Congresso de Métodos Numéricos em Engenharia 2015
Lisboa, 29 de Junho a 2 de Julho 2015

DEVELOPMENT OF A GENERAL GROUND CONTACT MODEL FOR ANALYSIS AND SIMULATION OF HUMAN GAIT

João A. Marques[1*], Sérgio Gonçalves[1] and Miguel T. Silva[1]

1: LAETA, IDMEC - Instituto de Engenharia Mecânica
Instituto Superior Técnico
Univeridade de Lisboa
Lisbon, Portugal
e-mail: joaoamorimmarques,sergio.goncalves,miguelsilva@tecnico.ulisboa.pt, web: http://biomechmov.tecnico.ulisboa.pt

Keywords: Contact detection, Multibody dynamics, Foot force, Implicit surfaces, Movement simulation, Contact forces

Abstract. *The correct calculation of foot and ground contact forces and their application points is a crucial and indispensable step in path planning or human movement simulation. The purpose of this work is the development of a computational model that, using forward dynamic analysis with natural coordinates, accurately calculates the contact forces and their application points generated during the contact between two bodies. With the purpose of simulate the soft tissue contact with the ground of a leg and foot, a contact force law with a moderated coefficient of restitution is used. Contact detection is assured through the use of an iterative method that recurs to implicit surfaces, in particular planar, quadric and superquadric surfaces, to find possible contact pairs. The contact detection method and contact force model are applied to simulate simple case test examples.*

1 INTRODUCTION

Rigid body contact, which can be defined as the sharing of the same physical space by two different bodies or objects, generating forces opposing body intersection [1], has various applications in computational physics and is an important step in mechanical modelling [2, 3]. It is of even greater relevance in forward dynamic simulations [4]. In this work we propose a contact detection method, as well as a force (and friction) model, to be used in a biomechanical foot model in order to correctly evaluate contact forces in dynamic simulations.
In a general sense, the analysis of rigid body contact can be divided in multiple sets: *i*) definition of the geometric constraints that define the bodies, *ii*) detection of possible contact bodies, areas or points, *iii*) application of the correct constitutive force laws [5–7].

1.1 Contact Detection

Contact detection is one of the most difficult and challenging tasks of mechanical simulation [5, 8]. It is also one of the more computationally expensive processes involved, increasing it's demands as the complexity of the objects increase [9, 10]. In order to decrease the demands of contact detection it is common practice to resort to bounding boxes or volumes [9, 10]. For such bounding volumes, there are various types of surfaces commonly used, being (super) ellipsoids between the most frequent [10, 11] as they allow for great flexibility in terms of shape generation [11] and present a low algebraic degree [10].
In the present work super(ellipsoids) are used, but not as bounding volumes but as contact surfaces as proposed by Lopes *et al.* [1], where contact detection is based in the common normal concept. In this method, the contact detection problem is reduced to determine a pair of points and respective normals and evaluate their collinearity. This kind of approach normally has the drawback that examining a single point for contact is usually insufficient [8]. On the other hand the detection of the correct contact candidate points is highly efficient, using the Newton-Raphson method, as it only requires a few number of iterations to find an accurate solution [12].
The work Lopes *et al.*[1] imposes as a condition for the method that the surfaces considered must be at least C^2 continuous. Although they focus on ellipsoids and superellipsoids, it is also referred that the method should be expanded to any kind of surface that is at least C^2 continuous and can be defined implicitly [1]. In this work the method developed is applicable to any surfaces that fulfil the above conditions, but we will focus on planar, quadric and superquadric surfaces.

1.2 Contact Force Model

A key point in contact-impact interactions is the constitutive force law utilized [13]. Force based models, also known as penalty models, have the advantage of being simple and effective [13] and can also easily approximate body deformation [3, 14].
The development of force models is made based on the Hertz law joined together with a hysteresis damping coefficient, in order to consider the energy dissipation that occurs during contact events [13, 15]. Different approximations of the hysteresis damping factor lead to different force models and the selection of the model has a strong impact on the prediction of dynamic behaviour [13]. It is important to refer that these models are typically a function of the coefficient of restitution, which is the ratio between the post- and pre-impact velocities [13, 15].
In this work we use both a purely elastic force model (Hertz law) and the model proposed by Flores *et al.* [15]. This last model has the particularity of performing well for medium and low coefficient of restitution values and can be easily included in the equations of motion of multibody systems [15].

1.3 Friction Force

Generally speaking, most contact force models do not consider friction[16]. However, in this work we implemented a very simple friction force model. The model implemented considers the sum of the Coulomb friction term, from Coulomb's law[17] with a viscous friction term, that depends on the tangent velocity[18].

2 METHODS

In this section we will describe the algorithm used for the contact detection, with special emphasis to the process of obtaining orthogonal bases for each contact candidate point. The force model used will also be detailed. In order to grant easier readability, this section will be divided in different parts where in each, a specific detail of the algorithm or an important step will be described.

2.1 Terminology and Nomenclature

In the process of detection of contact and force application ,three different reference frames need to be considered: *i*) a global reference frame, *ii*) a rigid body local frame, *iii*) a surface local reference frame. As such for a given vector or point, the following notation was used:

- For vectors, points and matrices in the global reference frame no special notation is applied. As such, the coordinates vector will be $\boldsymbol{q}$;
- For vectors, points and matrices in the rigid body local reference frame a prime (') is used. As such, the coordinates vector will be $\boldsymbol{q}'$;
- For vectors, points and matrices in the surface local reference frame a double prime (") is applied. As such, the coordinates vector will be $\boldsymbol{q}''$.

Other relevant notation will include:

- $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ for the coordinates of the contact candidate points for surfaces i and j (see figure 1);
- $\boldsymbol{q}$ as the coordinate vector containing the candidate points coordinates;

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{s}_i \\ \boldsymbol{s}_j \end{bmatrix} = \begin{bmatrix} x_i & y_i & z_i & x_j & y_j & z_j \end{bmatrix}^T \tag{1}$$

- $\boldsymbol{n}_i$, $\boldsymbol{t}_i$ and $\boldsymbol{b}_i$ as the normal, tangential and binormal vectors of the surface i, evaluated in point $\boldsymbol{s}_i$ (and the same for surface j). See a detailed representation of these vectors in figure 1;

- $\mathcal{F}_i$, $\mathcal{F}_j$ as the implicit definitions of surfaces i and j;
- $\boldsymbol{d}_{ij}$ for the distance vector between the points considered for contact;
- $\boldsymbol{c}$ as the geometric constraints vector;
- $\boldsymbol{D}$ as the geometric constraints Jacobian matrix;
- The subscript $_{\boldsymbol{q}}$ for any partial derivatives calculated in order to $\boldsymbol{q}$.

2.2 Minimum distance calculation

As described above, in the contact detection problem, the Newton-Raphson method was used in order to find a pair of points, belonging to the surfaces considered for contact, with collinear and symmetric external normals, which according to the work of Lopes *et al.* [1] will correspond to the closer pair of points of the surfaces considered and as such, the point where if it occurs, contact will happen. It is also required collinearity between the distance vector $\boldsymbol{d}_{ij}$ (connecting both candidate contact points represented in figure 1) and the normal surfaces. When translated mathematically, the conditions are:

$$\boldsymbol{n}_i \times \boldsymbol{n}_j = \boldsymbol{0} \tag{2}$$
$$\boldsymbol{d}_{ij} \times \boldsymbol{n}_i = \boldsymbol{0} \tag{3}$$
$$\boldsymbol{d}_{ij} \times \boldsymbol{n}_j = \boldsymbol{0} \tag{4}$$
$$\mathcal{F}_i(\boldsymbol{s}_i'') = 0 \tag{5}$$
$$\mathcal{F}_j(\boldsymbol{s}_j'') = 0 \tag{6}$$

The first equation ensures that the normals are collinear with each other. Equations 3 and 4 test the collinearity between each normal and the distance vector. The last two equations ensure that the candidate points are in the considered surface. It is important to notice that both Eqs. 3 and 4 imply the same if Eq. 2 is verified (if the normals are collinear and one of them is collinear with the distance vector, the other normal will automatically be collinear with the distance vector). As such one of them can be dropped to avoid redundant equations. The system of equations is also easier to solve for internal product conditions rather than for cross product ones. As such:

$$\boldsymbol{n}_i \times \boldsymbol{n}_j = \boldsymbol{0} \Leftrightarrow \begin{cases} \boldsymbol{n}_i \cdot \boldsymbol{t}_j = 0 \\ \boldsymbol{n}_i \cdot \boldsymbol{b}_j = 0 \end{cases} \tag{7}$$

$$\boldsymbol{d}_{ij} \times \boldsymbol{n}_j = \boldsymbol{0} \Leftrightarrow \begin{cases} \boldsymbol{d}_{ij} \cdot \boldsymbol{t}_j = 0 \\ \boldsymbol{d}_{ij} \cdot \boldsymbol{b}_j = 0 \end{cases} \tag{8}$$

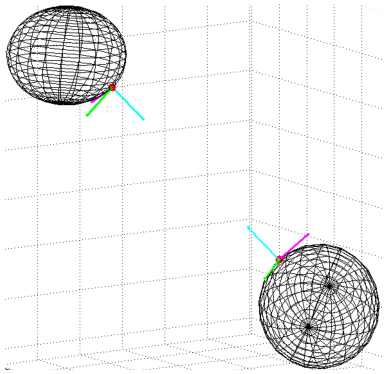

Figure 1: *Representation of a pair of candidate contact points and their respective normal (blue), tangent (green) and binormal (magenta) vectors at the correspondent surface.*

With Eqs. 5 - 8, the geometric constraints vector $\boldsymbol{c}$ can be defined:

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{n}_i \cdot \boldsymbol{t}_j \\ \boldsymbol{n}_i \cdot \boldsymbol{b}_j \\ \boldsymbol{d}_{ij} \cdot \boldsymbol{t}_j \\ \boldsymbol{d}_{ij} \cdot \boldsymbol{b}_j \\ \mathcal{F}_i \\ \mathcal{F}_j \end{bmatrix} \tag{9}$$

Eq. 9 defines a system of six equations that we want to solve in the homogeneous form in order to determine $\boldsymbol{q''}$. Hence:

$$\boldsymbol{c}(\boldsymbol{q''}) = \boldsymbol{0} \tag{10}$$

In order to solve the system of six equations presented above with the Newton-Raphson method it is necessary to calculate the Jacobian matrix of the constraints vector. The Jacobian matrix can be defined as:

$$\boldsymbol{D}(\boldsymbol{q''}) = \frac{\partial \boldsymbol{c}(\boldsymbol{q''})}{\partial \boldsymbol{q''}} \Leftrightarrow \boldsymbol{D}(\boldsymbol{q''}) = \begin{bmatrix} \boldsymbol{t}_j \cdot (\boldsymbol{n}_i)_{\boldsymbol{q''}} & + & \boldsymbol{n}_i \cdot (\boldsymbol{t}_j)_{\boldsymbol{q''}} \\ \boldsymbol{b}_j \cdot (\boldsymbol{n}_i)_{\boldsymbol{q''}} & + & \boldsymbol{n}_i \cdot (\boldsymbol{b}_j)_{\boldsymbol{q''}} \\ \boldsymbol{t}_j \cdot (\boldsymbol{d}_{ij})_{\boldsymbol{q''}} & + & \boldsymbol{d}_{ij} \cdot (\boldsymbol{t}_j)_{\boldsymbol{q''}} \\ \boldsymbol{b}_j \cdot (\boldsymbol{d}_{ij})_{\boldsymbol{q''}} & + & \boldsymbol{d}_{ij} \cdot (\boldsymbol{b}_j)_{\boldsymbol{q''}} \\ \boldsymbol{n}_i'' & & \boldsymbol{0} \\ \boldsymbol{0} & & \boldsymbol{n}_j'' \end{bmatrix} \tag{11}$$

The method used to calculate all the values, vectors and respective derivatives is described in the next subsections.

2.3 Implicit surface and respective normal

From now on, many of the calculations and derivatives will be made in the surface reference frame because, as any term in the global reference frame is a multiplication of a rotation matrix and the respective vector. As the rotation matrix are not dependent of the coordinates, they can be multiplied after all the calculations to put all variables in the global reference frame.
The calculation of the implicit surface equation, its normal and respective gradient is the only calculation that depends on the surface used for contact detection. As such, there is a need for a way to determine which surface is to be used (defined by the user) and based on that, make the necessary calculations.
With the objective of making the implemented code as easy to read as possible, for each kind of surface was created a separated subroutine that is responsible for making the calculations above referred, and the main routine only calls the right subroutine, in accordance with the surfaces chosen by the user.
As stated above, this subroutine must return (based on the geometrical parameters of the surface and the point that is being considered for contact):

1. **The surface evaluation** - Based on the point considered and in the implicit equation of the surface (e.g., for the case of an ellipsoid).

$$\mathcal{F} = \frac{x''^2}{a^2} + \frac{y''^2}{b^2} + \frac{z''^2}{c^2} - 1 \tag{12}$$

2. **The surface normal vector** - It is the gradient of the implicit definition, evaluated in the considered point.

$$\mathbf{n''} = \nabla \mathcal{F} \tag{13}$$

3. **The derivative of the normal vector with respect to the vector of contact points coordinates** - derivative of each term of the normal with regard to each one of the six coordinates considered. Corresponds to the Hessian matrix of the implicit function.

$$\boldsymbol{n''_{q''}} = \begin{bmatrix} \frac{\partial n''_x}{\partial x''_i} & \frac{\partial n''_x}{\partial y''_i} & \frac{\partial n''_x}{\partial z''_i} & \frac{\partial n''_x}{\partial x''_j} & \frac{\partial n''_x}{\partial y''_j} & \frac{\partial n''_x}{\partial z''_j} \\ \frac{\partial n''_y}{\partial x''_i} & \frac{\partial n''_y}{\partial y''_i} & \frac{\partial n''_y}{\partial z''_i} & \frac{\partial n''_y}{\partial x''_j} & \frac{\partial n''_y}{\partial y''_j} & \frac{\partial n''_y}{\partial z''_j} \\ \frac{\partial n''_z}{\partial x''_i} & \frac{\partial n''_z}{\partial y''_i} & \frac{\partial n''_z}{\partial z''_i} & \frac{\partial n''_z}{\partial x''_j} & \frac{\partial n''_z}{\partial y''_j} & \frac{\partial n''_z}{\partial z''_j} \end{bmatrix} \tag{14}$$

2.4 Non collinear vector and its derivative

This non collinear vector to the normal will be used, together with the normal, for the calculation of the tangent vector to the surface, which in turn allows for the calculation of the binormal vector. As such, it is assumed that:

$$\boldsymbol{u''} = \boldsymbol{n''} + \hat{\boldsymbol{u}}'' \tag{15}$$

where $\hat{\boldsymbol{u}}''$ is defined as:

$$\hat{\boldsymbol{u}}'' = \begin{bmatrix} \pm 1 \\ 1 \\ 0 \end{bmatrix} \cdot ||\boldsymbol{n''}|| \tag{16}$$

being the signal of the first component related with the quadrant in which the normal vector is (regarding (x'', y'')), in such a way that the vectors $\boldsymbol{u''}$ and $\boldsymbol{n''}$ are always in different quadrants:

- When $\boldsymbol{n''}$ is in even quadrants the signal is positive.
- When $\boldsymbol{n''}$ is in odd quadrants the signal is negative.
- When $\boldsymbol{n''}$ is collinear with one of the axis or zero in the plane $(x''Oy'')$ both signals ensure a non collinear vector. As such, and because there has to be a rule, a positive signal is assumed in these circumstances.

The calculation of the derivative of vector $\boldsymbol{u''}$ is also required, which by definition is the same as it was for vector $\boldsymbol{n''}$. This derivative will be used later on for the calculation of the derivatives of the tangential and binormal vectors.
Accordingly, it can be said that:

$$\boldsymbol{u''_{q''}} = \begin{bmatrix} \frac{\partial u''_x}{\partial x''_i} & \frac{\partial u''_x}{\partial y''_i} & \frac{\partial u''_x}{\partial z''_i} & \frac{\partial u''_x}{\partial x''_j} & \frac{\partial u''_x}{\partial y''_j} & \frac{\partial u''_x}{\partial z''_j} \\ \frac{\partial u''_y}{\partial x''_i} & \frac{\partial u''_y}{\partial y''_i} & \frac{\partial u''_y}{\partial z''_i} & \frac{\partial u''_y}{\partial x''_j} & \frac{\partial u''_y}{\partial y''_j} & \frac{\partial u''_y}{\partial z''_j} \\ \frac{\partial u''_z}{\partial x''_i} & \frac{\partial u''_z}{\partial y''_i} & \frac{\partial u''_z}{\partial z''_i} & \frac{\partial u''_z}{\partial x''_j} & \frac{\partial u''_z}{\partial y''_j} & \frac{\partial u''_z}{\partial z''_j} \end{bmatrix} \tag{17}$$

which based on Eq. 15 leads to:

$$\boldsymbol{u''_{q''}} = \boldsymbol{n''_{q''}} + (\hat{\boldsymbol{u}}'' \cdot ||\boldsymbol{n''}||)_{\boldsymbol{q''}} = \boldsymbol{n''_{q''}} + \begin{bmatrix} \pm\frac{\partial ||\boldsymbol{n''}||}{\partial x''_i} & \pm\frac{\partial ||\boldsymbol{n''}||}{\partial y''_i} & \pm\frac{\partial ||\boldsymbol{n''}||}{\partial z''_i} & \pm\frac{\partial ||\boldsymbol{n''}||}{\partial x''_j} & \pm\frac{\partial ||\boldsymbol{n''}||}{\partial y''_j} & \pm\frac{\partial ||\boldsymbol{n''}||}{\partial z''_j} \\ \frac{\partial ||\boldsymbol{n''}||}{\partial x''_i} & \frac{\partial ||\boldsymbol{n''}||}{\partial y''_i} & \frac{\partial ||\boldsymbol{n''}||}{\partial z''_i} & \frac{\partial ||\boldsymbol{n''}||}{\partial x''_j} & \frac{\partial ||\boldsymbol{n''}||}{\partial y''_j} & \frac{\partial ||\boldsymbol{n''}||}{\partial z''_j} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

The first term ($\boldsymbol{n''_{q'}}$) it is already calculated (Eq. 14). There is, however, the need of calculating the derivatives of the norm of the normal with regard to each coordinate.

In a generic way:

$$||\boldsymbol{n''}|| = \sqrt{{n''_{x''}}^2 + {n''_{y''}}^2 + {n''_{z''}}^2} \tag{19}$$

If we consider:

$$w = {n''_{x''}}^2 + {n''_{y''}}^2 + {n''_{z''}}^2 \tag{20}$$

we can say that:

$$||\boldsymbol{n''}|| = \sqrt{w} \tag{21}$$

and the derivative of the norm can be calculated using the derivative of a power function. For example, specifying for the x'' coordinate:

$$\frac{\partial ||\boldsymbol{n''}||}{\partial x''} = \frac{\partial}{\partial x''}(\sqrt{w}) = \frac{1}{2}\frac{\partial w}{\partial x''}\frac{1}{\sqrt{u''}} \tag{22}$$

which, after simplification, will lead to:

$$\frac{\partial ||\boldsymbol{n''}||}{\partial x''} = \frac{\partial n''_x}{\partial x''}\frac{n''_x}{||\boldsymbol{n''}||} \tag{23}$$

that only depends on previously calculated terms for the normal and respective derivative. As such, and without losing generality,we have:

$$\boldsymbol{u''_{q''}} = \boldsymbol{n''_{q''}} + \begin{bmatrix} \pm\frac{\partial n''_x}{\partial x''_i}\frac{n''_x}{||\boldsymbol{n''}||} & \pm\frac{\partial n''_y}{\partial y''_i}\frac{n''_y}{||\boldsymbol{n''}||} & \pm\frac{\partial n''_z}{\partial z''_i}\frac{n''_z}{||\boldsymbol{n''}||} & \pm\frac{\partial n''_x}{\partial x''_j}\frac{n''_x}{||\boldsymbol{n''}||} & \pm\frac{\partial n''_y}{\partial y''_j}\frac{n''_y}{||\boldsymbol{n''}||} & \pm\frac{\partial n''_z}{\partial z''_j}\frac{n''_z}{||\boldsymbol{n''}||} \\ \frac{\partial n''_x}{\partial x''_i}\frac{n''_x}{||\boldsymbol{n''}||} & \frac{\partial n''_y}{\partial y''_i}\frac{n''_y}{||\boldsymbol{n''}||} & \frac{\partial n''_z}{\partial z''_i}\frac{n''_z}{||\boldsymbol{n''}||} & \frac{\partial n''_x}{\partial x''_j}\frac{n''_x}{||\boldsymbol{n''}||} & \frac{\partial n''_y}{\partial y''_j}\frac{n''_y}{||\boldsymbol{n''}||} & \frac{\partial n''_z}{\partial z''_j}\frac{n''_z}{||\boldsymbol{n''}||} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

2.5 Tangential and Binormal vectors and respective derivatives

Once obtained the vector $\boldsymbol{u''}$ the tangential and binormal vectors can be calculated as:

$$\boldsymbol{t''} = \boldsymbol{n''} \times \boldsymbol{u''} \tag{25}$$

$$\boldsymbol{b''} = \boldsymbol{n''} \times \boldsymbol{t''} \tag{26}$$

The respective derivatives can be obtained by:

$$\boldsymbol{t''_q} = \tilde{\boldsymbol{n}}'' \cdot \boldsymbol{u''_q} - \tilde{\boldsymbol{u}}'' \cdot \boldsymbol{n''_q} \tag{27}$$

$$\boldsymbol{b''_q} = \tilde{\boldsymbol{n}}'' \cdot \tilde{\boldsymbol{n}}'' \cdot \boldsymbol{u''_q} - \tilde{\boldsymbol{n}}'' \cdot \tilde{\boldsymbol{u}}'' \cdot \boldsymbol{n''_q} - \boldsymbol{n''_q} \cdot (\tilde{\boldsymbol{n}''} \cdot \tilde{\boldsymbol{u}''} - \tilde{\boldsymbol{u}''} \cdot \tilde{\boldsymbol{n}''}) \tag{28}$$

where $\tilde{\boldsymbol{n}}''$ e $\tilde{\boldsymbol{u}}''$ represent the skew-symmetric matixes of $\boldsymbol{n''}$ e $\boldsymbol{u''}$, respectively.

2.6 Distance vector and derivative

The distance vector $\boldsymbol{d}_{ij}$ is calculated, resorting to the global coordinates of the candidate contact points $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$:

$$\boldsymbol{d}_{ij} = \boldsymbol{s}_j - \boldsymbol{s}_i \tag{29}$$

where $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ can be defined as:

$$\boldsymbol{s}_i = A_{bi}(A_{si}\boldsymbol{s}_i'' + \boldsymbol{r}'_{si}) + \boldsymbol{r}_{bi} \tag{30}$$

$$\boldsymbol{s}_i = A_{bj}(A_{sj}\boldsymbol{s}_j'' + \boldsymbol{r}'_{sj}) + \boldsymbol{r}_{bj} \tag{31}$$

where the terms A_{bi} and A_{bj} represent the rotation matrixes that transform the local reference frame of the bodies associated with the surfaces being tested for contact into the global reference frame, A_{si} and A_{sj} represent the rotation matrixes of the that transform the surface reference frame of the surfaces being tested for contact into the body local reference frame, $\boldsymbol{r}'_{si}$ and $\boldsymbol{r}'_{si}$ are the position of the surface in the locar reference frame of each body and $\boldsymbol{r}_{bi}$ and $\boldsymbol{r}_{bi}$ the position of the bodies center of mass in the global reference frame.
As only the terms $\boldsymbol{s}_i''$ and $\boldsymbol{s}_j''$ are dependent of the coordinates vector, the derivative of the distance vector can be calculated as:

$$(\boldsymbol{d}_{ij})_{\boldsymbol{q}''} = A_j(\boldsymbol{s}_j'')_{\boldsymbol{q}''} - A_i(\boldsymbol{s}_i'')_{\boldsymbol{q}''} \tag{32}$$

where A_i is the product of A_{bi} with A_{si} and A_j is the product of A_{bj} with A_{sj}.
Considering:

$$\boldsymbol{s}_i'' = \begin{bmatrix} x_i'' & y_i'' & z_i'' \end{bmatrix}^T \tag{33}$$

$$\boldsymbol{s}_j'' = \begin{bmatrix} x_j'' & y_j'' & z_j'' \end{bmatrix}^T \tag{34}$$

$(\boldsymbol{s}_i'')_{\boldsymbol{q}''}$ and $(\boldsymbol{s}_j'')_{\boldsymbol{q}''}$ can be easily calculated (keeping in mind the definition of $\boldsymbol{q}''$ in Eq. 1):

$$(\boldsymbol{s}_i'')_{\boldsymbol{q}''} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{35}$$

$$(\boldsymbol{s}_j'')_{\boldsymbol{q}''} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{36}$$

Joining the results of Eq. 32 with equations 35 and 36 one can see that the derivative of the distance vector is constant and does not depend on the pair of points considered:

$$(\boldsymbol{d}_{ij})_{\boldsymbol{q}''} = \begin{bmatrix} -A_i & A_j \end{bmatrix} \tag{37}$$

2.7 Contact Detection

Once the minimum distance is calculated, it is necessary to determine if there is contact or not. For that we use the value of pseudo-penetration, which can be defined as[1]:

$$\delta = \boldsymbol{n}_j \cdot \boldsymbol{d}_{ij} \tag{38}$$

taking into account that $\boldsymbol{n}_i$ should be a normalized vector. For values of δ higher than zero we consider that there is contact.
Once contact is detected, the pseudo-penetration velocity $\dot{\delta}$ needs to be calculated in order to calculate the reaction force. This velocity can be defined as:

$$\dot{\delta} = \frac{d\delta}{dt} = \dot{\boldsymbol{d}}_{ij} \cdot \boldsymbol{n}_j \tag{39}$$

2.8 Contact Force Computation

In this work we applied two different force models to our test cases: one purely elastic and other that takes into account some energy dissipation during contact. The force generated by Hertz contact law can be calculated by:

$$F_C = K\delta^n \tag{40}$$

where K is the generalized stiffness parameter of the materials and the exponent n is the term of non-linearity (if$n$ equal to 1 it is said to be Hookean, if n equal to $3/2$ it is said to be Hertzian).
For models with the dissipative term, the force can be calculated by:

$$F_C = K\delta^n + D\dot{\delta} = K\delta^n + \mathcal{X}\delta^n\dot{\delta} \tag{41}$$

For the model used in this work, the value of $\mathcal{X}$ can be determined by:

$$\mathcal{X} = \frac{8K(1-c_r)}{5c_r\dot{\delta^{(-)}}} \tag{42}$$

where $\dot{\delta^{(-)}}$ is the pseudo-penetration velocity immediately before contact.

2.9 Friction Force Computation

For the determination of the friction force the intensity of the contact force is necessary but is not sufficient. It is also required to know the direction and intensity of the velocity tangent to the contact event. This velocity is calculated by having the relative velocity

between the points involved in the contact event and the respective pseudo penetration velocity:

$$\boldsymbol{v}_t = \dot{\boldsymbol{d}}_{ij} - \dot{\delta}\boldsymbol{n}_i \tag{43}$$

where $\boldsymbol{v}_t$ is the tangent velocity and $\dot{\boldsymbol{d}}_{ij}$ the relative velocity between bodies.
The intensity of the friction force can be determined as:

$$F_A = \mu_1 F_C + \mu_2 ||\boldsymbol{v}_t|| \tag{44}$$

where μ_1 is the Coulomb friction coefficient and μ_2 is the tangent velocity friction coefficient. Once known the value of F_A and the direction of $\boldsymbol{v}_t$ all that remains is to calculate the corresponding vector:

$$\boldsymbol{F}_A = F_A \hat{\boldsymbol{v}}_t \tag{45}$$

where $\hat{\boldsymbol{v}}_t$ represents the tangent velocity direction.

2.10 Application Point

In order to simulate the deformation in the bodies involved in the contact event, the forces application points were not the points considered for contact, but an intermediate point, which was calculated based on a ratio of the stiffness of each body:

$$\boldsymbol{r}_{app} = \boldsymbol{s}_j + \delta \frac{E_i}{E_i + E_j} \boldsymbol{n}_j \tag{46}$$

From the previous equation results that if the stiffness of body i is much higher than the stiffness of body j, the application point will be closer the contact point of the body i (which implies that the major deformation would be from body j) and vice-versa. In the case of both bodies having the same stiffness, the application point would be the point between the contact points determined for i and j.

3 RESULTS

Three simple tests were made in order to evaluate the results of the implemented models. The first was a simple bouncing ball test, with different contact models, in order to compare with existing results. The second was a Newton cradle test with 2 pendulums. This test was carried to test the contact force model, as well as the detection algorithm. The third test was designed to test the friction model implemented and consisted in an ellipsoid rolling down an inclined plane.

3.1 Test 1 - Bouncing Ball

In the bouncing ball test, the force model described above was compared with the models proposed by Hunt and Crossley and by Lankarani and Nikravesh [15]. Various simulations were made, all with different coefficients of restitution, in order to compare the coefficients of pre- and post-restitution.

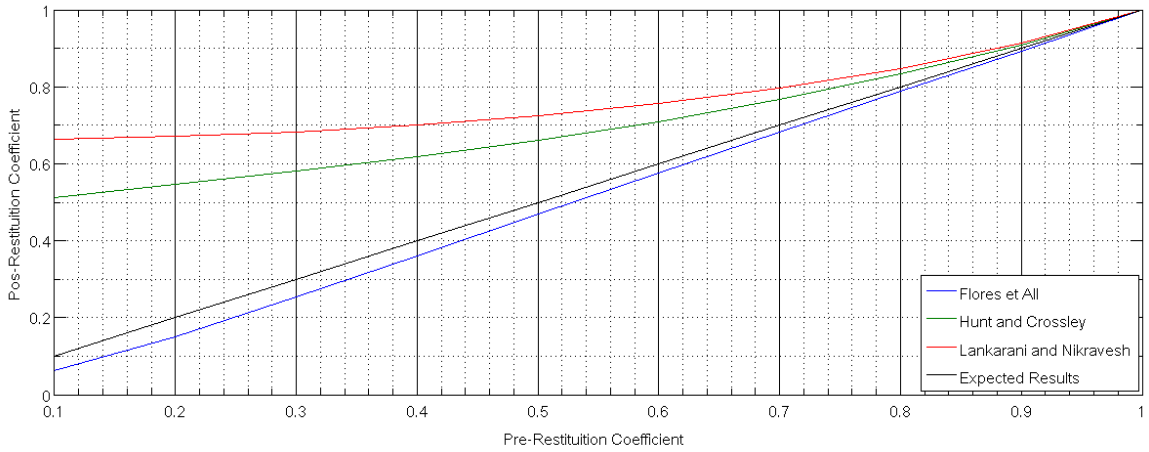

Figure 2: *Variation of the coefficient of post-restitution with the coefficient of pre-restitution*

The results presented in figure 2 are in accordance with what was expected for the models considered, which implies that the detection method and the contact models are correctly implemented.

3.2 Test 2 - Newton Cradle

Figures 3 to 6 show the behaviour of the Newton cradle tests with different coefficients of restitution. As expected, when the coefficient of restitution is 1 (the force model becomes the Hertz model) there is no loss of energy (figures 3 and 5). It is also noteworthy that all the energy of the impact is transmitted to the pendulum in rest, which means that the pendulum in rest achieves the height at which the first pendulum was released off.
However, when we reduce the coefficient of restitution to 0.5 (figures 4 and 6) the results are different as expected. Since the transfer of energy is not total, the first pendulum does not remain in rest when the impact occurs. For the same reason, the maximum height achieved by the second pendulum is not the height at which the first pendulum was released. It is also observed that after a given number of impacts (around eight) both

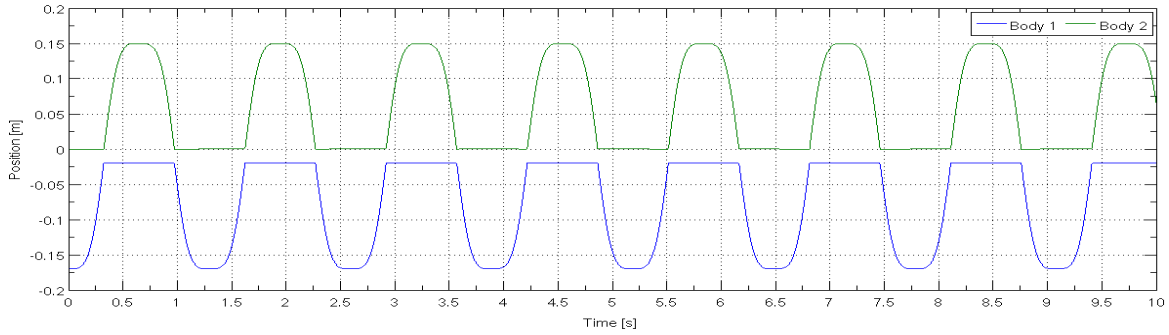

Figure 3: *Pendulum position in the yy axis with coefficient of restitution 1*

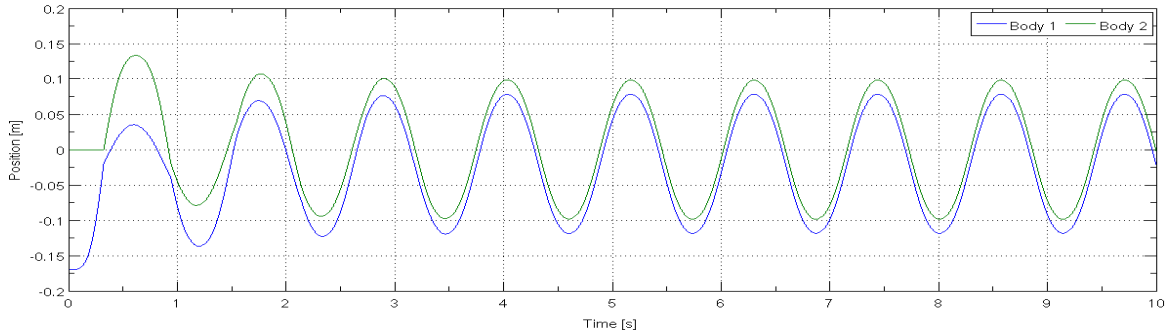

Figure 4: *Pendulum position in the yy axis with coefficient of restitution 1*

pendulums get synchronised. This implies that though they still remain together there is no more energy transfer or loss.
In terms of force and pseudo-penetration behaviour, figure 7 shows the results for both cases. It is easy to see the energy loss when the coefficient of restitution is 0.5 by the curves obtained.

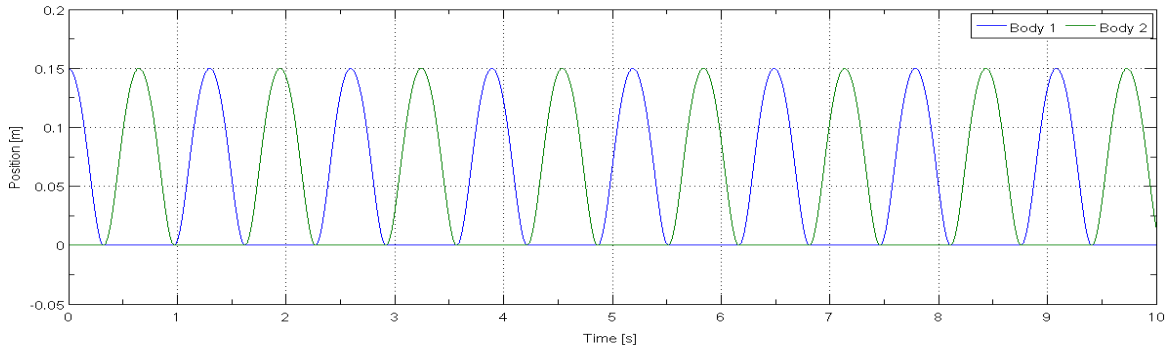

Figure 5: *Pendulum position in the zz axis with coefficient of restitution 0.5*

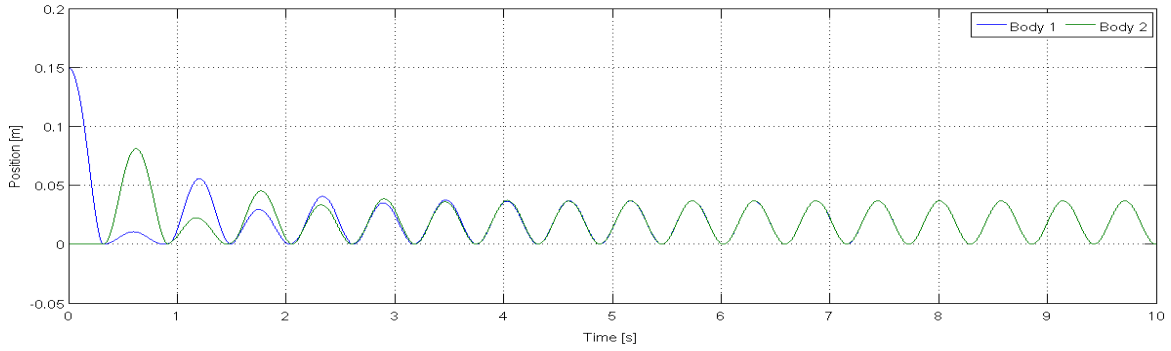

Figure 6: *Pendulum position in the zz axis with coefficient of restitution 0.5*

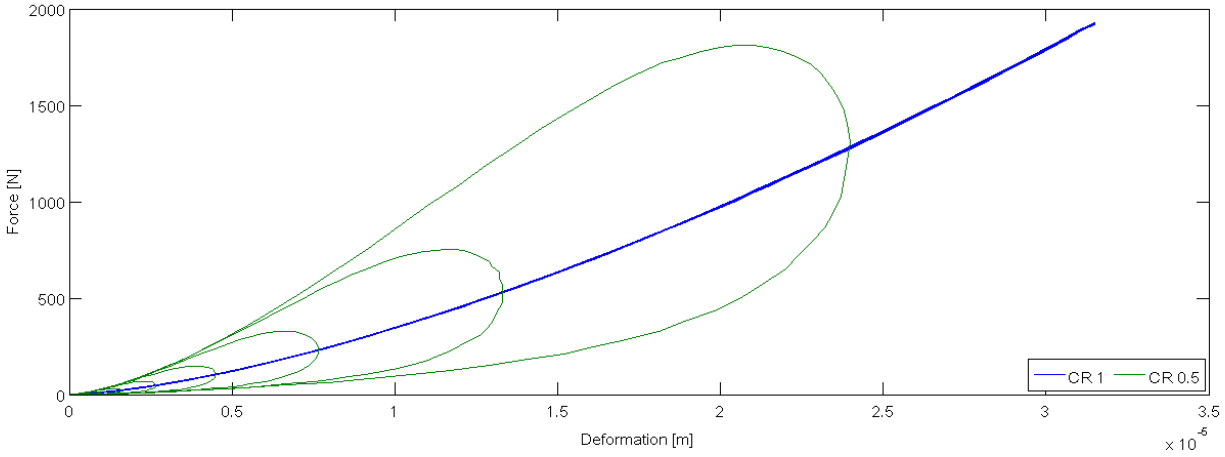

Figure 7: *Force variation with pseudo-penetration for* $CR = 1$ *and* $CR = 0.5$

3.3 Test 3 - Ellipsoid Rolling Down a Plane

In a round body in an inclined plane, the existence of friction is responsible for the rotation of the body when going down the plane. If there was no friction, the body would simply slide down the plane. The friction force is applied in the contact point, reducing greatly its velocity. The mass center, however, remains moving at the same velocity, which leads to the rotation of the body. In this specific case, as the body is an ellipsoid which axis (a, b and c) are all different from each other and this body is rotated along all the axis, the rotation leads to the mass center of the body not going down in a line parallel to the plane. That will eventually make the body lose contact with the plane, as its speed increases. In figure 8 the above described is illustrated through images of the ellipsoid simulation in different time steps.

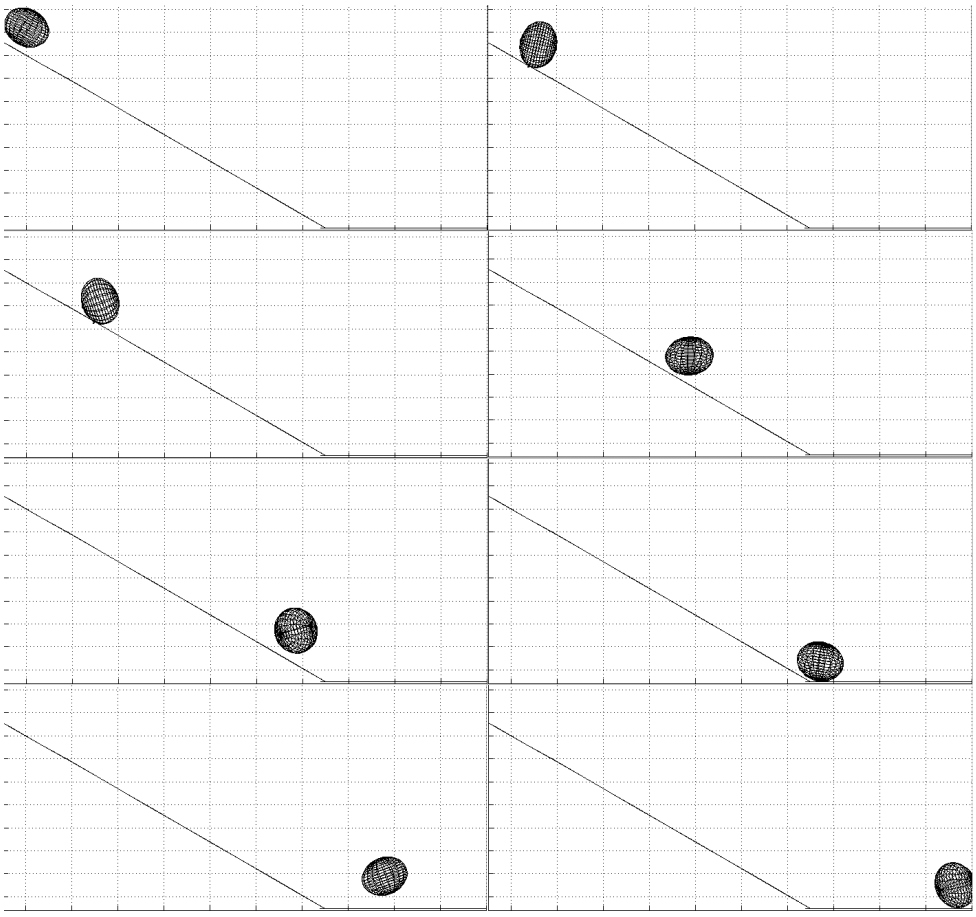

Figure 8: *Ellipsoid Rolling down a plane in different time steps*

CONCLUSIONS

In this work, an all purpose contact detection method was proposed and implemented along a contact force model that works not only with high restitution coefficients but with low as well. Additionally, a very simple friction model was implemented. The contact detection model was correctly implemented and showed very promising results in terms of efficiency and precision in detecting the right candidate points for contact. The force model used showed very consistent when compared with the literature and produced results in accordance with the physical phenomena. The energy transfer observed for different restitution coefficients was according to the expected. The addition of the friction model also generated good results. All the simulations run with these force model had the expected results.
The next steps of this work will be the application of the above described method and model to a biomechanical foot model in order to obtain a good estimation of foot-ground contact forces in the human gait. It is also considered the usage of this model for forward dynamic simulations.

REFERENCES

[1] Daniel S Lopes, Miguel T Silva, Jorge A Ambrósio, and Paulo Flores. A mathematical framework for rigid contact detection between quadric and superquadric surfaces. *Multibody System Dynamics*, 24(3):255–280, 2010.

[2] Sinan Kockara, Tansel Halic, Coskun Bayrak, K Iqbal, and RA Rowe. Contact detection algorithms. *Journal of Computers*, 4(10):1053–1063, 2009.

[3] Juhwan Choi, Han Sik Ryu, Chang Wan Kim, and Jin Hwan Choi. An efficient and robust contact algorithm for a compliant contact force model between bodies of complex geometry. *Multibody System Dynamics*, 23(1):99–120, 2010.

[4] H Cenk Güler, Necip Berme, and Sheldon R Simon. A viscoelastic sphere model for the representation of plantar soft tissue during simulations. *Journal of biomechanics*, 31(9):847–853, 1998.

[5] Brian Mirtich. Rigid body contact: Collision detection to force computation. In *IEEE International Conference on Robotics and Automation*, 1998.

[6] Pedro Moreira, Paulo Flores, and Miguel Silva. A biomechanical multibody foot model for forward dynamic analysis. In *Bioengineering (ENBENG), 2012 IEEE 2nd Portuguese Meeting in*, pages 1–6. IEEE, 2012.

[7] Ming C Lin et al. Collision detection: Algorithms and applications. 1996.

[8] Gerhard Hippmann. An algorithm for compliant contact between complexly shaped surfaces in multibody dynamics. *Multibody Dynamics, Jorge AC Ambrosio (Ed.), IDMEC/IST, Lisbon, Portugal, July*, 14, 2003.

[9] Yi-King Choi, Jung-Woo Chang, Wenping Wang, Myung-Soo Kim, and Gershon Elber. Continuous collision detection for ellipsoids. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):311–325, 2009.

[10] Xiaohong Jia, Yi-King Choi, Bernard Mourrain, and Wenping Wang. An algebraic approach to continuous collision detection for ellipsoids. *Computer Aided Geometric Design*, 28(3):164–176, 2011.

[11] Christian Wellmann, Claudia Lillie, and Peter Wriggers. A contact detection algorithm for superellipsoids based on the common-normal concept. *Engineering Computations*, 25(5):432–442, 2008.

[12] Daniel Simões Lopes. *Smooth convex surfaces for modeling and simulating multibody systems with compliant contact elements.* PhD thesis, INSTITUTO SUPERIOR TÉCNICO, 2013.

[13] Janete Alves, Nuno Peixinho, Miguel Tavares da Silva, Paulo Flores, and Hamid M Lankarani. A comparative study of the viscoelastic constitutive models for frictionless contact interfaces in solids. *Mechanism and Machine Theory*, 85:172–188, 2015.

[14] Juhwan Choi, Sungsoo Rhim, and Jin Hwan Choi. A general purpose contact algorithm using a compliance contact force model for rigid and flexible bodies of complex geometry. *International Journal of Non-Linear Mechanics*, 53:13–23, 2013.

[15] Paulo Flores, Margarida Machado, Miguel T Silva, and Jorge M Martins. On the continuous contact force models for soft materials in multibody dynamics. *Multibody System Dynamics*, 25(3):357–375, 2011.

[16] Kenneth R Gratz and Robert L Sah. Experimental measurement and quantification of frictional contact between biological surfaces experiencing large deformation and slip. *Journal of biomechanics*, 41(6):1333–1340, 2008.

[17] Yves Gonthier, John McPhee, Christian Lange, and Jean-Claude Piedboeuf. A regularized contact model with asymmetric damping and dwell-time dependent friction. *Multibody System Dynamics*, 11(3):209–233, 2004.

[18] Pedro Moreira. Development of a three-dimensional contact model for the foot-ground interaction in gait simulations, 2009.