A general contact algorithm for multibody system dynamics with complex non-conforming 3D geometry

D. Dopico* A. Luaces† U.Lugris‡ J.Cuadrado§
Universidad de A Coruña
Ferrol, Spain

Abstract— *Contact plays a key role in a large number of multibody system dynamics applications. The treatment of contact is still an open problem and a challenging topic for the multibody system dynamics community. There are two different aspects that must be taken into consideration to successfully simulate the behavior of multibody systems with contacting bodies. First, the contact model chosen has to be adequate for the applications to tackle. Second, independently of the contact model chosen, the geometrical detection of the contact events is necessary to feed the contact model.*

In this work, a general contact algorithm for rigid bodies is described, which includes continuous contact force model and contact detection between complex 3D geometries given by CAD models.

Keywords: Contact, detection, multibody dynamics.

I. Introduction

The multibody system dynamics methods, constitute general formalisms to write and solve the equations of motion of any mechanism or machine. There are a wide variety of formalisms that lead to different equations of motion and solution methods, see e.g. [1], but independently of the formalisms chosen, the forces and/or constraints of the system are always inputs to the problem.

A large number of industrial applications of multibody system dynamics require proper contact algorithms since contacts between bodies of the system (or between bodies and the environment), govern the dynamics of the system. To take into account the contact in the equations of motion is not so simple and can be divided in two different aspects: first, the contact model chosen has to be accurate, stable and robust enough for the applications to tackle; second, independently of the contact model chosen, the geometrical detection of the contact events is a previous condition for any contact model.

In a broad sense there are two families of methods to solve the normal contact problem in multibody systems composed of rigid bodies, see [2], [3]: the discontinuous and the continuous approaches. The discontinuous approaches are impulsive and for this reason suited to impact forces, while the continuous approaches are better suited to applications in which it is expected to occur permanent contacts or at least contacts of a significant duration; the normal force model described in this work is continuous and based on regularized forces. Moreover, it is worth to mention that, up to these days, there is not a universally accepted model to calculate the friction force between bodies under dry conditions. In this work, a tangential friction model developed by the authors in a previous work is described [4]: the model includes dry friction, sticktion at low velocities and a viscous component.

Independently of the contact model chosen, it is necessary to detect the contacts which turns to be a geometrical problem, see e.g. [5], [6]. In many cases, it is possible to make some assumptions, for example when the exact geometry of the bodies are known, it is possible to simplify the task of detecting the contacts between bodies by using the analytical equations of the geometry or by replacing the actual one by surrounding primitives like spheres or boxes. Nevertheless, when the exact geometries of the bodies are not known (because they are read from CAD models) or when the geometries are very complex, more general strategies have to be developed in order to detect the contacts and carry out all the calculations necessary to feed the contact model. The algorithms described in this work are able to detect contacts between bodies with complex 3D geometries given by CAD models in triangular mesh format.

Between the large number of formulations of the equations of motion existent (see e.g. [1]), the penalty and augmented Lagrangian formulations, [7] [8], are characterized by transforming the constraints into forces proportional to the constraints violation. This technique, used along this work, is similar and compatible to that of the continuous force models for normal contact,which relate the force and deformation of the bodies in contact to avoid the penetration between them.

II. Multibody formulation

The multibody formulation chosen for this work is an index-3 augmented Lagrangian with projections of velocities and accelerations onto the constraints manifolds. As integration scheme, the implicit single-step trapezoidal rule has been adopted. The mentioned formulation was extensively described in Cuadrado (2001 and 2004).

*ddopico@udc.es
†aluaces@udc.es
‡ulugris@udc.es
§javicuad@cdf.udc.es

III. Description of the contact model

The contact forces approach proposed for this work comprises two different models: the normal force model and the tangential force model. The two sub-models are presented separately in subsequent sections. A more detailed description of the contact model is given in [4].

For simplicity reasons, the contact model is going to be explained for the collision of a spherical and a flat body but it is easily generalized to bodies with arbitrary shapes.

A. Normal force model

The normal force model chosen for this work was the Hunt-Crossley model [9]. The model is suited to collisions between massive solids for which the assumption of quasi static contact holds and it can be supposed that the deformation is limited to a small region of the colliding bodies while the remainder of them are assumed to be rigid. The expression for the normal force, after some calculations, has the following form,

$$\mathbf{F}_n = k_n\,\delta^e\left(1+\frac{3\,(1-\epsilon)}{2}\,\frac{\dot{\delta}}{\dot{\delta}_0}\right)\mathbf{n} \tag{1}$$

where k_n is the equivalent stiffness of the contact and depends on the shape and material properties of the colliding bodies, e is the Hertz's exponent, $\delta = R_{sph} - |\mathbf{p}_{center} - \mathbf{p}_{contact}|$ is the indentation, $\dot{\delta}$ its temporal derivative, $\dot{\delta}_0$ is the relative normal velocity between the colliding bodies when the contact is detected, ϵ is the coefficient of restitution, and $\mathbf{n}$ is the direction of the force (see Fig.(1)). The subscript "*n*" comes from "*normal*".

The value of k_n depends on the shape and materials of the colliding bodies.

In figure 1 a collision between a spherical and a flat body is represented.

B. Tangential force model

The tangential force model developed for the friction force is based on Coulomb's law including sticktion. Moreover a viscous term is added to the dry friction force. The general form of this force is the following,

$$\mathbf{F}_t = \kappa\,\mathbf{F}_{stick} + (1-\kappa)\,\mathbf{F}_{slide} - \mu_{visc}\mathbf{v}_t \tag{2}$$

In the previous expression, the first two terms constitute the dry friction, while the third term accounts for the viscous friction. For the smooth transition between sticking and slipping the dry friction force is divided in two components coupled by a smooth function, following the ideas proposed in [10]. The subscript "*t*" comes from "*tangential*".

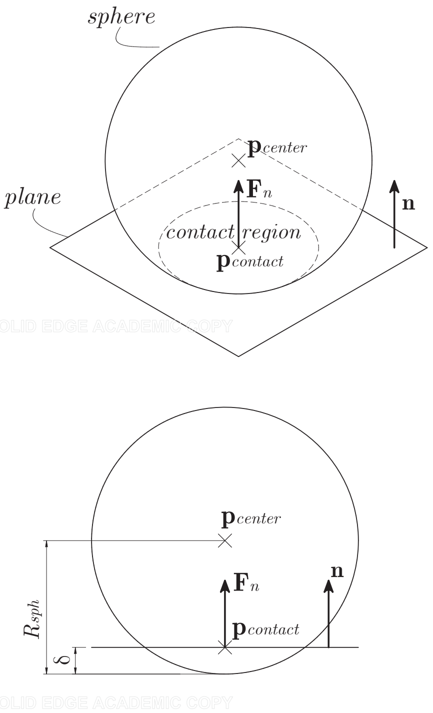

Fig. 1. Normal contact between sphere and plane: isometric and front views.

In (2), μ_{visc} is the viscous damping coefficient, $\mathbf{F}_{stick}$ and $\mathbf{F}_{slide}$ are the components of the sticktion and slipping forces, κ is a smooth function of the tangential velocity, $\mathbf{v}_t$, which is defined in terms of the central point of the contact region, $\mathbf{p}_{contact}$, and the normal vector at the contact,$\mathbf{n}$, as follows.

$$\mathbf{v}_t = \dot{\mathbf{p}}_{contact} - \left(\mathbf{n}^{\mathrm{T}}\,\dot{\mathbf{p}}_{contact}\right)\mathbf{n} \tag{3}$$

The mentioned function, κ, has to match the following conditions,

$$\kappa = \left\{\begin{array}{ll} 0; & |\mathbf{v}_t| >> v_{stick} \\ 1; & |\mathbf{v}_t| = 0 \end{array}\right\} \tag{4}$$

where v_{stick} is a parameter of the model accounting for the velocity of the stick-slip transition. A good choice for the transition function κ was given in [10] and has the following form.

$$\kappa = \mathrm{e}^{-\left(\mathbf{v}_t^{\mathrm{T}}\mathbf{v}_t\right)/v_{stick}^2} \tag{5}$$

Equation (2) showed that the total force is composed of three contributions: the sliding dry friction force at high velocities, the sticktion force at low velocities and the viscous friction force. The sliding force is given by the classical Coulomb expression, while the sticktion force is considered by means of viscoelastic elements acting between the colliding bodies. To see the detailed expressions of the sliding and sticktion forces see [4].

IV. Contact detection algorithms

The algorithms described in this section perform the detection of the existent contacts between bodies. Moreover,

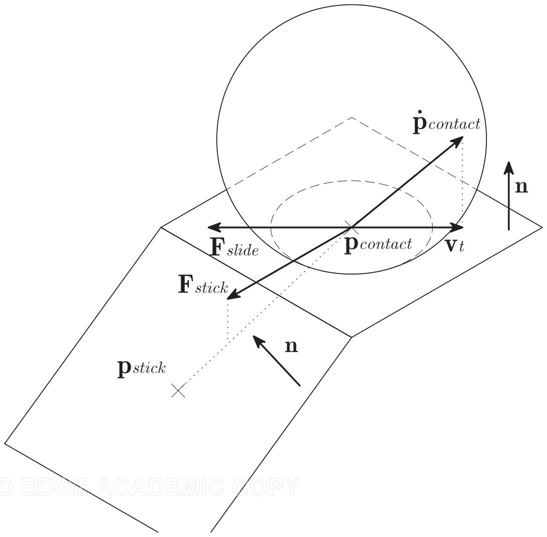

Fig. 2. Tangential contact between sphere and plane.

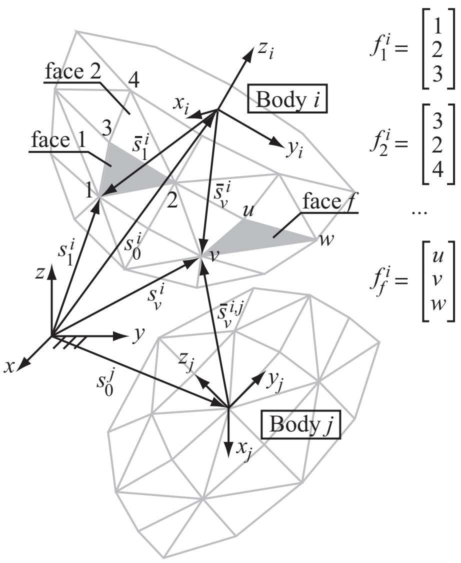

Fig. 3. Body mesh.

for each one of the contacts found, they perform the calculation of the necessary data to feed the contact model described in section III. The bodies have complex 3D geometries given by CAD models in triangular mesh format. All the existent CAD packages have translators from the native format to triangular mesh formats like *obj* or *stl*.

The mathematical representation of the triangular mesh of the body *i*, with n_v vertexes and n_f triangular faces, expressed in the local framework of the body, is the following (see figure 3).

$$\bar{\mathbf{s}}_v^i;\ 1 \leq v \leq n_v;\ n_v \geq 3 \text{ (list of vertexes)} \qquad (6)$$

$$\mathbf{f}_f^i = \begin{bmatrix} f_{f1}^i \\ f_{f2}^i \\ f_{f3}^i \end{bmatrix};\ 1 \leq f \leq n_f;\ n_f \geq 1 \text{ (list of triangles)} \qquad (7)$$

Thus, the three vertexes that compose the face *f* of the body are obtained by replacing each one of the indexes from (7) in (6): $\bar{\mathbf{s}}^i_{f^i_{f1}}$, $\bar{\mathbf{s}}^i_{f^i_{f2}}$ and $\bar{\mathbf{s}}^i_{f^i_{f3}}$.

The vertexes list of the mesh in global coordinates is obtained by means of the body transformation matrix $\mathbf{A}^{\,i}$, using homogeneous coordinates.

$$\mathbf{s}_v^{*i} = \mathbf{A}^i \bar{\mathbf{s}}_v^{*i} \Rightarrow \begin{bmatrix} \mathbf{s}_v^i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^i & \mathbf{s}_0^i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{s}}_v^i \\ 1 \end{bmatrix} \qquad (8)$$

It will be necessary, for some of the algorithms of this section, to express the vertexes of a body in the local frame of other body. Thus, the vertexes of body *i* in the local frame of body *j*.

$$\bar{\mathbf{s}}_v^{*i,j} = \left(\mathbf{A}^j\right)^{-1} \mathbf{s}_v^{*i} = \left(\mathbf{A}^j\right)^{-1} \mathbf{A}^i \bar{\mathbf{s}}_v^{*i} = \mathbf{A}^{i,j} \bar{\mathbf{s}}_v^{*i} \qquad (9)$$

where,

$$\left(\mathbf{A}^j\right)^{-1} = \begin{bmatrix} \left(\mathbf{R}^j\right)^{\mathrm{T}} & -\left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{s}_0^j \\ 0 & 1 \end{bmatrix} \qquad (10)$$

$$\mathbf{A}^{i,j} = \begin{bmatrix} \left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{R}^i & \left(\mathbf{R}^j\right)^{\mathrm{T}} \left(\mathbf{s}_0^i - \mathbf{s}_0^j\right) \\ 0 & 1 \end{bmatrix} \qquad (11)$$

Replacing equation (11) in equation (9).

$$\bar{\mathbf{s}}_v^{i,j} = \left(\mathbf{R}^j\right)^{\mathrm{T}} \left(\mathbf{s}_0^i - \mathbf{s}_0^j\right) + \left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{R}^i \bar{\mathbf{s}}_v^i \qquad (12)$$

A. CAD models preprocessing

This task is performed only once at the beginning of the simulation and carries out the preprocessing of the CAD models, which includes basically the calculation of the neighbors of each face and the creation of the data structures to calculate the pairs of triangles in collision. As was mentioned before, the geometry of each body, is given by a mesh of triangles.

B. Calculation of the collision pairs

For this task the open source library Opcode (see [**?**]) was partially used. The tree structures to order and subdivide the faces were taken from Opcode, but the algorithms to detect if two triangles collide were completely reprogrammed, since the original algorithms from Opcode offer inaccurate (and sometimes completely incorrect) results, not valid for multibody dynamics. The algorithms described in this section, return a list of n_p disordered colliding triangle pairs between bodies *i* and *j* (see figure 4)

$$\mathbf{pairs}^{ij} = \begin{bmatrix} f_1^i & f_1^j \\ f_2^i & f_2^j \\ \multicolumn{2}{c}{\cdots} \\ f_k^i & f_k^j \\ \multicolumn{2}{c}{\cdots} \\ f_{np}^i & f_{np}^j \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 4 \\ \multicolumn{2}{c}{\cdots} \\ 1 & f \\ \multicolumn{2}{c}{\cdots} \\ 1 & 2 \end{bmatrix}$$

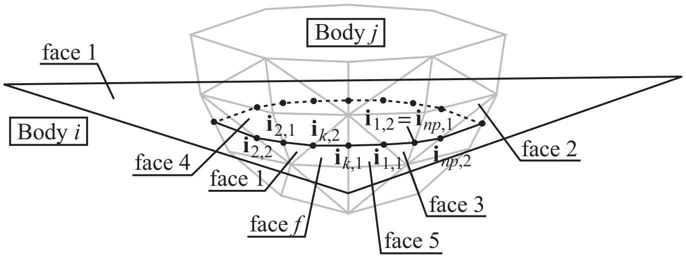

Fig. 4. Collision pairs and segments between bodies *i* and *j*

and a list of n_p disordered intersection segments between the pairs, expressed in the local frame of body *j*.

$$\mathbf{pairs}^{ij} = \left\{ f_k^i, f_k^j \right\};\ 0 \leq k \leq n_p \text{ (list of pairs)} \tag{13}$$

$$\overline{\mathbf{isects}}^{ij,j} = \left\{ \overline{\mathbf{i}_{k,1}\mathbf{i}_{k,2}} \right\};\ 0 \leq k \leq n_p \text{ (list of segments)} \tag{14}$$

In equations (13) and (14) the super index ^{ij}, indicates collision between bodies *i* and *j*, while in equation (14) the over line along with the superindex ^{,j} indicates local coordinates of body *j*.

B.1 Box-Box overlap algorithm

The original Opcode algorithm to test AABB (Axis Aligned Bounding Box) with AABB collisions offered incorrect results for local aligned or almost local aligned boxes. The new algorithm checks this alignment.

B.2 Triangle-Triangle overlap algorithm

The original Opcode algorithm was based on projections while the new programmed algorithm is based on the direct solution of edge-triangle intersections which is much more robust. Moreover the new algorithm was programmed in double precision. The intersection of the triangles is typically a straight line segment, the new algorithm includes also the calculation of the extreme points of the segment.

In figure 5 the triangles p^i (from body *i*) and q^j (from body *j*), are intersecting. The triangles are composed of the vertexes $\mathbf{f}_p^i$ and $\mathbf{f}_q^j$ respectively. To check the intersection between them, it is enough to check each edge of triangle p^i against q^j and vice versa. If the triangles overlap, two edge-triangle intersections exist. To illustrate the edge-triangle test, the intersection between edge $\overline{f_{q1}^j f_{q2}^j}$ and triangle p^i is calculated here.

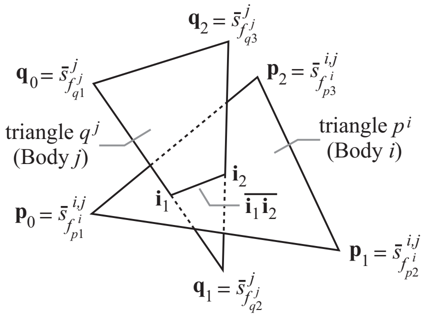
$\mathbf{q}_2 = \bar{s}^j_{f^j_{q3}}$
$\mathbf{q}_0 = \bar{s}^j_{f^j_{q1}}$
$\mathbf{p}_2 = \bar{s}^{i,j}_{f^i_{p3}}$
triangle q^j (Body j)
triangle p^i (Body i)
$\mathbf{i}_2$
$\mathbf{i}_1$
$\overline{\mathbf{i}_1\mathbf{i}_2}$
$\mathbf{p}_0 = \bar{s}^{i,j}_{f^i_{p1}}$
$\mathbf{p}_1 = \bar{s}^{i,j}_{f^i_{p2}}$
$\mathbf{q}_1 = \bar{s}^j_{f^j_{q2}}$

Fig. 5. Triangle-triangle intersection

Let's call $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$ the vertexes of triangle p^i and $\mathbf{q}_0$, $\mathbf{q}_1$, $\mathbf{q}_2$ the vertexes of triangle q^j. Using equation (12), it is possible to express all the vertexes in the local frame of body *j*.

$$\begin{aligned} \mathbf{p}_0 &= \bar{\mathbf{s}}^{i,j}_{f^i_{p1}} = \left(\mathbf{R}^j\right)^{\mathrm{T}} \left(\mathbf{s}^i_0 - \mathbf{s}^j_0\right) + \left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{R}^i \bar{\mathbf{s}}^i_{f^i_{p1}} \\ \mathbf{p}_1 &= \bar{\mathbf{s}}^{i,j}_{f^i_{p2}} = \left(\mathbf{R}^j\right)^{\mathrm{T}} \left(\mathbf{s}^i_0 - \mathbf{s}^j_0\right) + \left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{R}^i \bar{\mathbf{s}}^i_{f^i_{p2}} \\ \mathbf{p}_2 &= \bar{\mathbf{s}}^{i,j}_{f^i_{p3}} = \left(\mathbf{R}^j\right)^{\mathrm{T}} \left(\mathbf{s}^i_0 - \mathbf{s}^j_0\right) + \left(\mathbf{R}^j\right)^{\mathrm{T}} \mathbf{R}^i \bar{\mathbf{s}}^i_{f^i_{p3}} \end{aligned} \tag{15}$$

$$\begin{aligned} \mathbf{q}_0 &= \bar{\mathbf{s}}^j_{f^j_{q1}} \\ \mathbf{q}_1 &= \bar{\mathbf{s}}^j_{f^j_{q2}} \\ \mathbf{q}_2 &= \bar{\mathbf{s}}^j_{f^j_{q3}} \end{aligned} \tag{16}$$

The equations of triangle p^i.

$$\begin{gathered} \mathbf{r}_t = \mathbf{p}_0 + \mu_1 \mathbf{u}_1 + \mu_2 \mathbf{u}_2 \left\{ \begin{array}{l} \dfrac{\mu_1}{l_1} + \dfrac{\mu_2}{l_2} \leq 1 \\ \mu_1 \geq 0 \\ \mu_2 \geq 0 \end{array} \right\} \\ \mathbf{u}_1 = \frac{\mathbf{p}_1 - \mathbf{p}_0}{l_1};\ \mathbf{u}_2 = \frac{\mathbf{p}_2 - \mathbf{p}_0}{l_2} \\ l_1 = \left|\mathbf{p}_1 - \mathbf{p}_0\right|;\ l_2 = \left|\mathbf{p}_2 - \mathbf{p}_0\right| \end{gathered} \tag{17}$$

The equations of edge $\overline{f_{q1}^j f_{q2}^j}$.

$$\begin{gathered} \mathbf{r}_e = \mathbf{q}_0 + \eta \mathbf{v};\ 0 \leq \eta \leq d \\ \mathbf{v} = \frac{\mathbf{q}_1 - \mathbf{q}_0}{d};\ d = \left|\mathbf{q}_1 - \mathbf{q}_0\right| \end{gathered} \tag{18}$$

Making $\mathbf{r}_t = \mathbf{r}_e$.

$$\begin{gathered} \mathbf{p}_0 + \mu_1 \mathbf{u}_1 + \mu_2 \mathbf{u}_2 = \mathbf{q}_0 + \eta \mathbf{v} \Rightarrow \\ \left[-\mathbf{v}\ \mathbf{u}_1\ \mathbf{u}_2\right] \begin{bmatrix} \eta \\ \mu_1 \\ \mu_2 \end{bmatrix} = \left[\mathbf{q}_0 - \mathbf{p}_0\right] \Rightarrow \\ \mathbf{A}\mathbf{x} = \mathbf{b} \end{gathered} \tag{19}$$

There are 3 possible situations.

1. $\mathrm{rank}(\mathbf{A}) = 3 = \mathrm{rank}([\mathbf{A}|\mathbf{b}])$. The edge intersects the plane which contains the triangle. In case $\frac{\mu_1}{l_1} + \frac{\mu_2}{l_2} \leq 1$ with, $\mu_1, \mu_2 \geq 0$ and $0 \leq \eta \leq d$ the intersection lays into the triangle, otherwise the edge is discarded. In case of intersection, the intersection point can be easily calculated replacing η in (18).
2. $\mathrm{rank}(\mathbf{A}) = 2 = \mathrm{rank}([\mathbf{A}|\mathbf{b}])$. The edge is contained in the plane. The triangles might be coplanar or adjacent. The edge is discarded.
3. $\mathrm{rank}(\mathbf{A}) = 2 \neq \mathrm{rank}([\mathbf{A}|\mathbf{b}]) = 3$. The edge is parallel to the plane which contains the triangle. The edge is discarded.

The three edges of triangle q^j are successively checked against triangle p^i and after, the three edges of p^i against triangle q^j. In case two intersections are obtained, the triangles overlap and the intersection, $\overline{\mathbf{i}_1\mathbf{i}_2}$, is given by the segment composed of the intersection points.

C. Contact regions contour closure algorithm

From the disordered colliding triangles list (13) of section IV-B, this algorithm performs the closure of the contours of the different contact regions, grouping together the collision pairs by regions and ordering the collision pairs of each region (and its contour segments) in a way that permits to follow the contours from a segment to the adjacent one. The algorithm uses the topological information about the neighbors, calculated in section IV-A, to find out which collision pairs belong to the same region and to order the segments inside each region.

Once the segments are grouped by regions and the segments of each region are ordered, the algorithm merges the adjacent segments removing the coincident vertexes in (14), by means of a simple numerical procedure. Finally, the algorithm returns a list with the existent n_c 3D contours given by their ordered vertexes.

$$\overline{\mathbf{c}}_c^{ij,j} = \begin{bmatrix} \mathbf{r}_{c,1} & \mathbf{r}_{c,2} & \dots & \mathbf{r}_{c,n_{ic}} \end{bmatrix}; \quad 0 \leq c \leq n_c \text{ (list of contours)} \tag{20}$$

In (20), n_{ic} is the number of vertexes of the contour *c*, the super index ^{ij}, indicates collision between bodies *i* and *j*, and the over line along with the superindex $^{\cdot j}$ indicates local coordinates of body *j*.

D. Contact plane calculation algorithm

For each one of the contact regions identified in section IV-C, the algorithm calculates the equations of the contact plane that better fits the 3D contour, (20), of the region (see figure 6).

Replacing the vertexes of the contour given by (20) in the equations of the contact plane.

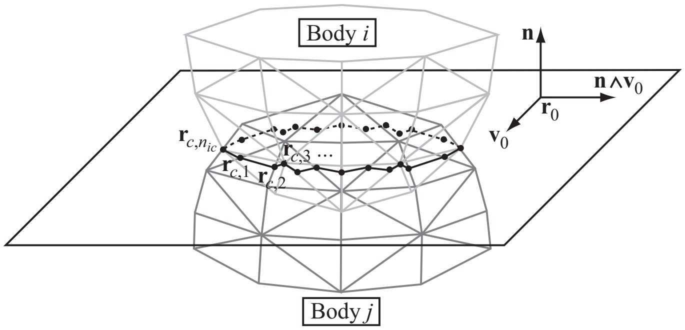

Fig. 6. Contact plane calculation

$$\begin{bmatrix} \mathbf{r}_{c,1}^{\mathrm{T}} & 1 \\ \mathbf{r}_{c,2}^{\mathrm{T}} & 1 \\ \dots & \dots \\ \mathbf{r}_{c,n_{ic}}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \overline{\mathbf{n}} \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \Rightarrow \mathbf{A}\mathbf{x} = \mathbf{0} \tag{21}$$

Where $\overline{\mathbf{n}}$ is a vector normal to the contact plane and $d = \; -\mathbf{r}_c^{\mathrm{T}}\overline{\mathbf{n}}$ being $\mathbf{r}_c$ a point that belongs to the contact plane.

In general the system of equations (21) has the only solution $\overline{\mathbf{n}} = \mathbf{0}; d = 0$, which obviously is not the desired solution. It is necessary to impose the condition $|\overline{\mathbf{n}}| = 1$ to obtain an incompatible system of equations that can be solved by least squares.

Writing the least squares system from (21).

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)\mathbf{x} = \mathbf{0} \tag{22}$$

Factoring the matrix $\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)$ and imposing the value (for example equal to 1) of the component of **x** corresponding to the minimum pivot of the factorization, the equations of the contact plane are obtained. Finally **x** has to be scaled to fulfill the condition $|\overline{\mathbf{n}}| = 1$, obtaining the final equations of the contact plane.

$$\overline{\mathbf{n}}^{\mathrm{T}}\mathbf{r} + d = 0 \tag{23}$$

Where $\overline{\mathbf{n}}$ is the unit normal vector to the plane and *d* is the distance from the plane to the origin measured along the normal vector.

All the calculations described in this section were performed with the contour of equation (20) expressed in the local reference frame of body *j*. The normal vector transformed to the global reference frame is obtained by means of the rotation matrix of body *j*.

$$\mathbf{n} = \mathbf{R}^j\overline{\mathbf{n}} \tag{24}$$

E. Contact region centroid algorithm

For each one of the contact regions identified in section IV-C, the algorithm described in this section, calculates the centroid of the projection of the contact region into the contact plane.

The centroid of a general 2D polygon of *N* vertexes, contained in the *X*-*Y* plane has the following expression.

$$\bar{\mathbf{r}}_c^{\triangle} = \frac{1}{6A}\sum_{i=0}^{N-1}\left[\begin{array}{c}(x_i + x_{i+1})\,(x_i y_{i+1} - x_{i+1} y_i)\\ (y_i + y_{i+1})\,(x_i y_{i+1} - x_{i+1} y_i)\\ 0\end{array}\right] \qquad (25)$$

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)$$

Nevertheless the contour, *c*, of equation (20) does not constitute a 2D polygon since its vertexes do not belong, in general, to the same plane. To assimilate the contour to a 2D polygon, the vertexes can be projected into the contact plane calculated in section IV-D. Moreover, the resulting 2D polygon has to be contained in the *X*-*Y* plane, what can be achieved by means of the transformation matrix $\mathbf{M}_t$, which transforms the *X*-*Y* plane into the contact plane of equation (23).

The mentioned transformation matrix has the following expression.

$$\mathbf{M}_t = \left[\begin{array}{cc}\mathbf{M}_r & \bar{\mathbf{r}}_0\\ 0 & 1\end{array}\right] \qquad (26)$$

$$\mathbf{M}_r = \left[\begin{array}{ccc}\bar{\mathbf{v}}_0 & \bar{\mathbf{n}}\wedge\bar{\mathbf{v}}_0 & \bar{\mathbf{n}}\end{array}\right]$$

Where $\bar{\mathbf{r}}_0$ and $\bar{\mathbf{v}}_0$ are a point and a vector contained in the contact plane, respectively, that can be chosen like follows.

$$\bar{\mathbf{r}}_0 = \left[\begin{array}{c}-d/n_x\\ 0\\ 0\end{array}\right];\ \bar{\mathbf{v}}_0 = \left[\begin{array}{c}-n_y\\ n_x\\ 0\end{array}\right];\quad n_x = \max(n_x, n_y, n_z)$$

$$\bar{\mathbf{r}}_0 = \left[\begin{array}{c}0\\ -d/n_y\\ 0\end{array}\right];\ \bar{\mathbf{v}}_0 = \left[\begin{array}{c}n_y\\ -n_x\\ 0\end{array}\right];\quad n_y = \max(n_x, n_y, n_z)$$

$$\bar{\mathbf{r}}_0 = \left[\begin{array}{c}0\\ 0\\ -d/n_z\end{array}\right];\ \bar{\mathbf{v}}_0 = \left[\begin{array}{c}n_z\\ 0\\ -n_x\end{array}\right];\quad n_z = \max(n_x, n_y, n_z) \qquad (27)$$

Being n_x, n_y, n_z the components of $\bar{\mathbf{n}}$. Expressing the contour in the local frame of the plane.

$$\bar{\mathbf{c}}_c^{ij,\triangle} = \mathbf{M}_r^{\mathrm{T}}\left(\bar{\mathbf{c}}_c^{ij,j} - \left[\begin{array}{cccc}\bar{\mathbf{r}}_0 & \bar{\mathbf{r}}_0 & \dots & \bar{\mathbf{r}}_0\end{array}\right]\right) \qquad (28)$$

Replacing the *x* and *y* components of (28) in (25), the centroid $\bar{\mathbf{r}}_c^{\triangle}$, in local coordinates of the plane, is obtained.

Finally, the centroid expressed in global coordinates has the following expression.

$$\mathbf{r}_c^{ij} = \mathbf{s}_0^j + \mathbf{R}^j\left(\bar{\mathbf{r}}_0 + \mathbf{M}_r\bar{\mathbf{r}}_c^{\triangle}\right) \qquad (29)$$

F. *Maximum indentation calculation algorithm*

For each one of the contact regions identified in section IV-C, the algorithm calculates the maximum indentation (or inter-penetration), δ (see figure 1). This algorithm travels along the colliding triangles (equation (13)) and their neighbors looking for the maximum indentation. In order to distinguish the faces of the first body that are inter-penetrating the second, it is necessary to check, for each neighbor not belonging to the colliding triangles list, two conditions: 1) the distance of each vertex of the triangle to the contact plane, calculated in section IV-D, is negative and 2) the projection each one of the vertex of the triangle into the contact plane lies inside the projected contact region contour of section IV-C (see figure 6). Between the lists of colliding and internal triangles of each body, the algorithm looks for the maximum indentation.

The checking of the condition 1) is straightforward while the condition 2) is checked by means of a ray casting algorithm: the number of intersections of a ray departing from a point is an even number if the point is outside the polygon, and it is odd if the point is inside the polygon.

V. Numerical examples: anchor maneuver of a warship

The numerical example chosen for this work is the anchor maneuver of a warship. The hull of the warship is static, while the anchor is composed of two bodies joined by a revolute joint and tied to cable modeled by ideal constraints. The total mass of the assembly is $6414kg$. The system is subjected to the gravity forces, the constraint forces, the tension force of the cable and the contact forces with the hull of the warship. The anchor and the hull are modelled in a commercial 3D CAD package and exported as triangular meshes.

The maneuver is shown in figure V. The green vectors represent the normal contact forces while the red ones represent the tangential forces due to the friction. The contact model implemented was described in section III.

VI. Conclusions

A general strategy to simulate the dynamics of multibody systems with contacting rigid bodies was described. The description includes: a contact force model composed of normal and tangential forces; a contact detection algorithm for bodies with complex non-conforming 3D geometries given by CAD models in triangular mesh format. Furthermore, the performance of the described strategy was tested with a realistic example: the anchor maneuver of a warship.

References

[1] Garcia de Jalon J. and Bayo E. *Kinematic and dynamic simulation of multibody systems: The real-time challenge.* Springer-Verlag, 1994.

[2] Lankarani H.M. and Nikravesh P.E. A contact force model with hysteresis damping for impact analysis of multibody systems. *Journal of Mechanical Design*, 112: 369–376, 1990.

[3] P. Flores, J. Ambrosio, J.C.P. Claro and H.M. Lankarani. *Kinematics and Dynamics of Multibody Systems with Imperfect Joints.* Springer-Verlag, Berlin, Heidelberg, 2008.

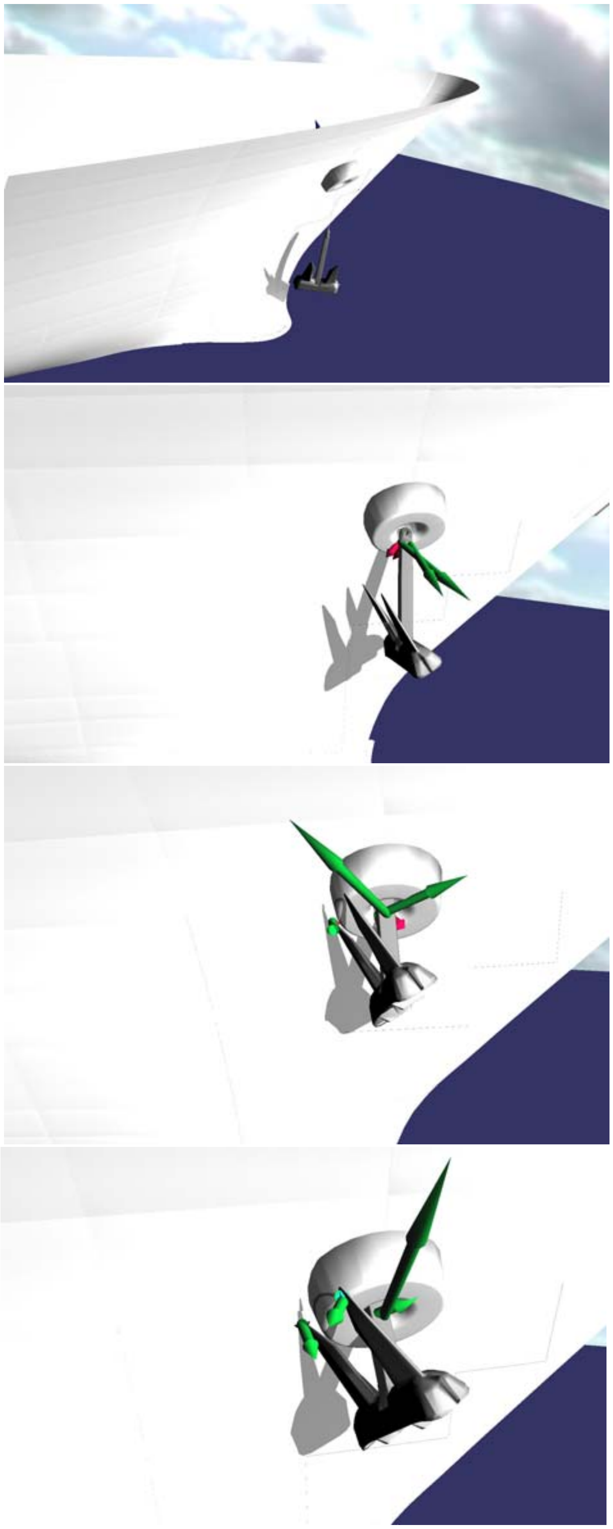

Fig. 7. Warship anchor maneuver

[4] Dopico D., Luaces A. and Gonzalez M. A soil model for a hydraulic excavator simulator based on real-time multibody dynamics. *5th Asian Conference on Multibody Dynamics 2010*, Kyoto, Japan, August, 23–26, 2010.

[5] Van Den Bergen G. *Collision detection in interactive 3D environments*. Elsevier, 2004.

[6] Ericson, C. *Real-time collision detection*.

[7] Cuadrado J., Gutierrez R., Naya M.A. and Morer P. A comparison in terms of accuracy and efficiency between a MBS dynamic formulation with stress analysis and a non-linear FEA code. *Int. Journal for Numerical Methods in Engineering*, 51: 1033–1052, 2001.

[8] Cuadrado J., Dopico D., Naya M.A. and Gonzalez M. Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators. *Multibody System Dynamics*, 12: 117–132, 2004.

[9] Hunt K.H. and Crossley F.R.E. Coefficient of restitution interpreted as damping in vibroimpact. *Journal of Applied Mechanics*, 7: 440–445, 1975.

[10] Gonthier Y., McPhee J., Lange C. and Piedboeuf J.C. A Regularized Contact Model with Asymmetric Damping and Dwell-Time Dependent Friction. *Multibody System Dynamics*, **11**, 209–233, 2004.