Design of optimized soft soles for humanoid robots

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

Design of Optimized Soft Soles for Humanoid Robots

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

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

Abstract

We describe a methodology to design foot soles for a humanoid robot given walking gait parameters (i.e. given center-of-mass and zero-moment-point trajectories). In order to obtain an optimized compliant sole, we devised a shape optimization framework which takes –among other inputs, an initial rough (simplified) shape of the sole and refines it through successive optimization steps under additional constraints and a cost function. The shape is optimized based on the simulation of the sole deformation during an entire walking step, taking time dependent input of the walking pattern generator into account. Our shape optimization framework is able to minimize the impact of the foot with the ground during the heel-strike phase and to limit foot rotation in case of perturbations. Indeed, low foot rotation enforces a vertical posture and secures the balance of the humanoid robot. Moreover, weight restriction (formulated as a constraint on the sole volume) is added to our optimization problem.

Keywords: Shape Optimizations, Compliant soles, Humanoid robots, Walking

1. Introduction

Bipedal walking on flat ground is a well-documented and thoroughly studied problem in the humanoid research field. An excellent review is presented in [1]. Recent studies show that humanoid walking on flat floors reached a good level of maturity. Indeed, humanoid robots can walk with human-in-loop [2], with human stylized gaits [3, 4] or with emotional behavior gaits [5]. In the latter works, the precise control of the contact interaction between the humanoid robot feet and the ground plays an important role. Reducing the impact at the landing is important to perform smooth walking gaits. For example, reducing impacts and having smooth walking is critical in human-in-the-loop walking under hard visual servoing constraints [6] or for precise stylized or emotional walking gaits.

To reduce the impact of the foot landing, different studies illustrate the importance of the foot shape design. Adamczyk *et al.* [7] show that in a human passive walking locomotion, the energy loss during heel contact with the ground varies with (i) the foot shape, and (ii) the first contact point between the foot and the ground [8]. They explain how a rolling foot shape considerably reduces impact-shock when the heel touches the ground. Recently, Fukushima *et al.* [9] show that foot shape, once optimized, allows passive walking on various slopes. Miyakoshi and Cheng [10] propose a model for biped walking with a curved surface foot (using an array of point contacts). This model improves the stability of walking and reduces impact forces during foot contact maintaining the same human-like walking characteristics. Aerts *et al.* [11] found by experimental measurements that the shock absorption in humans happens at the foot pads. In most people, the heel pad is 25mm thick, and compresses by 12mm during stance. To make the foot more controllable, the material is hyperelastic –it starts out being compliant, and then stiffen during stance. Furthermore, Sardain and Bessonnet [12] show how the sole mechanism influences the stabilization of biped robot walking.

The large majority of humanoid robots are embedded with force sensing devices at the ankles. In order to protect the robot structure and the force sensors from high impact forces during landing phases, two types of shock-absorbing mechanisms are commonly used.

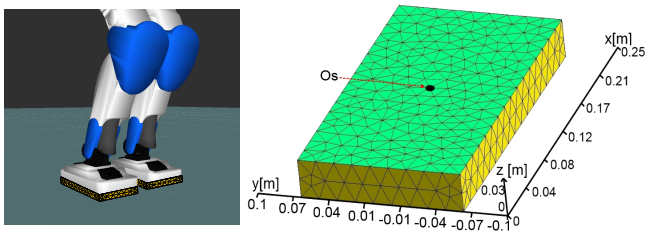
The first is an *internal passive mechanism.* This type of shock-absorbing mechanism is present in the feet of many humanoid robots such as the Honda's Asimo [13] and the Kawada's HRP families [14]. The Asimo's shock-absorbing system smooths the impact force deforming rubber bushing inserted into guides. HRP robots have three rubber bushing with dampers that are attached between the foot and the leg. Unfortunately, this non-measurable compliance makes the attitude of the robot difficult to control, especially in complex maneuvers [15]. For this reason, a dedicated software-compensator is necessary to stabilize the walking [16, 17]. Bruneau *et al.* [18] explain the difficulty in modeling and controlling these flexibilities.

The second shock-absorbing system consists in using *external passive mechanisms.* Yamaguchi *et. al* use a multi-sensor foot mechanism with shock absorbing material [19].

*Corresponding author:
Email address: giovanni_demagistris@hotmail.it (G. De Magistris)

The foot sensors detect the path surface during the landing phase, which is combined with an adaptive biped walking control method and a flexible material to reduce the impact forces. Of course, combination of both is possible. In fact, all humanoid robots have a thin patterned rubber sole mounted (glued) on the part of their feet in contact with the ground. Because it is thin, large part of the landing shocks are absorbed at the ankle compliance.

In this paper, we aim at designing an optimized compliant sole added under the rigid structure of each foot to reduce the impact force at the landing phase, while accounting for the difficulty of the control after the landing phase (by increasing rotational stiffness). In addition, the soft soles must also be designed to absorb landing impact shocks and ground unevenness with a better contact-surface casting. During walking, the sole's compliance depends on the contact area variations. This external flexible sole can also offer other benefits such as better adherence properties and adapting, through deformation, to the terrain irregularities. We already mentioned this problem and reported very preliminary results in [20], Fig. 1. We decided to pursue this idea further.



(a) (b)

Figure 1: (a): Rectangular parallelepiped soles mounted on HRP-4's feet; (b): Initial sole shape where O_s is the center of the surface attached to the foot

Our long-term objective is to optimize the sole together with the parameters of the walking pattern generator (WPG) to design soles that can adapt (in form and stiffness) to specific conditions and prerequisite needs for walking. In order to achieve this goal, we decided to proceed in three phases:

1. devise an approach by which we optimize the sole shape for given walking parameters and prerequisites,
2. extend the latter approach to combine tuning the walking parameters and sole shapes in a single optimization problem,
3. use the latter approach to obtain an optimized sole shape for different terrain conditions and requests.

This paper deals with the first phase of our work-plan. In Sec. 2, we explain our framework; in Sec. 3, we illustrate our optimization problem; in Sec. 4 and 5, we detail each module of the simulation framework and the methods used to define the sole geometry and the mechanical properties of the shape optimization problem; in Sec. 6 we describe the cost function of our optimization process; in Sec. 7 we present some solutions obtained using our approach.

2. Sole design framework

Shape optimization is very important in different engineering disciplines. When load carrying efficiency is tightly linked to the shape, the optimization of the structure is necessary. The common goal of a shape optimization problem is to find an object shape that minimizes an objective function and satisfies some constraints [21, 22]. Many examples show the importance of shape optimization: for example the design of fillets, structures with holes [23], general 2-D bodies [24], curvilinear-stiffened panels [25], hull design [26], electromagnetics [27], and automotive aerodynamics [28]. Particularly relevant are the studies on aerospace engineering [29]. These shape optimization problems adopt different approaches to find optimal design of statically-loaded elastic structures. In our study, the force applied to the structure/object (i.e. the sole) depends on both time and space. The simulation of the foot/sole movement in a specific instant of time depends on the previous ones due to the sliding friction. To take into account these factors, we developed a dedicated shape optimization framework (Fig. 2) that optimizes the sole shape for a given walking pattern parameters, see Fig. 1a. Our aim is to find the sole's shape that minimizes a certain cost function computed over the simulation of the foot movement during the contact phase, while bounding the volume of the sole.

Recently, adaptive or refinement-based parametrization have been developed to improve the efficiency of shape optimization algorithms [30, 31, 32, 33]. Zingg *et al.* [34] show that gradient-based optimization is more efficient than gradient-free optimization in aerodynamics. This difference becomes more important with a higher number of design variables. In our optimization problem, the decision variables (i.e. the parameters) are the spline coefficients describing the sole deformation. The number of these coefficients is chosen by the user. Since we already envision extending our approach to a single optimization problem of the sole shape and the walking pattern generator at the same time, we expect the number of parameters to increase significantly. For this reason, we use a gradient-based formulation of the sole optimization problem.

The Fig. 2 illustrates our framework. The sole shape and the walking pattern are optimized using two distinct modules (Sec. 1). The Zero Moment Point (ZMP) position and force obtained from the walking pattern generator are the input to the simulation of the movement and the deformation. The simulation module will be detailed later.

3. Problem formulation

Our objective is to obtain a sole shape Ω, seen as a connected subset of $\mathbb{R}^3$, which lowers at best (i) the impact force during the heel-strike phase (shock-absorbing),

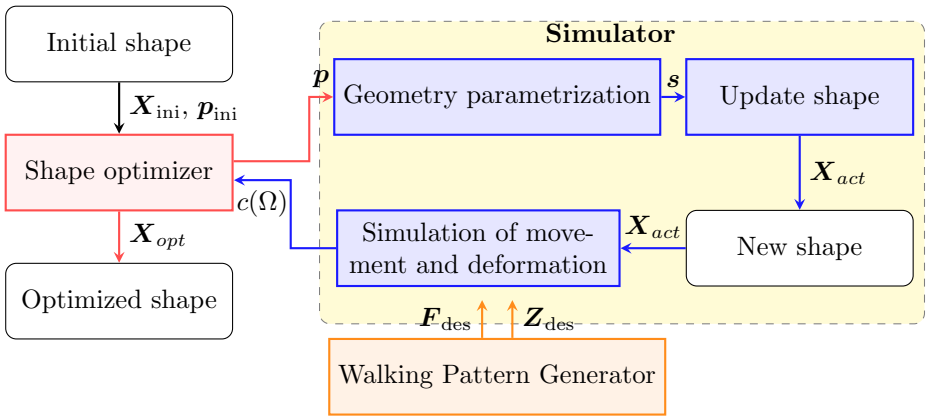


Figure 2: Framework to optimize the sole shape for the given walking parameters. $\boldsymbol{X}_{\text{ini}}$, $\boldsymbol{X}_{act}$ and $\boldsymbol{X}_{opt}$ are respectively the initial, actual (during shape optimization process) and optimized sole nodes positions defining its shape. $\boldsymbol{p}_{\text{ini}}$ and $\boldsymbol{p}$ are respectively the initial and intermediate (during shape optimization) sole shape parameters. $\boldsymbol{s}$ is the function transforming $\boldsymbol{X}_{\text{ini}}$ in $\boldsymbol{X}_{act}$. The Walking Pattern Generator (also formulated as an optimization problem) is an independent module that return the desired force $\boldsymbol{F}_{\text{des}}$ and ZMP positions $\boldsymbol{Z}_{\text{des}}$ as input to the simulation of the sole movement and deformation. The geometry parametrization module describes the shape of sole. The Shape optimizer finds numerically minimums of the cost function $c(\Omega)$.

and (ii) the foot rotation during the whole foot movement, while tracking a given resultant wrench of support reaction. We formulate this goal as an optimization program.

We specify a pair of functions $(\boldsymbol{F}_{\text{des}}(t), \boldsymbol{Z}_{\text{des}}(t))$ representing the total contact forces and ZMP position over the time interval $[t_0, t_f]$ during which we want the sole to be in contact with the floor. We defined the cost function as follows, using the Cartesian stiffness matrix K_c:

$$\begin{aligned} c(\Omega) &= \int_{t_0}^{t_h} \sigma_{\max}\left(K_c^{\text{tr}}(\Omega,t)\right)\, dt - w \int_{t_0}^{t_f} \sigma_{\min}\left(K_c^{\text{rot}}(\Omega,t)\right)\, dt \\ &\overset{\text{def}}{=} c^{\text{tr}} - w\, c^{\text{rot}} \end{aligned} \tag{1}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ denotes respectively the largest and smallest singular value of a matrix, which implies we optimize the stiffness in the worst direction. The two terms correspond respectively to:

1. minimization of the translational part of the Cartesian stiffness matrix K_c (maximize the compliance) during the heel-strike phase to reduce the impact force, and
2. maximization of the rotational part of the Cartesian stiffness matrix K_c during the whole foot movement to decrease foot rotation in case of perturbations. Low foot rotation enforces the vertical posture and secures the balance of the robot during walking.

t_h is the end of the heel-strike phase that we define as lasting 10% of the whole contact phase, when the vertical component of force almost reached its maximum, that is at the end of double support. Note that c and K_c are parametrized by $\boldsymbol{F}_{\text{des}}$ and $\boldsymbol{Z}_{\text{des}}$.

We also specify a maximum volume $v_{\max}$ (and hence the weight) for the sole, otherwise we would obtain very large and bulky soles. We thus have the following program:

$$\min_{\Omega} \quad c(\Omega), \text{ s.t. } v(\Omega) \leq v_{\max} \tag{2}$$

where $v(\Omega)$ denotes the volume of the sole.

4. Sole description

To make the problem (2) tractable, we need to describe the shape Ω with a finite number of parameters, and to simulate the movement and deformations undergone by a given shape. We do so by choosing a specific parametrization of the shape Ω, using Finite Element Model (FEM) to include the mechanical properties of the sole and a special algorithm to simulate the movement and deformations.

In the remaining of this paper, we divide the surface $\partial\Omega$ into two non-empty disjoint parts:

$$\partial\Omega = S_D \cup S_S, \quad \text{with} \quad S_D \cap S_S = \emptyset \tag{3}$$

S_D is the surface of the sole attached to the rigid part of the foot (Dirichlet surface) and S_S is the surface of the sole that could be in contact with the ground. We also denote by $\Omega_I := \Omega \setminus \partial\Omega$ the interior volume of Ω.

4.1. Geometry parametrization

An important part of the shape optimization problem is the choice of basis in the parametrization. In this paper, we choose to describe the sole geometry by giving an initial shape that is deformed according to a function parametrized by a 2D spline. The coefficients of this spline are the optimization decision variables. The obtained shape is used to simulate the whole foot movement imposed by the walking pattern generator outputs.

Therefore, we describe Ω by (i) an initial volume Ω_0 and (ii) a deformation of this volume according to a parametrized function s. Fig. 3 gives a 2D illustration of this process.

The shape Ω_0 is discretized into finite elements and we define a point O_s on S_D (Fig. 1b), typically at its center.

To each node X_i of the FEM with Cartesian coordinates (x_i, y_i, z_i) in a frame attached to O_s, we associate spherical coordinates $(r_i, \theta_i, \varphi_i)$ with the following convention

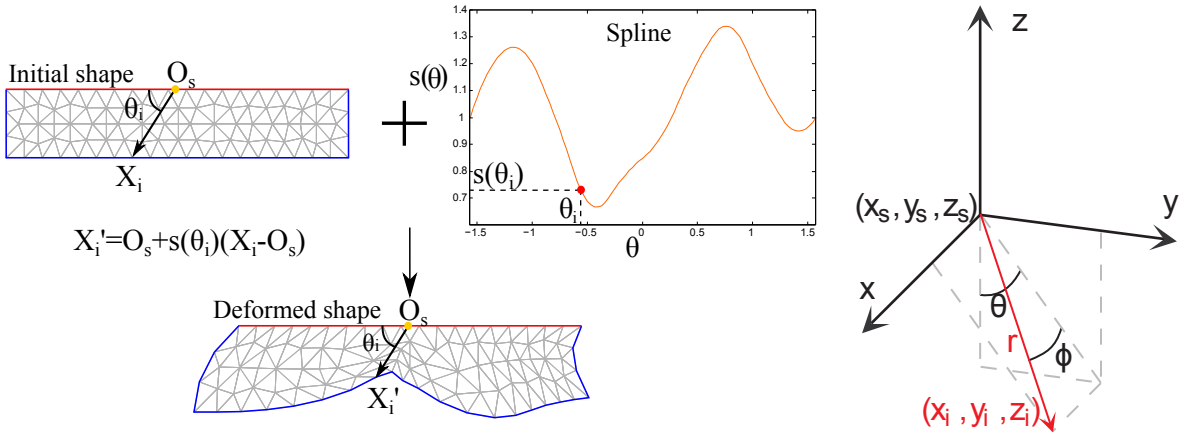


Figure 3: Geometry parametrization module in 2D case at the left; convention for the 3D spherical coordinates at the right.

(Fig. 3):

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

$$\theta_i = \text{atan2}\left(y_i, -z_i\right), \ \varphi_i = \text{atan2}\left(x_i, \sqrt{y_i^2 + z_i^2}\right)$$

Then the coordinates of the node after deformation are

$$X_i' = s(p, \theta_i, \varphi_i) X_i \tag{4}$$

We define s as a 2D B-spline with parameters p:

$$s(p, \theta, \varphi) = \sum_{j=0}^{n_\theta} \sum_{k=0}^{n_\varphi} p_{j,k} N_j(\theta) N_k(\varphi) \tag{5}$$

where the N_j and N_k are B-spline basis function of degree 2 scaled appropriately so that s is spatially defined on $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2$. The variables n_θ and n_φ are user-defined; they control the precision of the spline. The spherical mapping has the singular points (the poles) for $\phi = \pm\frac{\pi}{2}$. In proximity of the poles, close points X_i have very different θ_i and thus may have very different images X_i' through equation (4). To avoid this problem, we impose that in proximity of the pole coordinates the spline function is constant, by constraining the $p_{j,0}$ (respectively n_φ) to have the same value.

We used a spherical parametrization, rather than a Cartesian one, since it allows a 2D shape parametrization of the surface, rather than a 3D one. The Cartesian parameterization would then have more parameters and would be redundant (tangential deformations to the surface of the body having no effect on the shape).

To avoid that the sole size is too large or too small, we bound the shape optimization parameters $p_{j,k}$.

4.2. Finite Element Model

To simulate the deforming motion given by the flexible sole and obtain a desired ZMP force and position, we use FEM and contact modeling from mechanical laws (Sec. 5).

The simplest FEM is the static linear elastic model under the assumption of small deformations [35]. This model is valid when the node displacements w.r.t the object size is relatively low and the sole deformation response is fast enough. In this paper, we assume that under a given force, the dynamic sole deformations are faster than the external dynamic forces and that the sole has small deformations.

In the static case for linear elasticity, a characteristic stiffness matrix is extracted from each element. These elementary matrices are then assembled in a large stiffness matrix K of the sole.

Once discretized by the finite element method, the displacements at each point of the mesh depend on external forces applied to the same points. Therefore, we can write the following linear relationship:

$$\boldsymbol{K}\ {}^l\underline{\boldsymbol{U}} = {}^l\underline{\boldsymbol{F}} \tag{6}$$

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

The displacement of the Dirichlet nodes are known and for this reason, we can write the system (6) as:

$$\begin{bmatrix} \boldsymbol{K}_{DD} & \boldsymbol{K}_{DI} & \boldsymbol{K}_{DS} \\ \boldsymbol{K}_{ID} & \boldsymbol{K}_{II} & \boldsymbol{K}_{IS} \\ \boldsymbol{K}_{SD} & \boldsymbol{K}_{SI} & \boldsymbol{K}_{SS} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{0}} \\ {}^l\underline{\boldsymbol{U}_I} \\ {}^l\underline{\boldsymbol{U}_S} \end{bmatrix} = \begin{bmatrix} {}^l\underline{\boldsymbol{F}_D} \\ {}^l\underline{\boldsymbol{F}_I} \\ {}^l\underline{\boldsymbol{F}_S} \end{bmatrix} \tag{7}$$

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

To simplify the mechanical model of sole, as in [36], we add that internal nodes are not subject to external forces ${}^l\underline{\boldsymbol{F}_I} = \underline{0}$. Therefore, keeping the two last lines, the system becomes:

$$\begin{bmatrix} \boldsymbol{K}_{II} & \boldsymbol{K}_{IS} \\ \boldsymbol{K}_{SI} & \boldsymbol{K}_{SS} \end{bmatrix} \begin{bmatrix} {}^l\underline{\boldsymbol{U}_I} \\ {}^l\underline{\boldsymbol{U}_S} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{0}} \\ {}^l\underline{\boldsymbol{F}_S} \end{bmatrix} \tag{8}$$

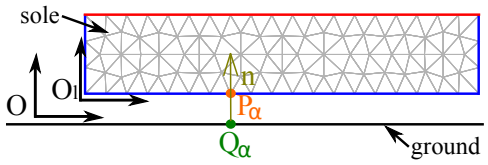


Figure 4: Contact between the sole and the ground.

From (8), we can write a condensed linear elasticity law [36], to be used in first analysis of our algorithm:

$$\begin{aligned} {}^l\underline{\boldsymbol{U}_S} &= \left(\boldsymbol{K}_{SS} - \boldsymbol{K}_{SI}\boldsymbol{K}_{II}^{-1}\boldsymbol{K}_{IS}\right)^{-1} {}^l\underline{\boldsymbol{F}_S} \\ &= \boldsymbol{K}_S^{-1}\, {}^l\underline{\boldsymbol{F}_S} \\ {}^l\underline{\boldsymbol{U}_I} &= -\boldsymbol{K}_{II}^{-1}\boldsymbol{K}_{IS}\, {}^l\underline{\boldsymbol{U}_S} \end{aligned} \tag{9}$$

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

As in [36], we consider that external forces apply to the sole surface nodes only. Hence, to find the displacement of each nodes, we divide the simulation into two steps:

1. find the displacement ${}^l\underline{\boldsymbol{U}_S}$ induced by the contact with the ground (Sec. 5), and then
2. compute the displacement ${}^l\underline{\boldsymbol{U}_I}$ using the second equation in (9).

5. Contacts

We describe how we model the contact between the sole and the (flat) ground. To model the contact, we use a relationship between the two unknowns of the system (9): the contact force and the relative position between the sole node position and the ground position (Fig. 4). This relative position is called contact space displacement vector $\underline{\boldsymbol{\delta}}$. For a point contact α, the normal component of $\boldsymbol{\delta}$ is:

$$\boldsymbol{\delta}_{n,\alpha} = \mathbf{n}\,(\boldsymbol{P}_\alpha - \boldsymbol{Q}_\alpha) \tag{10}$$

where $\mathbf{n} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$ is the normal vector.

We denote with the subscript n the normal component of the vector $\boldsymbol{v}_n = \mathbf{n}\boldsymbol{v}$ and with the subscript t the tangential components of the vector $\boldsymbol{v}_t = \mathbf{t}\boldsymbol{v}$, where $\mathbf{t} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$.

We need to compute the forces $\underline{\boldsymbol{F}}$ applying to the contact points $\underline{\boldsymbol{P}}$, which impose non-interpenetration between the sole and the ground. We use the two following laws:

1. **Signorini's law** establishes for each contact node α the complementarity between $\boldsymbol{F}_\alpha$ and $\boldsymbol{\delta}_\alpha$ along each contact normal [37]:

$$\mathbf{0} \le \boldsymbol{F}_{n,\alpha} \perp \boldsymbol{\delta}_{n,\alpha} \ge \mathbf{0} \tag{11}$$

2. **Coulomb's law** characterizes the dry friction. If $\boldsymbol{F}_\alpha$ is the contact force applied to the contact node α and μ the friction coefficient, we can distinguish two complementarity conditions for *stick* and *slip* motions:
   (a) $\|\boldsymbol{F}_{t,\alpha}\| < \mu|F_{n,\alpha}|$ and $\boldsymbol{\delta}_{t,\alpha} = 0$ (**stick condition**)
   (b) $\boldsymbol{F}_{t,\alpha} = -\mu F_{n,\alpha}\frac{\boldsymbol{\delta}_{t,\alpha}}{\|\boldsymbol{\delta}_{t,\alpha}\|}$ (**slip condition**)

Based on the framework in Fig. 2, we want to find a possible position and orientation of the sole together with the configuration of the contact points to obtain $\boldsymbol{F}_{\text{des}}$ and $\boldsymbol{Z}_{\text{des}}$ given by the walking pattern generator. Our algorithm is illustrated in Fig. 5.

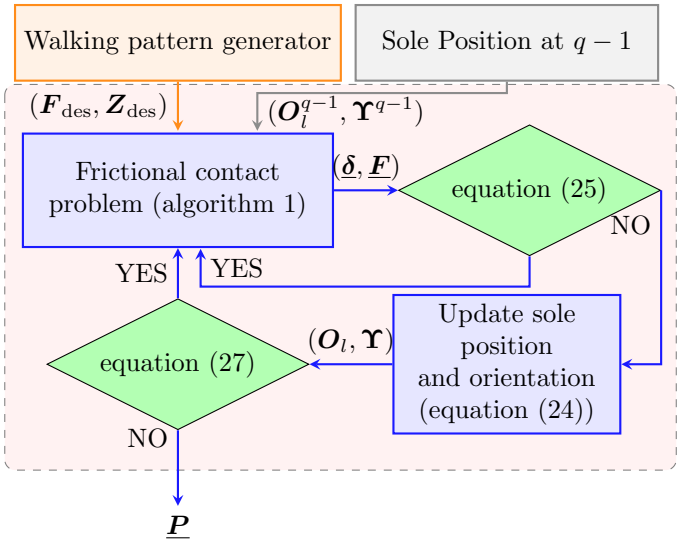


Figure 5: Framework for contact handling to obtain $\boldsymbol{F}_{\text{des}}$ and $\boldsymbol{Z}_{\text{des}}$. $q-1$ is the previous time step.

5.1. Signorini's law

To find the contact space displacements and forces, we can formulate the contact problem as a linear complementarity problem (LCP). Various methods, direct or iterative are possible to solve LCP [38]. For our applications, we use an iterative Gauss-Seidel solver based on [39]. Using this contact resolution, LCP takes into account the coupling between the different contact points.

Given the foot orientation and position defined respectively by three angles $\boldsymbol{\Upsilon} = (\theta, \phi, \psi)$ and a vector $\boldsymbol{O}_l$, the sole surface node positions in the absolute frame $\underline{\boldsymbol{P}}^{\text{free}}$ are given by the following relationship (Fig. 4):

$$\underline{\boldsymbol{P}}^{\text{free}} = [\boldsymbol{O}_l \cdots \boldsymbol{O}_l]^T + \boldsymbol{H}\left[{}^l\boldsymbol{P}_1^{\text{free}} \cdots {}^l\boldsymbol{P}_m^{\text{free}}\right]^T \stackrel{\text{def}}{=} \underline{\boldsymbol{O}_l} + \boldsymbol{H}\,{}^l\underline{\boldsymbol{P}}^{\text{free}} \tag{12}$$

where ${}^l\underline{\boldsymbol{P}}^{\text{free}}$ is the vector of the free node positions in the sole frame. $\boldsymbol{H} = \text{diag}(\boldsymbol{R}, \ldots, \boldsymbol{R})$, where $\boldsymbol{R}$ is the foot orientation matrix defined by three Euler rotation angles $\boldsymbol{\Upsilon} = (\theta, \phi, \psi)$. The subscript m is the number of nodes of suface S_S of the sole that can potentially be in contact with the ground. The superscript T denotes the transpose operator.

When the sole comes into contact with the ground, we can express the displacement $\underline{\boldsymbol{U}_S}$ as:

$$\underline{\boldsymbol{U}_S} = \underline{\boldsymbol{P}} - \underline{\boldsymbol{P}}^{\text{free}} \tag{13}$$

Using the linear relationship (9) tranformed in the global frame, the position of each surface node is composed of two terms:

$$\underline{\boldsymbol{P}} = \boldsymbol{H}\boldsymbol{K}_S^{-1}\boldsymbol{H}^T\underline{\boldsymbol{F}} + \underline{\boldsymbol{P}}^{\text{free}} \stackrel{\text{def}}{=} \boldsymbol{W}\underline{\boldsymbol{F}} + \underline{\boldsymbol{O}_l} + \boldsymbol{H}\ {}^l\underline{\boldsymbol{P}}^{\text{free}} \tag{14}$$

$\boldsymbol{W}$ is the Delassus operator [40].

By definition of $\boldsymbol{\delta}_\alpha$, linking a surface node to a reference point $\boldsymbol{Q}_\alpha$ on the ground, we can write:

$$\underline{\boldsymbol{\delta}} = \underline{\boldsymbol{P}} - \underline{\boldsymbol{Q}} = \boldsymbol{W}\underline{\boldsymbol{F}} + \underline{\boldsymbol{O}_l} + \boldsymbol{H}\ {}^l\underline{\boldsymbol{P}}^{\text{free}} - \underline{\boldsymbol{Q}} \stackrel{\text{def}}{=} \boldsymbol{W}\underline{\boldsymbol{F}} + \underline{\boldsymbol{\delta}}^{\text{free}} \tag{15}$$

where $\underline{\boldsymbol{Q}} = \begin{bmatrix}\boldsymbol{Q}_1 \dots \boldsymbol{Q}_m\end{bmatrix}$.

Once $\underline{\boldsymbol{\delta}}$ is computed as presented next, ${}^l\underline{\boldsymbol{U}_S}$ is obtained by:

$${}^l\underline{\boldsymbol{U}_S} = \boldsymbol{H}^T\underline{\boldsymbol{U}_S} = \boldsymbol{H}^T(\underline{\boldsymbol{\delta}} + \underline{\boldsymbol{Q}}) - {}^l\underline{\boldsymbol{P}}^{\text{free}} \tag{16}$$

Then, the normal component of the displacement vector (15) for each contact point α is:

$$\begin{aligned}\boldsymbol{\delta}_{n,\alpha} &= \mathbf{n}\boldsymbol{O}_l + \mathbf{n}\boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{n}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) \\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}_{n,\alpha}^{\text{free}} + \mathbf{n}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta)\end{aligned} \tag{17}$$

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. Q_α does not appear in (17) since we consider the absolute frame on the flat floor, and hence $\mathbf{n}\boldsymbol{Q}_\alpha = 0$.

Using (11) and (17), the Signorini's problem can be solved using a LCP in the frictionless case.

5.2. Coulomb's law

We now explain how we add the Coulomb's law. In the 3D case, this law adds a non-linearity to the initial contact problem. To solve friction contact cases, two approaches are commonly used in the literature: (i) keeping the problem linear by using a multi-sided pyramid approximates of the friction cones [38] and (ii) try solving the non-linear problem like in [41].

Using the first method, we can stack multiple contacts with friction in a global LCP. However, if we choose to model the friction with pyramids of 4 facets, each contact generates six new lines in the LCP and the computational cost of the problem can become very high. In addition, the accuracy is very low with 4 facets pyramids.

The non-linear approach in [41] is based on the Newton's method. For the paper to be self-contained, we explain this method for one node in contact (we also derive gradients from them later-on).

From (15), we find the tangential component of the contact space displacement:

$$\begin{aligned}\boldsymbol{\delta}_{t,\alpha} &= \mathbf{t}\boldsymbol{O}_{l,\alpha} + \mathbf{t}\boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) - \mathbf{t}\boldsymbol{Q}_\alpha \\ &\approx \mathbf{t}\boldsymbol{O}_{l,\alpha} + \mathbf{t}\boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) - \mathbf{t}\boldsymbol{P}_\alpha^{(q-1)} \\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}_{t,\alpha}^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta)\end{aligned} \tag{18}$$

where $\mathbf{t}$ is the tangential vector, $\boldsymbol{P}_\alpha^{(q-1)}$ is the node position in absolute frame at the previous time step and q is the current time step. In case of a node already in contact at $q-1$, the contact reference point is the previous node position $\boldsymbol{P}_\alpha^{(q-1)}$. In case of a new contact, we project on the ground the node position $\boldsymbol{P}_\alpha^{(q-1)}$ as an approximation of the reference contact position. In fact, real contact takes place at an intermediate time between previous and current time step, and a node may have moved a little in the tangent direction. Precise contact position could be found, but at the price of a variable step size and event driven simulation, which is computationally costly when many contacts occur. Additionally, this approximation is good enough when the time-step size is sufficiently small.

From (17) and (18), we obtain:

$$\begin{aligned}\boldsymbol{\delta}_\alpha &= \boldsymbol{O}_l + \boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} - \begin{bmatrix}\boldsymbol{P}_{t,\alpha}^{(q-1)} \\ 0\end{bmatrix} + \sum_{\beta=1}^{m}\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta \\ &\stackrel{\text{def}}{=} \boldsymbol{\delta}_\alpha^{\text{free}} + \sum_{\beta=1}^{m}\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta\end{aligned} \tag{19}$$

Using (19) and Coulomb's law, we build for each contact α an application $\Phi \in \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}$. This method to solve friction contacts is taken by [42, 39]. In this section, we detail these equations below to compute the gradient of our algorithm in Sec. 5.4.

$$\begin{cases}\Phi_1(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha) = \boldsymbol{\delta}_{t,\alpha} - \boldsymbol{\delta}_{t,\alpha}^{\text{free}} - \boldsymbol{W}_{tn}F_{n,\alpha} - \boldsymbol{W}_{tt}\boldsymbol{F}_{t,\alpha} \\ \Phi_2(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha) = \delta_{n,\alpha} - \delta_{n,\alpha}^{\text{free}} - W_{nn}F_{n,\alpha} - \boldsymbol{W}_{nt}\boldsymbol{F}_{t,\alpha} \\ \Phi_3(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha) = \boldsymbol{F}_{t,\alpha} - \dfrac{\mu F_{n,\alpha}\,(\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha})}{\|\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha}\|} \\ \Phi_4(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha) = q_n\delta_{n,\alpha}\end{cases} \tag{20}$$

The first three equations (two for Φ_1 and one for Φ_2) are given by (19) and the last three equations (two for Φ_3 and one for Φ_4) by the Coulomb's law. To solve (20), we need to find a couple $(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha)$ such as $\Phi(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha) = \mathbf{0}$.

Alart and Curnier [43] explains that a Newton method can be applied to this type of function. Detail about the convergence are given in [44]. Based on [42], we choose the coefficients q_n and q_t as:

$$q_n = \frac{1}{W_{nn}},\ q_t = \frac{\lambda_{\min}}{\lambda_{\max}^2} \tag{21}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are respectively the min and max eigenvalue of the matrix $\boldsymbol{W}_{tt}$.

To solve (20), we also use the Newton's method:

$$\begin{bmatrix}\boldsymbol{\delta}_\alpha^{(r)} \\ \boldsymbol{F}_\alpha^{(r)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\delta}_\alpha^{(r-1)} \\ \boldsymbol{F}_\alpha^{(r-1)}\end{bmatrix} - \left[\partial\Phi\left(\boldsymbol{\delta}_\alpha^{(r-1)}, \boldsymbol{F}_\alpha^{(r-1)}\right)\right]^{-1}\left[\Phi\left(\boldsymbol{\delta}_\alpha^{(r-1)}, \boldsymbol{F}_\alpha^{(r-1)}\right)\right]$$

where $\partial\Phi$ is the Jacobian matrix of the application Φ and it is illustrated in AppendixA.

If the first derivatives are sufficiently smooth and the initial point is not too far from one of the roots of the equations, the convergence of the Newton's Method is quadratic.

5.3. Gauss-Seidel method for contacts

In multi-contact case (m contacts), we need to take into account the coupling between the contact nodes. Using (19), we have $3m$ equations with $3m$ force unknowns and $3m$ movement unknowns.

Rewriting (19) as:

$$\boldsymbol{\delta}_\alpha - \boldsymbol{W}_{\alpha\alpha}\boldsymbol{F}_\alpha = \sum_{\beta=1,\beta\neq\alpha}^{m} \boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta + \boldsymbol{\delta}_\alpha^{\text{free}} \qquad (22)$$

As in [39], we can use an iterative Gauss-Seidel method to obtain a real-time solution of the contact problem under the Signorini's and Coulomb's laws. This method is well-adapted to sparse matrices and it converges, at least locally, when the system is over-constrained and there are multiple solutions. The convergence of this method is shown in [45].

The principle is to test the contacts one by one, and always in the same order, blocking all others to their most recently calculated value (algorithm 1). ϵ_2, ϵ_3 are respectively the desired accuracy for Signorini's law and Coulomb's law.

Algorithm 1: $\underline{\boldsymbol{F}} = \text{contact}(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}})$

Data: $\boldsymbol{W}$, μ
Parameters: ϵ_2, ϵ_3
for $1 \leq \alpha \leq m$ **do**
  $\boldsymbol{\delta}_\alpha \leftarrow \sum_{\beta=1,\beta\neq\alpha}^{m} \boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta + \boldsymbol{\delta}_\alpha^{\text{free}}$
  if $\delta_{n,\alpha} < \epsilon_2$ **then**
    $\boldsymbol{F}_\alpha \leftarrow -\boldsymbol{W}_{\alpha\alpha}^{-1}\boldsymbol{\delta}_\alpha$
    if $\|\boldsymbol{F}_{\alpha,t}\| > \mu\, F_{\alpha,n}$ **then**
      $F_{\alpha,n} \leftarrow -\delta_{\alpha,n}^{\text{free}}/W_{nn}$
      $\boldsymbol{\delta}_\alpha \leftarrow \boldsymbol{W}_{\alpha\alpha}\boldsymbol{F}_\alpha + \boldsymbol{\delta}_\alpha^{\text{free}}$
      do
        $c \leftarrow \left[\partial\Phi\left(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha\right)\right]^{-1}\left[\Phi\left(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha\right)\right]$
        $\begin{bmatrix}\boldsymbol{\delta}_\alpha \\ \boldsymbol{F}_\alpha\end{bmatrix} \leftarrow \begin{bmatrix}\boldsymbol{\delta}_\alpha \\ \boldsymbol{F}_\alpha\end{bmatrix} - c$
      while $\Phi\left(\boldsymbol{\delta}_\alpha, \boldsymbol{F}_\alpha\right) > \epsilon_3$
  else
    $\boldsymbol{F}_\alpha \leftarrow \boldsymbol{0}$

To find the appropriate contact node forces and then the node positions for a given foot 6D position (translation and rotation), we can use the frictional contact algorithm 3 using Gauss-Seidel method [39]. q is the time step and r is the total number of simulation steps; ϵ_1 is the desired accuracy for the contact problem. The initialization procedure of the frictional contact algorithm is described in algorithm 2.

Algorithm 2: $(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}}) = \text{init}(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$

Data: $\boldsymbol{O}_l^{\text{ini}}$, ${}^l\underline{\boldsymbol{P}}^{\text{free}}$, $\underline{\boldsymbol{P}}^{(q-1)}$, $\underline{\boldsymbol{F}}^{(q-1)}$
compute $\boldsymbol{R}(\boldsymbol{\Upsilon})$
if $q = 1$ **then**
  for $\alpha = 1 \ldots m$ **do**
    $\boldsymbol{F}_\alpha \leftarrow \boldsymbol{0}$
    $\boldsymbol{\delta}_\alpha^{\text{free}} \leftarrow \boldsymbol{O}_l + \begin{bmatrix} -\mathbf{t}\boldsymbol{O}_l^{\text{ini}} \\ \mathbf{n}\boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} \end{bmatrix}$
if $q = 2 \ldots r$ **then**
  for $\alpha = 1 \ldots m$ **do**
    $\boldsymbol{F}_\alpha \leftarrow \boldsymbol{F}_\alpha^{(q-1)}$
    $\boldsymbol{\delta}_\alpha^{\text{free}} \leftarrow \boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} + \boldsymbol{O}_l - \begin{bmatrix} \boldsymbol{P}_{t,\alpha}^{(q-1)} \\ 0 \end{bmatrix}$

Algorithm 3: $(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}) = \text{AlgoContacts}(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$

Parameters: ϵ_1
$(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}}) \leftarrow \text{init}(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$ //algorithm 2
do
  $\underline{\boldsymbol{F}}^{\text{old}} \leftarrow \underline{\boldsymbol{F}}$
  $\underline{\boldsymbol{F}} \leftarrow \text{contact}(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}})$ //algorithm 1
while $\dfrac{\left\|\underline{\boldsymbol{F}} - \underline{\boldsymbol{F}}^{old}\right\|}{\|\underline{\boldsymbol{F}}\|} > \epsilon_1$
compute $\underline{\boldsymbol{\delta}}$ using equation 19

5.4. Desired force and ZMP position algorithm

In the previous section, we detailed the frictional contact problem where the inputs are the foot position and orientation and the outputs are the contact forces and the contact space displacements. In this section, we propose an algorithm to compute foot position and orientation to obtain the desired $\boldsymbol{F}_{\text{des}}(t)$ and $\boldsymbol{Z}_{\text{des}}(t)$ and null vertical torque at ZMP $\boldsymbol{\Gamma}_{Z_{\text{des}},n} = 0$ mentioned in Fig. 5.

Let f be the function computing the resultant force $\boldsymbol{F}_{\text{tot}}$, the ZMP $\boldsymbol{Z}$ and vertical torque at desired ZMP $\boldsymbol{\Gamma}_{Z_{\text{des}},n}$ from $(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}})$. We want to solve the equation

$$f(\text{AlgoContacts}(\boldsymbol{O}_l, \boldsymbol{\Upsilon})) = \begin{bmatrix} \boldsymbol{F}_{\text{des}}(t) \\ \boldsymbol{Z}_{\text{des}}(t) \\ 0 \end{bmatrix} \qquad (23)$$

We solve it by Newton method.

Therefore, we derive a Newton step by finding a differential relationship between $(\boldsymbol{F}_{\text{tot}}, \boldsymbol{Z}, \boldsymbol{\Gamma}_{Z_{\text{des}},n})$ and $(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$:

$$\begin{bmatrix} \boldsymbol{F}_{\text{des}} - \boldsymbol{F}_{\text{tot}} \\ \boldsymbol{Z}_{\text{des}} - \boldsymbol{Z} \\ -\boldsymbol{\Gamma}_{Z_{\text{des}},n} \end{bmatrix} = \begin{bmatrix} d\boldsymbol{F}_{\text{tot}} \\ d\boldsymbol{Z} \\ d\boldsymbol{\Gamma}_{Z_{\text{des}},n} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\Upsilon} \end{bmatrix} \qquad (24)$$

where $d\boldsymbol{\Upsilon} = \begin{bmatrix} d\theta & d\phi & d\psi \end{bmatrix}^T$; $\boldsymbol{J}$ is the Jacobian matrix that links ZMP Force and position with foot orientation and position.

In AppendixB, we detail the terms of equation (24).

The method to obtain the node positions for a desired ZMP position and force is illustrated in algorithm 4. ϵ_4 is the desired accuracy for ZMP position and force. In algorithm 4, we relax the convergence of AlgoContact 3 through a user-defined parameter $\lambda \in [0\ 1]$:

$$\frac{\left\| \underline{\boldsymbol{F}} - \underline{\boldsymbol{F}}^{\text{old}} \right\|}{\|\underline{\boldsymbol{F}}\|} > \lambda \left\| \begin{array}{l} \boldsymbol{F}_{\text{des}} - \boldsymbol{F}_{\text{tot}} \\ \boldsymbol{Z}_{\text{des}} - \boldsymbol{Z} \\ -\Gamma_{Z_{\text{des}},n} \end{array} \right\| \tag{25}$$

Using this relaxed method, we reduce the simulation time by 30%.

The convergence of our algorithm is achieved when the following condition is met:

$$\frac{\left\| \underline{\boldsymbol{F}} - \underline{\boldsymbol{F}}^{\text{old}} \right\|}{\|\underline{\boldsymbol{F}}\|} < \epsilon_1 \tag{26}$$

and:

$$\left\| \begin{array}{l} \boldsymbol{F}_{\text{des}} - \boldsymbol{F}_{\text{tot}} \\ \boldsymbol{Z}_{\text{des}} - \boldsymbol{Z} \\ -\Gamma_{Z_{\text{des}},n} \end{array} \right\| < \epsilon_4 \tag{27}$$

Algorithm 4: $\underline{\boldsymbol{P}} = \text{ForcePos}(\boldsymbol{F}_{\text{des}}, \boldsymbol{Z}_{\text{des}})$

Data: $\boldsymbol{O}_l$, $\boldsymbol{\Upsilon}$, ${}^l\underline{\boldsymbol{P}}^{\text{free}}$
Parameters: ϵ_1, ϵ_4, λ
$(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}}) \leftarrow \text{init}(\boldsymbol{O}_l, \boldsymbol{\Upsilon})$ //algorithm 2
do
  do
    $\underline{\boldsymbol{F}}^{\text{old}} \leftarrow \underline{\boldsymbol{F}}$
    $\underline{\boldsymbol{F}} \leftarrow \text{contact}(\underline{\boldsymbol{F}}, \underline{\boldsymbol{\delta}}^{\text{free}})$ //algorithm 1
    compute $\boldsymbol{F}_{\text{tot}}$ using equation (B.1)
    compute $\boldsymbol{Z}$, $\Gamma_{Z_{\text{des}},n}$ using equation (B.2)
  while *equation* (25) *is true*
  compute $\boldsymbol{J}$ using equation (B.8)
  $\begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\Upsilon} \end{bmatrix} \leftarrow \boldsymbol{J}^{-1} \begin{bmatrix} \boldsymbol{F}_{\text{des}} - \boldsymbol{F}_{\text{tot}} \\ \boldsymbol{Z}_{\text{des}} - \boldsymbol{Z} \\ -\Gamma_{Z_{\text{des}},n} \end{bmatrix}$
  $\boldsymbol{O}_l \leftarrow d\boldsymbol{O}_l + \boldsymbol{O}_l$
  $\boldsymbol{\Upsilon} \leftarrow d\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}$
while *equation* (27) *is true*
compute $\underline{\boldsymbol{U}}$ using equation (9)
compute $\underline{\boldsymbol{P}}^{\text{free}}$ using equation (12)
$\underline{\boldsymbol{P}} \leftarrow \underline{\boldsymbol{U}} + \underline{\boldsymbol{P}}^{\text{free}}$

5.5. Initial conditions

To solve the problem, we have to set the condition at the first time step. We impose an initial position $\boldsymbol{O}_l^{\text{ini}}$ and yaw orientation of the foot just before the first contact that will be used for the reference contact positions $\boldsymbol{Q}$. Initial roll and pitch orientation of the foot just before first contact is not imposed since it could be discontinuous with the roll and pitch orientation found from desired ZMP position. Initial roll and pitch orientation of the foot are then taken to be the same as those obtained from the desired ZMP position. Hence, tangential reference contact positions are:

$$\boldsymbol{Q}_t = \mathbf{t}\boldsymbol{O}_l^{\text{ini}} + \mathbf{t}\boldsymbol{R}^{\text{ini}}(\theta, \phi, \psi^{\text{ini}})\ {}^l\boldsymbol{P}_\alpha^{\text{free}} \tag{28}$$

where ψ_{ini} is the desired fixed yaw orientation of the foot at first contact with the ground. For the normal component, $\boldsymbol{\delta}_{n,\alpha}^{\text{ini}} = \boldsymbol{\delta}_{n,\alpha}$. For the tangential term, equation (18) becomes:

$$\begin{array}{rcl} \boldsymbol{\delta}_{t,\alpha}^{\text{ini}} & = & \mathbf{t}\boldsymbol{O}_{l,\alpha} + \mathbf{t}\boldsymbol{R}\ {}^l\boldsymbol{P}_\alpha^{\text{free}} + \mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) - \mathbf{t}\boldsymbol{Q}_\alpha \\ & \approx & \mathbf{t}(\boldsymbol{O}_{l,\alpha} - \boldsymbol{O}_{l,\alpha}^{\text{ini}}) + \mathbf{t}(\boldsymbol{R} - \boldsymbol{R}^{\text{ini}}){}^l\boldsymbol{P}_\alpha^{\text{free}} \\ & & +\mathbf{t}\sum_{\beta=1}^{m}(\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta) \end{array} \tag{29}$$

From (17) and (29), we obtain:

$$\begin{array}{rcl} \boldsymbol{\delta}_\alpha & = & O_{l,\alpha} + R\ {}^lP_\alpha^{\text{free}} - \begin{bmatrix} \mathbf{t}(\boldsymbol{O}_{l,\alpha}^{\text{ini}} + \boldsymbol{R}^{\text{ini}}{}^l\boldsymbol{P}_\alpha^{\text{free}}) \\ 0 \end{bmatrix} \\ & & +\sum_{\beta=1}^{m}\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta \end{array} \tag{30}$$

The Jacobian matrix for the first step is detailed in AppendixC.

6. Cost function

As explained in Sec. 3, our cost function is based on the Cartesian stiffness computed in the ZMP position. Here, we explain how we can find this matrix. For an iterative algorithm we need to find a differential relationship between $(\boldsymbol{F}_{\text{tot}}, \boldsymbol{\Gamma}_Z)$ and $({}^z\boldsymbol{O}_l, \boldsymbol{\omega})$, where ω is angular velocity and ^{z} denotes the ZMP frame. We consider ${}^z\boldsymbol{O}_l$ rather than $\boldsymbol{O}_l$ so that reference point for small rotation is Z rather than the absolute origin. Defining a small rotation $d\omega = \omega\ dt$, we want the following matrix $\boldsymbol{K}_c$:

$$\begin{bmatrix} d\boldsymbol{F}_{\text{tot}} \\ d\boldsymbol{\Gamma}_Z \end{bmatrix} = \boldsymbol{K}_c \begin{bmatrix} {}^z d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_c^{\text{tr}} & \boldsymbol{K}_c^{\text{tr,rot}} \\ \boldsymbol{K}_c^{\text{rot,tr}} & \boldsymbol{K}_c^{\text{rot}} \end{bmatrix} \begin{bmatrix} {}^z d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} \tag{31}$$

The detailed description with computational details of the Cartesian stiffness is in AppendixD.

7. Results

We first present the conditions of the sole optimization problems, followed by a presentation of the obtained soles and discussions.

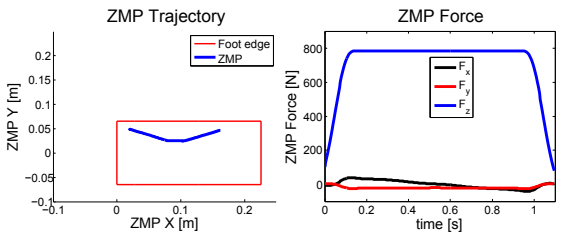


Figure 6: ZMP trajectory at left and ZMP force at right for the energy-minimizing case.

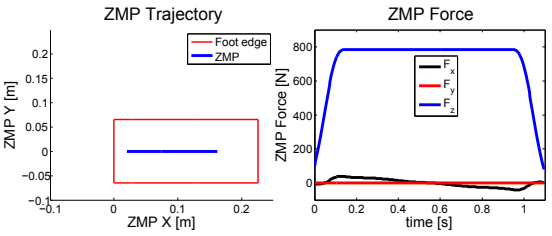


Figure 7: ZMP trajectory at left and ZMP force at right for the straight case.

7.1. Optimization description

We used smooth ZMP and force trajectories of Fig. 6 obtained with the walking pattern generator [46] based on a minimization of the robot energy consumption. Pattern generator was used on a single cyclic step. To obtain the whole ZMP trajectory under each foot, the ZMP repartition is computed during the double-support-phase of the walking gait. The ZMP trajectory is defined as a twice differentiable piecewise polynomial function whose parameters are obtained by solving a QP optimization problem. The force trajectory is a by-product of the computation. More details are given in [47] and [46].

Table 1 summarizes the different cases used to obtain optimized sole shapes. They were chosen to study the influence of different parameters on the sole shape obtained. Here is a detailed description of studied cases:

1. A reference case that can be applied to the robot.
2. A case giving more weight to the rotation criterion for better stability during the walk. Used to see the influence of relative weight given to impact force and foot rotation reduction.
3. A case with a different material, to see the influence of the choice of material on the sole shape and criteria.
4. A case with a different ZMP trajectory simply going from heel to toe in the middle of the foot, to see the influence of the ZMP trajectory.
5. A case with a non-modified surface attached to the rigid lower part of the robot foot S_D (Dirichlet nodes with no displacement) during optimization, as opposed to the other cases where the shape of this surface of Dirichlet nodes is modified and optimized (but kept flat). In this case this surface remains constant during optimization, in order to obtain a sole that can be used on real robot without changing the existing rigid part of the foot. This case allows to see the cost of this constraint on the foot shape and criteria.

To obtain dynamic properties close to the original robot set-up, we impose that the existing robot foot weight with ankle flexibility is similar to our proposed new foot with the flexible sole. This is enforced by the constraint on the total volume in problem (2)[1].

[1]This constraint is implemented as an inequality instead of an equality; yet this is equivalent at the solution, because the constraint is active (but an inequality enhances the convergence).

Table 1: Different cases for the different optimized sole shapes. The Young modulus of neoprene is 1 MPa and of butyl rubber is 1.4 MPa [48]. The cost function is the equation (1); WPG denotes the ZMP force and trajectory obtained using the walking pattern generator in [47].

Case	WPG	Material	Cost	Dirichlet
1-Ref	Fig. 6	Neoprene	$w = 1$	Free
2-Rot	Fig. 6	Neoprene	$w = 1000$	Free
3-Rubber	Fig. 6	Butyl rubber	$w = 1$	Free
4-Straight	Fig. 7	Neoprene	$w = 1$	Free
5-Fixed	Fig. 6	Neoprene	$w = 1$	Fixed

In the case of the neoprene for a maximum sole weight of 1 kg, the maximum volume is:

$$v_{\max} = \frac{m_{\max}}{\rho} = \frac{1\,\mathrm{kg}}{1230\,\mathrm{kg\,m^{-3}}} = 813\,\mathrm{cm}^3 \qquad (32)$$

where ρ is the material density [48]. In the case of butyl rubber $v_{\max} = 1098\,\mathrm{cm}^3$.

Starting from the initial shape in Fig. 1b, we obtain convergence to an optimized sole shape. We then refine the optimized shape using `iso2mesh`'s library[2] and we run a second optimization with the refined mesh as initial shape, etc. During the shape optimization process, the mesh could be invalid because of some reversed tetrahedrons (with a negative volume) [49]. In this case, we stop automatically the optimization, remesh and restart. To obtain one optimized sole, our optimization takes between 1 day and 1.5 days of computation with a partially optimized code on a personal laptop. The library `iso2mesh` is very useful in our case because it is able to create quality volumetric mesh from isosurface patches.

Fig. 8, 9, 10, 11, 12 shows the optimized soles for the different cases in Table 1 and a dry friction of 0.8.

7.2. Discussions

Notice for all optimized soles, that to minimize the impact during the heel-strike phase, the optimizer creates a

[2]`http://iso2mesh.sourceforge.net/`

flexible appendix with minimum connection with the rigid part of the foot, in a shape similar to the human heel. This "heel" part is different for the different cases in Table 1. In particular, notice that for the trajectory in Fig. 6, this part is in the back-left part of the sole, for the trajectory in Fig. 7 it is in the back-center part of the sole. One can conclude that the optimizer creates a heel where the ZMP trajectory is along the sole.

The difference of the heel part for different cost functions is also remarkable. In the case 1-Ref, this part is more rounded, shorter and with less connection with the rigid part of the foot in comparison with the optimized sole in the case 2-Rot. This interesting result shows that if we want to enforce lower foot rotation to keep the vertical posture, we need a less rounded, longer heel with rigid support. Those characteristics to minimize foot rotation are also observed at the front of the optimized sole, where only rotational part of the Cartesian stiffness is maximized. We then obtained a large front part of the optimized sole similar to front part of human feet.

To fulfill the volume constraint of the problem (2), the optimizer whittle many sole parts down. Especially, less material is present in the middle x-axis, which increases rotational stiffness along x-axis.

More precisely, from Fig. 13 and Table 2, comparing the different cases, one can observe the followings:

1. For a given sole, Cartesian stiffness in translation and rotation are rather linked, since their evolution with respect to time is similar in Fig. 13. Hence minimizing translational stiffness and maximizing rotational stiffness are antagonistic objectives. Different compromises can be found between both criteria. Since translational stiffness is minimized only at the beginning, we can observe that stiffnesses are lower at the beginning and larger later.
2. For the case 1-Ref compared to the initial sole, c^{tr} has been divided by almost 5, while c^{rot} is about the same. So shape optimization improves one of the two sole criteria by a large factor.
3. For the case 2-Rot compared to the case 1-Ref, the rotational stiffness is improved while the impact absorption is worsen, which is another compromise with more importance given to maximizing the rotational stiffness, as expected with increased weight on this criterion.
4. For the case 3-Rubber compared to case the 1-Ref, what is lost in translational stiffness minimization is gained in rotational stiffness maximization, which is another compromise. Criteria for this butyl-rubber sole are another compromise close to neoprene sole, so this change of material has no significant impact on the sole performance.
5. For the case 4-Straight compared to the case 1-Ref. Both c^{tr} and c^{rot} are worse. Having a straight ZMP trajectory seems worse than a trajectory starting and finishing in a corner of the foot. This is caused by a poor rotational stiffness around the y-axis due to an entirely soft heel. It shows that ZMP trajectory has a strong influence on sole shape optimization. In future works, we plan to optimize the sole shape and the ZMP trajectories simultaneously.
6. For the case 5-Fixed compared to the case 1-Ref, c^{tr} and c^{rot} are about 40% worse. It shows that it is worth optimizing the rigid part of the foot and redesign robot feet accordingly.

Additionally, as explained in Sec. 4.1, we bound the shape optimization parameters $p_{j,k}$. This bound in combination with the initial parallelepiped shape can create some sharp shapes in the optimized soles. We notice that for example, in the front part in Fig. 8, 9, 10, 11, 12.

These results can be visualized in the accompanying video available at `https://youtu.be/wivyn5klAMw`. This video presents the deformation simulations of the soles. One can notice that the deformation of the heel appendices at the beginning of foot contact are undergoing deformations of the order of cm, with rotations of about 45°. The small deformation hypothesis is compromised, and results in this case are not precise. For more precise results, we plan to use a large-deformation FEM code in the future.

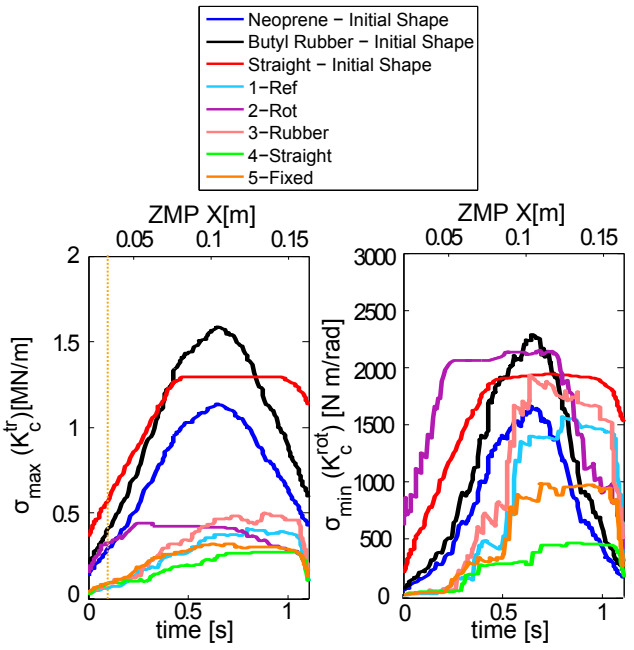


Figure 13: Cost functions for the five cases in Table 1.

8. Conclusion

In this paper, we devised a framework to optimize the sole shape of a humanoid robot for a given cost function. To obtain faster and easier to implement algorithms, some model approximations have been done. In particular, we used a simplest FEM for static linear elastic model under the assumption of small deformations to model the soles.

1-Ref

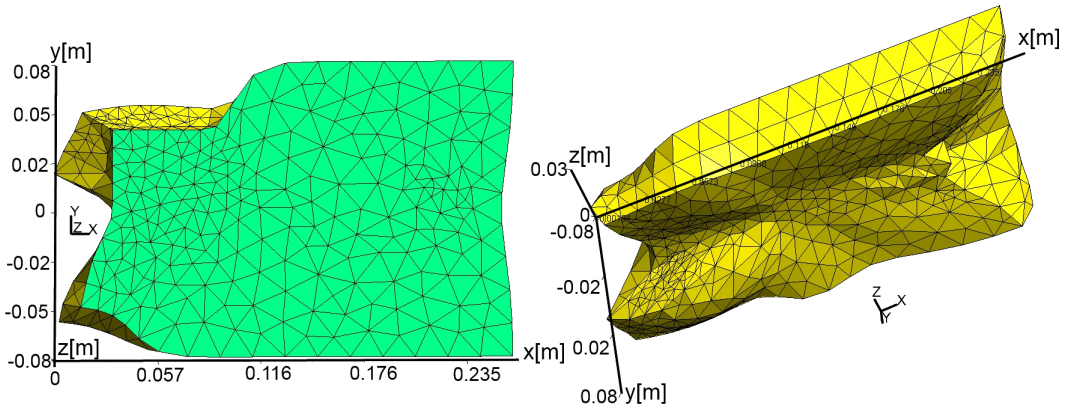


Figure 8: Optimized sole for the case 1-Ref: ZMP trajectory illustrated in Fig. 6, material neoprene and $w = 1$ in the cost function (1)

2-Rot

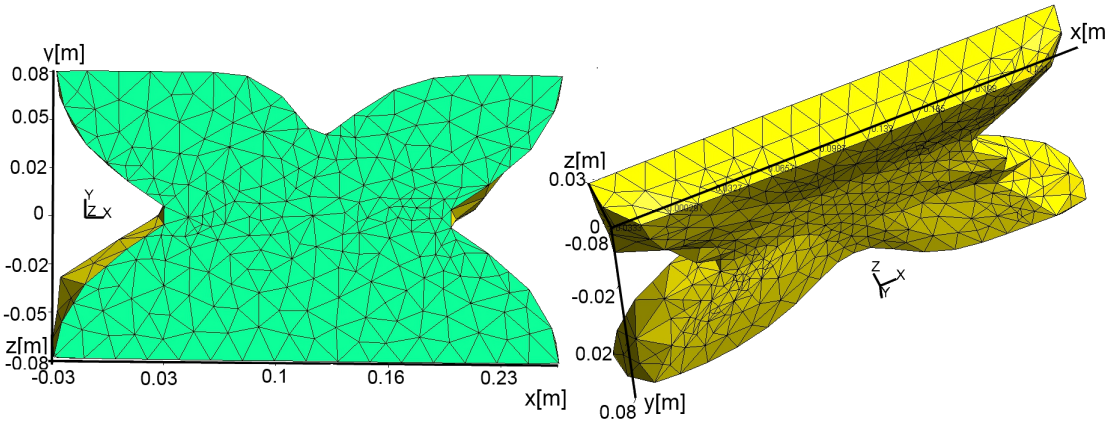


Figure 9: Optimized sole for the case 2-Rot: ZMP trajectory illustrated in Fig. 6, material neoprene and $w = 1000$ in the cost function (1)

3-Rubber

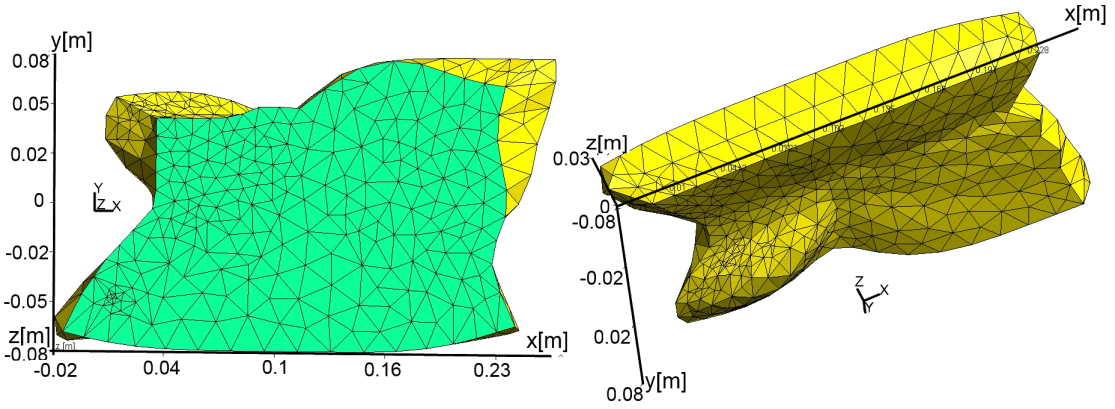


Figure 10: Optimized sole for the case 3-Rubber: ZMP trajectory illustrated in Fig. 6, material butyl rubber and $w = 1$ in the cost function (1)

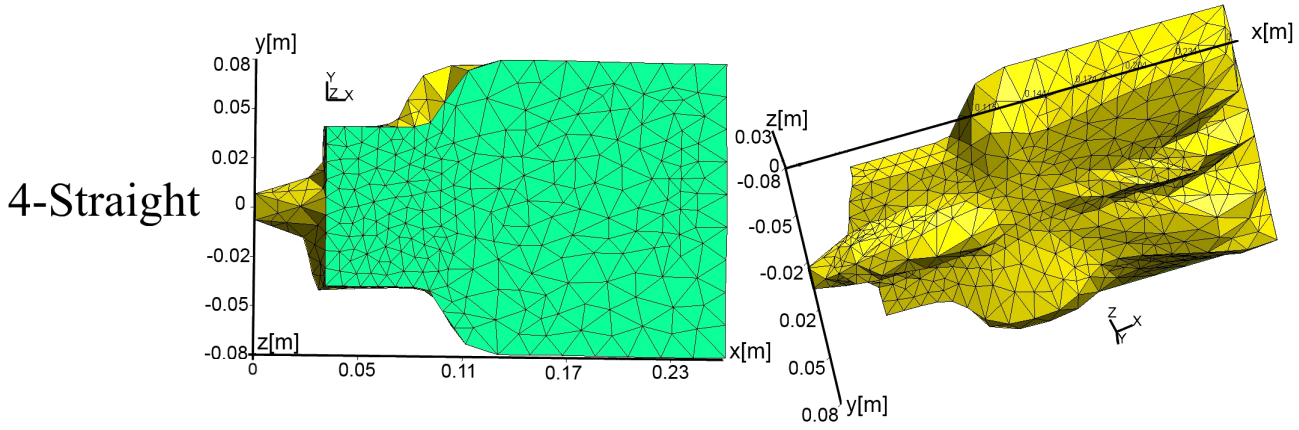


Figure 11: Optimized sole for the case 4-Straight: ZMP trajectory illustrated in Fig. 7, material neoprene and $w = 1$ in the cost function (1)

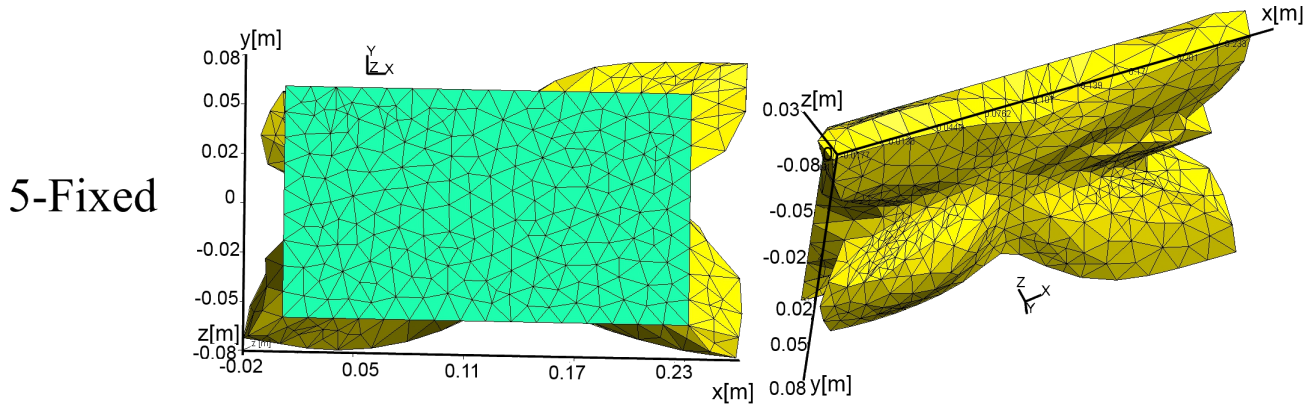


Figure 12: Optimized sole for the case 5-Fixed: ZMP trajectory illustrated in Fig. 6, material neoprene and $w = 1$ in the cost function (1). Case with a non-modified surface attached to the rigid lower part of the robot foot S_D (Dirichlet nodes with no displacement) during optimization

Table 2: Comparison of the cost function in equation 1 for the initial and optimized shape for the five cases in Table 1. The subscript ini denotes the plain rectangular brick shaped sole in Fig. 1b and opt denotes the optimized sole shape. $\mathbf{c}^{\mathbf{tr}}_{\mathbf{ini}}$ and $\mathbf{c}^{\mathbf{tr}}_{\mathbf{opt}}$ are in [N/m], while $\mathbf{c}^{\mathbf{rot}}_{\mathbf{ini}}$ and $\mathbf{c}^{\mathbf{rot}}_{\mathbf{opt}}$ in [N.m/rad]. The column with "comp." are comparisons with optimized case 1-Ref, in the most interpretable format (multiplied or divided by a coefficient, increased or decreased by an amount in percent compared to criterion of 1-Ref). $\mathbf{c}^{\mathbf{tr}}_{\mathbf{opt}}$ is minimized, so increase is worse. $\mathbf{c}^{\mathbf{rot}}_{\mathbf{opt}}$ is maximized, so increase is better and decrease is worse.

Case	$\mathbf{c}^{\mathbf{tr}}_{\mathbf{ini}}$	$\mathbf{c}^{\mathbf{tr}}_{\mathbf{opt}}$	$\mathbf{c}^{\mathbf{tr}}_{\mathbf{opt}}$ comp.	$\mathbf{c}^{\mathbf{rot}}_{\mathbf{ini}}$	$\mathbf{c}^{\mathbf{rot}}_{\mathbf{opt}}$	$\mathbf{c}^{\mathbf{rot}}_{\mathbf{opt}}$ comp.	$\mathbf{c}_{\mathbf{ini}}$	$\mathbf{c}_{\mathbf{opt}}$
1-Ref	$3.88 \cdot 10^6$	$8.06 \cdot 10^5$	/	$1.79 \cdot 10^5$	$1.74 \cdot 10^5$	/	$3.70 \cdot 10^6$	$6.33 \cdot 10^5$
2-Rot	$3.88 \cdot 10^6$	$4.65 \cdot 10^6$	$\times 5.77$	$1.79 \cdot 10^5$	$3.50 \cdot 10^5$	$\times 2.01$	$-1.75 \cdot 10^8$	$-3.45 \cdot 10^8$
3-Rubber	$5.39 \cdot 10^6$	$9.59 \cdot 10^5$	$+19.0\%$	$2.48 \cdot 10^5$	$2.14 \cdot 10^5$	$+23.0\%$	$5.15 \cdot 10^6$	$7.45 \cdot 10^5$
4-Straight	$9.14 \cdot 10^6$	$1.02 \cdot 10^6$	$+26.6\%$	$3.41 \cdot 10^5$	$5.84 \cdot 10^4$	$/2.98$	$8.80 \cdot 10^6$	$9.62 \cdot 10^5$
5-Fixed	$3.88 \cdot 10^6$	$1.21 \cdot 10^6$	$+50.1\%$	$1.79 \cdot 10^5$	$1.16 \cdot 10^5$	-33.3%	$3.70 \cdot 10^6$	$1.09 \cdot 10^6$

The obtained soles appear having some similar characteristics with the human foot. We also showed that the shape optimization improves significantly the criteria considered for the sole, and that the ZMP trajectory and the shape of the contact with rigid part of the foot have a strong influence on the performance of the sole, but not the different material tested.

The next step of this research is to extend the latter approach to optimize the walking pattern parameters and sole shape in a single optimization problem formulation. This proved to be pertinent and was investigated in simpler 2D predefined forms in [50].

In the papers [51][46], we investigated adding a plain rectangular compliant sole to a humanoid robot. In the near future, we will build up and test the optimized soles on a real robot to validate the procedure and criteria considered. We will also use more accurate FEM, for example considering large-displacement and nonlinear material behaviors.

In the current state of our framework, the gradient of the cost function is obtained by finite differences, which impose numerous simulations of the sole movement for each iteration of the shape optimization process. We are currently implementing the analytical gradient. We expect this will speed-up the presented computation by a factor 20–100.

After those improvements, we also plan to develop a controller for stabilization with a sole prior to experiments on the HRP-4 humanoid robot.

9. Acknowledgments

The research leading to these results has received funding from the EU 7th Framework Program (FP7/2007-2013) under grant agreement no 611909 (KoroiBot).

AppendixA. Jacobian of the application Φ

$$\partial\Phi = \begin{bmatrix} \boldsymbol{I}_{2\times2} & \boldsymbol{0}_{2\times1} & -\boldsymbol{W}_{tt} & -\boldsymbol{W}_{tn} \\ \boldsymbol{0}_{1\times2} & 1 & -\boldsymbol{W}_{nt} & -W_{nn} \\ \dfrac{\partial\Phi_3}{\partial\boldsymbol{\delta}_{t,\alpha}} & \boldsymbol{0}_{2\times1} & \dfrac{\partial\Phi_3}{\partial\boldsymbol{F}_{t,\alpha}} & \dfrac{\partial\Phi_3}{\partial F_{n,\alpha}} \\ \boldsymbol{0}_{1\times2} & q_n & \boldsymbol{0}_{1\times2} & 0 \end{bmatrix} \tag{A.1}$$

in which:

$$\frac{\partial\Phi_3}{\partial\boldsymbol{\delta}_{t,\alpha}} = q_t\mu F_{n,\alpha}\Pi\left(\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha}\right),$$
$$\frac{\partial\Phi_3}{\partial\boldsymbol{F}_{t,\alpha}} = \boldsymbol{I}_{2\times2} - \mu F_{n,\alpha}\Pi\left(\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha}\right),$$
$$\frac{\partial\Phi_3}{\partial F_{n,\alpha}} = -\frac{\mu\left(\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha}\right)}{\|\boldsymbol{F}_{t,\alpha} - q_t\boldsymbol{\delta}_{t,\alpha}\|}$$
$$\Pi(\boldsymbol{V}) = \frac{\partial}{\partial V}\left(\frac{V}{\|V\|}\right) = \frac{1}{\|V\|^3}\begin{bmatrix} (v_2)^2 & -v_1v_2 \\ -v_1v_2 & (v_1)^2 \end{bmatrix}$$

AppendixB. Relationship between ZMP force-position and foot position-orientation

From (24), we can straightforwardly obtain $\boldsymbol{F}_{\text{tot}}$ in function of the force applied at the contact node α:

$$\boldsymbol{F}_{\text{tot}} = \sum_{\alpha=1}^{m} \boldsymbol{F}_\alpha \tag{B.1}$$

Then for $\boldsymbol{Z}$ and $\boldsymbol{\Gamma}_{Z_{\text{des}},n}$ we start with equation of resultant torque at a point A given by:

$$\boldsymbol{\Gamma}_A = \sum_{\beta=1}^{m} (\boldsymbol{P}_\beta - \boldsymbol{A}) \times \boldsymbol{F}_\beta \tag{B.2}$$

Using the definition of the ZMP and the vertical resultant torque at Z_{des} we get:

$$\begin{bmatrix} \Gamma_{Z,t_1} = 0 \\ \Gamma_{Z,t_2} = 0 \\ \Gamma_{Z_{\text{des}},n} \end{bmatrix} = \sum_{\beta=1}^{m} \begin{bmatrix} (\boldsymbol{P}_{t_2,\beta} - \boldsymbol{Z}_{t_2}) \cdot F_{n,\beta} \\ -(\boldsymbol{P}_{t_1,\beta} - \boldsymbol{Z}_{t_1}) \cdot F_{n,\beta} \\ (\boldsymbol{P}_{t_1,\beta} - \boldsymbol{Z}_{des,t_1}) \cdot F_{t_2,\beta} - (\boldsymbol{P}_{t_2,\beta} - \boldsymbol{Z}_{des,t_2}) \cdot F_{t_1,\beta} \end{bmatrix} \tag{B.3}$$

The subscripts t_1 and t_2 denote respectively the first and second component of the vector.

We now have relations (B.1) and (B.3) between $(\boldsymbol{F}_{\text{tot}}, \boldsymbol{Z}, \boldsymbol{\Gamma}_{Z_{\text{des}},n})$ and nodes position and force, and relation (19), Signorini's and Coulomb's laws between nodes position and force as well as sole position and orientation. To find the differential relationship (24), we have to take the differential of previously cited equations.

Considering $\boldsymbol{F}_\alpha$, $\boldsymbol{\delta}_\alpha$, $\boldsymbol{O}_l$ and (θ,ϕ,ψ) as variables, the differential of (19) is:

$$d\boldsymbol{\delta}_\alpha = d\boldsymbol{O}_l + \sum_{\beta=1}^{m} \boldsymbol{W}_{\alpha\beta} d\boldsymbol{F}_\beta + \begin{bmatrix} \nu^\theta & \nu^\phi & \nu^\psi \end{bmatrix} d\boldsymbol{\Upsilon} \tag{B.4}$$

where ν^x for each angle $x \in \{\theta, \phi, \psi\}$:

$$\nu^x = \frac{\partial \boldsymbol{R}}{\partial x}{}^l\boldsymbol{P}_\alpha^{\text{free}} + \sum_{\beta=1}^{m} \left(\frac{\partial \boldsymbol{R}}{\partial x}\boldsymbol{C}_{\alpha\beta}\boldsymbol{R}^T + \boldsymbol{R}\boldsymbol{C}_{\alpha\beta}\frac{\partial \boldsymbol{R}}{\partial x}^T \right) \boldsymbol{F}_\beta$$

Following the Coulomb's law, we can distinguish two cases for each contact node α:

1. **Stick**: We have three equations given by (B.4) and three equations given by $\boldsymbol{\delta}_\alpha = \boldsymbol{0} \rightarrow d\boldsymbol{\delta}_\alpha = \boldsymbol{0}$.
2. **Slip**: We have three equations given by (B.4) and three equations given by the differential of the Coulomb's law (A.1).

Using those considerations, we can write the following system that relates contact forces and contact space displacements to the foot position and orientation:

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} d\underline{\boldsymbol{F}} \\ d\underline{\boldsymbol{\delta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{0}_{3s\times3} & \boldsymbol{0}_{3s\times3} \end{bmatrix} \begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\Upsilon} \end{bmatrix} \tag{B.5}$$

where:
$d\underline{\boldsymbol{\delta}}$ includes only the slipping contacts, m is the number

of contacts, s is the number of slipping contacts, $p(j)$ $j \in [1, ..., s]$ is the j^{th} slip contact and:

$$\begin{cases} \boldsymbol{A}_{11} = -\boldsymbol{W}, \boldsymbol{A}_{12} = \left[\boldsymbol{a}_1 \cdots \boldsymbol{a}_j \cdots \boldsymbol{a}_s\right], \\ \boldsymbol{a}_j = \left[\boldsymbol{0}^1_{3\times3} \cdots \boldsymbol{I}^{p(j)}_{3\times3} \cdots \boldsymbol{0}^s_{3\times3}\right]^T \\ \boldsymbol{A}_{21} = \left[\boldsymbol{c}_1^T \cdots \boldsymbol{c}_j^T \cdots \boldsymbol{c}_s^T\right]^T, \\ \boldsymbol{c}_j = \begin{bmatrix} \boldsymbol{0}^1_{3\times3} \cdots \begin{bmatrix} \dfrac{\partial \Phi_3}{\partial \boldsymbol{F}_{t,j}} & \dfrac{\partial \Phi_3}{\partial \boldsymbol{F}_{n,j}} \\ \boldsymbol{0}_{1\times2} & 0 \end{bmatrix}^{p(j)} \cdots \boldsymbol{0}^m_{3\times3} \end{bmatrix} \\ \boldsymbol{A}_{22} = \operatorname{diag}(d_j), d_j = \begin{bmatrix} \dfrac{\partial \Phi_3}{\partial \boldsymbol{\delta}_{t,j}} & \boldsymbol{0}_{2\times1} \\ \boldsymbol{0}_{1\times2} & q_n \end{bmatrix} \\ \boldsymbol{B}_{11} = \left[\boldsymbol{I} \cdots \boldsymbol{I}\right]^T \\ \boldsymbol{B}_{12} = \left[\boldsymbol{b}_1 \cdots \boldsymbol{b}_i \cdots \boldsymbol{b}_m\right]^T, \forall i \in [1\, m] \\ \boldsymbol{b}_i = \left[\nu_i^\theta \quad \nu_i^\phi \quad \nu_i^\psi\right]^T \end{cases}$$

The differential of (B.3) gives:

$$\begin{array}{l} \sum_{\beta=1}^{m} \frac{\partial \boldsymbol{P}_{t_2,\beta}}{\partial \delta_{t_2,\beta}} d\delta_{t_2,\beta} \boldsymbol{F}_{n,\beta} + \sum_{\beta=1}^{m} (\boldsymbol{P}_{t_2,\beta} - \boldsymbol{Z}_{t_2}) dF_{n,\beta} - d\boldsymbol{Z}_{t_2} \sum_{\beta=1}^{m} F_{n,\beta} = 0 \\ -\sum_{\beta=1}^{m} \frac{\partial \boldsymbol{P}_{t_1,\beta}}{\partial \delta_{t_1,\beta}} d\delta_{t_1,\beta} \boldsymbol{F}_{n,\beta} - \sum_{\beta=1}^{m} (\boldsymbol{P}_{t_1,\beta} - \boldsymbol{Z}_{t_1}) dF_{n,\beta} + d\boldsymbol{Z}_{t_1} \sum_{\beta=1}^{m} \boldsymbol{F}_{n,\beta} = 0 \\ \sum_{\beta=1}^{m} \frac{\partial \boldsymbol{P}_{t_1,\beta}}{\partial \delta_{t_1,\beta}} d\delta_{t_1,\beta} \boldsymbol{F}_{t_2,\beta} + \sum_{\beta=1}^{m} (\boldsymbol{P}_{t_1,\beta} - \boldsymbol{Z}_{des,t_1}) dF_{t_2,\beta} \\ -\sum_{\beta=1}^{m} \frac{\partial \boldsymbol{P}_{t_2,\beta}}{\partial \delta_{t_2,\beta}} d\delta_{t_2,\beta} \boldsymbol{F}_{t_1,\beta} - \sum_{\beta=1}^{m} (\boldsymbol{P}_{t_2,\beta} - \boldsymbol{Z}_{des,t_2}) dF_{t_1,\beta} = d\boldsymbol{\Gamma}_{Z_{\text{des}},n} \end{array} \tag{B.6}$$

With $\dfrac{\partial \boldsymbol{P}_{t,\beta}}{\partial \delta_{t,\beta}} = \boldsymbol{I}$ (obtained by differentiating (15)) it can be presented in the following matrix form:

$$\begin{bmatrix} \boldsymbol{F}_{\text{des}} - \boldsymbol{F}_{\text{tot}} \\ \boldsymbol{Z}_{\text{des}} - \boldsymbol{Z} \\ -\boldsymbol{\Gamma}_{Z_{\text{des}},n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{0}_{3\times3s} \\ \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{bmatrix} \begin{bmatrix} d\underline{\boldsymbol{F}} \\ d\underline{\boldsymbol{\delta}} \end{bmatrix} \tag{B.7}$$

where for $p(j)$ $j \in [1\, s]$ is the j^{th} slip contact:

$$\begin{cases} \boldsymbol{D}_{11} = \left[\boldsymbol{I}^1_{3\times3} \cdots \boldsymbol{I}^i_{3\times3} \cdots \boldsymbol{I}^m_{3\times3}\right], \forall i \in [1\, m] \\ \boldsymbol{D}_{21} = \left[\boldsymbol{d}_1 \cdots \boldsymbol{d}_i \cdots \boldsymbol{d}_m\right], \forall i \in [1\, m] \\ \boldsymbol{d}_i = \begin{bmatrix} 0 & 0 & \dfrac{1}{\boldsymbol{F}_{\text{tot},n}}(\boldsymbol{P}_{it_1} - \boldsymbol{Z}_{t_1}) \\ 0 & 0 & \dfrac{1}{\boldsymbol{F}_{\text{tot},n}}(\boldsymbol{P}_{it_2} - \boldsymbol{Z}_{t_2}) \\ (-\boldsymbol{P}_{it_2} + \boldsymbol{Z}_{t_2}) & (\boldsymbol{P}_{it_1} - \boldsymbol{Z}_{t_1}) & 0 \end{bmatrix} \\ \boldsymbol{D}_{22} = \left[\boldsymbol{e}_1 \cdots \boldsymbol{e}_j \cdots \boldsymbol{e}_s\right], j \in [1\, s] \\ \boldsymbol{e}_j = \begin{bmatrix} \dfrac{1}{\boldsymbol{F}_{\text{tot},n}} F_{p(j)n} & 0 & 0 \\ 0 & \dfrac{1}{\boldsymbol{F}_{\text{tot},n}} F_{p(j)n} & 0 \\ F_{p(j)t_2} & -F_{p(j)t_1} & 0 \end{bmatrix} \end{cases}$$

Combining (B.5) with (B.7), we finally get $\boldsymbol{J}$ in (24):

$$\boldsymbol{J} = \boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B} \tag{B.8}$$

Considering $\boldsymbol{\Gamma}_{Z_{\text{des}},n}$ rather than $\boldsymbol{\Gamma}_{Z,n}$ gives a simpler differential (B.6) since $\boldsymbol{Z}_{\text{des}}$ is constant. Since $\boldsymbol{Z}$ converges to $\boldsymbol{Z}_{\text{des}}$, the same solution is obtained in the end.

AppendixC. Jacobian for the first step

Considering $\boldsymbol{F}_\alpha$, $\boldsymbol{\delta}_\alpha$, $\boldsymbol{O}_l$ and (θ,ϕ,ψ) as variables, the differential of (30) is:

$$d\boldsymbol{\delta}_\alpha = d\boldsymbol{O}_{l,\alpha} + \sum_{\beta=1}^{m} \boldsymbol{W}_{\alpha\beta} d\boldsymbol{F}_\beta + \begin{bmatrix} \nu^\theta_{\text{ini}} & \nu^\phi_{\text{ini}} & \nu^\psi \end{bmatrix} d\boldsymbol{\Upsilon} \tag{C.1}$$

where:

$$\nu^\theta_{\text{ini}} = \nu^\theta - \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \theta} {}^l\boldsymbol{P}^{\text{free}}_\alpha \\ 0 \end{bmatrix}, \; \nu^\phi_{\text{ini}} = \nu^\phi - \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \phi} {}^l\boldsymbol{P}^{\text{free}}_\alpha \\ 0 \end{bmatrix}$$

This gives a modification of (B.5) for the initial step only for b_i as follows:

$$\boldsymbol{b}_i = \begin{bmatrix} \nu_i^\theta - \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \theta} {}^l\boldsymbol{P}^{\text{free}}_i \\ 0 \end{bmatrix} & \nu_i^\phi - \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \phi} {}^l\boldsymbol{P}^{\text{free}}_i \\ 0 \end{bmatrix} & \nu_i^\psi \end{bmatrix}^T \tag{C.2}$$

For the first time step, the differential of (15) gives:

$$d\boldsymbol{P}_\beta = d\delta_\beta + \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \theta} {}^l\boldsymbol{P}^{\text{free}}_\alpha & \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \phi} {}^l\boldsymbol{P}^{\text{free}}_\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} d\boldsymbol{\Upsilon} \tag{C.3}$$

Differential (B.6) now have additional terms that finally gives initial step $\boldsymbol{J}^{\text{ini}} = \boldsymbol{J} + \tilde{J}$ in (24), where:

$$\begin{array}{l} \tilde{J} = \begin{bmatrix} \boldsymbol{0}_{3,3} & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \begin{bmatrix} \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \theta} h_n & \mathbf{t} \frac{\partial \boldsymbol{R}^{\text{ini}}}{\partial \phi} h_n & 0 \\ \frac{\partial \boldsymbol{R}^{\text{ini}}_{t_1}}{\partial \theta} h_{t_2} - \frac{\partial \boldsymbol{R}^{\text{ini}}_{t_2}}{\partial \theta} h_{t_1} & \frac{\partial \boldsymbol{R}^{\text{ini}}_{t_1}}{\partial \phi} h_{t_2} - \frac{\partial \boldsymbol{R}^{\text{ini}}_{t_2}}{\partial \phi} h_{t_1} & 0 \end{bmatrix} \end{bmatrix} \\ h_{t_1} = \sum_{\beta=1}^{m} {}^l\boldsymbol{P}^{\text{free}}_\beta F_{t_1,\beta}, \; h_{t_2} = \sum_{\beta=1}^{m} {}^l\boldsymbol{P}^{\text{free}}_\beta F_{t_2,\beta}, \\ h_n = \frac{1}{F_{\text{tot},n}} \sum_{\beta=1}^{m} {}^l\boldsymbol{P}^{\text{free}}_\beta F_{n,\beta} \end{array}$$

AppendixD. Details of the cost function

To obtain the equation (31), we start by differentiating (19) with respect to t instead of (θ, ϕ, ψ). To do so, we use the following derivation of a vector applied on a rotation matrix

$$dR = \widehat{d\omega}\boldsymbol{R} \tag{D.1}$$

where $\widehat{d\omega}$ denotes the skew matrix of $d\omega$.

(D.1) applied to the differential of (19) gives after factorization in $d\omega$:

$$\begin{array}{rcl} d\boldsymbol{\delta}_\alpha & = & d\boldsymbol{O}_l + \sum_{\beta=1}^{m} \boldsymbol{W}_{\alpha\beta} d\boldsymbol{F}_\beta - (\widehat{\boldsymbol{R}\boldsymbol{P}^{\text{free}}_\alpha}) d\omega \\ & & + \sum_{\beta=1}^{m} \left(\boldsymbol{W}_{\alpha\beta} \hat{\boldsymbol{F}}_\beta - (\widehat{\boldsymbol{W}_{\alpha\beta}\boldsymbol{F}_\beta}) \right) d\omega \end{array} \tag{D.2}$$

Therefore in matrix form we have:

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} d\underline{\boldsymbol{F}} \\ d\underline{\boldsymbol{\delta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_{\omega 11} & \boldsymbol{B}_{\omega 12} \\ \boldsymbol{0}_{3s\times3} & \boldsymbol{0}_{3s\times3} \end{bmatrix} \begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} \tag{D.3}$$

where A_{ij} are same as in (B.5) and:

$$\begin{cases} \boldsymbol{B}_{\omega 11} = \left[\boldsymbol{I}_1 \cdots \boldsymbol{I}_i \cdots \boldsymbol{I}_m\right]^T \ \forall i \in [1\ m] \\ \boldsymbol{B}_{\omega 12} = \left[\boldsymbol{b}_1 \cdots \boldsymbol{b}_i \cdots \boldsymbol{b}_m\right]^T \ \forall i \in [1\ m] \\ \boldsymbol{b}_i^T = -(\widehat{\boldsymbol{R}\boldsymbol{P}_i^{\text{free}}}) + \sum_{\beta=1}^{m} \left(\boldsymbol{W}_{i\beta}\hat{\boldsymbol{F}}_\beta - (\widehat{\boldsymbol{W}_{i\beta}\boldsymbol{F}_\beta})\right) \end{cases}$$

Using differential of (B.1) and (B.2) with (D.3), we have the following relationship:

$$\begin{bmatrix} d\boldsymbol{F}_{\text{tot}} \\ d\boldsymbol{\Gamma} \end{bmatrix} = \boldsymbol{G}\begin{bmatrix} \underline{d\boldsymbol{F}} \\ \underline{d\boldsymbol{\delta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11} & \boldsymbol{0}_{3\times 3s} \\ \boldsymbol{G}_{21} & \boldsymbol{G}_{22} \end{bmatrix} \boldsymbol{A}^{-1}\boldsymbol{B}_\omega \begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} \quad \text{(D.4)}$$

where for $p(j)\ j \in [1\ s]$ is the j^{th} slip contact:

$$\begin{cases} \boldsymbol{G}_{11} = \left[\boldsymbol{I}_{3\times 3}^1 \cdots \boldsymbol{I}_{3\times 3}^i \cdots \boldsymbol{I}_{3\times 3}^m\right], \forall i \in [1\ m] \\ \boldsymbol{G}_{21} = \left[\boldsymbol{g}_1 \cdots \boldsymbol{g}_i \cdots \boldsymbol{g}_m\right], \forall i \in [1\ m] \\ \boldsymbol{g}_i = \begin{bmatrix} 0 & 0 & (\boldsymbol{P}_{it_2} - \boldsymbol{Z}_{it_2}) \\ 0 & 0 & -(\boldsymbol{P}_{it_1} - \boldsymbol{Z}_{it_1}) \\ -(\boldsymbol{P}_{it_2} - \boldsymbol{Z}_{it_2}) & (\boldsymbol{P}_{it_1} - \boldsymbol{Z}_{it_1}) & 0 \end{bmatrix} \\ \boldsymbol{G}_{22} = \left[\boldsymbol{\zeta}_1 \cdots \boldsymbol{\zeta}_j \cdots \boldsymbol{\zeta}_s\right], \\ \boldsymbol{\zeta}_j = \begin{bmatrix} 0 & F_{p(j),n} & 0 \\ -F_{p(j),n} & 0 & 0 \\ F_{p(j),t_2} & -F_{p(j),t_1} & 0 \end{bmatrix} \end{cases}$$

Using (D.4) and the transport law for the screw, we can find the Cartesian stiffness as:

$$\begin{aligned} \begin{bmatrix} d\boldsymbol{F}_{\text{tot}} \\ d\boldsymbol{\Gamma}_Z \end{bmatrix} &= \boldsymbol{G}\boldsymbol{A}^{-1}\boldsymbol{B}_\omega \begin{bmatrix} d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} = \boldsymbol{G}\boldsymbol{A}^{-1}\boldsymbol{B}_\omega \boldsymbol{\Xi} \begin{bmatrix} {}^z d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} \\ &= \boldsymbol{K}_c \begin{bmatrix} {}^z d\boldsymbol{O}_l \\ d\boldsymbol{\omega} \end{bmatrix} \end{aligned} \quad \text{(D.5)}$$

where $\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{I}_{(3,3)} & \hat{Z}_{(3,3)} \\ \boldsymbol{0}_{(3,3)} & \boldsymbol{I}_{(3,3)} \end{bmatrix}$ and $\hat{Z}$ is the skew matrix:

$$\hat{Z} = \begin{bmatrix} 0 & \boldsymbol{O}_{l,n} & -\boldsymbol{O}_{l,t_2} + \boldsymbol{Z}_{t_2} \\ -\boldsymbol{O}_{l,n} & 0 & \boldsymbol{O}_{l,t_1} - \boldsymbol{Z}_{t_1} \\ \boldsymbol{O}_{l,t_2} - \boldsymbol{Z}_{t_2} & -\boldsymbol{O}_{l,t_1} + \boldsymbol{Z}_{t_1} & 0 \end{bmatrix} \quad \text{(D.6)}$$

References

[1] P.-B. Wieber, S. Kuindersma, R. Tedrake, Modeling and control of legged robots, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Berlin Heidelberg, 2nd edn., 2016.

[2] A. Bussy, A. Kheddar, A. Crosnier, F. Keith, Human-humanoid haptic joint object transportation case study, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 3633–3638, 2012.

[3] K. Miura, M. Morisawa, S. Nakaoka, F. Kanehiro, K. Harada, K. Kaneko, S. Kajita, Robot motion remix based on motion capture data –Towards human-like locomotion of humanoid robots– , in: IEEE/RAS International conference on Humanoid Robots, Paris, France, 596–603, 2009.

[4] K. Mombaur, A. Truong, J.-P. Laumond, From human to humanoid locomotion– an inverse optimal control approach, Autonomous Robot 28 (2010) 369–383.

[5] M. L. Felis, K. Mombaur, H. Kadone, A. Berthoz, Modeling and identification of emotional aspects of locomotion, Journal of Computational Science 4 (2013) 255–261.

[6] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, A. Kheddar, Collaborative human-humanoid carrying using vision and haptic sensing, in: IEEE International Conference on Robotics and Automation, Hong Kong, China, 607–612, 2014.

[7] P. Adamczyk, S. Collins, A. Kuo, The advantages of a rolling foot in human walking, The Journal of Experimental Biology 209 (Pt 20) (2006) 3953–3963.

[8] M. Kwan, M. Hubbard, Optimal foot shape for a passive dynamic biped, Journal of Theoretical Biology 248 (2007) 331–339.

[9] T. Fukushima, K. Hyodo, M. Kawanishi, T. Narikiyo, Gait Stabilization of Passive Dynamic Walking by Foot Shape Optimization, Transactions of the Society of Instrument and Control Engineers 50 (1) (2014) 51–57.

[10] S. Miyakoshi, G. Cheng, Examining Human Walking Characteristics with a Telescopic Compass-like Biped Walker Model, in: IEEE International Conference on Systems, Man and Cybernetics, 1538–1543, 2004.

[11] P. Aerts, R. Ker, D. D. Clercq, D. Ilsleys, R. Alexander, The mechanical properties of the human heel pad: a paradox resolved, Journal of Biomechanics 28 (11) (1995) 1299–1308.

[12] P. Sardain, G. Bessonnet, Gait Analysis of a Human Walker wearing Robot Feet as Shoes, in: IEEE International Conference on Robotics and Automation, Seoul, 2285–2292, 2001.

[13] 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.

[14] 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.

[15] 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.

[16] S. Kajita, K. Yokoi, M. Saigo, K. Tanie, Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback, in: IEEE International Conference on Robotics and Automation, vol. 4, 3376–3382, 2001.

[17] S. Nakaoka, S. Hattori, F. Kanehiro, S. Kajita, H. Hirukawa, Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms, in: IEEE International Conference on Intelligent Robots and Systems, San Diego, USA, 3641–3647, 2007.

[18] 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.

[19] 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.

[20] A. David, J.-R. Chardonnet, A. Kheddar, K. Kaneko, K. Yokoi, Study of an external passive shock-absorbing mechanism for walking robots, in: IEEE/RAS International conference on Humanoid Robots, Daejeon, Republic of Korea, 435–440, 2008.

[21] G. Allaire, Conception optimale de structures, Springer-Verlag Berlin Heidelberg, 2007.

[22] J. Arora, Q. Wang, Review of formulations for structural and mechanical system optimization, Structural and Multidisciplinary Optimization 30 (4) (2005) 251–272.

[23] R. T. Haftka, R. V. Grandhi, Structural shape optimization – A survey, Computer Methods in Applied Mechanics and Engineering 57 (1) (1985) 91–106.

[24] F. Espiga, L. Gracia, M. Doblare, Shape optimization of elastic homogeneous 2D bodies by the boundary element method, Computers and Structures 33 (5) (1989) 1233.

[25] T. D. Dang, R. K. Kapania, W. C. H. Slemp, M. Bhatia, S. P. Gurav, Optimization and Postbuckling Analysis of Curvilinear-

Stiffened Panels Under Multiple-Load Cases, Journal of Aircraft 47 (5) (2010) 1656–1671.
[26] S. Percival, D. Hendrix, F. Noblesse, Hydrodynamic optimization of ship hull forms, Applied Ocean Research 23 (6) (2001) 337–355.
[27] V. Akcelik, G. Biros, O. Ghattas, D. Keyes, K. Ko, L.-Q. Lee, E. G. Ng, Adjoint methods for electromagnetic shape optimization of the low-loss cavity for the International Linear Collider, Computers & Fluids 16 (1) (2005) 435–445.
[28] F. Muyl, L. Dumas, V. Herbert, Hybrid method for aerodynamic shape optimization in automotive industry, Computers & Fluids 33 (5-6) (2004) 849–858.
[29] T. L. Holst, Genetic Algorithms Applied to Multi-Objective Aerospace Shape Optimization, Journal of Aerospace Computing, Information, and Communication 2 (4) (2005) 217–235.
[30] H. S. Kohli, G. F. Carey, Shape optimization using adaptive shape refinement, International Journal for Numerical Methods in Engineering 36 (14) (1993) 2435–2451.
[31] J.-A. Desideri, B. A. E. Majd, A. Janka, Nested and self-adaptive Bezier parameterizations for shape optimization, Journal of Computational Physics 224 (14) (2007) 117–131.
[32] A. P. Nagy, M. M. Abdalla, Z. Gurdal, Isogeometric sizing and shape optimisation of beam structures, Computer Methods in Applied Mechanics and Engineering 199 (17-20) (2010) 1216–1230.
[33] X. Han, D. W. Zingg, An Evolutionary Geometry Parametrization for Aerodynamic Shape Optimization, in: AIAA-2011-3536, 2011.
[34] D. Zingg, M. Nemec, T. H. Pulliam, Comparative Evaluation of Genetic and Gradient-Based Algorithms Applied to Aerodynamic Optimization, Shape design in aerodynamics REMN 17 (2008) 103–126.
[35] 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.
[36] 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.
[37] A. Signorini, Sopra alcune questioni di elastostatica, Atti della Societa Italiana per il Progresso delle Scienze, 1933.
[38] K. Murty, Linear Complementarity, Linear and Nonlinear Programming, Internet Edition URL `http://www-personal.umich.edu/~murty/books/linear_complementarity_webbook/`.
[39] 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.
[40] 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.
[41] M. Jean, Numerical methods for three dimensional dynamical problems, in: Conference in Contact Mechanics, Southampton, 71, 1993.
[42] M. Jean, Documentation sur LMGC: Logiciel de Mécanique Gérant le Contact, Tech. Rep., Université la Méditerranée, Marseilles, 1996.
[43] P. Alart, A. Curnier, Time-stepping for threedimensal rigid body dynamics, Computer Methods in Applied Mechanics and Engineering 92 (3) (1991) 353–375.
[44] P. Alart, Critères d'injectivité et de surjectivité pour certaines applications de $\mathbb{R}^n$ dans lui-même; application à la mécanique du contact, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 27 (2) (1993) 203–222, URL `http://eudml.org/doc/193701`.
[45] 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.
[46] G. De Magistris, A. Pajon, S. Miossec, A. Kheddar, Optimized humanoid walking with soft soles, Robotics and Autonomous Systems (RAS) .
[47] 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.
[48] M. C. in Part I of the Engineering Tripos, Materials Data Book, Cambridge University Engineering Department, 2003.
[49] F. Hecht, O. Pironneau, K. Ohtsuka, FreeFem++ Manual, `http://www.freefem.org/ff++/ftp/old-v1/full-ff++-manual.pdf`, 2004.
[50] K. Yamane, L. Trutoiu, Effect of foot shape on locomotion of active biped robots, in: IEEE-RAS International Conference on Humanoid Robots, Paris, France, 230–236, 2009.
[51] 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.