World Scientific
www.worldscientific.com

Impulsive Pedipulation of a Spherical Object with 3D Goal Position by a Humanoid Robot: A 3D Targeted Kicking Motion Generator[*]

Rafael Cisneros[†]

Humanoid Research Group,
National Institute of Advanced Industrial Science
and Technology (AIST),
Tsukuba Central 2, 1-1-1 Umezono,
Tsukuba, Ibaraki 305-8568, Japan
rafael.cisneros@aist.go.jp

Kazuhito Yokoi

Intelligent Systems Research Institute,
Department of Information Technology and Human Factors,
National Institute of Advanced Industrial Science
and Technology (AIST),
Tsukuba Central 2, 1-1-1 Umezono,
Tsukuba, Ibaraki 305-8568, Japan
kazuhito.yokoi@aist.go.jp

Eiichi Yoshida

CNRS–AIST JRL (Joint Robotics Laboratory), UMI3218/CRT,
National Institute of Advanced Industrial Science
and Technology (AIST),
Tsukuba Central 2, 1-1-1 Umezono,
Tsukuba, Ibaraki 305-8568, Japan
e.yoshida@aist.go.jp

This paper describes an algorithm that enables a humanoid robot to perform an impulsive pedipulation task on a spherical object in the environment; that is, by using its feet to exert an impulsive force capable of driving the object to a 3D goal position while achieving certain motion characteristics. This is done by planning a suitable motion for the legs of the humanoid,

capable to exert the required impact conditions on the spherical object while maintaining the dynamic stability of the robot. As an example of this algorithm implementation we consider the free kick in soccer and take it as a case study. Finally, we provide some simulation and experimental results that intend to show the validity of this algorithm.

1. Introduction

Manipulation, from the Latin *manus* "hand" plus the root *plere* "to fill", is defined as the sense of "skillful handling of objects"; that is, a dexterous method or process by which objects are moved, operated or controlled by using the hands, or any other mechanical means.[1] One of the main purposes of traditional industrial robots is the manipulation of objects in the environment by means of an *end effector*, whose shape and functionality depend highly on the task it is meant for. One common approach is to use grippers, "hand"-like end effectors intended for skillfully handling targeted objects by means of a pick-and-place strategy; that is, by grasping them, carrying them to a new location, and releasing them.[2]

On the other hand, humanoid robots may or may not have a proper grasping system, as the focus of most of the research nowadays is on the locomotion and the whole body motion itself. Furthermore, even if they had such a system providing the functionality of the hand, this one may lack the size, strength, or dexterity needed to grasp objects in the same way that humans do.[3] These objects in the environment can be located at different heights, or even placed at floor level, such that if the robot is required to change the position or layout of these objects without lifting them from the floor, the robot would have to bend down to perform the manipulation task. However, this is not efficient from the point of view of the energy required. In such a situation, humans even choose to push (or tap) the object by means of their feet. These objects may represent an obstruction on the path, which can be easily cleared by using the feet if the objects are light enough, especially if the grasping system is dedicated to another task. Then, it is worth to consider the use of the feet (or any part of the legs) of the humanoid robot as an alternative end effector.

This last kind of "manipulation" which uses the feet instead of the hands can be termed as *pedipulation*, from the Latin *pes* (genitive *pedis*) "foot",[1] and defined as the process by which objects in the environment are moved, operated or controlled by using the feet, especially in a skillful manner. This term has been previously used in the robotics literature, referring to the footstep planning for a humanoid robot and to describe the action performed by the foot of a human or a robot when it is used to juggle objects, but not in a generalized way of handling them with the feet while maintaining its balance.[4,5] In order to perform these pedipulation tasks, the robot could simply make use of articulated toes intended to grasp objects. However, this behavior is not typical on humans and not practical either, as it would require complex additional hardware to be installed at the feet. That is, conventional grasp-based pedipulation is not practical. As a feasible alternative it is possible for the

actual feet of the robot to make use of graspless strategies. These ones exploit the mechanics of the task to achieve a goal state without the need of specialized hardware, allowing simple mechanisms to accomplish complex tasks.[3]

Graspless strategies are characterized by the use of repulsive forces, which may be *impulsive* or *non-impulsive*. Impulsive forces are mainly used by *striking* strategies which basically provide the initial linear and angular velocities to an object by means of an impact. Then, this object continues its motion subject to forces and constraints imposed by the environment.[6] Non-impulsive forces, on the other hand, are mainly used by *pushing*, *throwing*, *rolling* and other strategies. Pushing is preferable when a fine handling is required, or in case that the initial and the goal locations share the same support face and the distance between them be short enough.[3] In this case the movement of the object coincides with the motion of the robot, as detaching is not desirable. The other strategies smoothly accelerate the object and release it when the required linear and angular velocities are reached, letting the object to continue its motion.[7,6]

Pedipulation tasks that use non-impulsive or impulsive strategies are not uncommon in our daily lives. For example, when pushing a pedal or operating any mechanism placed at floor level. It is also possible, as mentioned before, to push or strike an object at floor level to move it, in order to line it up, bring it closer or push it away, without bending down. Other typical applications can be found in some sports as in soccer, being the most representative one, whose primary objective is to drive the ball to the opposite goal by using the feet, involving tasks such as stopping ("killing"), dribbling and kicking the ball. The kicking motion is not a new issue for the research in humanoid robots. It has been previously addressed several times. For example, Müller *et al.* focused on creating complex motions by dividing them into simple phases like kicking to the front or to the side,[8] while Wenk and Röfer successfully calculated how to kick a ball in a certain 2D direction by using a mathematical model of the contour of the swing foot.[9] Others just took this motion as an example to validate optimization and planning algorithms,[10–12] stabilization or whole body motion generation.[13,14] That is, the respective research has mainly been focused on the humanoid motion itself and the stability of the humanoid robot without considering to achieve any specific goal position, not even in 3D. On the other hand, some works dealt with the trajectory of the ball without considering the complexity of a humanoid robot, which has no fixed link. For example, Schempf *et al.* developed a fixed-base robotic leg capable of kicking a ball to a distance of about 20 m and with a height of 3–5 m, without having a particular position as a target.[15] Choi *et al.* also used a similar fixed-based leg but they focused on the trajectory of the ball resulting from a given kicking motion without solving the inverse problem; that is, without considering the way of producing the necessary impact conditions on a ball for it to follow some desired trajectory.[16]

It is our concern for a humanoid robot to perform this type of pedipulation tasks in order to achieve a desired behavior on the pedipulated object. For this purpose, the main contribution of this paper is the solution of the corresponding inverse

mechanics problem and the planning of a suitable stable motion for the humanoid robot capable to achieve the required impact conditions.

Having this purpose in mind, we chose to deal with the kicking motion problem such that a ball, idealized as a rigid body, could reach a specific goal position with certain motion characteristics. Solving this problem can improve the capabilities of the actual humanoid robots that play soccer in tournaments by using simplistic methods. One of these tournaments, *RoboCup*, has as an official goal that "*by the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup.*"[17] But, besides that direct application, we believe that if this problem is successfully solved, our approach can be applied to other pedipulation tasks involving non-spherical objects that can be modeled as rigid ones, improving the "manipulation" capabilities of the robot without using additional hardware.

2. Problem Statement

Considering the free kick motion in soccer as a case study on impulsive pedipulation, the present work focuses on driving a ball to a desired 3D goal position while achieving certain motion characteristics and considering the constraints imposed by the nature of the impact. The required impact should be exerted by the foot of a humanoid robot, whose motion needs to be properly planned to perform the desired task while ensuring the stability of the robot.[18]

In order to solve this problem we propose to follow the process represented by the flowchart shown in Fig. 1 and explained as follows:

(i) Given a 3D *goal position* for the ball, as well as some desired motion characteristics, it is first necessary to calculate the required *initial linear and angular velocities* of the ball; that is, to solve its *inverse motion model*.

(ii) Assuming that the ball is originally steady, it is necessary to calculate the *impulse* required to produce the change in momentum leading to the desired motion, as well as where to apply it; that is, some *impact coordinates* defined on the ball. Both the impulse vector and its point of application are the *required impact conditions*.

(iii) By taking into account both the *impulsive model* of the ball and the impulsive model of the robot, the required *approaching velocity* of some selected operational point on the swing foot is then calculated.

(iv) By knowing the contact point where the impulse should be applied and, proposing feasible attitudes for the support foot and the waist, the *configuration of the humanoid robot* at the instant of the impact can be calculated.

(v) By proposing proper detaching and landing footprints, a *trajectory for the swing foot* can be generated, as well as some *desired ZMP trajectory* that assures the *dynamic stability* of the humanoid robot.
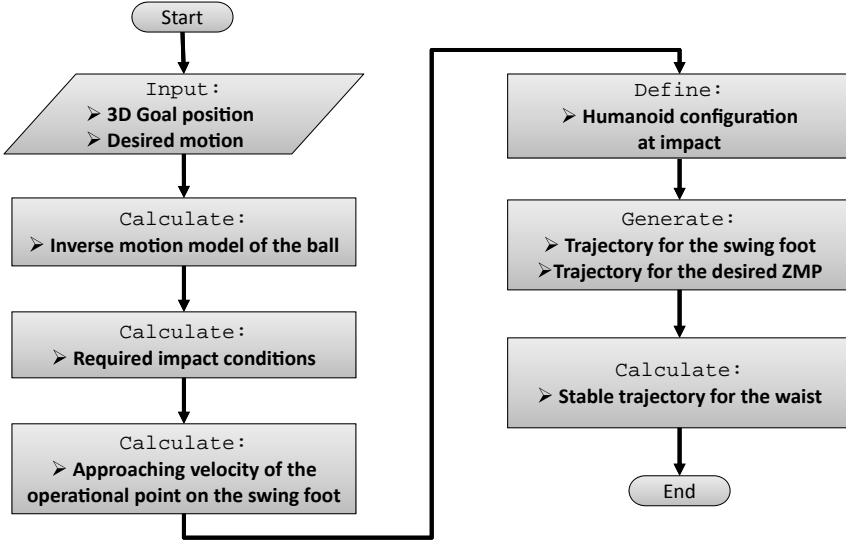
Fig. 1.   Steps needed to solve the impulsive pedipulation problem.

(vi) Finally, the proposed generated motion is stabilized by means of a proper *stable trajectory for the waist*, capable of realizing the ZMP trajectory.

The next sections deal with every one of these stages.

3. Projectile Motion

Let us consider a world reference frame $\{W\}$ whose Z-axis points upwards, as well as a local reference frame $\{B\}$ rigidly attached to the center of mass (CoM) of the ball. Any vector that is described in $\{B\}$ will be denoted with a leading superscript indicating that it is referenced to it. Vectors that are not denoted by any leading superscript should be considered as being described in $\{W\}$. Position vectors are always denoted by p plus a descriptive subscript.

Also, let us assume that the ball will follow a projectile motion under the sole influence of gravity; that is, the effect of various aerodynamical forces (the drag force and the "Magnus effect") will be neglected.[16] Under these circumstances the magnitude of the angular velocity of the ball during the trajectory cannot change. Then, the magnitude of its initial angular velocity, $\boldsymbol{\omega}_{B,0}$, should be equal to the final one, $\boldsymbol{\omega}_{B,f}$; that is,

$$\boldsymbol{\omega}_{B,0} = \boldsymbol{\omega}_{B,f}. \tag{1}$$

Now, given a 3D goal position for the ball, $\boldsymbol{p}_G$, we can use the standard projectile motion equations under the sole influence of gravity that describe the trajectory of its

CoM, $\boldsymbol{p}_{\boldsymbol{O}_{\boldsymbol{B}}}(t)$, to find a suitable initial velocity vector, $\boldsymbol{v}_{\boldsymbol{O}_{\boldsymbol{B}},\boldsymbol{0}}$. This trajectory is given by

$$\boldsymbol{p}_{\boldsymbol{O}_{\boldsymbol{B}}}(t) = \boldsymbol{p}_{\boldsymbol{O}_{\boldsymbol{B}}}(t_k) + \boldsymbol{v}_{\boldsymbol{O}_{\boldsymbol{B}},\boldsymbol{0}}t - \frac{1}{2}\boldsymbol{g}t^2, \tag{2}$$

where t_k is the instant of the impact, $\boldsymbol{g} = [0 \quad 0 \quad -g]^T$ is the gravity vector and g is the acceleration due to gravity, whereas the initial linear velocity vector, $\boldsymbol{v}_{\boldsymbol{O}_{\boldsymbol{B}},\boldsymbol{0}}$, may be described by the magnitude $v_{O_B,0}$ and two angles representing its direction (ϕ_m, $\theta_{m,0}$) (Fig. 2): ϕ_m, the *azimuthal angle* between the ground projection of the trajectory and the X-axis of $\{W\}$, and $\theta_{m,0}$, the *angle of launch*. In such a way that

$$\boldsymbol{v}_{\boldsymbol{O}_{\boldsymbol{B}},\boldsymbol{0}} = \begin{bmatrix} v_{O_B,0}\cos\phi_m\cos\theta_{m,0} \\ v_{O_B,0}\sin\phi_m\cos\theta_{m,0} \\ v_{O_B,0}\sin\theta_{m,0} \end{bmatrix}; \tag{3}$$

then, considering the goal is attained at time T, we can get $\boldsymbol{p}_{\boldsymbol{G}} = \boldsymbol{p}_{\boldsymbol{O}_{\boldsymbol{B}}}(T)$.

Now, let us define $\boldsymbol{r}_{\boldsymbol{G}/\boldsymbol{O}_{\boldsymbol{B}0}} := \boldsymbol{p}_{\boldsymbol{G}} - \boldsymbol{p}_{\boldsymbol{O}_{\boldsymbol{B}},\boldsymbol{0}}$, such that

$$\boldsymbol{r}_{\boldsymbol{G}/\boldsymbol{O}_{\boldsymbol{B}0}} = \begin{bmatrix} r_{G/O_{B0x}} \\ r_{G/O_{B0y}} \\ r_{G/O_{B0z}} \end{bmatrix} = \begin{bmatrix} v_{O_B,0}T\cos\phi_m\cos\theta_{m,0} \\ v_{O_B,0}T\sin\phi_m\cos\theta_{m,0} \\ v_{O_B,0}T\sin\theta_{m,0} - \frac{1}{2}gT^2 \end{bmatrix}. \tag{4}$$

The angle ϕ_m represents the orientation of the *plane of motion* (where the projectile motion actually occurs) with respect to the *XZ*-plane of $\{W\}$, remaining constant during the whole trajectory. Then, it can be directly calculated from the goal position as

$$\phi_m = \arctan 2(r_{G/O_{B0y}}, r_{G/O_{B0x}}). \tag{5}$$

However, the other three variables (T, $v_{O_B,0}$, $\theta_{m,0}$) cannot be directly calculated because the three equations are not independent. There is an infinite set of possible
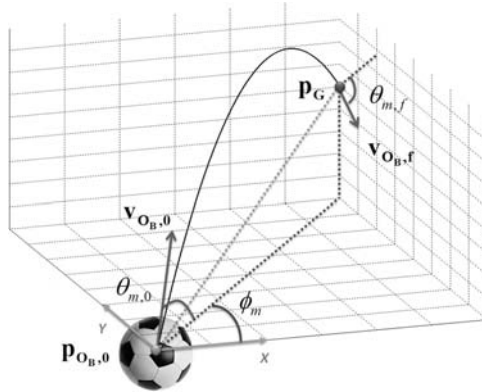

Fig. 2. Projectile motion.

trajectories capable of accomplishing the required goal, unless we specify another constraint for the problem.

Let us select the *reaching angle* $\theta_{m,f}$ (Fig. 2) as a constraint. This one is defined as the direction of the motion of the projectile at the goal position, and within the plane of motion; that is, the angle between the velocity vector at that point, $\boldsymbol{v_{O_B,f}}$, and the XY-plane of $\{W\}$.

The velocity of the ball at the goal position is obtained by $\boldsymbol{v_{O_B,f}} = \frac{d}{dt}\boldsymbol{p_{O_B}}(T) = [v_{O_{Bx},f} \;\; v_{O_{By},f} \;\; v_{O_{Bz},f}]^T$, such that

$$\tan\theta_{m,f} = \frac{v_{O_{Bz},f}}{\sqrt{v_{O_{Bx},f}^2 + v_{O_{By},f}^2}} = \frac{v_{O_B,0}\sin\theta_{m,0} - gT}{v_{O_B,0}\cos\theta_{m,0}}. \tag{6}$$

Now, let us focus in Eq. (4). From the expression for $r_{G/O_{B0x}}$ we can solve for the required time to accomplish the goal, T, as

$$T = \frac{r_{G/O_{B0x}}}{v_{O_B,0}\cos\phi_m\cos\theta_{m,0}}, \tag{7}$$

which can be substituted into the expression for $r_{G/O_{B0z}}$ to get

$$r_{G/O_{B0z}} = \frac{r_{G/O_{B0x}}}{\cos\phi_m}\tan\theta_{m,0} - \frac{1}{2}g\left(\frac{r_{G/O_{B0x}}^2}{\cos^2\phi_m}\right)\left(\frac{1+\tan^2\theta_{m,0}}{v_{O_B,0}^2}\right). \tag{8}$$

Substituting T of Eq. (7) into Eq. (6) the velocity $v_{O_B,0}$ can be calculated as

$$v_{O_B,0} = \sqrt{\frac{r_{G/O_{B0x}}}{\cos\phi_m}\frac{g(1+\tan^2\theta_{m,0})}{\tan\theta_{m,0} - \tan\theta_{m,f}}}. \tag{9}$$

Then, by considering the expression given in Eq. (8) and substituting this value, $v_{O_B,0}$, it is possible to compute the required angle of launch, $\theta_{m,0}$, as

$$\theta_{m,0} = \arctan\left(2r_{G/O_{B0z}}\frac{\cos\phi_m}{r_{G/O_{B0x}}} - \tan\theta_{m,f}\right). \tag{10}$$

4. Impulsive Contact Models

After finding the required initial linear and angular velocity vectors of the ball ($\boldsymbol{v_{O_B,0}}$ and $\boldsymbol{\omega_{B,0}}$), and assuming that the ball is originally steady as well as rigid, it is necessary to calculate the *impulse* $\tilde{\boldsymbol{f}}_{c_B}$ needed to produce the change in momentum required to achieve the desired motion, as well as the point on the ball to apply that impulse; that is, the *impact coordinates*, (ϕ_0, θ_0) (Fig. 3).

Then, we are able to calculate the approaching velocity of some selected *operational point* on the swing foot, which will exert the necessary impulse to start the motion of the ball. This is done by first formulating and solving the dynamical model of both the ball and the robot at the instant of the impact; that is, considering the
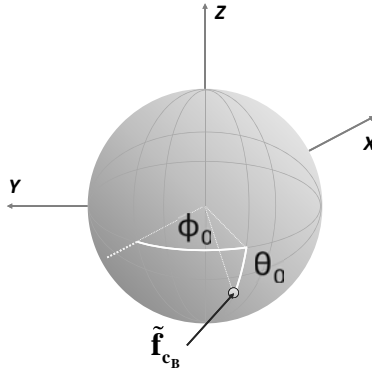
Fig. 3.   Definition of the impact coordinates in the world reference frame.

presence of the external impulse and a point contact model, given that the current humanoid robots cannot kick fast enough to produce considerable deformation on the ball.

4.1. *Orientation of the ball*

By idealizing the ball as a plain sphere of radius r, it does not have any inherent orientation. However, in order to introduce the impact coordinates, (ϕ_0, θ_0), into the model we can consider that its local reference frame, $\{B\}$, is oriented in such a way that the ball is always kicked at the same point described in the local reference frame; that is, the contact point on the ball, $^B\boldsymbol{p}_{c_B}$, has a constant representation on $\{B\}$ (see Fig. 4), given by

$$^B\boldsymbol{p}_{c_B} = [\,-r_B \quad 0 \quad 0\,]^T. \tag{11}$$

Then, if we describe the orientation of $\{B\}$ with respect to $\{W\}$ by using the Euler ZYX convention and the orientation coordinates $(\phi_B, \theta_B, \psi_B)$, we will have the
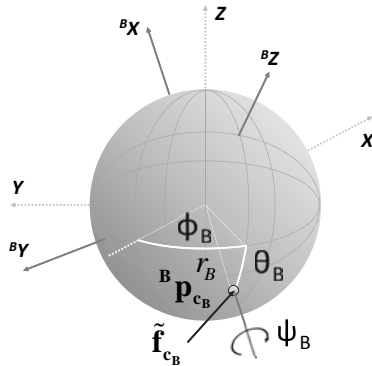

Fig. 4.   Relationship between the ball reference frame and the world reference frame.

following correspondence with the impact coordinates: $\phi_0 = \phi_B(t_0)$ and $\theta_0 = \theta_B(t_0)$, where t_0 stands for the initial time. The rotation matrix that describes the orientation of $\{B\}$ with respect to $\{W\}$ is denoted by $\boldsymbol{R_B}(\phi_B, \theta_B, \psi_B)$, and expressed as

$$\boldsymbol{R_B} = \begin{bmatrix} c_\phi\, c_\theta & -s_\phi\, c_\psi + c_\phi\, s_\theta\, s_\psi & s_\phi\, s_\psi + c_\phi\, s_\theta\, c_\psi \\ s_\phi\, c_\theta & c_\phi\, c_\psi + s_\phi\, s_\theta\, s_\psi & -c_\phi\, s_\psi + s_\phi\, s_\theta\, c_\psi \\ -s_\theta & c_\theta\, s_\psi & c_\theta\, c_\psi \end{bmatrix}, \tag{12}$$

where $c_\phi = \cos\phi_B$, $s_\phi = \sin\phi_B$, $c_\theta = \cos\theta_B$, $s_\theta = \sin\theta_B$, $c_\psi = \cos\psi_B$ and $s_\psi = \sin\psi_B$. Let us remark that only when $\theta_B = \pm\frac{\pi}{2}$ the matrix $\boldsymbol{R_B}$ becomes singular; that is, when describing the highest and lowest points of the ball, which are not of interest to the task as no useful kicks can be attained at those points.

4.2. *Lagrange equation of the ball*

Let us describe the configuration of the ball, $\boldsymbol{q_B} \in \mathbb{R}^6$, by using the set of *generalized coordinates* given by the position of its CoM, $\boldsymbol{p_{O_B}}$, and its orientation coordinates, as

$$\boldsymbol{q_B} = \begin{bmatrix} \boldsymbol{q_{B_p}^T} & \boldsymbol{q_{B_r}^T} \end{bmatrix}^T = \begin{bmatrix} p_{O_{Bx}} & p_{O_{By}} & p_{O_{Bz}} & \phi_B & \theta_B & \psi_B \end{bmatrix}^T. \tag{13}$$

If the mass of the ball is denoted as m_B and its inertia tensor with respect to its center of gravity as $\boldsymbol{I_B}$, then its kinetic energy, $\mathcal{T}_B$, is given by

$$\mathcal{T}_B = \frac{1}{2} m_B \boldsymbol{v_{O_B}^T} \boldsymbol{v_{O_B}} + \frac{1}{2} \boldsymbol{\omega_B}^T \boldsymbol{I_B} \boldsymbol{\omega_B}, \tag{14}$$

where $\boldsymbol{v_{O_B}}$ and $\boldsymbol{\omega_B}$ are the linear velocity of CoM of the ball and its angular velocity, respectively. The linear velocity is related to the rate of change of the position coordinates by

$$\boldsymbol{v_{O_B}^T} \boldsymbol{v_{O_B}} = \dot{p}_{O_B}^T \dot{p}_{O_B} = \dot{p}_{O_{Bx}}^2 + \dot{p}_{O_{By}}^2 + \dot{p}_{O_{Bz}}^2. \tag{15}$$

The angular velocity, on the other hand, is related to the rate of change of the orientation coordinates $(\dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B)$ given that the *skew-symmetric matrix* of the angular velocity vector of the ball, denoted as $\hat{\boldsymbol{\omega}}_B$, is a function of $\dot{\boldsymbol{R}}_B$:

$$\hat{\boldsymbol{\omega}}_B = (\dot{\boldsymbol{R}}_B)(\boldsymbol{R_B})^T. \tag{16}$$

Remembering that a *skew-symmetric matrix* $\hat{\boldsymbol{\omega}}$ of a vector $\boldsymbol{\omega}$ is defined as

$$\hat{\boldsymbol{\omega}} := \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad \forall\, \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tag{17}$$

then the vector $\boldsymbol{\omega}_B$ can be extracted from Eq. (16) and written as

$$\boldsymbol{\omega}_B = \begin{bmatrix} 0 & -s_\phi & c_\phi\, c_\theta \\ 0 & c_\phi & s_\phi\, c_\theta \\ 1 & 0 & -s_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi}_B \\ \dot{\theta}_B \\ \dot{\psi}_B \end{bmatrix} = \boldsymbol{\Omega}_B \dot{\boldsymbol{q}}_{\boldsymbol{B}_r}. \tag{18}$$

Finally, the inertia tensor of the ball can be expressed as

$$\boldsymbol{I_B} = \rho_B m_B r_B^2 \boldsymbol{E_3}, \tag{19}$$

where $\boldsymbol{E_n} \in \mathbb{R}^{3\times3}$ is an identity matrix and ρ_B is termed as the *construction coefficient* of the ball. This one lets us generalize the inertia tensor to any type of ball, whether it is a filled sphere with constant density, $\rho_B = \frac{2}{5}$, an ideal hollow sphere, $\rho_B = \frac{2}{3}$, or any other spherical construction. For example, according to Brody,[19] for a pressurized tennis ball $\rho_B = 0.535$, whereas for a pressureless one $\rho_B = 0.509$.

On the other hand, the potential energy is given by

$$\mathcal{V}_B = m_B \boldsymbol{g}^T \boldsymbol{p_{O_B}}. \tag{20}$$

Having done this, the Lagrangian is calculated as $L_B = \mathcal{T}_B - \mathcal{V}_B$, such that the dynamical equation of the ball during the impact can be expressed by

$$\frac{d}{dt}\frac{\partial L_B}{\partial \dot{\boldsymbol{q}}_B} - \frac{\partial L_B}{\boldsymbol{q}_B} = \boldsymbol{Q_B}, \tag{21}$$

where $\boldsymbol{Q_B} \in \mathbb{R}^6$ stands for the resultant of all the *generalized forces* at the moment of the impact. Assuming that only two generalized forces are present: one related to a non-impulsive normal force exerted by the ground at the lowest part of the ball, $\boldsymbol{Q_{s_B}} \in \mathbb{R}^6$ (assuming that the ball is not hit against the ground), and one related to an impulsive force created by the impact, $\boldsymbol{Q_{c_B}} \in \mathbb{R}^6$, then we have $\boldsymbol{Q_B} = \boldsymbol{Q_{s_B}} + \boldsymbol{Q_{c_B}}$.

4.3. *Description of the impulsive force*

In order to calculate the *generalized impulsive force* created by the impact, we need first to express $^B\boldsymbol{p_{c_B}}$ defined in Eq. (11) in $\{W\}$ as

$$\boldsymbol{p_{c_B}} = \boldsymbol{p_{O_B}} + \boldsymbol{R_B}{}^B\boldsymbol{p_{c_B}} = \begin{bmatrix} p_{O_{Bx}} - r_B c_\phi c_\theta \\ p_{O_{By}} - r_B s_\phi c_\theta \\ p_{O_{Bz}} + r_B s_\theta \end{bmatrix}, \tag{22}$$

from which the Jacobian $\boldsymbol{J_{B,c,v}} \in \mathbb{R}^{3\times6}$ that relates the velocity of $\boldsymbol{p_{c_B}}$, denoted as $\boldsymbol{v_{c_B}}$, with $\dot{\boldsymbol{q}}_B$ is calculated by differentiating the last equation with respect to t as

$$\boldsymbol{J_{B,c,v}} = \begin{bmatrix} 1 & 0 & 0 & r_B s_\phi c_\theta & r_B c_\phi s_\theta & 0 \\ 0 & 1 & 0 & -r_B c_\phi c_\theta & r_B s_\phi s_\theta & 0 \\ 0 & 0 & 1 & 0 & r_B c_\theta & 0 \end{bmatrix}. \tag{23}$$

The transpose of this Jacobian, $\boldsymbol{J_{B,c,v}^T}$, relates the force $\boldsymbol{f_{c_B}}$ applied to the point $\boldsymbol{p_{c_B}}$ with the *generalized impulsive force* $\boldsymbol{Q_{c_B}}$; that is,

$$\boldsymbol{Q_{c_B}} = \boldsymbol{J_{B,c,v}^T} \boldsymbol{f_{c_B}}. \tag{24}$$

Also, in a similar way, $\boldsymbol{Q_{s_B}}$ can be written as $\boldsymbol{Q_{s_B}} = \boldsymbol{J_{B,s,v}^T} \boldsymbol{f_{s_B}}$.

4.4. *Impulsive contact model of the ball*

One way of computing the impulsive contact model of the ball is suggested by Walker.[20] The dynamical equation of the ball in Eq. (21) can also be expressed as

$$\boldsymbol{M_B}\ddot{\boldsymbol{q}}_B + \mathcal{C}_B + \mathcal{G}_B = \boldsymbol{J}_{B,s,v}^T \boldsymbol{f}_{s_B} + \boldsymbol{J}_{B,c,v}^T \boldsymbol{f}_{c_B}, \tag{25}$$

where $\boldsymbol{M_B} \in \mathbb{R}^{6\times6}$ corresponds to the mass matrix, whereas $\boldsymbol{C_B} \in \mathbb{R}^6$ and $\boldsymbol{G_B} \in \mathbb{R}^6$ correspond to the coupling and gravitational vectors, respectively.

This dynamical model is valid only during the short time that the collision lasts; that is, from t_k (the instant of the impact) to $t_k + \Delta t$. Let us integrate Eq. (25) in this period of time and assume that Δt is too small that it is possible to idealize it as $\Delta t \to 0$. Under this assumption, the position and orientation of the ball remain constant during the period of contact. Also, all the velocities, the gravitational effect and the non-impulsive forces are assumed to be finite, such that the integral terms $\int_{t_k}^{t_k+\Delta t} \mathcal{C}_B d\tau$, $\int_{t_k}^{t_k+\Delta t} \mathcal{G}_B d\tau$ and $\int_{t_k}^{t_k+\Delta t} \boldsymbol{J}_{B,s,v}^T \boldsymbol{f}_{s_B} d\tau$ become zero as $\Delta t \to 0$.[16,20] On the other hand, $\int_{t_k}^{t_k+\Delta t} \boldsymbol{f}_{c_B} d\tau$ produces a finite impulse, denoted by $\tilde{\boldsymbol{f}}_{c_B}$, such that

$$\boldsymbol{M_B}\Delta\dot{\boldsymbol{q}}_B = \boldsymbol{J}_{B,c,v}^T \tilde{\boldsymbol{f}}_{c_B}, \tag{26}$$

where $\Delta\dot{\boldsymbol{q}}_B = \dot{\boldsymbol{q}}_B(t_k^+) - \dot{\boldsymbol{q}}_B(t_k^-)$ (as $\Delta t \to 0$, t_k^- and t_k^+ will be used to refer to the instants just before and after the impact, respectively). The expression Eq. (26) corresponds to the principle of generalized impulse and momenta, which can also be written as

$$\Delta\dot{\boldsymbol{q}}_B = \boldsymbol{M}_B^{-1} \boldsymbol{J}_{B,c,v}^T \tilde{\boldsymbol{f}}_{c_B}, \tag{27}$$

given that $\boldsymbol{M_B}$ is a positive-definite matrix and its inverse exists.

4.5. *Impulsive contact model of the humanoid*

Let us describe the configuration of a humanoid robot of n structural degrees of freedom (dof), $\boldsymbol{q_R} \in \mathbb{R}^{n+6}$, by using the set of joint variables, $\boldsymbol{q}_{R_\theta} \in \mathbb{R}^n$, plus the position and orientation of a link of the robot chosen as a reference, $\boldsymbol{q}_{R_p} \in \mathbb{R}^3$ and $\boldsymbol{q}_{R_r} \in \mathbb{R}^3$, respectively. This reference link can be the waist of the robot or its support foot. In this way, $\boldsymbol{q_R} = [\boldsymbol{q}_{R_\theta}^T \ \boldsymbol{q}_{R_p}^T \ \boldsymbol{q}_{R_r}^T]^T$. Then, the dynamic model of the humanoid robot can be expressed in a similar way as for the ball, Eq. (25), as

$$\boldsymbol{M_R}\ddot{\boldsymbol{q}}_R + \mathcal{C}_R + \mathcal{G}_R = \begin{bmatrix} \boldsymbol{\tau_R} \\ \boldsymbol{0} \end{bmatrix} + \boldsymbol{J}_{R,s}^T \begin{bmatrix} \boldsymbol{f}_{s_R} \\ \boldsymbol{\tau}_{s_R} \end{bmatrix} - \boldsymbol{J}_{R,c,v}^T \boldsymbol{f}_{c_B}. \tag{28}$$

In a similar way, $\boldsymbol{M_R} \in \mathbb{R}^{(n+6)\times(n+6)}$ corresponds to the mass matrix of the robot, whereas $\mathcal{C}_R \in \mathbb{R}^6$ and $\mathcal{G}_B \in \mathbb{R}^6$ correspond to the coupling and gravitational vectors, respectively. Also, $\boldsymbol{\tau_R} \in \mathbb{R}^n$ comprises the joint torques, whereas $\boldsymbol{f}_{s_R}$ and $\boldsymbol{\tau}_{s_R}$ represent the force and moment exerted by the ground to the sole of the support foot and $-\boldsymbol{f}_{c_B}$, the reaction exerted by the ball (same magnitude and opposite direction) on the selected *operational point*; that is, the contact point on the foot of the robot, $\boldsymbol{p}_{c_R}$. The Jacobian that relates the velocity of this point, $\boldsymbol{v}_{c_R}$, with the rate of change of the

generalized coordinates of the robot, $\dot{\boldsymbol{q}}_R$, is denoted as $\boldsymbol{J}_{R,c,v} \in \mathbb{R}^{3 \times (n+6)}$, whereas the Jacobian that relates the linear and angular velocities of the sole of the support foot to these generalized coordinates is denoted as $\boldsymbol{J}_{R,s} = [\boldsymbol{J}_{R,s,v}^T \; \boldsymbol{J}_{R,s,\omega}^T]^T \in \mathbb{R}^{6 \times (n+6)}$.

Let us assume that the generalized impulsive force is small compared to the joint torques required to produce the motion of the robot (a realistic assumption). Then, we can imagine the robot at the moment of the impact as just one rigid object (not an assembly of links) and propose an alternative dynamic model by considering the generalized coordinates $\bar{\boldsymbol{q}}_R \in \mathbb{R}^6$, such that $\bar{\boldsymbol{q}}_R = [\boldsymbol{q}_{R_p}^T \; \boldsymbol{q}_{R_r}^T]^T$. In this way,

$$\bar{\boldsymbol{M}}_R \ddot{\bar{\boldsymbol{q}}}_R + \bar{\mathcal{C}}_R + \bar{\mathcal{G}}_R = \bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \boldsymbol{f}_{s_R} \\ \boldsymbol{\tau}_{s_R} \end{bmatrix} - \bar{\boldsymbol{J}}_{R,c,v}^T \boldsymbol{f}_{c_B}, \tag{29}$$

where $\bar{\boldsymbol{M}}_R \in \mathbb{R}^{6 \times 6}$, $\bar{\mathcal{C}}_R \in R^6$, $\bar{\mathcal{G}}_R \in \mathbb{R}^6$, $\bar{\boldsymbol{J}}_{R,s} \in \mathbb{R}^{6 \times 6}$ and $\bar{\boldsymbol{J}}_{R,c,v} \in \mathbb{R}^{3 \times 6}$. The "bar" over each variable indicates that it corresponds to the alternative dynamic model. By integrating Eq. (29) from t_k to $t_k + \Delta t$ such that $\Delta t \to 0$ and considering similar assumptions as for the case of the ball we get

$$\bar{\boldsymbol{M}}_R \Delta \dot{\bar{\boldsymbol{q}}}_R = \bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_{s_R} \\ \tilde{\boldsymbol{\tau}}_{s_R} \end{bmatrix} - \bar{\boldsymbol{J}}_{R,c,v}^T \tilde{\boldsymbol{f}}_{c_B}, \tag{30}$$

where we considered that the ground reaction was also impulsive. The impulsive force exerted by the ball on the robot will be directed against the floor or parallel to it and compensated by the ground, as no penetration is allowed and supposing that the friction between them is large enough to prevent sliding.

Assuming that every link of the robot has nonzero mass, length, and moment of inertia about its CoM, $\boldsymbol{M}_R$ will always be a positive-definite matrix,[21] and Eq. (30) may be expressed in a similar way as for the ball, as

$$\Delta \dot{\bar{\boldsymbol{q}}}_R = \bar{\boldsymbol{M}}_R^{-1} \left(\bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_{s_R} \\ \tilde{\boldsymbol{\tau}}_{s_R} \end{bmatrix} - \bar{\boldsymbol{J}}_{R,c,v}^T \tilde{\boldsymbol{f}}_{c_B} \right). \tag{31}$$

5. Required Impact Conditions

Let us consider Eq. (26). This equation is a nonlinear function of $\tilde{\tilde{\boldsymbol{f}}}_{c_B}$, ϕ_B, θ_B, ψ_B and $\Delta \dot{\boldsymbol{q}}_B$. However, by considering that the ball is steady just before the impact we have $\dot{\boldsymbol{q}}_B(t_k^-) = \boldsymbol{0}$, so that $\Delta \dot{\boldsymbol{q}}_B = \dot{\boldsymbol{q}}_B(t_k^+)$, which is a function of the required initial linear and angular velocities of the ball, $\boldsymbol{v}_{O_B,0}$ and $\boldsymbol{\omega}_{B,0}$. In particular, $\boldsymbol{\omega}_{B,0}$ is closely related to $\dot{\phi}_B(t_k^+)$, $\dot{\theta}_B(t_k^+)$ and $\dot{\psi}_B(t_k^+)$ by means of $\boldsymbol{\Omega}_B$. By expanding Eq. (26) we get six equations corresponding to each of its components. The first three equations relate the applied impulse to the initial linear momentum of the ball, such that we can directly calculate $\tilde{\tilde{\boldsymbol{f}}}_{c_B}$ as

$$\tilde{\tilde{\boldsymbol{f}}}_{c_B} = m_B \boldsymbol{v}_{O_B,0}. \tag{32}$$

On the other hand, the following three equations give us the relationship for its initial angular momentum, from where it is possible to calculate the necessary impact coordinates $\phi_0 = \phi_B(t_k^+)$ and $\theta_0 = \theta_B(t_k^+)$ by solving:

$$
\begin{aligned}
\tilde{f}_{c_{B_x}} s_\phi c_\theta - \tilde{f}_{c_{B_y}} c_\phi c_\theta + \rho_B m_B r_B \dot{\psi}_{B_0} s_\theta &= \rho_B m_B r_B \dot{\phi}_{B_0}, \\
\tilde{f}_{c_{B_x}} c_\phi s_\theta + \tilde{f}_{c_{B_y}} s_\phi s_\theta + \tilde{f}_{c_{B_z}} c_\theta &= \rho_B m_B r_B \dot{\theta}_{B_0}, \\
\dot{\phi}_{B_0} s_\theta &= \dot{\psi}_{B_0},
\end{aligned}
\tag{33}
$$

where $c_\phi = \cos\phi_0$, $s_\phi = \sin\phi_0$, $c_\theta = \cos\theta_0$ and $s_\theta = \sin\theta_0$. However, this last system of equations is overdetermined as we have three equations but just two unknowns. This means that we cannot specify the rate of change for the three orientation coordinates independently, but only for two of them. We arbitrarily choose to specify $\dot{\phi}_{B_0}$ and $\dot{\theta}_{B_0}$, and to treat $\dot{\psi}_{B_0}$ as a variable whose value depends on the evolution of the system. In this way, we shall solve:

$$
\begin{aligned}
\tilde{f}_{c_{B_x}} s_\phi - \tilde{f}_{c_{B_y}} c_\phi - \rho_B m_B r_B \dot{\phi}_{B_0} c_\theta &= 0, \\
\tilde{f}_{c_{B_x}} c_\phi s_\theta + \tilde{f}_{c_{B_y}} s_\phi s_\theta + \tilde{f}_{c_{B_z}} c_\theta - \rho_B m_B r_B \dot{\theta}_{B_0} &= 0.
\end{aligned}
\tag{34}
$$

5.1. *Main components of the impulse*

When two rigid bodies collide, a point on the surface of one of them is coincident with a point on the surface of the other at the *contact point* $\boldsymbol{p_c}$ during the brief period of contact. If at least one of the bodies has a surface that is topologically smooth at the contact point, there is a plane tangent to this surface at this point. If both bodies are convex and the surfaces have continuous curvature near the contact point, then this tangent plane is tangential to both surfaces that touch at $\boldsymbol{p_c}$; i.e., the surfaces of the colliding bodies have a *common tangent plane*.[22]

Then, the impulse $\tilde{\boldsymbol{f}}_{c_B}$ may be resolved into two orthogonal components (Fig. 5) with respect to this common tangent plane: (i) a *normal component* ($\tilde{\boldsymbol{f}}_{c_{Bn}}$) which is directly related to the change in relative motion of the *points* that come into contact, and (ii) a *tangential component* ($\tilde{\boldsymbol{f}}_{c_{Bt}}$) which has to be produced just by the effect of
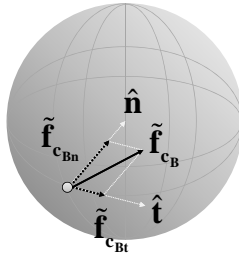

Fig. 5.   Normal and tangential components of the impulse.

friction[22,23] so that

$$\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}} = \tilde{\boldsymbol{f}}_{\boldsymbol{c_{Bn}}} + \tilde{\boldsymbol{f}}_{\boldsymbol{c_{Bt}}} = \tilde{f}_{c_{Bn}} \boldsymbol{\check{n}} + \tilde{f}_{c_{Bt}} \boldsymbol{\check{t}}, \tag{35}$$

where $\boldsymbol{\check{n}}$ and $\boldsymbol{\check{t}}$ are the unit vectors directed in the directions of the orthogonal components. Also, it is worth to notice that the magnitude of the tangential component is limited by the *coefficient of friction*, μ, as

$$\tilde{f}_{c_{Bt}} \leq \mu \tilde{f}_{c_{Bn}}. \tag{36}$$

This inequality must be considered when solving Eq. (34), as explained later.

The vector $\boldsymbol{\check{n}}$ can be calculated from Eq. (22) as

$$\boldsymbol{\check{n}} = \frac{\boldsymbol{p}_{O_B} - \boldsymbol{p}_{c_B}}{\|\boldsymbol{p}_{O_B} - \boldsymbol{p}_{c_B}\|} = \begin{bmatrix} c_\phi c_\theta \\ s_\phi c_\theta \\ -s_\theta \end{bmatrix}, \tag{37}$$

such that for the magnitude of the normal component of the impulse we have

$$\tilde{f}_{c_{Bn}} = \tilde{\boldsymbol{f}}_{\boldsymbol{c_B}} \cdot \boldsymbol{\check{n}} = f_{c_{Bx}} c_\phi c_\theta + f_{c_{By}} s_\phi c_\theta - f_{c_{Bz}} s_\theta. \tag{38}$$

The vector $\boldsymbol{\check{t}}$ can be found by first realizing that it must lie inside of a unit circle perpendicular to $\boldsymbol{\check{n}}$. Let us parametrize this circle by using the angle γ, such that

$$\boldsymbol{\check{t}} = \begin{bmatrix} -c_\gamma s_\phi + s_\gamma c_\phi s_\theta \\ c_\gamma c_\phi + s_\gamma s_\phi s_\theta \\ s_\gamma c_\theta \end{bmatrix}, \tag{39}$$

where $c_\gamma = \cos\gamma$ and $s_\gamma = \sin\gamma$. Then, for the magnitude of the tangential component of the impulse we have

$$\tilde{f}_{c_{Bt}} = \tilde{\boldsymbol{f}}_{\boldsymbol{c_B}} \cdot \boldsymbol{\check{t}} = -f_{c_{Bx}} c_\gamma s_\phi + s_\gamma c_\phi s_\theta + f_{c_{By}} c_\gamma c_\phi + f_{c_{Bz}} s_\gamma s_\phi s_\theta + s_\gamma c_\theta, \tag{40}$$

from which the angle γ can be found by solving $\frac{\partial \tilde{f}_{c_{Bt}}}{\partial \gamma} = 0$, such that

$$\gamma = \pm\frac{\pi}{2} + \arctan\left(\frac{f_{c_{Bx}} s_\phi - f_{c_{By}} c_\phi}{f_{c_{Bx}} c_\phi s_\theta + f_{c_{By}} s_\phi s_\theta + f_{c_{Bz}} c_\theta}\right). \tag{41}$$

5.2. *Solving the impulsive contact model of the ball for ϕ_0, θ_0*

The nonlinear system of equations shown in Eq. (34) shall be solved for ϕ_0 and θ_0 by using a multi-objective optimization procedure as it is the *goal attain* method.[24] This one solves the multi-objective optimization problem defined as

$$\min_{\boldsymbol{y} \in \aleph} \mathcal{F}(\boldsymbol{y}), \tag{42}$$

where $\boldsymbol{y}$ is the design parameter vector, $\aleph$ is the feasible parameter space and $\mathcal{F}$ is the objective vector function, by converting it into the following nonlinear programming

problem:

$$\min_{\zeta, \boldsymbol{y} \in \aleph} \zeta \quad \text{such that} \quad \begin{cases} \mathcal{F}(\boldsymbol{y}) - \boldsymbol{w} \cdot \zeta \leq \boldsymbol{\eta} \\ \boldsymbol{c}(\boldsymbol{y}) \leq \boldsymbol{0} \end{cases}. \tag{43}$$

Here, $\boldsymbol{\eta}$ is a vector of goals for the designed objective vector function $\mathcal{F}$, $\boldsymbol{w}$ is a vector of weights such that $w_i > 0 \forall i$, and $\boldsymbol{c}(\boldsymbol{y}) \leq \boldsymbol{0}$ represents a set of nonlinear inequalities that constrain the solution of the problem. The minimization of the scalar ζ leads to the acquisition of a non-dominated solution which under- or over-attains the specified goals to a degree represented by the quantities $w_i \cdot \zeta$.[25]

For this optimization problem, based on Eq. (34), we define

$$\mathcal{F}(\boldsymbol{y}) = \begin{bmatrix} \tilde{f}_{c_{Bx}} s_\phi - \tilde{f}_{c_{By}} c_\phi - \rho_B m_B r_B \dot{\phi}_{B_0} c_\theta \\ \tilde{f}_{c_{Bx}} c_\phi s_\theta + \tilde{f}_{c_{By}} s_\phi s_\theta + \tilde{f}_{c_{Bz}} c_\theta - \rho_B m_B r_B \dot{\theta}_{B_0} \end{bmatrix}, \tag{44}$$

$$\boldsymbol{y} = \begin{bmatrix} \theta_0 & \phi_0 \end{bmatrix}^T, \tag{45}$$

$$\boldsymbol{\eta} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \tag{46}$$

whereas for $\boldsymbol{w}$, we set an arbitrary value of $w_i = 0.001 \forall i$. The set of nonlinear inequalities $\boldsymbol{c}(\boldsymbol{y}) \leq \boldsymbol{0}$ is used to ensure that Eq. (36) holds.

Let us rewrite Eq. (35) by considering Eq. (38), such that

$$\tilde{f}_{c_{Bt}} = \left\| \tilde{\boldsymbol{f}}_{\boldsymbol{c_B}} - \left(\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}^{\boldsymbol{T}} \check{\boldsymbol{n}} \right) \check{\boldsymbol{n}} \right\|, \tag{47}$$

which can be substituted into the inequality shown in Eq. (36) and squared to get

$$\left\| \tilde{\boldsymbol{f}}_{\boldsymbol{c_B}} - \left(\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}^{\boldsymbol{T}} \check{\boldsymbol{n}} \right) \check{\boldsymbol{n}} \right\|^2 \leq \mu^2 \left(\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}^{\boldsymbol{T}} \check{\boldsymbol{n}} \right)^2. \tag{48}$$

Then, after expanding this expression we have

$$\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}^{\boldsymbol{T}} \hat{\boldsymbol{n}} \geq \sqrt{\frac{\tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}^{\boldsymbol{T}} \tilde{\boldsymbol{f}}_{\boldsymbol{c_B}}}{\mu^2 + 1}} := \Gamma_B, \tag{49}$$

which can be expressed, for this optimization problem, as

$$\boldsymbol{c}(\boldsymbol{y}) = \Gamma_B - \tilde{f}_{c_{Bx}} c_\phi c_\theta - \tilde{f}_{c_{By}} s_\phi c_\theta + \tilde{f}_{c_{Bz}} s_\theta \leq 0. \tag{50}$$

This inequality represents a circle on the surface of the ball, over which the impact should occur in order to accomplish the constraint imposed by the coefficient of friction.

An efficient set of initial values for ϕ_0 and θ_0 for the optimization problem is the one that would be required in case of a direct impact; that is, if we ask for an angular velocity of $\boldsymbol{\omega_{B,0}} = \boldsymbol{0}$, which is equivalent to $\dot{\phi}_{B_0}(t_k^+) = \dot{\theta}_{B_0}(t_k^+) = \dot{\psi}_{B_0}(t_k^+) = 0$. In this case, the impulse is fully described by its normal component, resulting in an easy

calculation of these initial values, given by

$$\phi_0 = \arctan 2\left(v_{O_{By},0}, v_{O_{Bx}}\right), \tag{51}$$

$$\theta_0 = -\arctan 2\left(v_{O_{Bz},0}, \sqrt{v_{O_{Bx},0}^2 + v_{O_{By},0}^2}\right). \tag{52}$$

5.3. *Calculation of the approaching velocity*

Assuming that the bodies in contact at the moment of the collision are rigid, the mathematical theory of rigid body collisions can be used: when two rigid bodies collide, the relationship between the normal velocities of both *at the contact point* ($\boldsymbol{p}_c$) just before and after the impact can be expressed by[16,20,22]:

$$\left(\boldsymbol{\Delta v}_{c_B} - \boldsymbol{\Delta v}_{c_R}\right)^T \check{\boldsymbol{n}} = -(1+e)\left(\boldsymbol{v}_{c_B} - \boldsymbol{v}_{c_R}\right)^T \check{\boldsymbol{n}}. \tag{53}$$

where e is the coefficient of restitution (CoR) between the ball and the swing foot, and $\boldsymbol{v}_{c_B}$, $\boldsymbol{v}_{c_R}$ stand for the velocities of the point $\boldsymbol{p}_c$ on the ball and on the foot (the operational point), respectively. The increments of velocities $\boldsymbol{\Delta v}_{c_B}$ and $\boldsymbol{\Delta v}_{c_R}$ are calculated by means of the results shown in Eqs. (27) and (31), respectively, and the corresponding Jacobians, as

$$\boldsymbol{\Delta v}_{c_B} = \boldsymbol{J}_{B,c,v} \boldsymbol{\Delta \dot{q}}_B$$
$$= \boldsymbol{J}_{B,c,v} \boldsymbol{M}_B^{-1} \boldsymbol{J}_{B,c,v}^T \tilde{\boldsymbol{f}}_{c_B}, \tag{54}$$

$$\boldsymbol{\Delta v}_{c_R} = \bar{\boldsymbol{J}}_{R,c,v} \boldsymbol{\Delta \dot{\bar{q}}}_R$$
$$= \bar{\boldsymbol{J}}_{R,c,v} \bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_{s_R} \\ \tilde{\boldsymbol{\tau}}_{s_R} \end{bmatrix} - \bar{\boldsymbol{J}}_{R,c,v} \bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,c,v}^T \tilde{\boldsymbol{f}}_{c_B}. \tag{55}$$

But also, a similar relationship applies for the sole of the foot, given by

$$\begin{bmatrix} \boldsymbol{\Delta v}_{s_R} \\ \boldsymbol{\Delta \omega}_{s_R} \end{bmatrix} = \bar{\boldsymbol{J}}_{R,s} \boldsymbol{\Delta \dot{\bar{q}}}_R$$
$$= \bar{\boldsymbol{J}}_{R,s} \bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_{s_R} \\ \tilde{\boldsymbol{\tau}}_{s_R} \end{bmatrix} - \bar{\boldsymbol{J}}_{R,s} \bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,c,v}^T \tilde{\boldsymbol{f}}_{c_B}. \tag{56}$$

Let us assume that the relative motion between the sole of the foot and the ground produced by the impact is negligible, and that $\bar{\boldsymbol{J}}_{R,s}$ is not singular. Then, from Eq. (56) we approximate

$$\bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,s}^T \begin{bmatrix} \tilde{\boldsymbol{f}}_{s_R} \\ \tilde{\boldsymbol{\tau}}_{s_R} \end{bmatrix} \approx \bar{\boldsymbol{M}}_R^{-1} \bar{\boldsymbol{J}}_{R,c,v}^T \tilde{\boldsymbol{f}}_{c_B}, \tag{57}$$

and substitute into Eq. (55) to get

$$\boldsymbol{\Delta v}_{c_R} \approx 0. \tag{58}$$

Substituting these expressions into Eq. (53) and assuming that the ball is originally steady we compute the *normal* component of $\boldsymbol{v_{c_R}}$, $v_{c_{Rn}}$; that is, the desired normal approaching velocity of the operational point, as

$$
\begin{aligned}
v_{c_{Rn}} &= \boldsymbol{v_{c_R}^T} \check{\boldsymbol{n}} \\
&= \frac{\tilde{f}_{c_{Bn}}}{1+e} \check{\boldsymbol{n}}^T \mathcal{Y}_B \check{\boldsymbol{n}} + \frac{\tilde{f}_{c_{Bt}}}{1+e} \check{\boldsymbol{n}}^T \mathcal{Y}_B \check{\boldsymbol{t}},
\end{aligned}
\tag{59}
$$

where $\tilde{f}_{c_{Bn}}$ and $\tilde{f}_{c_{Bt}}$ are the *normal* and the *tangential* components of the impulse and $\mathcal{Y}_B$ is the *mobility matrix* of the ball, given by

$$
\mathcal{Y}_B = \boldsymbol{J_{B,c,v}} \boldsymbol{M_B^{-1}} \boldsymbol{J_{B,c,v}^T}.
\tag{60}
$$

In order to calculate the tangential component of $\boldsymbol{v_{c_R}}$, $v_{c_{Rt}}$, it is necessary to consider that the ball slides at the beginning of the impact, accelerating its angular motion until the ball starts rolling. If Eq. (36) holds,[26] the ball starts rolling without sliding before the impact ends. It means that the contact points on both the foot and the ball should be moving with the same velocity (no relative motion) at the end of the impact. As $\boldsymbol{\Delta v_{c_R}} \approx 0$, as seen in (58), $\boldsymbol{v_{c_R}}$ will be approximately the same at the end of the impact, then

$$
\begin{aligned}
v_{c_{Rt}} &= \boldsymbol{v_{c_R}^T} \check{\boldsymbol{t}} = \boldsymbol{\Delta v_{c_B}^T} \check{\boldsymbol{t}} \\
&= \tilde{f}_{c_{Bn}} \check{\boldsymbol{n}}^T \mathcal{Y}_B \check{\boldsymbol{t}} + \tilde{f}_{c_{Bt}} \check{\boldsymbol{t}}^T \mathcal{Y}_B \check{\boldsymbol{t}}.
\end{aligned}
\tag{61}
$$

In this way we can calculate the required *approaching velocity* of the operational point on the foot at the instant of the impact as

$$
\boldsymbol{v_{c_R}} = v_{c_{Rn}} \check{\boldsymbol{n}} + v_{c_{Rt}} \check{\boldsymbol{t}}.
\tag{62}
$$

But this velocity depends on

$$
\check{\boldsymbol{n}}^T \mathcal{Y}_B \check{\boldsymbol{n}} = m_{B_{nn}}^{-1}, \quad \check{\boldsymbol{t}}^T \mathcal{Y}_B \check{\boldsymbol{t}} = m_{B_{tt}}^{-1}, \quad \check{\boldsymbol{n}}^T \mathcal{Y}_B \check{\boldsymbol{t}} = -m_{B_{nt}}^{-1};
\tag{63}
$$

that is, the corresponding *inverse of the effective mass* perceived at the contact point in response to the application of the impulse along each axis, or interaction between axes.[27] These ones are constant values, calculated as

$$
m_{B_{nn}}^{-1} = m_B^{-1}, \quad m_{B_{tt}}^{-1} = m_B^{-1}\left(1+\rho^{-1}\right), \quad m_{B_{nt}}^{-1} = 0.
\tag{64}
$$

6. Humanoid Configuration During the Impact

This section describes our proposed approach for defining a configuration for the humanoid robot at the instant of the impact. Let us define three frames: $\{Sup\}$, $\{C\}$ and $\{Swg\}$, rigidly attached to the support foot, the waist and the swinging foot of the robot, respectively, as depicted in Fig. 6. By knowing the initial position of the ball, $\boldsymbol{p_{O_B,0}}$, and the impact coordinates, (ϕ_0, θ_0), we can calculate the contact point
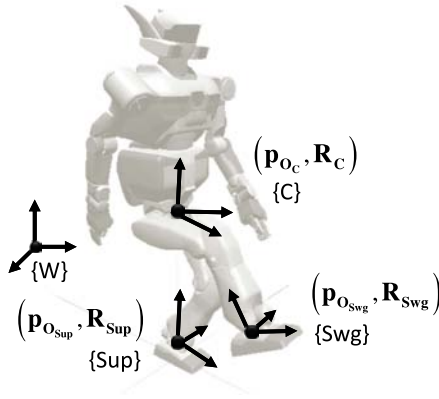
Fig. 6.   Frames of the robot used to describe its configuration.

described in the world frame, $\boldsymbol{p_c}$, as

$$\boldsymbol{p_c} = \boldsymbol{p_{O_B,0}} - r_B \boldsymbol{\check{n}}. \tag{65}$$

Then, by proposing for the humanoid robot: (i) the attitude of the support foot, $(\boldsymbol{p_{O_{Sup}}}, \boldsymbol{R_{Sup}})$, (ii) the attitude of the waist at the instant of the impact, $(\boldsymbol{p_{O_C}}(t_k), \boldsymbol{R_C}(t_k))$, (iii) the operational point on the swing foot described in $\{Swg\}$, $^{Swg}\boldsymbol{p_{c_R}}$ (Fig. 7), along with the orientation of this foot also at the instant of the impact, $\boldsymbol{R_{Swg}}(t_k)$, and (iv) the configuration of all the remaining kinematic chains (trunk, head and arms), we can calculate the robot's configuration $\boldsymbol{q_R}$ at t_k by solving its *inverse kinematics* problem.

The attitude of these links can be chosen arbitrarily, as long as the resulting configuration of the robot be feasible and dynamically stable. However, for the swing foot some remarks should be considered. Ideally, any point on the swing foot could be chosen as the operational point for the impact. But, as it is mentioned in Sec. 5.1, there is a close relationship between the chosen operational point and the orientation of the swing foot: the tangent to the surface of the foot at the operational point should coincide with the tangent to the surface of the ball at the contact point; that is, the normal vector to the surfaces of both objects at the contact point should have
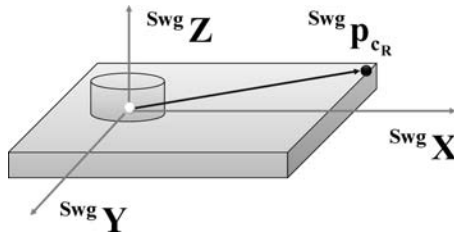

Fig. 7.   Description of the *operational point* in the swinging foot frame $\{Swg\}$.

the same orientation. Let us select a feasible operational point leading to a minimum change in orientation for the foot in order to meet the requirement stated above, by means of the procedure explained in the following.

Suppose that we have a 3D model of the swing foot represented as a polygonal mesh and described as a graph structure $\mathcal{M} = (\mathcal{T}, \mathcal{V})$. This one consists of a set of vertices, $\mathcal{V} = \{{}^{Swg}\boldsymbol{p}_{\boldsymbol{v}_1}, \ldots, {}^{Swg}\boldsymbol{p}_{\boldsymbol{v}_V}\}$, ${}^{Swg}\boldsymbol{p}_{\boldsymbol{v}_i} \in \mathbb{R}^3$, and a set of triangular faces connecting them, $\mathcal{T} = \{\boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_T\}$, $\boldsymbol{\Gamma}_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$, where every vertex is described in $\{Swg\}$.[28] As an example, the 3D model of the foot of the HRP-2 humanoid robot is depicted in Fig. 8. After heuristically selecting the triangles composing the mesh that are feasible for kicking the ball, $\mathcal{T}_{\text{kick}} \subset \mathcal{T}$, we calculate the centroid ${}^{Swg}\boldsymbol{c}_{\boldsymbol{\Gamma}_i}$ and the unitary normal vector ${}^{Swg}\check{\boldsymbol{n}}_{\boldsymbol{\Gamma}_i}$ of each triangle $\boldsymbol{\Gamma}_i \in \mathcal{T}_{\text{kick}}$, as shown in Fig. 9.

After representing each normal vector in the world reference frame, we compute the angle between each $\check{\boldsymbol{n}}_{\boldsymbol{\Gamma}_i}$ and the normal vector $\hat{\boldsymbol{n}}$ calculated in Eq. (37), in order to find the triangle $\boldsymbol{\Gamma}_{\text{sel}}$ whose orientation is the most similar to the tangent plane at the contact point on the surface of the ball; that is,

$$\boldsymbol{\Gamma}_{\text{sel}} = \arg \min_{\boldsymbol{\Gamma}_i \in \mathcal{T}_{\text{kick}}} \arccos\left(\check{\boldsymbol{n}}_{\boldsymbol{\Gamma}_i}^T \check{\boldsymbol{n}}\right). \tag{66}$$

The centroid of this triangle $\boldsymbol{\Gamma}_{\text{sel}}$, ${}^{Swg}\boldsymbol{c}_{\boldsymbol{\Gamma}_{\text{sel}}}$, is chosen as the operational point, ${}^{Swg}\boldsymbol{p}_{\boldsymbol{c}_R}$, whereas the orientation of the foot is calculated such that its normal vector, $\check{\boldsymbol{n}}_{\boldsymbol{\Gamma}_{\text{sel}}}$, coincide with the direction of $\check{\boldsymbol{n}}$. For that purpose, let us describe the orientation of the swing foot by using a set of orientation coordinates (Euler angles) following the Euler ZYX convention, $(\phi_{Swg}, \theta_{Swg}, \psi_{Swg})$, upon which $\boldsymbol{R_{Swg}}$ is built.

During the impact, at $t = t_k$, $\boldsymbol{R_{Swg,k}}(\phi_{Swg,k}, \theta_{Swg,k}, \psi_{Swg,k})$ is calculated as:

$$\phi_{Swg,i} = \arctan 2(\check{n}_y \check{n}_x) - \arctan 2(\check{n}_{\Gamma_{\text{sel}} y} \check{n}_{\Gamma_{\text{sel}} x}), \tag{67}$$

$$\theta_{Swg,i} = \arccos(\check{n}_z) - \arccos(\check{n}_{\Gamma_{\text{sel}} z}), \tag{68}$$

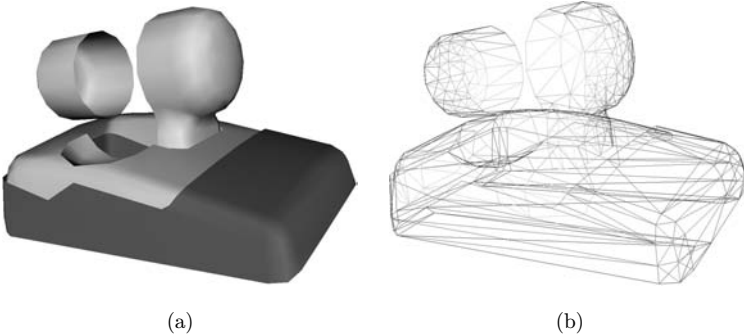$$\psi_{Swg,i} = 0. \tag{69}$$

(a)   (b)

Fig. 8. The 3D model of the foot of the HRP-2 represented as: (a) a solid object and (b) a wireframe.
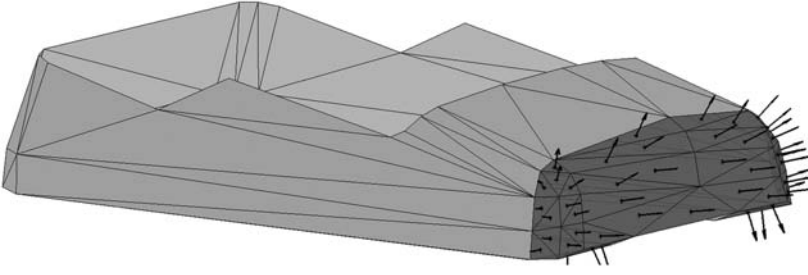
Fig. 9. The 3D model of the foot showing the triangles that were heuristically selected (the dark ones), as well as the unitary normal vector of each one placed at the corresponding triangle's centroid.

7. Kicking Motion Planning

The function shown in Algorithm 1 implements all the processes described until now by specifying as input parameters the initial conditions of the robot and the ball, as well as the goal specifications, and returning the required impact conditions: the impact point, p_c, the selected operational point, $^{Swg}p_{c_R}$, the orientation of the foot at the instant of the impact, $R_{Swg,k}$, and the approaching velocity, v_{c_R}.

Then, once this velocity is known, it is possible to propose a suitable 3D trajectory satisfying some specific position/velocity values at the instant of the impact. This can be accomplished by shaping the trajectory with *cubic B-spline curves*, as these ones have superb properties: (i) the curve is contained entirely inside of the convex hull of its control polyline and (ii) the curve can be locally controlled.[29]

7.1. *Definition of the cubic B-spline*

A cubic B-spline curve made up of n segments, $s_i(u)$, and delimited by $n+1$ knots, p_i, is governed by $n+3$ control points, q_i.[30] This curve is defined as a weighted sum

---

Algorithm 1 Function for computing the required impact conditions
---
function IMPACT_CONDITIONS(initial conditions and goal)

$\quad v_{O_B,0} \Leftarrow$ BALL_TRAJECTORY($p_{O_B,0}, p_G$) $\qquad\qquad\qquad$ ▷ Projectile motion

$\quad (\phi_0, \theta_0, \tilde{f}_{c_B}) \Leftarrow$ APPLIED_IMPULSE($v_{O_B,0}, \omega_{B,0}$) $\qquad$ ▷ Impact coordinates and
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ impulse

$\quad p_c \Leftarrow$ CONTACT_POINT($p_{O_B,0}, \phi_0, \theta_0$)

$\quad \left(^{Swg}p_{c_R}, R_{Swg,k}\right) \Leftarrow$ SWG_FOOT_CONFIG($\phi_0, \theta_0, \tilde{f}_{c_B}$) $\qquad$ ▷ Select operational
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ point

$\quad v_{c_R} \Leftarrow$ APPROACHING_VELOCITY($e, m_B, \rho_B, \phi_0, \theta_0, \tilde{f}_p$)

$\quad$ **return** $p_c, {}^{Swg}p_{c_R}, R_{Swg,k}, v_{c_R}$

end function
---

of its control points, such that

$$s_i(u) = \sum_{j=-2}^{1} b_j(u) q_{i+j}, \quad \forall i = 1, \ldots, n,$$

$$b = [b_{-2} \quad b_{-1} \quad b_0 \quad b_1] = \frac{1}{6}[u^3 \quad u^2 \quad u \quad 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \tag{70}$$

Also, $u = [0, 1]$ in order to achieve C^1 continuity; that is, $s_i(1) = s_{i+1}(0)$ and $s'_i(1) = s'_{i+1}(0)$. This parameter u can be scaled in time, by doing $u = t/\tau$, where $t = [0, \tau]$ and τ is the time required to travel *one segment*. Then, traveling a curve made up of n segments requires a total time $T_s = n\tau$. The extreme points of each segment (the knots) are calculated by

$$p_i = \frac{1}{6}(q_{i-1} + 4q_i + q_{i+1}), \quad \forall i = 0, \ldots, n, \tag{71}$$

while the velocities at these extreme points are given by

$$v_i = \frac{1}{2\tau}(q_{i+1} - q_{i-1}), \quad \forall i = 0, \ldots, n. \tag{72}$$

This cubic B-spline is an approximating curve. Its shape is determined by its control points, q_i, but the curve itself does not pass through those. Instead, it passes through the knots, p_i.[30] By using Eqs. (71) and (72) it is possible to interpolate a set of $n + 1$ data points, $p_0, \ldots, p_n$, with a curve made up of n segments, which starts and ends with given velocities v_0 and v_n.[30] Rearranging Eqs. (71) and (72) into a matrix form we get

$$A[q_{-1} \quad q_0 \quad \cdots \quad q_n \quad q_{n+1}]^T = [v_0 \quad p_0 \quad \cdots \quad p_n \quad v_n]^T, \tag{73}$$

where $A \in \mathbb{R}^{(n+3)\times(n+3)}$ is given by

$$A = \frac{1}{6}\begin{bmatrix} -3/\tau & 0 & 3/\tau & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & -3/\tau & 0 & 3/\tau \end{bmatrix}, \tag{74}$$

and τ is calculated by dividing the total time T_s by the number of segments in equal parts; that is, $\tau = T_s/n$. This coefficient matrix is non-singular and therefore Eq. (73) has a unique solution.

7.2. *Trajectory planning*

By using cubic B-splines we can now construct a trajectory for the swing foot, in order to take the operational point defined on it from an initial position to a final one while passing by the desired contact point on the ball with the required approaching velocity. Although there are many ways to achieve this motion, generally it will be divided into two phases,[31] each one described with a different cubic B-spline: (i) the *swing phase* which detaches the foot from the ground (with an initial zero velocity) and drives the operational point from its initial position, $p_{c_R,0}$, to the contact point, p_c, with the required velocity, v_{c_R}, and (ii) the *follow-through phase* which drives the operational point on the foot moving with a velocity v_{c_R} from the contact point, p_c, to the final landing position (with zero velocity).

The landing position can be chosen arbitrarily as long as it is inside of the robot's workspace. We arbitrarily chose it to be the same as the initial one. Then, for each cubic B-spline we know already the extreme points of every segment and the velocities that should be attained at these ones. In addition, to have a better control of the curve we propose to add an auxiliary knot to each phase, p_{aux}, manually tuned to avoid hitting the floor, the support leg or the ball (before the desired instant of impact). See Fig. 10.

It is worth to mention that sometimes the required trajectory may not be contained inside of the corresponding workspace, unless we shorten the time span T_s of each phase. However, this action may lead to trajectories that are not feasible on a real robot. In general, a shorter time span will lead to shorter curves, so that they can fit in the workspace. However, it implies that the foot will reach the desired velocity in a shorter time; that is, they will require higher acceleration values and consequently, a higher joint torque, which may be physically impossible to achieve by using the actuators of a real robot.

8. Stabilization

Once the trajectory for the foot is decided and the leg motion is produced, the robot should maintain its stability while this motion is being performed. This can be done by moving the waist of the robot horizontally (or its upper limbs) such that the *zero moment point* (ZMP) remain inside of the polygon of support.[32] In this work, we used
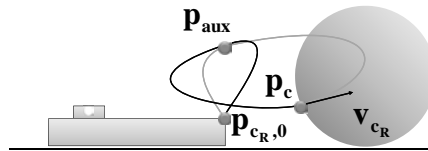

Fig. 10.   Proposed trajectory for the operational point as seen on the sagittal plane. The darker trajectory corresponds to the *swing phase*, whereas the lighter one to the *follow-through phase.*

the method proposed by Nishiwaki *et al.* to generate a waist trajectory capable of stabilizing the robot's motion.[33–35]

Let the motion of the humanoid robot be characterized by the trajectory of its COM, $\boldsymbol{p_{cm}}$, and its linear and angular *momenta* about the origin of the world reference frame $(\boldsymbol{O_W})$, $\mathcal{P_R}$ and $\mathcal{L}_{\boldsymbol{R},\boldsymbol{O_W}}$, respectively. The ZMP for this motion, $\boldsymbol{p_{zm}} = [p_{zm_x} \quad p_{zm_y} \quad p_{zm_z}]^T$, considering $p_{zm_z} = 0$, is calculated as

$$p_{zm_x} = \frac{\tilde{p}_{cm_x}\tilde{m}_R g - \dot{\mathcal{L}}_{R_y,O_W}}{\tilde{m}_R g + \dot{P}_z}, \quad p_{zm_y} = \frac{\tilde{p}_{cm_y}\tilde{m}_R g + \dot{\mathcal{L}}_{R_x,O_W}}{\tilde{m}_R g + \dot{P}_z}, \tag{75}$$

where m_R is the total mass of the robot and g is the acceleration due to gravity. However, in order to use the method of Nishiwaki *et al.*, it is first necessary to assume a simplified model of the robot, regarding it as a single point mass located at $\bar{\boldsymbol{p}}_{\boldsymbol{cm}}$.[33] Here, the upper "bar" is used to distinguish the corresponding variables from the ones of the multi-body model. This method assumes that the vertical motion of this point is known, i.e., $\bar{p}_{cm_z}$, $\dot{\bar{p}}_{cm_z}$ and $\ddot{\bar{p}}_{cm_z}$ are known. Then, it is just necessary to calculate the horizontal motion, $\bar{p}_{cm_x}$ and $\bar{p}_{cm_y}$, capable to attain a desired ZMP.

The corresponding linear and angular *momenta* for this model are given by

$$\bar{\mathcal{P}}_{\boldsymbol{R}} = \tilde{m}_R \dot{\bar{\boldsymbol{p}}}_{\boldsymbol{cm}}, \tag{76}$$

$$\bar{\mathcal{L}}_{\boldsymbol{R},\boldsymbol{O_W}} = \bar{\boldsymbol{p}}_{\boldsymbol{cm}} \times \tilde{m}_R \dot{\bar{\boldsymbol{p}}}_{\boldsymbol{cm}}. \tag{77}$$

These ones can be substituted in Eq. (75) to get

$$\bar{p}_{zm_x} = \bar{p}_{cm_x} - \frac{\bar{p}_{cm_z}\ddot{\bar{p}}_{cm_x}}{\ddot{\bar{p}}_{cm_z} + g}, \tag{78}$$

where we have considered only the x-component (a similar procedure applies for the y-component). Let us discretize the horizontal acceleration of the single point mass by considering a sampling time Δt:

$$\ddot{\bar{p}}_{cm_x,i} = \frac{\bar{p}_{cm_x,i-1} - 2\bar{p}_{cm_x,i} + \bar{p}_{cm_x,i+1}}{\Delta t^2}, \tag{79}$$

where $\bar{p}_{cm_x,i} := \bar{p}_{cm_x}(i\Delta t)$ and $i = 1, \ldots, n$ with n being the total number of samples of the trajectory. Also, it is assumed that $\bar{p}_{cm_x,-1} = \bar{p}_{cm_x,0}$ and $\bar{p}_{cm_x,n+1} = \bar{p}_{cm_x,n}$. Using this approximation, the discretized ZMP equation is expressed as

$$\bar{p}_{zm_x,i} = a_i\bar{p}_{cm_x,i-1} + b_i\bar{p}_{cm_x,i} + c_i\bar{p}_{cm_x,i+1}, \tag{80}$$

where the coefficients a_i, b_i and c_i are calculated as:

$$a_i = c_i = -\bar{p}_{cm_z,i}/[(\ddot{\bar{p}}_{cm_z,i} + g)\Delta t^2], \tag{81}$$

$$b_i = 2\bar{p}_{cm_z,i}/[(\ddot{\bar{p}}_{cm_z,i} + g)\Delta t^2] + 1. \tag{82}$$

Let us notice that the same coefficients a_i, b_i and c_i apply for $\bar{p}_{cm_x,i}$ or $\bar{p}_{cm_y,i}$.

Now, let us define $\bar{\boldsymbol{p}}'_{\boldsymbol{cm,i}} = [\,\bar{p}_{cm_x,i} \quad \bar{p}_{cm_y,i}\,]^T$ and $\bar{\boldsymbol{p}}'_{\boldsymbol{zm,i}} = [\,\bar{p}_{zm_x,i} \quad \bar{p}_{zm_y,i}\,]^T$. Then, it is possible to rearrange Eq. (80) in matrix form to get:

$$\begin{bmatrix} \bar{\boldsymbol{p}}'_{\boldsymbol{zm,1}} & \cdots & \bar{\boldsymbol{p}}'_{\boldsymbol{zm,n}} \end{bmatrix}^T = \boldsymbol{A} \begin{bmatrix} \bar{\boldsymbol{p}}'_{\boldsymbol{cm,n}} & \cdots & \bar{\boldsymbol{p}}'_{\boldsymbol{cm,n}} \end{bmatrix}^T, \tag{83}$$

$$\bar{\boldsymbol{p}}'_{\boldsymbol{zm,all}} = \boldsymbol{A}(\bar{\boldsymbol{p}}'_{\boldsymbol{cm,all}}), \tag{84}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} a_1 + b_1 & c_1 & \cdots & 0 & 0 \\ a_2 & b_2 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & a_n & b_n + c_n \end{bmatrix}. \tag{85}$$

This coefficient matrix is non-singular and therefore Eq. (84) has a unique solution. $\boldsymbol{A}$ is a huge square matrix consisting of several thousands of rows and columns, given that n is the total number of samples taken for a certain motion. Still, it is a *tridiagonal matrix* whose elements are all zeros except of its main diagonal and the adjacent diagonals. For such a matrix, there is an efficient algorithm to compute its inverse, reported in Ref. 36. The result, however, is just an approximation.

Let us suppose that we have defined an arbitrarily preliminary trajectory for the waist of the robot that ensures static stability but not dynamic one. Then, the trajectory for the ZMP corresponding to the actual motion of the multi-body model, $\boldsymbol{p}_{zm}$, can be calculated by using Eq. (75), discretized by considering a sampling time Δt, such that all the samples be arranged in matrix form as

$$\boldsymbol{p}_{\boldsymbol{zm,all}} = \begin{bmatrix} \boldsymbol{p}_{\boldsymbol{zm,1}} & \cdots & \boldsymbol{p}_{\boldsymbol{zm,n}} \end{bmatrix}^T. \tag{86}$$

Let us arbitrarily specify a desired ZMP trajectory, discretized in the same way, $\boldsymbol{p}^d_{\boldsymbol{zm,all}}$, which ensures the dynamic stability of the multi-body model. The error between the desired ZMP trajectory and the one for the actual motion is given by

$$\boldsymbol{e}_{\boldsymbol{zmp,all}} = \boldsymbol{p}^d_{\boldsymbol{zm,all}} - \boldsymbol{p}_{\boldsymbol{zm,all}}. \tag{87}$$

By using this error in place of the approximated ZMP trajectory for the single point mass shown in Eq. (84), we can calculate the horizontal variation of the CoM capable of compensating the ZMP error as

$$\Delta \boldsymbol{p}_{\boldsymbol{cm,all}} = \boldsymbol{A}^{-1} \boldsymbol{e}_{\boldsymbol{zmp,all}}, \tag{88}$$

where $\boldsymbol{A}$ is calculated by using the actual values of $p_{cm_z,i}$ and $\ddot{p}_{cm_z,i}$ of the multi-body model. Then, the trajectory for the CoM is updated as

$$\boldsymbol{p}^{new}_{\boldsymbol{cm,all}} := \boldsymbol{p}^{old}_{\boldsymbol{cm,all}} + \Delta \boldsymbol{p}_{\boldsymbol{cm,all}}. \tag{89}$$

However, the trajectory for this point cannot be directly manipulated as it is the result of the configuration of the robot evolving in time. Still, it is possible to

---

Algorithm 2 Function for computing a stable waist trajectory

---

function BALANCER($\boldsymbol{p}_{O_C}^{old}[i], \boldsymbol{R}_C[i], \boldsymbol{q}_R[i], \boldsymbol{p}_{zm}^d[i]$)

    $\boldsymbol{p}_{O_C}^{new}[i] \leftarrow \boldsymbol{p}_{O_C}^{old}[i]$

    for $j \leftarrow 1, 2$ **do**

        $(\boldsymbol{p}_{zm}[i], \boldsymbol{p}_{cm}[i], \ddot{\boldsymbol{p}}_{cm}[i]) \Leftarrow$ CALC_DYN($\boldsymbol{p}_{O_C}^{new}[i], \boldsymbol{R}_C[i], \boldsymbol{q}_R[i]$)

                                                 ▷ Actual COM and ZMP

        $\boldsymbol{e}_{zmp}[i] \leftarrow \boldsymbol{p}_{zm}^d[i] - \boldsymbol{p}_{zm}[i]$ ▷ ZMP error

        $(a[i], b[i], c[i]) \Leftarrow$ CALC_COEFFICIENTS($p_{cm_z}[i], \ddot{p}_{cm_z}[i], \Delta t$)

                                         ▷ Coefficients' calculation

        $\Delta \boldsymbol{p}_{O_C}[i] \Leftarrow$ TRIDIAG($a[i], b[i], c[i], \boldsymbol{e}_{zmp}[i]$)   ▷ Calculation of $\boldsymbol{A}^{-1}\boldsymbol{e}_{zmp,all}$

        $\boldsymbol{p}_{O_C}^{new}[i] \leftarrow \boldsymbol{p}_{O_C}^{new}[i] + \Delta \boldsymbol{p}_{O_C}[i]$ ▷ Update values

    end for

    return $\boldsymbol{p}_{O_C}^{new}[i]$

end function

---

approximate the horizontal variation of the CoM to the horizontal variation of the waist position, $\Delta \boldsymbol{p}_{O_C,all}^{new} \approx \Delta \boldsymbol{p}_{cm,all}^{new}$, and repeat this process iteratively until the ZMP error becomes small enough (typically only two iterations are necessary to produce a good approximation); that is,

$$\boldsymbol{p}_{O_C,all}^{new} := \boldsymbol{p}_{O_C,all}^{old} + \boldsymbol{A}^{-1}\boldsymbol{e}_{zmp,all}. \tag{90}$$

The function shown in Algorithm 2 implements this process by specifying as input parameters the discretized trajectories for: the preliminary waist attitude ($\boldsymbol{p}_{O_C}^{old}[i], \boldsymbol{R}_C[i]$), all the joint angles of the robot $\boldsymbol{q}_R[i]$ and the desired ZMP $\boldsymbol{p}_{zm}^d[i]$, and returning an updated waist position trajectory, $\boldsymbol{p}_{O_C}^{new}[i]$, that attains the desired one for the ZMP with enough precision. The complete algorithm for kicking the ball is shown in Algorithm 3.

9. Simulation and Experimental Results

In order to assess the validity of our approach we simulated a set of kicking motions performed by the HRP-2 humanoid robot on a regular volleyball. The motion of the ball after the impact was analyzed in every case, and compared to the desired one, in order to statistically evaluate the performance of the proposed algorithm. These analyses are reported in Sec. 9.1.

Having done this, one kicking motion is taken into consideration to be tested on the real robot, and also simulated for effects of comparison and detailed analysis. This one is reported in Sec. 9.2.

Algorithm 3 Impulsive pedipulation algorithm

Require: Ball information, robot information and CoR(e)
Require: Initial ball position $\boldsymbol{p_{O_B}}[0]$ and initial robot configuration $\boldsymbol{q_R}[0]$
Require: Goal position of the ball $\boldsymbol{p_G}$, reaching angle $\theta_{m,f}$ and angular velocity
$\qquad \boldsymbol{\omega_{B,0}} = \boldsymbol{\omega_{B,f}}$
Require: Kicking time t_k and sampling Δt
  Generate temporal waist position trajectory $\boldsymbol{p_{O_C}^{old}}[i]$
  Generate desired ZMP $\boldsymbol{p_{zm}^d}[i]$
  $(\boldsymbol{p_c}, \boldsymbol{v_{c_R}}, {}^{\boldsymbol{Swg}}\boldsymbol{p_{c_R}}, \boldsymbol{R_{Swg,k}}) \Leftarrow \text{KICK\_CONDITIONS}(\text{initial ones and goal})$
      ▷ Contact point, approaching velocity, operational point and foot orientation
                                                                    ▷ See Algorithm 1
  $(\boldsymbol{p_{c_R}}[i], \boldsymbol{R_{Swg}}[i]) \Leftarrow \text{FOOT\_ATTITUDE\_TRAJECTORY}(\boldsymbol{p_c}, \boldsymbol{v_{c_R}}, {}^{\boldsymbol{Swg}}\boldsymbol{p_{c_R}}, \boldsymbol{R_{Swg,k}})$
  $\boldsymbol{q_R}[i] \Leftarrow \text{INVERSE\_KINEMATICS}(\boldsymbol{p_{O_C}^{old}}[i], \boldsymbol{R_C}[i], \boldsymbol{p_{c_R}}[i], \boldsymbol{R_{Swg}}[i])$
  $\boldsymbol{p_{O_C}^{new}}[i] \Leftarrow \text{BALANCER}(\boldsymbol{p_{O_C}^{old}}[i], \boldsymbol{R_C}[i], \boldsymbol{q_R}[i], \boldsymbol{p_{zm}^d}[i])$ ▷ Stabilizer
                                                                    ▷ See Algorithm 2

9.1. *Set of simulation experiments*

The humanoid robot HRP-2 is 1.539 m height and weighs 58 kg (Fig. 11(a)).[37] It has 30 dof: six in each leg, six in each arm, one in each hand, two at the waist and two at the neck, arranged as shown in Fig. 11(b).

   In order to simulate the kicking motion we used the parameters for the volleyball reported in Table 1. These ones are based on a real ball available in the laboratory. The radius (r_B) and the mass (m_B) were directly measured, while the construction coefficient (ρ_B) was estimated, supposing that it may be similar to a tennis ball, given that both are formed by layers.[23] The coefficient of friction (μ) between the foot and the ball is arbitrarily considered to be small, given the slippery interaction between the corresponding surfaces. Finally, the CoR (e) was directly measured by using the bouncing height of the ball over the frontal surface of the foot, maintained as horizontal as possible. The ball is initially located over the origin of the world reference frame, such that for its initial position we have $\boldsymbol{p_{O_B,0}} = \begin{bmatrix} 0 & 0 & r_B \end{bmatrix}^T$.

   Let us place the humanoid robot in such a way that while standing in its reference configuration its cardinal sagittal plane be parallel to the *XZ*-plane of the world reference frame, and its swing foot before detaching be -0.35 m behind the ball. This distance was heuristically selected given that it allowed more kicking trajectories to be feasible. Also, the height of the waist was heuristically selected ($p_{O_{C_z}} = 0.6$ m), a little bit lower than in its reference configuration to increase the workspace of the swing leg but not so low to demand so much torque from the knee. These parameters, reported in Table 2, lead to the setup shown in Fig. 12.

   Then, a set of motions was designed by specifying seven parameters, six of them related to the desired 3D goal and angular velocity of the ball and one related to the
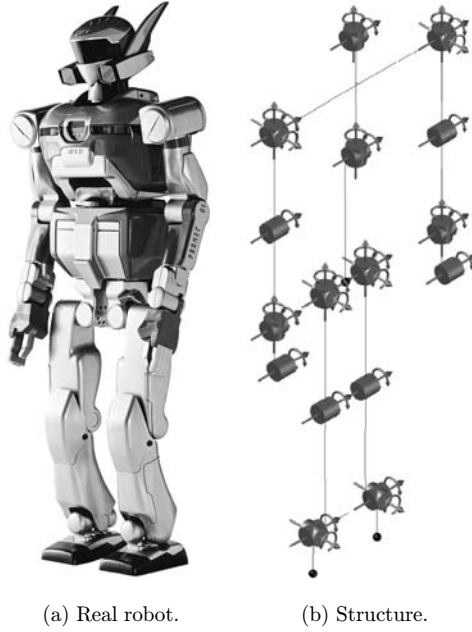
(a) Real robot.   (b) Structure.

Fig. 11.   HRP-2 humanoid robot.

Table 1.   Volleyball parameters.

Radius(r)	$0.099\,\mathrm{m}$
Mass(m_B)	$0.25\,\mathrm{kg}$
Construction coefficient(ρ)	0.535
CoR(e)	0.84
Coefficient of friction(μ)	0.07

Table 2.   Initial robot parameters.

Support ankle position ($p_{O_{Sup}}$)	$[\,-0.35,\,-0.19,\,0.105\,]^{T}\,\mathrm{m}$
Initial waist position ($p_{O_C,0}$)	$[\,-0.35,\,-0.095,\,0.6\,]^{T}\,\mathrm{m}$
Initial swing ankle position ($p_{O_{Swg},0}$)	$[\,-0.35,\,0.0,\,0.105\,]^{T}\,\mathrm{m}$

kicking trajectory: the time span for performing the swing phase of the kicking motion, T_{swing}. These parameters are listed in Table 3 together with the tested values. Each motion was defined by varying only one parameter. This means that for the test bench shown 840 different motions can be generated. However, not all of them are feasible for five principal reasons: (i) the swing foot leaves the corresponding workspace in at least one point of the trajectory, (ii) at some point of the produced kicking trajectory the sole of the foot (the rear end or the front end) collides with the floor, (iii) the swing foot collides against the support leg (self-collision), (iv) the
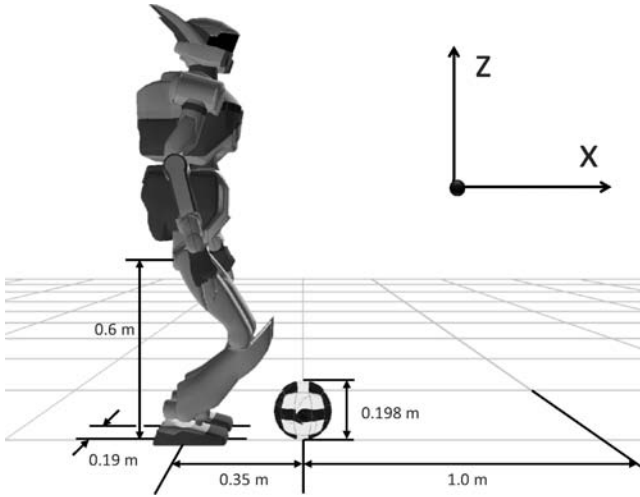
Fig. 12.   Setup for the set of kicking motions.

Table 3.   Test bench parameters.

Swing phase time span (T^d_{swing}) [s]	0.7, 1.0
Ball's yaw velocity ($\dot{\phi}^d_{B0}$) [°/s]	0
Ball's pitch velocity ($\dot{\theta}^d_{B0}$) [°/s]	0
Reaching angle ($\theta^d_{m,f}$) [°]	$-15, -10, -5, 0$
Goal's frontal distance ($p^d_{G_x}$) [m]	0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0
Goal's lateral distance ($p^d_{G_y}$) [m]	$-0.2, 0.0, 0.2$
Goal's height ($p^d_{G_z}$) [m]	0.1, 0.2, 0.3, 0.4, 0.5

produced joint trajectories are not completely inside of the joint limits, or (v) the joint velocities surpass the corresponding maximum values. These limitations severely reduce the feasible set of motions. Still, in order to evaluate the algorithm and take advantage that in simulation we can ignore the last three cases, we will drop the motions not complying with the first two: when the inverse kinematics fail or when the swing sole fails to clear the floor by at least 0.01 m. In this way, 149 feasible motions were produced by computing in each case the joint trajectories of the robot and a desired ZMP reference trajectory.

Each motion was simulated by using `OpenHRP3`,[38] an integrated simulation platform capable of performing realistic dynamic simulations and that was previously improved to include the capacity of simulating the ball dynamics.[39] The virtual model of HRP-2 implements the same controller built in the real robot, as well as the dynamical effect of its impact absorption mechanism consisting of rubber bushes and dampers. In this way, it behaves closely to the physical platform with the advantage of having control of the physical properties of the environment.

`OpenHRP3` is able to release as output data the sequence of positions and orientations for any object of the virtual world (the ball or any link), as well as the joint values for the robot and the sequence of readings of each sensor. The sequence of positions of the ball for each kicking motion was analyzed in `Matlab`, from which the attained goal position, p_G^s, could be calculated. This one was taken at the moment in which the centroid of the ball crossed the goal plane. Here, the superscript s stands for each "simulated" value, whereas d will be used for the "desired" one. The remaining trajectory parameters were calculated by means of a curve fitting process, by assuming a fixed quadratic coefficient $(-\frac{1}{2}g)$ for the Z-component. In this way, it was possible to calculate a very accurate value for the initial linear velocity $v_{O_B,0}^s$, described by its components $v_{O_B,0}^s$, ϕ_m^s and $\theta_{m,0}^s$, as well as the reaching angle $\theta_{m,f}$, by using Eq. (6).

The attained goal position was compared with the desired one by computing the goal position error, considered as $e_{p_G} = \|p_G^s - p_G^d\|$. Then, every error (in meters or in terms of the *radius* of the ball, r_B) was assigned a grayscale code according to Fig. 13 and plotted at the desired goal position, p_G^d. The interpolated results are shown in Figs. 14 and 15. These figures show the goal position errors grouped by blocks with respect to the desired reaching angle, $\theta_{mf}^d = \{-15, -10, -5, 0\}°$, and in columns with respect to the time span for the swing phase, $T_{\text{swing}} = \{0.7, 1.0\}$. Then, each block shows the results plotted in different grayscale maps, corresponding to each desired height, p_{Gz}^d. Each grayscale map represents the accuracy attained at each XY-coordinate, represented by p_{Gx}^d and p_{Gy}^d. Notice that only the heights that were feasible for producing the kicking motion are shown. For example, by considering $\theta_{mf}^d = -15°$ only the grayscale maps for $p_{G_z}^d = 0.1$ are shown, as with that reaching angle attaining other heights implied a collision of the kicking foot against the floor. On the other hand, by considering $\theta_{mf}^d = 0°$ no grayscale map for $p_{G_z}^d = 0.1$ is shown. This is due to the fact that if the center of the ball is at that height, its lower part is almost touching the floor. Then, the required approaching velocity would be so big that the foot would have to travel a large trajectory to reach an almost infinite velocity to hit the ball horizontally.

The goal position error, analyzed in terms of the radius of the ball r_B, gives a practical insight of the accuracy of the results. Let us calculate the percentage of the experiments that were successful, where the success depends on hitting a target whose size is less than $0.5r_B$, r_B, $1.5r_B$ or $2r_B$. The results are shown in Table 4, grouped with respect to the time span T_{swing}. The accuracy of the reaching angle can be analyzed by calculating the corresponding mean values $(\mu_{\theta_{m,f}^s})$ and standard deviations $(\sigma_{\theta_{m,f}^s})$ obtained in simulation for each desired value of the angle and also grouped with respect to T_{swing}. The results are shown in Table 5.
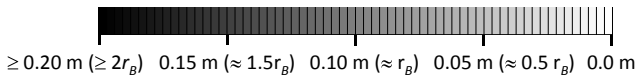

$\geq 0.20\ \text{m}\ (\geq 2r_B)\quad 0.15\ \text{m}\ (\approx 1.5r_B)\quad 0.10\ \text{m}\ (\approx r_B)\quad 0.05\ \text{m}\ (\approx 0.5\ r_B)\quad 0.0\ \text{m}$

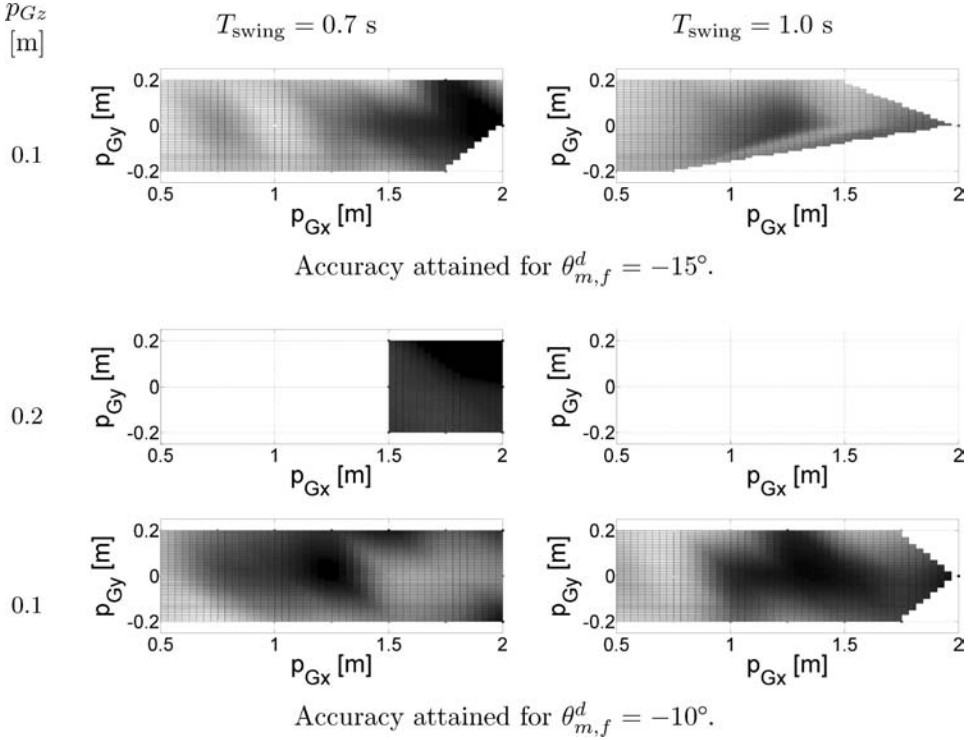Fig. 13. Grayscale code for representing the accuracy attained for the set of experiments.

Fig. 14.   Accuracy attained at each $\boldsymbol{p_G}$ for $\theta^d_{m,f} = \{-15, -10\}^\circ$ and $T_{\text{swing}} = \{0.7, 1.0\}$ s.

As can be seen from Table 4, the larger the time span for the swing phase is, the larger the probability of success becomes. However, even with a short time span, the percentage of complete failure is less than 10%. On the other hand, by taking a look to Table 5 it seems that there is not any obvious statistical influence of T_{swing} on the achieved reaching angle.

Every goal position, together with a reaching angle, requires a different initial linear velocity vector $\boldsymbol{v_{O_B,0}}$, making it not feasible to evaluate for each desired value as in the previous cases. Instead, let us evaluate the errors regarding to its *magnitude* $(e_{v_{O_B,0}} = v^s_{O_B,0} - v^d_{O_B,0})$, to its *azimuthal angle* $(e_{\phi_m} = \phi^s_m - \phi^d_m)$ and to its *launch angle* $(e_{\theta_{m,0}} = \theta^s_{m,0} - \theta^d_{m,0})$ by means of their corresponding statistical values, grouped by the time span for the swing phase T_{swing} in order to evaluate its influence on the results. These ones are shown in Table 6 and, as can be seen, it seems that the larger T_{swing} is, the more the standard deviation of each error decreases; that is, the kicking motion is more precise.

9.2. *Single experiment analysis*

Let us slightly modify the experimental setup as shown in Fig. 16, by adding a disk between the ball and the floor, as well as a goal frame (detailed in Fig. 17).
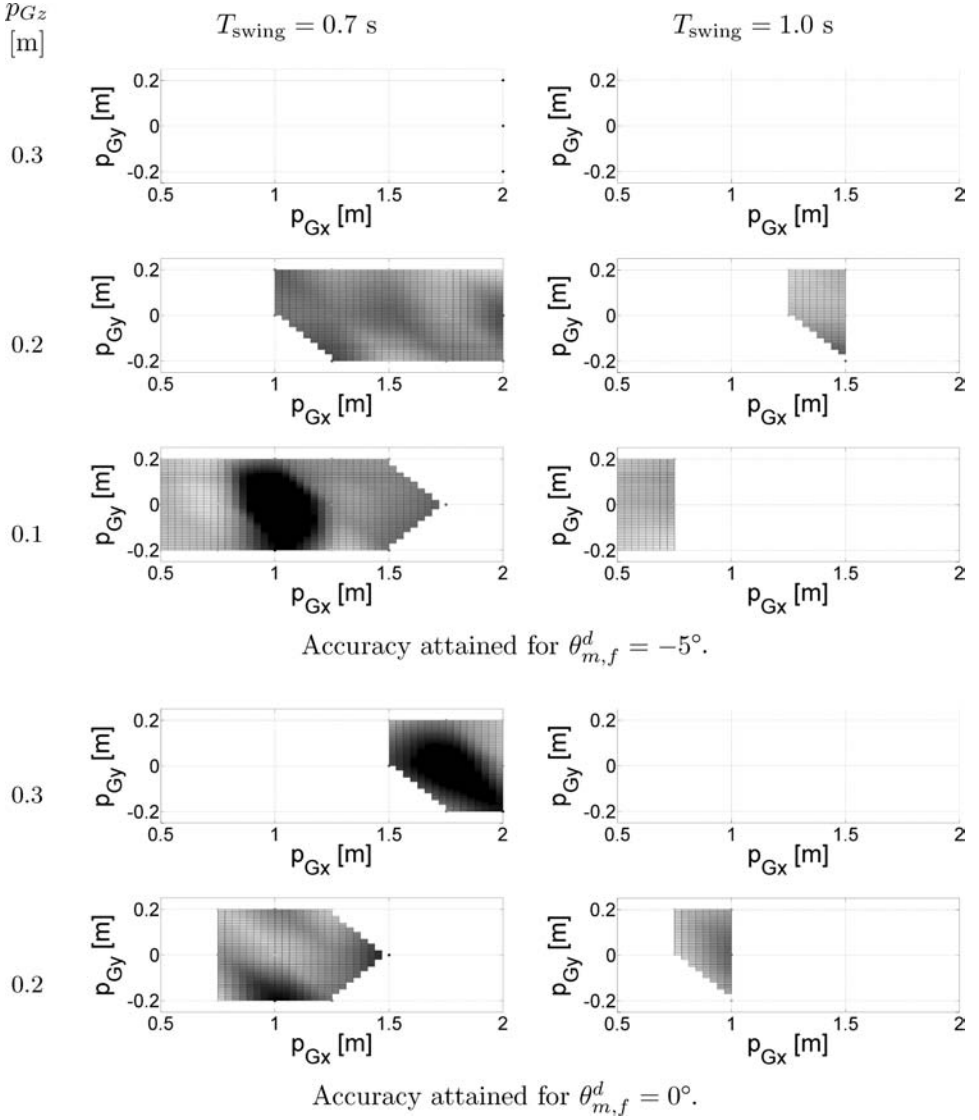
Fig. 15.   Accuracy attained at each $\boldsymbol{p_G}$ for $\theta_{m,f}^d = \{-5, 0\}^{\circ}$ and $T_{\text{swing}} = \{0.7, 1.0\}$ s.

Table 4.   Percentages of the N experiments for those that hit the target $\boldsymbol{p_G^d}$ within each tolerance.

Tolerance of error	$T_{\text{swing}} = 0.7$ s, $N = 100$	$T_{\text{swing}} = 1.0$ s, $N = 49$
≤ 0.05 m ($\leq 0.5r$)	30.0%	42.9%
≤ 0.10 m ($\leq r$)	58.0%	69.4%
≤ 0.15 m ($\leq 1.5r$)	75.0%	89.8%
≤ 0.20 m ($\leq 2r$)	92.0%	100.0%

Table 5. Statistical values for the reaching angles $\theta^s_{m,f}$ obtained in simulation.

Desired reaching angle	$T_{\text{swing}} = 0.7\,\text{s}\ (N = 100)$		$T_{\text{swing}} = 1.0\,\text{s}\ (N = 49)$	
	$\mu_{\theta^s_{m,f}}$	$\sigma_{\theta^s_{m,f}}$	$\mu_{\theta^s_{m,f}}$	$\sigma_{\theta^s_{m,f}}$
$-15°$	$-20.3°$	$4.2°$	$-22.7°$	$4.7°$
$-10°$	$-14.4°$	$4.1°$	$-15.2°$	$3.9°$
$-5°$	$-6.6°$	$3.9°$	$-7.3°$	$3.2°$
$0°$	$-0.2°$	$3.8°$	$-1.7°$	$2.9°$

Table 6. Statistical values for the errors in velocities obtained in simulation.

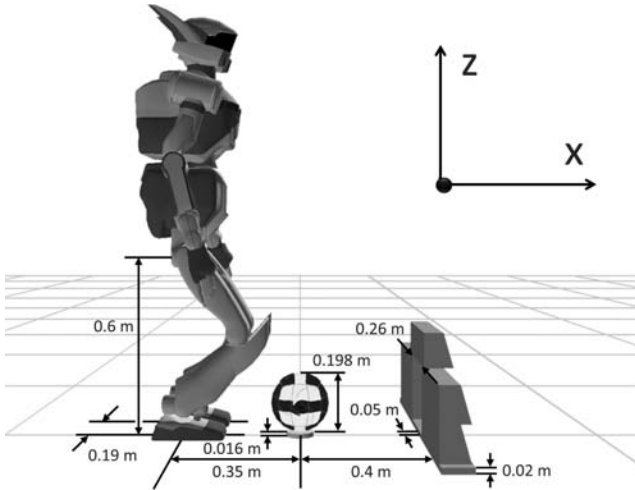| Error in velocity	$T_{\text{swing}} = 0.7\,\text{s}\ (N = 100)$		$T_{\text{swing}} = 1.0\,\text{s}\ (N = 49)$	
	μ_e	σ_e	μ_e	σ_e
$e_{v^s_{O_B,0}}$	$-0.38\,\text{m/s}$	$0.75\,\text{m/s}$	$-0.40\,\text{m/s}$	$0.54\,\text{m/s}$
$e_{\phi^s_m}$	$-1.9°$	$2.5°$	$-1.2°$	$1.5°$
$e_{\theta^s_{m,0}}$	$1.4°$	$5.1°$	$0.0°$	$4.2°$

Fig. 16. Setup for the single experiment.
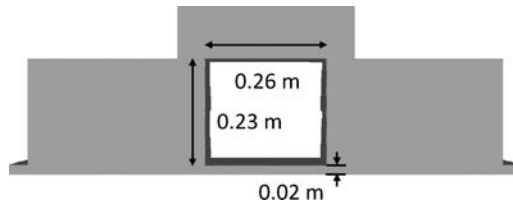
Fig. 17. Goal frame dimensions.

Table 7.   Parameters for the single experiment.

Swing phase time span (T_{swing}^d) [s]	1.3
Ball's yaw velocity ($\dot{\phi}_{B0}^d$) [°/s]	0
Ball's pitch velocity ($\dot{\theta}_{B0}^d$) [°/s]	-100
Reaching angle ($\theta_{m,f}^d$) [°]	-15
Goal's frontal distance ($p_{G_x}^d$) [m]	0.4
Goal's lateral distance ($p_{G_y}^d$) [m]	0.05
Goal's height ($p_{G_z}^d$) [m]	$r_B+0.03$

The mentioned disk was required in the real experiment to make it easier to place the ball at the desired XY-location. However, it raised the ball's initial height, such that for its initial position we have $\boldsymbol{p}_{O_B,0} = \begin{bmatrix} 0 & 0 & (r_B + 0.016\,\text{m}) \end{bmatrix}^T$, which was also considered in calculations and in simulation.

The parameters used for this kicking motion are shown in Table 7. These were specifically chosen to produce a motion inside of the corresponding workspace and to comply with the joint limits and the maximum joint velocities of the robot.

The desired goal position was marked out by using a goal frame over a hurdle of $0.02\,\text{m}$ height, creating a narrow space of $0.26\,\text{m}$ width centered at this goal position, that should be traveled through by the ball, whose diameter is almost $0.2\,\text{m}$; that is, with a margin of $0.03\,\text{m}$ for each side of the ball (Fig. 17). By including a hurdle at the goal position, the desired arrival condition of the ball to the goal was forced to be a requirement, as the ball would not be able to enter to the goal without attaining the desired height (within $0.01\,\text{m}$ of margin). It is worth to mention why the goal position is very near to the robot. This is because the joint velocities required for farther 3D goals were surpassing the current maximum joint velocities. Also, it is worth to remark that the ball's pitch velocity ($\dot{\theta}_{B0}^d$) is given a value of $-100°/\text{s}$. This is because in the case of $\dot{\theta}_{B0}^d = 0$, the required approaching velocity vector was inducing a trajectory for the swing foot in which the sole was colliding with the ground. By switching it to $-100°/\text{s}$, this vector turned to be less steeper, rising, in this way, the lowest point of the trajectory of the sole of the swing foot.

After calculating the desired impulse and the impact coordinates, an operational point on the swing foot is selected together with the foot's orientation at the moment of the impact. Then, the approaching velocity is calculated. These calculations, together with the specifications of the task, are used to define the trajectory of the operational point and the swing foot. The desired trajectory for the operational point is shown in Fig. 18.

On the other hand, the trajectory for the horizontal position of the waist is at first proposed to shift horizontally from the reference configuration to be placed exactly over the support (left) ankle position, and let it remain there until the motion finishes. Then, it shifts back to the original position at the end of the kicking motion. However, this trajectory does not ensure dynamic stability. By following the iterative
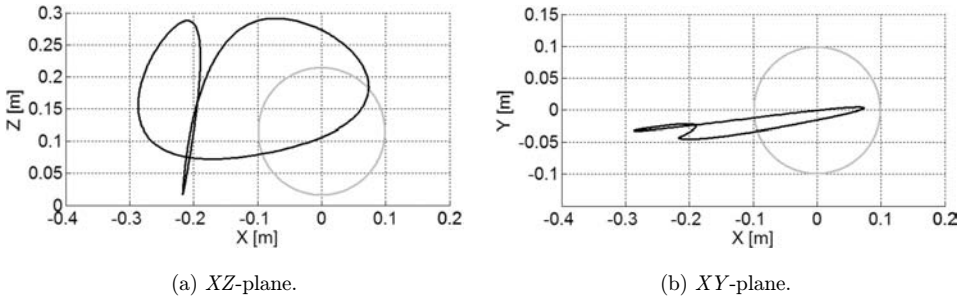
(a) *XZ*-plane.

(b) *XY*-plane.

Fig. 18. Desired operational point's trajectory.

process described in Sec. 8, and after two iterations, we get a balancing motion trajectory for the waist that almost fixes the trajectory for the ZMP as desired; that is, exactly under the left ankle throughout the duration of the kicking motion. The calculated trajectory for the waist is depicted in Fig. 19, compared to the proposed one (the initial condition of the iterative process).

In order to compute this trajectory, a delay of 2 s was originally considered at the beginning and at the end of the kicking motion, followed by the motion of the waist that transfers the ZMP to a position below the left ankle, which lasts 1 s; that is, the kicking motion actually starts at $t = 3$ s, in such a way that the instant of the impact occurs 1.3 s later, at $t = 4.3$ s, as it is marked in Fig. 19. The sampling time is $\Delta t = 0.005$ s. The obtained ZMP trajectory is depicted in Fig. 20, compared to the desired one. However, as we can see, the difference between these signals is extremely small and practically negligible, given that the mean error between them was 1×10^{-6} m, achieved with only two iterations.

The whole calculation lasted approximately 65 s, by using non-optimized `Matlab` code and running on a Laptop PC with Intel(R) Core(TM) i7-2670QM CPU @ 2.20 GHz, 8 GB of RAM and Windows 7 (64-bit).

This kicking motion was simulated in `OpenHRP3`,[38] as well as tested with the real robot. The resulting simulated motion is shown through a selection of frames in Fig. 21, as seen on the sagittal and frontal planes, respectively, and labeled with the
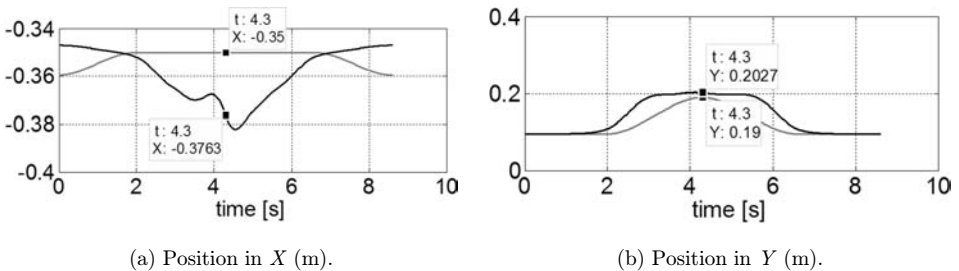

(a) Position in X (m).

(b) Position in Y (m).

Fig. 19. Proposed waist trajectory (gray) versus computed one after the iterative process (black).

(a) Position in X (m).

(b) Position in Y (m).

Fig. 20.   Desired ZMP trajectory (gray) versus attained one after the iterative process (black).

(a) 0.000 s   (b) 3.750 s   (c) 5.250 s   (d) 5.900 s   (e) 6.250 s

(f) 0.000 s   (g) 3.750 s   (h) 5.250 s   (i) 5.900 s   (j) 6.250 s

(k) 6.572 s   (l) 6.650 s   (m) 6.733 s   (n) 7.200 s   (o) 7.750 s

(p) 6.572 s   (q) 6.650 s   (r) 6.733 s   (s) 7.200 s   (t) 7.750 s

Fig. 21.   Simulated kicking motion as seen on the sagittal and the frontal planes.

corresponding simulation timestamp. At the beginning of the simulation the robot has a predefined configuration. This one evolves to the desired initial one in 3.750 s. Then, the robot transfers the ZMP to a position under the left ankle, such that the kicking motion starts at $t = 5.250$ s. The foot makes contact with the ball at $t = 6.572$ s; that is, $T^s_{\text{swing}} = 1.322$ s instead of 1.3 s, as it was planned. This is attributed to the *following error* of the controller. The ball starts its motion exactly after the impact, hitting the hurdle at $t = 6.733$. Figure 22 shows a closer view of the ball motion, where it can be seen that the ball succeeds to enter into the goal frame.

The same kicking motion was also performed by the real robot. This one is shown through a selection of snapshots of the video taken during the experiment on Fig. 23, labeled with the corresponding video timestamp; that is, relative to the time that the video camera started recording. As can be seen, the real kicking motion starts at $t = 8.490$ s and the foot makes contact with the ball at approximately $t = 9.790$ s; that is, $T^r_{\text{swing}} = 1.3$ s as it was planned. Here, the superscript r stands for a "real" value (from the actual experiment). The ball arrives to the goal at $t = 9.960$ s. The video was carefully taken such that the image plane was as parallel as possible to the sagittal plane of the robot. In this way, it was possible to estimate the Z-position of the ball on every frame by measuring distances in the image and relating them to known real distances, carefully considering the perspective. Then, by means of a polynomial curve fitting process with given initial position (p_{B0_z}) and fixed quadratic coefficient ($-\frac{1}{2}g$), it was possible to estimate the initial linear velocity vector of the ball, $\boldsymbol{v}^r_{\boldsymbol{O_B},0}$, and construct the corresponding curve. This one is shown in Fig. 24, together with the desired one and the simulated one. The considered goal position is marked as a *data-tip* at the end of each curve. It is worth to remark that the X-value for the goal position in those three curves is not exactly 0.4 m. This is due to the sampling time. Also, it is possible to notice that, in fact, during the real experiment the ball did not hit the hurdle, but it passed over it, almost reaching the desired position and performing better than in the simulation. Table 8 shows a comparison between the calculated, the simulated and the real data (that could be measured) regarding to the motion of the ball.

As there is no collision data released by the simulator, the simulated impact coordinates ϕ^s_0 and θ^s_0 were obtained by using the sequence of positions and orientations of the swing foot, together with the collision detection algorithm developed by us to search for the first single collision point on the ball,[39] from which the impact

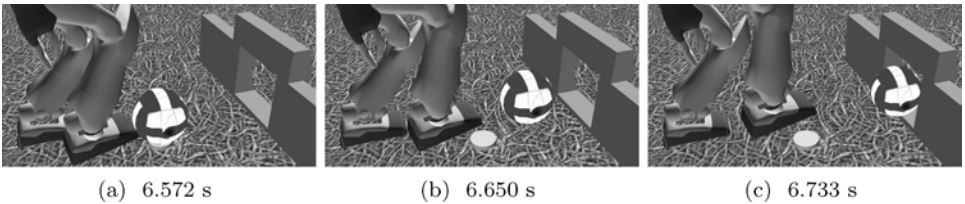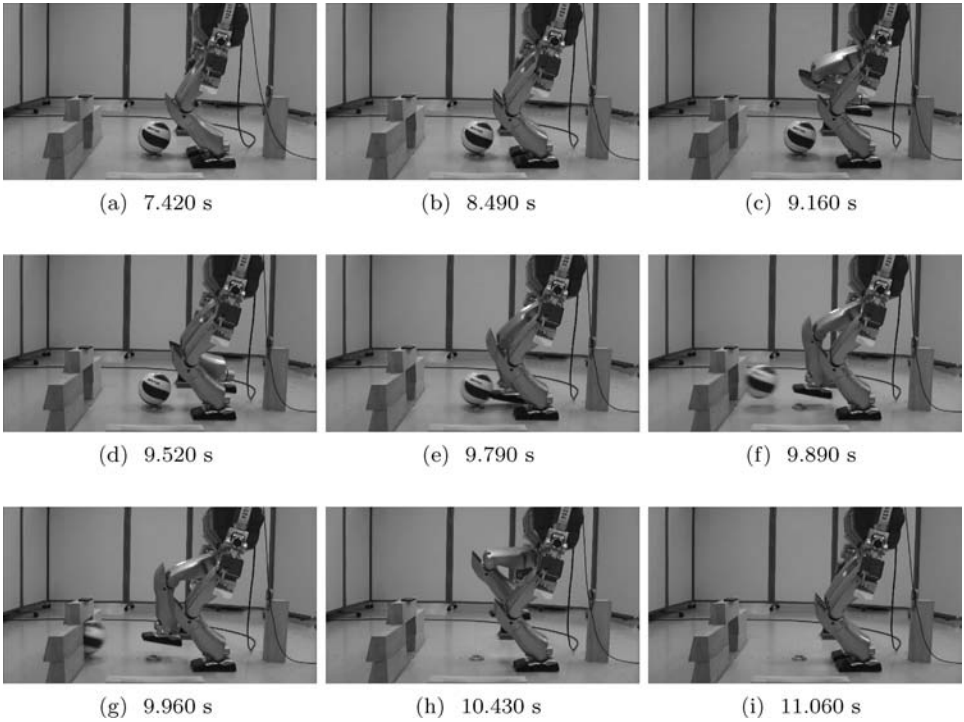| (a) 6.572 s | (b) 6.650 s | (c) 6.733 s |

Fig. 22. Simulated ball motion.

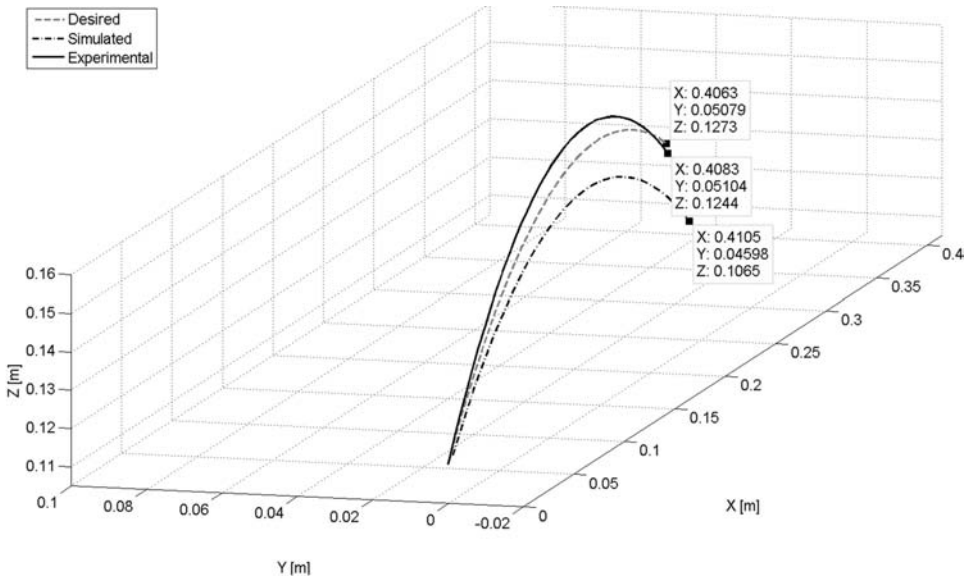Fig. 23.   Kicking motion implemented on the real humanoid robot.

Fig. 24.   Desired, simulated and experimental trajectories of the center of the ball.

Table 8. Comparison of calculated, simulated and real data.

Data	Desired (d)	Simulated (s)	Real (r)
Time span for the swing phase T_{swing} [s]	1.300	1.322	1.300
Azimuthal impact coordinate ϕ_0	7.1	7.5	7.1
Polar impact coordinate θ_0	-20.5	-20.4	-21.0
Magnitude of $\boldsymbol{v_0}$ (v_0) [m/s]	2.70	2.55	2.47
Azimuthal angle of $\boldsymbol{v_0}$ (ϕ_m) [°]	7.1	6.5	7.1
Polar angle of $\boldsymbol{v_0}$ (θ_{m0}) [°]	18.6	17.3	22.4
Time span to reach the goal (T_{goal}) [s]	0.160	0.168	0.170
Reaching angle $\theta_{m,f}$ [°]	-15.0	-20.8	-20.0
Goal's frontal distance (p_{G_x}) [m]	0.400	0.411	0.408
Goal's lateral distance (p_{G_y}) [m]	0.050	0.046	0.050
Goal's height (p_{G_z}) [m]	0.129	0.107	0.127
Ball's yaw velocity ($\dot{\phi}_{B0}^{d}$) [°/s]	0	-65	—
Ball's pitch velocity ($\dot{\theta}_{B0}^{d}$) [°/s]	-100	-59	—

coordinates were calculated. The initial linear velocity of the ball as well as the rate of change of its orientation coordinates were calculated by means of a polynomial curve fitting process.

On the other hand, the real (experimental) data cannot be calculated in the same way, as there is no sensor providing the position and orientation of the swing foot or the ball. The experimental impact coordinates were estimated by means of approximate measures taken from the snapshots of the video recorded by a second camera making a close-up of the feet of the robot. Some of the snapshots are shown in Fig. 25, particularly showing an instant before the impact, the instant of the impact itself and an instant after the impact. A ruler behind the robot was used as a reference measure. The rate of change of the orientation coordinates could not be measured from the video. The frames in Fig. 25 also show that the support foot has a negligible motion with respect to the floor. This validates the hypothesis made in Eq. (57), which caused the inverse of the effective mass of the robot to be neglected during the computation of the approaching velocity.

Finally, we show the comparison between the desired ZMP trajectory and the experimental one. These ones are shown in Fig. 26 along with the boundaries of the foot (in each direction) during the single and the double support phases, as a way to validate the stability of the motion. Let us point out that the disturbance shown at
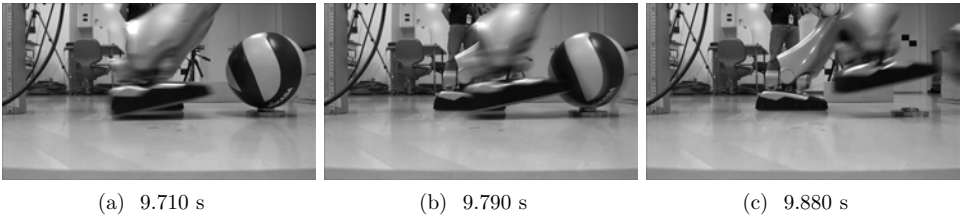

(a) 9.710 s   (b) 9.790 s   (c) 9.880 s

Fig. 25. Close-up for the swing foot motion during the real experiment.

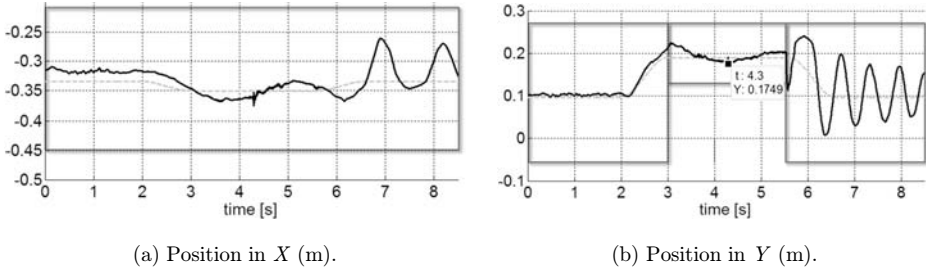(a) Position in X (m).   (b) Position in Y (m).

Fig. 26.   Desired ZMP trajectory (dashed) versus experimental one (solid). The boundaries of the foot (in each direction) during the single support phase are also shown to validate the stability.

$t = 4.3$ s is regarded as the effect that the collision with the ball had on the robot. As can be seen, this disturbance is comparable in magnitude to other uncertainties, and it is not large enough to destabilize the robot.

10. Conclusions and Future Work

In this paper, we presented a method to generate a proper stable motion for a humanoid robot that has to perform a specific pedipulation task, consisting of driving a spherical object to a specified 3D goal position with certain motion characteristics, by exerting some proper impact conditions. However, the accuracy of the results is highly dependent on how demanding the task is; that is, on the magnitude and orientation of the approaching velocity vector that is required to exert the necessary impact conditions. If the magnitude is so big, the trajectory that the foot has to follow is longer in order to accelerate from zero velocity to the desired one. This means that the trajectory of the foot may lie out of the workspace. One solution is to shorten the time needed to perform the motion, which actually demands a higher acceleration of the foot and, consequently, higher joint accelerations. On its behalf, the robot controller may not be able to cope with this trajectory as accurately as needed, so that the resultant one may induce an error on the impact conditions. With respect to the orientation of the approaching velocity vector, if it is very steep, the swing foot would need to go below the ground level to attain the required impact conditions. One solution is to relax the angular velocity requirements of the pedipulated object, in order to modify the impact coordinates and change the orientation of the approaching velocity vector. As a consequence, the impact conditions are more dependent on the friction between foot and ball, and on the tangential component of this velocity. In general, the farther (and higher) the goal position is, the less accurate the results are.

The method presented in this paper solved a specific impulsive pedipulation task, but it can be adapted to any other target object which can be modeled as a rigid object and for which the physical properties are known, as well as its geometry. Depending on which task we want to achieve by means of impulsive pedipulation, it

is just the matter to find the required impulse on the object applied to some operational point, whose velocity is related to the linear and angular ones of the object by means of the corresponding Jacobian. Then, the rest of the algorithm (from the side of the robot) can be worked out in the same way, by planning a dynamic motion capable to attain the required velocity of the operational point in the foot while keeping it balanced. Besides that, it is worth to mention that by the way this method was developed, it is possible to also adapt it to be used for any link of the robot.

As a future work we want to consider strategies for the robot to approach to a feasible position relative to the spherical object, and even to perform the pedipulation task while a dynamic motion is taking place. For example, kicking the ball while walking fast without stopping. This can make the kicking motion to look less clumsy and more natural. But also, this will require acquisition of the relative position of the initial position of the object by using vision data. Also, we want to consider the case in which the impact is large enough to produce some significant motion at the support foot. This would produce an inverse of the effective mass of the robot (in each direction) not negligible anymore, affecting the total momentum transmitted to the object, as well as compromise the stability of the robot. Finally, we would like to generate an appropriate movement for the upper body to compensate the vertical moment induced by the robot's motion, as well as to optimize the algorithm (on C++) to reduce the computation time and improve the accuracy of the results.

References

1. D. Harper, Online etymology dictionary (2010), http://www.Dictionary.com.
2. K. M. Lynch, The mechanics of fine manipulation by pushing, *IEEE Int. Conf. Robotics and Automation (ICRA)*, Vol. 3, May 1992, Nice, France, pp. 2269–2276.
3. K. M. Lynch and M. T. Mason, Stable pushing: Mechanics, controllability, and planning, *Int. J. Robot. Res.* **15**(6) (1996) 533–556.
4. T. Sugihara and Y. Nakamura, Boundary condition relaxation method for stepwise pedipulation planning of biped robots, *IEEE Trans. Robot.* **25**(3) (2009) 658–669.
5. M. T. Mason and K. M. Lynch, Dynamic manipulation, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vol. 1, July 1993, Yokohama, Japan, pp. 152–159.
6. W. Huang and M. T. Mason, Experiments in impulsive manipulation, *IEEE Int. Conf. Robotics and Automation (ICRA)*, Vol. 2, May 1998, Leuven, Belgium, pp. 1077–1082.
7. C. Zhu, Y. Aiyama, T. Chawanya and T. Arai, Releasing manipulation, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, November 1996, Osaka, Japan, pp. 911–916.
8. J. Müller, T. Laue and T. Röfer, Kicking a ball — Modeling complex dynamic motions for humanoid robots, *RoboCup 2010*, June 2010, Singapore, pp. 109–120.
9. F. Wenk and T. Röfer, Online generated kick motions for the NAO balanced using inverse dynamics, *RoboCup 2013*, June 2013, Eindhoven, Netherlands, pp. 25–36.
10. S. Miossec, K. Yokoi and A. Kheddar, Development of a software for motion optimization of robots — Application to kick motion of the HRP-2 robot, *IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, December 2006, Kunming, China, pp. 299–304.
11. S. Lengagne, P. Fraisse and N. Ramdani, Planning and fast re-planning of safe motions for humanoid robots: Application to a kicking motion, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, October 2009, St. Louis, USA, pp. 441–446.

12. S. Lengagne, J. Vailant, E. Yoshida and A. Kheddar, Generation of whole-body optimal dynamic multi-contact motions, *Int. J. Robot. Res.* **32**(9–10) (2013) 1104–1119.
13. S. J. Yi, S. McGill and D. D. Lee, Improved online kick generation method for humanoid soccer robots, *IEEE-RAS Int. Conf. Humanoid Robots*, October 2013, Atlanta, USA.
14. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirokawa, Resolved momentum control: Humanoid motion planning based on the linear and angular momentum, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vol. 2, October 2003, Las Vegas, USA, pp. 1644–1650.
15. H. Schempf, C. Kraeuter and M. Blackwell, Roboleg: A robotic soccer-ball kicking leg, *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 1995, Nagoya, Japan, pp. 1314–1318.
16. J. Y. Choi, B. R. So, B. J. Yi, W. Kim and I. H. Suh, Impact-based trajectory planning of a soccer ball in a kicking robot, *IEEE Int. Conf. Robotics and Automation (ICRA)*, April 2005, Barcelona, Spain, pp. 2834–2840.
17. The RoboCup Federation, Robocup (2015), http://www.robocup.org.
18. R. Cisneros *et al.*, Impulsive pedipulation of a spherical object for reaching a 3D goal position, *IEEE-RAS Int. Conf. Humanoid Robots*, October 2013, Atlanta, USA, pp. 154–160.
19. H. Brody, The moment of inertia of a tennis ball, *Phys. Teach.* **43** (2005) 503–505.
20. I. D. Walker, Impact configurations and measures for kinematically redundant and multiple armed robot systems, *IEEE Trans. Robot. Autom.* **10**(5) (1994) 670–683.
21. E. R. Westervelt *et al.*, *Feedback Control of Dynamical Bipedal Robot Locomotion* (CRC Press, 2007).
22. W. J. Stronge, *Impact Mechanics* (Cambridge University Press, 2000).
23. C. White, *Projectile Dynamics in Sports: Principles and Applications* (Routledge, 2011).
24. F. W. Gembicki, Vector optimization for control with performance and parameter sensitivity indices (Ph.D. thesis, Case Western Reserve University, 1974).
25. C. M. Fonseca and P. J. Fleming, Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization, *Int. Conf. Genetic Algorithms*, June 1993, Illinois, USA, pp. 416–423.
26. Y. Wang and M. T. Mason, Two-dimensional rigid-body collisions with friction, *J. Appl. Mech.* **59**(3) (1991) 635–642.
27. O. Khatib, Inertial properties in robotic manipulation: An object-level framework, *Int. J. Robot. Res.* **14**(1) (1994) 19–36.
28. M. Botsch, L. Kobbelt, M. Pauly, P. Alliez and B. Lévy, *Polygon Mesh Processing* (Taylor and Francis, 2010).
29. S. Lengagne, P. Mathieu, A. Kheddar and E. Yoshida, Generation of dynamic motions under continuous constraints: Efficient computation using B-splines and Taylor polynomials, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, October 2010, Taipei, Taiwan, pp. 698–703.
30. D. Salomon, *Computer Graphics and Geometric Modeling* (Springer-Verlag, 1999).
31. W. R. Barfield, The biomechanics of kicking in soccer, *Clin. Sports Med.* **17**(4) (1998) 711–728.
32. M. Vukobratović and B. Borovac, Zero-moment point — Thirty-five years of its life, *Int. J. Huma. Robot.* **1**(1) (2004) 157–173.
33. K. Nishiwaki *et al.*, Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vol. 3, October 2002, Lausanne, Switzerland, pp. 2684–2689.
34. S. Nakaoka, S. Kajita and K. Yokoi, Intuitive and flexible user interface for creating whole body motions of biped humanoid robots, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, October 2010, Taipei, Taiwan, pp. 1675–1681.

35. S. Kajita, H. Hirukawa, K. Harada and K. Yokoi, *Introduction to Humanoid Robotics* (Springer, 2005).
36. W. H. Press *et al.*, *Numerical Recipes in C* (Cambridge University Press, 1992).
37. K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi and T. Isozumi, Humanoid robot HRP-2, *IEEE Int. Conf. Robotics and Automation (ICRA)*, April 2004, New Orleans, USA, pp. 1083–1090.
38. S. Nakaoka, S. Hattori, F. Kanehiro, S. Kajita and H. Hirokawa, Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, October 2007, San Diego, USA, pp. 3641–3647.
39. R. Cisneros, E. Yoshida and K. Yokoi, Ball dynamics simulation on `OpenHRP3`, *IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, December 2012, Guangzhou, China, pp. 871–877.

Rafael Cisneros received the B.Eng. degree in Electronics and Computers Engineering from the University of the Americas-Puebla (UDLA-P), Mexico in 2006, the M.Sc. degree in Automatic Control from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), Mexico City in 2009 and the Ph.D. degree in Intelligent and Interaction Technologies from the University of Tsukuba, Japan in 2015. In 2008, he made a research stay at the Institute of Industrial Automation (IAI), Spain. From 2009–2011, he worked as a Research Assistant at the Department of Automatic Control of CINVESTAV-IPN and as a Professor at La Salle University, Mexico City. From 2011–2015, he was a member of the CNRS–AIST JRL, UMI3218/CRT at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. Since 2015, he has been working as a Postdoc at the Humanoid Research Group of AIST.

Kazuhito Yokoi is the Director of the Intelligent Systems Research Institute, Department of Information Technology and Human Factors, National Institute of Advanced Industrial Science and Technology (AIST) in Japan. He received the M.E. and Ph.D. degrees in Mechanical Engineering Science from the Tokyo Institute of Technology in 1986 and 1994, respectively. In 1986, he joined the Mechanical Engineering Laboratory, Ministry of International Trade and Industry, Japan. He is currently the Director of Intelligent Systems Research Institute, Department of Information Technology and Human Factors, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He is also a Guest Professor of Faculty of Science and Engineering at Waseda University. From April 2005 to July 2015, he was an Adjunctive Professor of the Cooperative Graduate School at the University of Tsukuba. From August 2013 to March 2015, he was a General Manager of the International Research Institute for Nuclear Decommissioning. From November

1994 to October 1995, he was a Visiting Scholar at the Robotics Laboratory, Computer Science Department, Stanford University. He has received the best video awards of ICRA'03 and ICRA'04, the best paper awards of Society of Instrument and Control Engineers (SICE), Robotics Society of Japan (RSJ), and Japan Society of Mechanical Engineers (JSME). He is a Fellow of RSJ and JSME and an AdCom member (2013–2015) of the IEEE Robotics and Automation Society.

Eiichi Yoshida received M.E. and Ph.D. degrees on Precision Machinery Engineering from Graduate School of Engineering, the University of Tokyo in 1993 and 1996, respectively. In 1996, he joined the former Mechanical Engineering Laboratory, later reorganized as the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He served as Co-Director of AIST/IS–CNRS/ST2I Joint French–Japanese Robotics Laboratory (JRL) at LAAS–CNRS, Toulouse, France, from 2004–2008. Since 2009, he is Co-Director of CNRS–AIST JRL (Joint Robotics Laboratory), UMI3218/RL, and since 2015 he serves as Deputy Director of the Intelligent Systems Research Institute (IS-AIST), AIST, Tsukuba, Japan. His research interests include robot task and motion planning, human modeling, and humanoid robots.