the state just after the sole stop to slip. Therefore, slip occur until a static state is reached. This gives a non-reversible sole simulation and a sole state that depends on the history of its deformation.

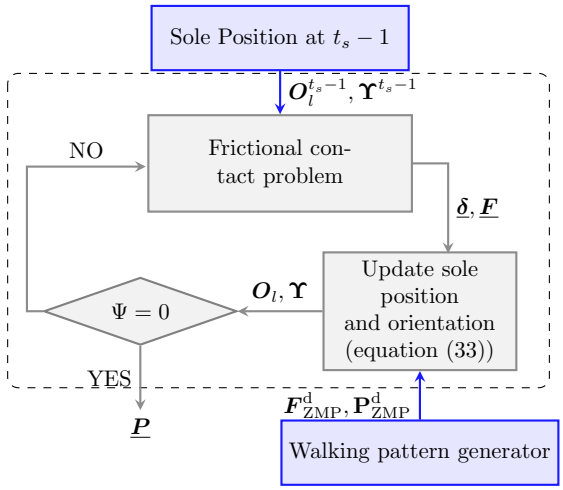


Figure 6: Framework for contact handling to obtain $\boldsymbol{F}^{\text{d}}_{\text{ZMP}}$ and $\mathbf{P}^{\text{d}}_{\text{ZMP}}$. $t_s - 1$ is the previous time step and $\Psi = \left[\boldsymbol{F}_{\text{ZMP}} - \boldsymbol{F}^{\text{d}}_{\text{ZMP}},\ \mathbf{P}_{\text{ZMP}} - \mathbf{P}^{\text{d}}_{\text{ZMP}},\ \boldsymbol{\Gamma}_{\mathbf{P}_{\text{ZMP}},n}\right]^T$

In order to obtain $\boldsymbol{P}_\alpha$ and $\boldsymbol{Q}_\alpha$ and solve this problem, the displacement $\underline{\boldsymbol{U}_S}$ of equation (20) (in absolute frame) is used:

$$\underline{\boldsymbol{U}_S} = \underline{\boldsymbol{P}} - \underline{\boldsymbol{P}}^{\text{free}} \tag{25}$$

and then, the sole surface node positions are defined as:

$$\underline{\boldsymbol{P}} = \boldsymbol{H}\boldsymbol{C}_S\boldsymbol{H}^T\underline{\boldsymbol{F}} + \underline{\boldsymbol{P}}^{\text{free}} \stackrel{\text{def}}{=} \boldsymbol{W}\underline{\boldsymbol{F}} + \underline{\boldsymbol{O}_l} + \boldsymbol{H}\ {}^{l}\underline{\boldsymbol{P}}^{\text{free}} \tag{26}$$

$\boldsymbol{W}$ is the Delassus' operator [17]; $\boldsymbol{H} = \text{diag}(\boldsymbol{R}, \ldots, \boldsymbol{R})$, where $\boldsymbol{R}$ is the foot orientation matrix defined by $\boldsymbol{\Upsilon} = (\theta, \phi, \psi)$.

During the contact $\mathbf{n}\boldsymbol{Q}_\alpha = 0$ and then the normal component of $\boldsymbol{\delta}_\alpha$ is:

$$\begin{aligned}\boldsymbol{\delta}_{n,\alpha} &= \mathbf{n}\boldsymbol{O}_l + \mathbf{n}\boldsymbol{R}\ {}^{l}\boldsymbol{P}^{\text{free}}_\alpha + \mathbf{n}\sum_{\beta=1}^{c_n}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta)\\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}^{\text{free}}_{n,\alpha} + \mathbf{n}\sum_{\beta=1}^{c_n}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta)\end{aligned} \tag{27}$$

where $\boldsymbol{W}_{\alpha\beta} \stackrel{\text{def}}{=} \begin{bmatrix} W_t \\ W_n \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} W_{tt} & W_{tn} \\ W_{nt} & W_{nn} \end{bmatrix}$ is the 3×3 Delassus' operator coupling the contacts α and β and c_n is the number of contact nodes.

The tangential component of $\boldsymbol{\delta}_\alpha$ is:

$$\begin{aligned} \boldsymbol{\delta}_{t,\alpha} &= \mathbf{t}\boldsymbol{O}_{l,\alpha} + \mathbf{t}\boldsymbol{R}\, {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{c_n}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) - \mathbf{t}\boldsymbol{Q}_\alpha \\ &\approx \mathbf{t}\boldsymbol{O}_{l,\alpha} + \mathbf{t}\boldsymbol{R}\, {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{c_n}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) - \mathbf{t}\boldsymbol{P}_\alpha^{(q-1)} \\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}_{t,\alpha}^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) \end{aligned} \tag{28}$$

If at $t_s - 1$ a node α is already in contact, the reference node at t_s is $\boldsymbol{P}_\alpha^{(t_s-1)}$. If α is not in contact at $t_s - 1$, the reference contact node at t_s is approximated by the normal projection of α on the ground at $t_s - 1$. In reality, the contact creation occurs at an intermediate time step between $t_s - 1$ and t_s. The contact creation position could be found with a variable step size and an event driven simulation. This can be computationally costly for a large number of new contact nodes. However, with a small time-step, the approximation error is negligible.

Based on (27) and (28), a contact node α has a contact space displacement defined by:

$$\begin{aligned} \boldsymbol{\delta}_\alpha &= \boldsymbol{O}_l + \boldsymbol{R}\, {}^l\boldsymbol{P}_\alpha^{\text{free}} - \begin{bmatrix} \boldsymbol{P}_{t,\alpha}^{(t_s-1)} \\ 0 \end{bmatrix} + \sum_{\beta=1}^{c_n}\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta \\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}_\alpha^{\text{free}} + \sum_{\beta=1}^{c_n}\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta \end{aligned} \tag{29}$$

The concatenation of (29) for all α gives the frictional contact problem solved by:

$$\underline{\boldsymbol{\delta}} = \underline{\boldsymbol{O}_l} + \boldsymbol{H}\, {}^l\underline{\boldsymbol{P}}^{\text{free}} - \underline{\boldsymbol{Q}} + \boldsymbol{W}\underline{\boldsymbol{F}} \tag{30}$$

where $\underline{\boldsymbol{Q}} = \left[\begin{bmatrix} \boldsymbol{P}_{t,1}^{(t_s-1)} \\ 0 \end{bmatrix}^T \cdots \begin{bmatrix} \boldsymbol{P}_{t,c_n}^{(t_s-1)} \\ 0 \end{bmatrix}^T \right]^T$.

The system (30) has $3c_n$ equations with $6c_n$ unknowns defined by $\underline{\boldsymbol{\delta}}$ and $\underline{\boldsymbol{F}}$. Additional $3c_n$ equations are obtained considering that each node in contact α can be in two different conditions:

1. **Stick**: (23), $\boldsymbol{\delta}_\alpha = 0 \Rightarrow 3c_{\text{stick}}$ equations where c_{stick} is the number of contacts in stick case
2. **Slip**: $\delta_{n,\alpha} = 0 \Rightarrow c_{\text{slip}}$ equations where c_{stip} is the number of contacts in slip case.

To define the frictional contact problem, $2c_{\text{slip}}$ equations are still needed. They are obtained using the Coulomb's law (24) in [18]. From [18] and (29), a system is built for each contact node α and solved using a Newton's method [19].

Finally, an iterative Gauss-Seidel method in [20, 21] is used to solve the system (30). This method is very fast and applicable for real-time solution [20].

4.2. Update sole position and orientation

In the frictional contact problem of the previous section, the foot position and orientation are the inputs when the contact forces and the contact space displacements are the outputs. Here, a new algorithm is developed to find the good foot position and orientation to match a desired ZMP force $\boldsymbol{F}_{\text{ZMP}}^{\text{d}}(t)$ and position $\mathbf{P}_{\text{ZMP}}^{\text{d}}(t)$ (see Fig. 6). Seeing the sole as a (complex 6-dof) spring, the principle is to invert the relation between the spring deformation and the force applied to it.

The *inputs* of this algorithm are the geometric and mechanical properties of the sole and the desired ZMP: i) ${}^l\underline{\boldsymbol{P}}^{\text{free}}$, ii) $\boldsymbol{C}_S$, iii) ${}^l\underline{\boldsymbol{P}}^{t_s-1}$, iv) $\left(\boldsymbol{F}_{\text{ZMP}}^{\text{d}}, \mathbf{P}_{\text{ZMP}}^{\text{d}}\right)$.

The *outputs* of this algorithm are the foot position and orientation: i) $\boldsymbol{O}_l$, ii) $\boldsymbol{\Upsilon} = (\theta, \phi, \psi)$. Recall that the contact node forces $\boldsymbol{F}$ and the displacements $\boldsymbol{\delta}$ are computed solving the frictional contact problem explained in the previous section.

Therefore, we can describe our algorithm using a function f:

$$f(\underline{\boldsymbol{O}_l, \boldsymbol{\Upsilon}}) = \begin{bmatrix} \boldsymbol{F}_{\text{ZMP}}^{\text{d}}(t) \\ \mathbf{P}_{\text{ZMP}}^{\text{d}}(t) \\ \Gamma_{P_{\text{ZMP}},n}^{\text{d}} = 0 \end{bmatrix} \tag{31}$$

We impose null vertical torque at ZMP $\Gamma_{P_{\text{ZMP}},n}^{\text{d}} = 0$.

From the system (30) and the Signorini's and Coulomb's laws, $6c_n + 6$ unknowns (considering now position and orientation of the sole as unknowns) and $6c_n$ equations are given (see section 4.1). The remaining 6 equations are obtained using the total force at ZMP $\boldsymbol{F}_{\text{ZMP}}$:

$$\boldsymbol{F}_{\text{ZMP}} = \sum_{\alpha=1}^{c_n}\boldsymbol{F}_\alpha \tag{32}$$

and the ZMP definition:

$$\begin{bmatrix} \Gamma_{P_{\text{ZMP}},t_1} = 0 \\ \Gamma_{P_{\text{ZMP}},t_2} = 0 \\ \Gamma_{P_{\text{ZMP}},n} \end{bmatrix} = \sum_{\beta=1}^{c_n} \begin{bmatrix} (P_{t_2,\beta} - P_{\text{ZMP},t_2}) \cdot F_{n,\beta} \\ -(P_{t_1,\beta} - P_{\text{ZMP},t_1}) \cdot F_{n,\beta} \\ (P_{t_1,\beta} - P_{\text{ZMP},t_1}) \cdot F_{t_2,\beta} - (P_{t_2,\beta} - P_{\text{ZMP},t_2}) \cdot F_{t_1,\beta} \end{bmatrix}$$

where $\Gamma_{P_{\text{ZMP}}}$ is the resultant torque at ZMP. The subscripts t_1 and t_2 denote respectively the first and second component of the vector.

To solve the problem (31), we use a Newton's method. A differential relationship between $(\boldsymbol{F}_{\text{ZMP}}, \mathbf{P}_{\text{ZMP}}, \Gamma_{P_{\text{ZMP}},n})$ and $(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$ is used to derive a Newton step:

$$\begin{bmatrix} \boldsymbol{O}_l^{(t_s)} \\ \boldsymbol{\Upsilon}^{(t_s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{O}_l^{(t_s-1)} \\ \boldsymbol{\Upsilon}^{(t_s-1)} \end{bmatrix} - \boldsymbol{J}^{-1} \begin{bmatrix} \boldsymbol{F}_{\text{ZMP}} - \boldsymbol{F}_{\text{ZMP}}^{\text{d}} \\ \mathbf{P}_{\text{ZMP}} - \mathbf{P}_{\text{ZMP}}^{\text{d}} \\ \Gamma_{P_{\text{ZMP}},n} \end{bmatrix} \tag{33}$$

where $\boldsymbol{J}$ is the Jacobian matrix.

The algorithm in Fig. 6 loops on frictional contact problem followed by a Newton step that gives a new foot position and orientation. To increase the speed of the algorithm, the precision of the frictional contact problem is increased with the convergence of the Newton steps.

The reference ankle position and orientation in the control scheme 2 is computed during the contact phase with:

$$\boldsymbol{P}_a = \boldsymbol{O}_l + \boldsymbol{R}\ {}^l\boldsymbol{P}_a \tag{34}$$

where ${}^l\boldsymbol{P}_a$ is the ankle position in the sole frame.

During the SSP, ankle positions and orientations of the foot in the air are computed by interpolating with 5th order polynomials the absolute ankle orientation, position, velocity and acceleration between take-off and landing phases.

4.3. Initial conditions

The previous problem is solved by imposing the initial conditions on position $\boldsymbol{O}_l^{\text{init}}$ and yaw orientation ψ^{init} of the foot. They are set at $t_s = 0$ to compute the first reference contact position $\boldsymbol{Q}$. Initial roll and pitch orientation could be discontinuous to fulfill the desired ZMP position. Thus initial roll and pitch orientation of the foot are set as those obtained from the desired ZMP position. The Newton step (33) at the first time is changed accordingly to the tangential reference contact positions:

$$\boldsymbol{Q}_t = \mathbf{t}\boldsymbol{O}_l^{\text{init}} + \mathbf{t}\boldsymbol{R}^{\text{init}}(\theta, \phi, \psi^{\text{init}})\ {}^l\boldsymbol{P}_\alpha^{\text{free}} \tag{35}$$

5. Multi-objective quadratic program controller

The reference trajectories for the COM, the ankle position and orientation are obtained offline. Those references are online tracked as good as possible by a QP that takes also into account the robot constraints. The obtained motions are then compatible with the COM dynamics and the model of the flexibility. A task-space QP formulation with a weighted hierarchy [5] is used to generate the whole body motion. The optimization variables are $\boldsymbol{x} = \left[\ddot{\boldsymbol{q}}^T\ \boldsymbol{\lambda}^T\right]^T$ where $\ddot{\boldsymbol{q}}$ is the joint acceleration vector and $\boldsymbol{\lambda}$ is the force intensity along with the linearized friction cone.

The QP is formulated as follows:

$$\min_{\boldsymbol{x}}.\ \left(\frac{1}{2}\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}\right) = \min_{\boldsymbol{x}}.\ \sum_{i=1}^{N} \alpha_i \|E_i(\boldsymbol{x})\|^2 + \alpha_\lambda \|\lambda\|^2$$

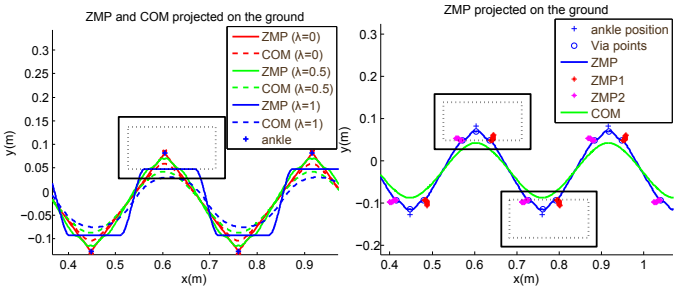
where N is the number of tasks put in the cost part, $E_i(\boldsymbol{x})$ is the task errors, α_i is the task weight and $\alpha_\lambda\|\lambda\|^2$ is a damping term to ensure positive definite Hessian matrix.

From [5], the tasks put as constraints are:

1. **Joint torque limits**
2. **Joint position/velocity limits**
3. **No-sliding contacts**: zero acceleration for the contact body i
4. **Collision avoidance**

More specifically, the tasks we used are:

1. **Ankle**: desired ankle trajectories given by the deformation estimator
2. **COM**: tracking of the desired COM position, velocity and acceleration given by the WPG
3. **Torso**: fixed torso absolute orientation
4. **Posture**: fixed posture for the whole upper body (torso, head, and arms)



(a) Different values of λ ($\mu = 15$) (b) With DSP ($\lambda = 0.5$, $\mu = 15$)

Figure 7: ZMP and COM trajectories with fixed ankle position

6. Results

6.1. WPG optimization results

The Fig. 7a shows part of the obtained ZMP and COM reference trajectories with fixed ankle position for different values of λ:

- $\lambda = 0$: minimization of ankle torque only
- $\lambda = 0.5$: minimization with equal weights on COM force and on ankle torque
- $\lambda = 1$: minimization of COM force only

For $\lambda = 0$ we obtain ZMP and COM trajectories similar to [22][8] with large lateral variations of the COM. For $\lambda = 1$ trajectories are similar to [7][23] with small lateral variations of the COM and with a ZMP close to the feet edge. The mid-compromise solution $\lambda = 0.5$ gives limited lateral variations of the COM, the ZMP trajectory is smoother like human ZMP trajectories [24]. All solutions result in walk with right and left feet as close as possible to minimize lateral variations of COM (and hence COM force) and variation of ZMP in DSP.

The Fig. 7b shows also ZMP_1 and ZMP_2 trajectories during the DSP. ZMP_1 and ZMP_2 are local ZMPs in DSP. When we have ZMP_1 under one foot, we have a ZMP_2 under the opposite one. In Fig. 7b several foot step are drawn and each foot step has a ZMP_1 and ZMP_2 but they don't exist during the same phase.

We chose μ equal to 15 which experimentally gave us the best compromise to have ZMP going from heel to toe under each foot.

6.2. HRP-4 walking with compliant sole using the new controller

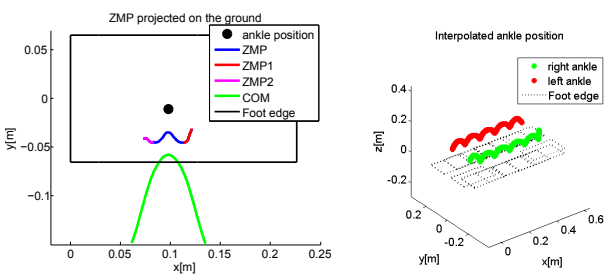
Here, we show the results of walking experiments with the HRP-4's new feet illustrated in Fig. 8. We removed the original shock absorbing springs located in the ankle (hence the ankle leg link is rigid) and added a flexible sole to absorb the landing impact shock and eventually cast terrain uncertainties. To do this experiments we used the WPG and deformation estimator explained before.

(a) (b)

Figure 8: (a): new HRP-4's feet; (b): photo of HRP-4's feet with flexible soles

The sole material is a flexible foam. Coupling compression test and FEM simulation, we obtained the Young's modulus $E = 0.32$MPa and Poisson's ratio $\nu = 0.31$ of the sole (which is softer than flexible soles used in [25] where Young's modulus were 5MPa, so that an estimator was not necessary).

Experiments consisted of HRP-4's straightforward walking for 10 footsteps (distance of 0.5 m) with average speeds of 3.5 cm/s and 4.2 cm/s. To validate our control framework in Fig. 2, we used ZMP and COM trajectories obtained using the WPG described in section 2 with $\lambda = 0.8$ and $\mu = 15$ in (4). These trajectories are shown in Fig. 9a. Fig. 9b shows the ankle trajectories when the foot is not in contact with the floor.



(a) (b)

Figure 9: (a): ZMP trajectory under the left foot and COM trajectory during the 6th foot step (SSP+DSP); (b): ankle trajectories when the foot is not in contact with the floor

The video[1] and Fig. 10 clearly show the improvements given by the control scheme in Fig. 2. Figs. 10a, 10b and 10c show the COM acceleration given by the robot Inertia Measurement Unit (IMU) for the experiment with the flexible foam. When we did not use the deformation estimator the robot fell down (the problem of the COM acceleration tracking is clearly identifiable around 4 s). When we used the deformation estimator the COM acceleration along the y-axis is close to the reference acceleration (see Fig. 10b). On the contrary, the COM acceleration along x-axis and z-axis are quite different from the reference acceleration (see Figs. 10a and 10c). These tracking errors arise from the use of the HRP-4's built-in stabilizer: a black box that adjust the desired robot state from the ones we compute and tuned for the original feet design. Hence, it does not take into account the new sole flexibilities. We also believe that few tracking errors result from some conflicting tasks and constraints of the multi-objective QP controller (see section 5).

We looked also at the ankle force to validate our control framework in Fig. 2. As we can see from Figs. 10e and 10f, that represent respectively the norm of horizontal plane ankle forces and the ankle force along z-axes, our controller estimates well the forces. As we can see in Fig. 10d, we also obtained ankle positions close to reference, which shows that the stabilizer does not need to change a lot the relative feet and waist positions to obtain stability, and this is thanks to the estimator.

The most important estimation of our new controller to enhance ZMP stabilization is given on foot orientations in Figs. 10g, 10h and 10i (in particular along x- and y-axes).

We can therefore state that our novel WPG and deformation estimator are a good model of the robot with flexible foam soles and enhances the native stabilizer to keep the desired robot attitude during the walking task.

6.3. Optimized HRP-4 walking with compliant soles for different ZMP trajectories

To find the best weight values in (4), we performed experiments on the HRP-4 with different weights using the new control framework in Fig. 2. Experiments consisted of humanoid walking with four different values of $\lambda = \{0.2, 0.4, 0.6, 0.8\}$ in (4), robot performed a straightforward walking for 10 footsteps (distance of 0.5 m) with average speed of 3.5 cm/s.

To choose experimentally the best weight λ of the cost function (4), we analyzed the ankle torques (Fig. 10j) and COM accelerations (Fig. 10k). The theoretical values are obtained from the WPG without taking into account the flexibility.

The Fig. 10j shows the right foot ankle torques for different values of the WPG (results are similar for the left foot). Theoretical ankle torques increase with λ since the weight given to the criterion dependent on the ankle torque decreases. Analyzing the Fig. 10j, we can make the following observations:

- For $\lambda = 0.2$, the mean torque is bigger than the theoretical one. In this case, the ZMP trajectory of each foot is shorter than for $\lambda > 0.2$ and it is close to the

[1] https://youtu.be/XByYfO3vcuc

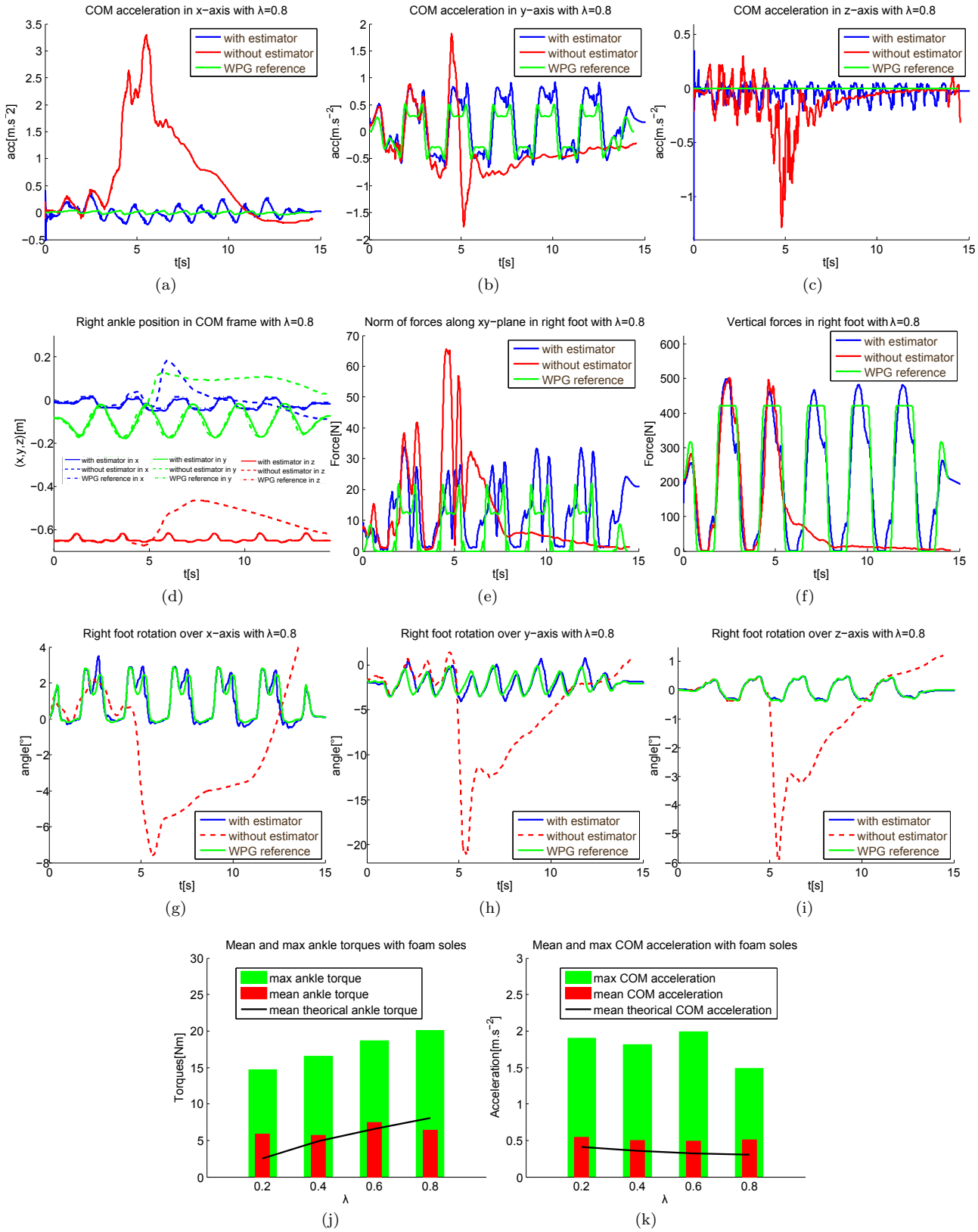


Figure 10: (a,b,c) COM acceleration compared to the reference acceleration along x/y/z axes with and without the deformation estimator; (d) Right ankle position along x/y/z axes; (e) Norm of horizontal plane ankle forces and (f) ankle force along z axes with and without the deformation estimator; (g,h,i) Ankle orientation along x/y/z axes with and without the deformation estimator; (j) Ankle torque for different cost functions; (k) COM acceleration for different cost functions.

ankle position (see Fig. 7a). This position of the ZMP must be more difficult to stabilize because it is the position of minimum sole flexibility when all the foam is equally (fully) compressed.

- From $\lambda = 0.4$ to $\lambda = 0.6$, mean torques are close to theoretical ones.
- For $\lambda = 0.8$, mean torques are lower than theoretical one. In this case, the ZMP trajectory of each foot is closer to the edge of the feet, which makes it difficult to move it further to the edge to stabilize the robot. Hence this walk is not as stable as for $\lambda = 0.4$ and $\lambda = 0.6$. Maximum ankle torques are then larger, but we have no explanation why the mean torque is lower than the theoretical one.
- Maximum ankle torques have a similar evolution to the theoretical mean ankle torques.

Fig. 10k shows the COM accelerations (which is also the image of COM force). In (4), λ mainly affect the COM acceleration in the lateral direction. Theoretical COM acceleration decreases with λ since the weight given to the minimization of the criterion dependents on the COM force increase. Analyzing the Fig. 10k, we can make the following observations:

- From $\lambda = 0.2$ to $\lambda = 0.6$, the evolution of COM acceleration is similar to the theoretical one, albeit with some variations due to the HRP-4 built-in stabilizer
- For $\lambda = 0.8$, the COM acceleration is not decreasing for the same reason as before; it is a case more difficult to stabilize.

Analyzing these results, we noticed that for $\lambda = 0.4$ and $\lambda = 0.6$ we obtained results closer to the theoretical expected ones as compared to $\lambda = 0.2$ and $\lambda = 0.8$, which might be due to problems of stability for extreme λ values.

7. Conclusion and future work

We investigated adding only flexible soles to a humanoid robot as compliant elements. Using a new control approach, we validate our approach on the HRP-4 robot. Experiments showed how our new WPG and deformation estimator enhances the ZMP stabilization during humanoid walking with flexible/soft soles. In the paper [14], we illustrate a new method to optimize the sole shape for improving stability and walking performances. In future work, we will use the controller illustrated in this paper to walk with the optimized sole.

The WPG minimizes the energy consumption and finds smooth heel-toe human-like ZMP force and trajectory under each foot.

We modeled the contact between the sole and the ground using a FEM model of the sole and a contact model accounting for both Signorini's and Coulomb's laws. These models are taken into account in the deformation estimator.

In near future work, we will develop an optimization that takes into account the WPG and the sole deformation at the same time. As a result, this will allow to obtain a more precise and optimized ankle movements. We also plan to design a control of COM with the sole flexibility model to better compensate for perturbations or uncertainty in the model.

8. Acknowledgments

This research is supported in part from the EU 7th Framework Program (FP7/2007-2013) under grant agreement No 611909 KoroiBot project `www.koroibot.eu`.

AppendixA. System to find COM trajectories

Equation (9) has $2m$ unknowns with $m = 3n+4$ phases. These unknowns satisfy the following boundary conditions for the COM position and velocity:

1. Initial

$$x^{(1)}(T_0) = V^{(1)} + A_0^{(1)} \tag{A.1}$$

$$\dot{x}^{(1)}(T_0) = W^{(1)} + A_1^{(1)} \tag{A.2}$$

2. Relationship between two successive sequences

$$VW^{(j)} + \sum_{i=0}^{5} A_i^{(j)} (\Delta T_j)^i = V^{(j+1)} + A_0^{(j+1)} \tag{A.3}$$

$$VW\omega^{(j)} + \sum_{i=1}^{5} iA_i^{(j)} (\Delta T_j)^{i-1} = W^{(j+1)}\omega_j + A_1^{(j+1)} \tag{A.4}$$

where

$$\begin{aligned} &VW^{(j)} = V^{(j)}C_j + W^{(j)}S_j \\ &VW\omega^{(j)} = V^{(j)}\omega_j S_j + W^{(j)}\omega_j C_j \\ &C_j = \cosh(\omega_j \Delta T_j), \quad S_j = \sinh(\omega_j \Delta T_j) \end{aligned}$$

3. Final

$$x^{(m)}(T_m) = VW^{(m)} + \sum_{i=0}^{5} A_i^{(m)} (\Delta T_m)^i \tag{A.5}$$

$$\frac{dx^{(m)}}{dt}(T_m) = VW\omega^{(m)} + \sum_{i=1}^{5} iA_i^{(m)} (\Delta T_m)^{i-1} \tag{A.6}$$

where $\Delta T_j = t_j - t_{j-1}$.

From the boundary conditions (A.1)-(A.5), the total conditions are $2m + 2$. Removing COM velocity conditions on initial and final phases (they are solved at the pattern optimization level), $2m$ conditions remain. The unknowns can be calculated then by the following system:

$$\mathbf{G} \cdot \mathbf{y} = \mathbf{N} \cdot \mathbf{x} + \mathbf{H} \cdot \mathbf{l} \tag{A.7}$$

where

$$\mathbf{y} = \begin{bmatrix} V^{(1)} \; W^{(1)} \; \cdots \; V^{(j)} \; W^{(j)} \; \cdots \; V^{(m)} \; W^{(m)} \end{bmatrix}^T$$
$$\mathbf{x} = \begin{bmatrix} x^{(1)}_{\mathrm{COM}}(t_0) \; \dot{x}^{(1)}_{\mathrm{COM}}(t_0) \; x^{(m+1)}_{\mathrm{COM}}(t_m) \; \dot{x}^{(m+1)}_{\mathrm{COM}}(t_m) \end{bmatrix}^T$$
$$\mathbf{l} = \begin{bmatrix} A^{(1)}_0 \; \cdots \; A^{(1)}_5 \; \cdots \; A^{(m)}_0 \; \cdots \; A^{(m)}_5 \end{bmatrix}^T$$
$$\mathbf{G}_{h,q} = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix} & \text{for } (h,q) = (1,1) \\ \begin{bmatrix} C_m & S_m \end{bmatrix} & \text{for } (h,q) = (m+2,m) \\ G_{1,h} & \text{for } q = h-1 \\ G_{2,h} & \text{for } q = h \\ 0 & \text{otherwise} \end{cases}$$
$$\mathbf{N}_{h,1} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} & \text{for } h = 1 \\ \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} & \text{for } h = m+2 \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} & \text{otherwise} \end{cases}$$
$$\mathbf{H}_{h,q} = \begin{cases} -\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \text{for } (h,q) = (1,1) \\ -\begin{bmatrix} (\Delta T_m)^0 & \cdots & (\Delta T_m)^5 \end{bmatrix} & \text{for } (h,q) = (m+2,m) \\ H_{1,h} & \text{for } q = h-1 \\ H_{2,h} & \text{for } q = h \\ 0 & \text{otherwise} \end{cases}$$

with $h = 1, \ldots, m+2$, $q = 1, \ldots, m$, $x^{(1)}_{\mathrm{COM}}(T_1)$ the initial COM position, $\dot{x}^{(1)}_{\mathrm{COM}}(T_1)$ the initial COM velocity, $x^{(m+1)}_{\mathrm{COM}}(T_{m+1})$ the final COM position and $\dot{x}^{(m+1)}_{\mathrm{COM}}(T_{m+1})$ the final COM velocity. $\mathbf{G}_{1,h} = \begin{bmatrix} C_j & S_h \\ \omega_h S_h & \omega_h C_h \end{bmatrix}$, $\mathbf{G}_{2,h} = \begin{bmatrix} -1 & 0 \\ 0 & -\omega_h \end{bmatrix}$,

$$\mathbf{H}_{1,h} = -\begin{bmatrix} (\Delta T_m)^0 & (\Delta T_m)^1 & \cdots & (\Delta T_m)^5 \\ 0 & (\Delta T_m)^0 & \cdots & (\Delta T_m)^4 \end{bmatrix},$$
$$\mathbf{H}_{2,h} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

AppendixB. System to find the distribution of force under the feet in DSP

Based on the simplified robot model used in [10], the relationship between $F^z_{\mathrm{ZMP}_1}$ and $F^z_{\mathrm{ZMP}_2}$ is:

$$\begin{aligned} F^{z(j)}_{\mathrm{ZMP}_1} + F^{z(j)}_{\mathrm{ZMP}_2} &= F^{z(j)}_{\mathrm{ZMP}} = Mg \\ F^{z(j)}_{\mathrm{ZMP}_1} &= Mg - F^{z(j)}_{\mathrm{ZMP}_2} \end{aligned} \tag{B.1}$$

where the reaction force $F^z_{\mathrm{ZMP}_2}$ in (B.1) is defined by:

$$F^{z(j)}_{\mathrm{ZMP}_2} = k^{(j)} \cdot F^{z(j)}_{\mathrm{ZMP}} = k^{(j)} \cdot Mg \tag{B.2}$$

k in (B.2) is a 5th order polynomial representing the distribution of ZMP force under each foot.

From (12), (B.1) and (B.2), we obtain:

$$x^{(j)}_{\mathrm{ZMP}_2} = x^{(j)}_{\mathrm{ZMP}_1} - \frac{1}{k^{(j)}}(x^{(j)}_{\mathrm{ZMP}_1} - x^{(j)}_{\mathrm{ZMP}}) \tag{B.3}$$

The boundary conditions that must be satisfied at the start of the DSP during a walking cycle are respectively a C^2 contact force and ZMP (due to the soft sole), an arbitrary new ZMP $x^{(j)}_{\mathrm{ZMP}_2}$ inside the new foot position and bounded $\dot{x}^{(j)}_{\mathrm{ZMP}_2}$ $\ddot{x}^{(j)}_{\mathrm{ZMP}_2}$:

$$\begin{aligned} & F^{z(j)}_{\mathrm{ZMP}_1} = F^{z(j)}_{\mathrm{ZMP}} = Mg, \; \dot{F}^{z(j)}_{\mathrm{ZMP}_1} = 0, \; \ddot{F}^{z(j)}_{\mathrm{ZMP}_1} = 0, \\ & x^{(j)}_{\mathrm{ZMP}_1} = x^{(j)}_{\mathrm{ZMP}} = x^{(j)}_B, \; \dot{x}^{(j)}_{\mathrm{ZMP}_1} = \dot{x}^{(j)}_{\mathrm{ZMP}} = \dot{x}^{(j)}_B, \\ & \ddot{x}^{(j)}_{\mathrm{ZMP}_1} = \ddot{x}^{(j)}_{\mathrm{ZMP}} = \ddot{x}^{(j)}_B \\ & x^{(j)}_{\mathrm{ZMP}_2} \neq x^{(j)}_B \text{ and bounded } \dot{x}^{(j)}_{\mathrm{ZMP}_2} \text{ and } \ddot{x}^{(j)}_{\mathrm{ZMP}_2} \end{aligned} \tag{B.4}$$

To satisfy (11) and (B.1) and their first and second time derivatives, (B.4) gives at the start of the DSP:

$$\begin{bmatrix} F^{z(j)}_{\mathrm{ZMP}_1} \\ \dot{F}^{z(j)}_{\mathrm{ZMP}_1} \\ \ddot{F}^{z(j)}_{\mathrm{ZMP}_1} \end{bmatrix} = \begin{bmatrix} Mg \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} F^{z(j)}_{\mathrm{ZMP}_2} \\ \dot{F}^{z(j)}_{\mathrm{ZMP}_2} \\ \ddot{F}^{z(j)}_{\mathrm{ZMP}_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.5}$$

In the same way we obtain at the end of the DSP:

$$\begin{bmatrix} F^{z(j)}_{\mathrm{ZMP}_1} \\ \dot{F}^{z(j)}_{\mathrm{ZMP}_1} \\ \ddot{F}^{z(j)}_{\mathrm{ZMP}_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} F^{z(j)}_{\mathrm{ZMP}_2} \\ \dot{F}^{z(j)}_{\mathrm{ZMP}_2} \\ \ddot{F}^{z(j)}_{\mathrm{ZMP}_2} \end{bmatrix} = \begin{bmatrix} Mg \\ 0 \\ 0 \end{bmatrix} \tag{B.6}$$

Equations (B.5) and (B.6) are the initial and final conditions to interpolate k in (B.3). k is a C^2 function which varies from 0 to 1.

To define the function k during *start* sequence, we took into consideration experimental studies on human walking [24]. During *start* sequence, humans move their COM laterally to be under the stance limb (we noted this foot '2'). The starting sequence begins with an equal load distribution in each limb under each foot. Then, the lateral movement is induced by a momentary load of the swing limb (we noted this foot '1'). At the final part of the walking phases, the stance limb is fully loaded and the swing limb is unloaded to begin the first SSP. Using these considerations, we chose the function k equal to 0.5 on C'_1, 0.25 on C''_1 and 1 on C_1.

The human behavior during *stop* sequence is symmetric to *start* sequence [24]. During the *stop* sequence, humans move laterally their COM and decelerate their movement. This lateral movement begins with a full load of the stance limb (foot 1) from the end of the SSP. Then, the deceleration is induced by a momentary load of the swing limb (foot 2). The stopping sequence ends with the same load distribution on each limb (under each foot in our case) [24]. For this reason, we chose the function k equal to 0 on B_{n+2}, 0.75 on B''_{n+2} and 0.5 on B'_{n+2}.

References

[1] K. Bouyarmane, A. Kheddar, Humanoid robot locomotion and manipulation step planning, Advanced Robotics (26) (2012) 1099–1126, doi:\bibinfo{doi}{10.1080/01691864.2012.686345}.

[2] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of honda humanoid robot, in: IEEE International Conference on Robotics and Automation, vol. 2, IEEE, Leuven, Belgium, 1321–1326, 1998.

[3] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, T. Isozumi, Humanoid robot hrp-2, in: IEEE International Conference on Robotics and Automation, New Orleans, USA, 1083–1090, 2004.

[4] O. Bruneau, F. B. Ouezdou, J.-G. Fontaine, Dynamic walk of a bipedal robot having flexible feet, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, IEEE, Maui, USA, 512–517, 2001.

[5] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, F. Kanehiro, Multi-contact vertical ladder climbing with an HRP-2 humanoid, Autonomous Robots 40 (3) (2016) 561–580, ISSN 1573-7527, doi:\bibinfo{doi}{10.1007/s10514-016-9546-4}.

[6] J. Yamaguchi, A. Takanishi, Multisensor foot mechanism with shock absorbing material for dynamic biped walking adapting to unknown uneven surfaces, in: IEEE International Conference on Multisensor Fusion and Integration for Intelligent systems, Washington DC, USA, 233–240, 1996.

[7] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, M. Diehl, Online Walking Motion Generation with Automatic Foot Step Placement, Advanced Robotics 24 (5-6) (2010) 719–737.

[8] S. Kajita, B. Espiau, Handbook of Robotics, chap. Legged Robots, Springer-Verlag, 361–389, 2008.

[9] H.-K. Shin, B. K. Kim, Energy-Efficient Gait Planning and Control for Biped Robots Utilizing the Allowable ZMP Region, Robotics, IEEE Transactions on 30 (4) (2014) 986–993, ISSN 1552-3098, doi:\bibinfo{doi}{10.1109/TRO.2014.2305792}.

[10] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, S. Nakaoka, H. Hirukawa, A Biped Pattern Generation Allowing Immediate Modification of Foot Placement in Real-time, in: IEEE-RAS International Conference on Humanoid Robots, 581 – 586, 2006.

[11] P.-B. Wieber, On the stability of walking systems, in: Proceedings of the international workshop on humanoid and human friendly robotics, 2002.

[12] M. B. Popovic, A. Goswani, H. Herr, Ground reference points in legged locomotion : definitions, biological trajectories and control implications, International Journal of Robotics Research 24 (12) (2005) 1013–1032.

[13] G. De Magistris, A. Pajon, S. Miossec, A. Kheddar, Humanoid walking with compliant soles using a deformation estimator, in: International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016.

[14] G. De Magistris, S. Miossec, A. Escande, A. Kheddar, Design of optimized soft soles for humanoid robots, Robotics and Autonomous Systems (RAS) (2017) 1.

[15] M. Bonnet, A. Frangi, C. Rey, The finite element method in solid mechanics, McGraw Hill Education, URL `https://hal.archives-ouvertes.fr/hal-01083772`, 2014.

[16] M. Bro-Nielsen, S. Cotin, Real-time volumetric deformable models for surgery simulation using finite elements and condensation, Computer Graphics Forum (Eurographics'96) 15 (1996) 17.

[17] J.-J. Moreau, M. Jean, Numerical Treatment of Contact and Friction: The Contact Dynamics Method, Engineering Systems Design and Analysis 4 (1) (1996) 201–208.

[18] M. Jean, Numerical methods for three dimensional dynamical problems, in: Conference in Contact Mechanics, Southampton, 71, 1993.

[19] P. Alart, A. Curnier, Time-stepping for threedimensal rigid body dynamics, Computer Methods in Applied Mechanics and Engineering 92 (3) (1991) 353–375.

[20] C. Duriez, F. Dubois, A. Kheddar, C. Andriot, Realistic haptic rendering of interacting deformable objects in virtual environments, IEEE Transactions on Visualization and Computer Graphics 12 (1) (2006) 36–47.

[21] F. Jourdan, P. Alart, M. Jean, A gauss-seidel like algorithm to solve frictional contact problems, Computer Methods in Applied Mechanics and Engineering (1998) 33–47.

[22] K. Harada, S. Kajita, K. Kaneko, H. Hirukawa, An analytical Method on Real-time Gait Planning for a Humanoid Robot, International Journal of Humanoid Robotics .

[23] P.-B. Wieber, Trajectory free linear model predictive control for stable walking in the presence of strong perturbations, in: IEEE-RAS International Conference on Humanoid Robots, 137 – 142, 2006.

[24] D. A. Winter, Human balance and posture control during standing and walking, Gait and Posture 3 (1995) 193–214.

[25] A. Pajon, G. De Magistris, S. Miossec, K. Kaneko, A. Kheddar, A humanoid walking pattern generator for sole design optimization, in: International Conference on Advanced Robotics (ICAR), Istanbul, Turkey, 2015.