THE ARCHIVE OF MECHANICAL ENGINEERING

VOL. LVII 2010 Number 2

10.2478/v10180-010-0009-8

Key words: contact detection, superquadrics, optimization, multibody dynamics

RICARDO JOSÉ FONTES PORTAL, JOÃO MANUEL PEREIRA DIAS, LUÍS ALBERTO GONÇALVES DE SOUSA *

CONTACT DETECTION BETWEEN CONVEX SUPERQUADRIC SURFACES

This paper presents a methodology for contact detection between convex quadric surfaces using its implicit equations. With some small modifications in the equations, one can model superellipsoids, superhyperboloids of one or two sheets and supertoroids. This methodology is to be implemented on a multibody dynamics code, in order to simulate the interpenetration between mechanical systems, particularly, the simulation of collisions with motor vehicles and other road users, such as cars, motorcycles and pedestrians.

The contact detection of two bodies is formulated as a convex nonlinear constrained optimization problem that is solved using two methods, an Interior Point method (IP) and a Sequential Quadratic Programming method (SQP), coded in MATLAB and FORTRAN environment, respectively. The objective function to be minimized is the distance between both surfaces. The design constraints are the implicit superquadrics surfaces equations and operations between its normal vectors and the distance itself. The contact points or the points that minimize the distance between the surfaces are the design variables. Computational efficiency can be improved by using Bounding Volumes in contact detection pre-steps. First one approximate the geometry using spheres, and then Oriented Bounding Boxes (OBB).

Results show that the optimization technique suits for the accurate contact detection between objects modelled by implicit superquadric equations.

1. Introduction

This paper studies contact detection problem between two convex objects described as implicit superquadric surfaces [1,2]. The contact detection algorithm developed here is used in the narrow phase of the problem, where the minimum distance or the maximum interference between the objects has

* *IDMEC – Institute of Mechanical Engineering Instituto Superior Técnico – Technical University of Lisbon Av. Rovisco Pais, 1049-001 Lisbon, Portugal. E-mail: {ricardo.portal, jdias, lsousa}@dem.ist.utl.pt*

interest for future methodology purposes. Before this narrow phase, computational efficiency can be improved by using Bounding Volumes in contact detection pre-steps, where the bodies geometry is approximated using spheres and Oriented Bounding Boxes (OBB) [3,4].

The contact detection methodology can be used in several applications, for instance, in computer graphics, robotic arms path planning and haptics. The main objective for this methodology is the simulation of mechanical systems in multibody dynamics [5], in particular, the simulation of collisions between cars and motorcycles and other road vehicles, as well as pedestrian impacts with them [6].

The contact detection between two superquadric implicit surfaces is formulated as a convex nonlinear constrained optimization problem that is solved using two different methods, an interior point method (IP) [7-10] and a *Sequential Quadratic Programming* method (SQP) [11,12], coded in MATLAB and FORTRAN environment, respectively. The objective function to minimize is the distance between both surfaces. The design constraints (equality) are the implicit superquadrics surfaces equations, so that the contact points or the points that minimize the distance are located over the surface. Other equality constraints are added using the surfaces normal vectors, to guarantee that the algorithms converge to the minimum. The contact points or the points that minimize the distance between the surfaces are the design variables.

2. Contact detection broad phase

This section describes the broad phase of contact detection [3,4], where bodies are first approximated by Spheres and then by Oriented Bounding Boxes. Consider, for example, the bodies shown in Fig. 1, the radii of the spheres that approximate those bodies are obtained from the largest dimension of the bodies (Fig. 2a).

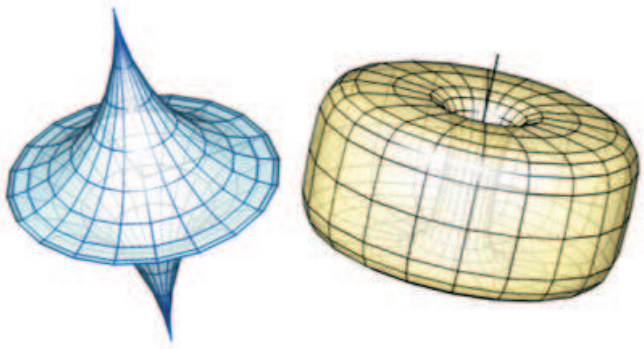


Fig. 1. Example of contact pairs for contact detection

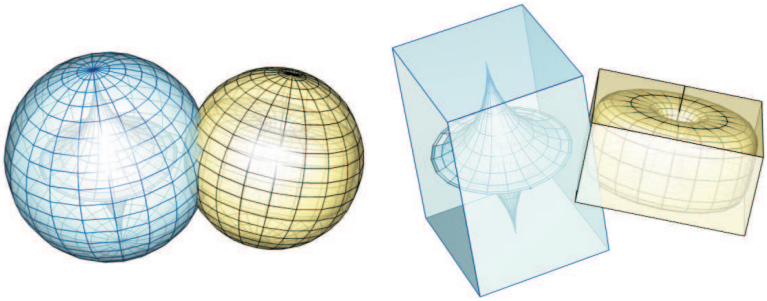


Fig. 2. a) Spheres' approximation of the contact pairs; b) OBB approximation of the contact pair for contact detection

If the distance between the centres of the spheres is less or equal than the sum of their radii, then the spheres intersect each other. Then, the Spheres' test consists in evaluating if the distance $\mathbf{r}_A B = \|\mathbf{r}_A - \mathbf{r}_B\|$ is lower or equal to the sum of the sphere's radii $(\rho_A + \rho_B)$, Eq. (1).

$$\|\mathbf{r}_\mathrm{A} - \mathbf{r}_\mathrm{B}\| \leq (\rho_\mathrm{A} + \rho_\mathrm{B}) \tag{1}$$

The following stage survey for the existence of a *separating axis* between the oriented bounding boxes that approximate the bodies (Fig. 2b). For non-intersection convex objects, a *separating axis* always exists. An arbitrary vector $\mathbf{u}$ is a *separating axis* if the projections of those boxes onto $\mathbf{u}$ do not overlap.

Being $\mathbf{r}_\mathrm{A} = (x_\mathrm{A}, y_\mathrm{A}, z_\mathrm{A})$ and $\mathbf{r}_\mathrm{B} = (x_\mathrm{B}, y_\mathrm{B}, z_\mathrm{B})$ the coordinate reference frames of boxes A and B, the distances (box's scalar projections) h_A and h_B onto $\mathbf{u}$ are given by, (see Fig. 3).

$$h_\mathrm{A} = a_1 |x_\mathrm{A} u_x| + a_2 |y_\mathrm{A} u_y| + a_3 |z_\mathrm{A} u_z| \tag{2}$$

$$h_\mathrm{B} = b_1 |x_\mathrm{B} u_x| + b_2 |y_\mathrm{B} u_y| + b_3 |z_\mathrm{B} u_z| \tag{3}$$

where (a_1, a_2, a_3) and (b_1, b_2, b_3) are the half dimensions of the boxes.

Defining $\|\mathbf{r}_\mathrm{AB}\| = \|\mathbf{r}_\mathrm{B} - \mathbf{r}_\mathrm{A}\|$ as the distance between the origins of both boxes (relatively to the local reference frame of box A), $\mathbf{u}$ forms a separating axis if the value of $\mathbf{r}_\mathrm{AB}$ projected onto $\mathbf{u}$ is lower than the sum of the projections of each box onto $\mathbf{u}$, as

$$|\mathbf{r}_\mathrm{AB} \cdot \mathbf{u}| > h_\mathrm{A} + h_\mathrm{B} \tag{4}$$

Figure 3 illustrates the *Separating Axis Test* (SAT) between two boxes. In the presented case, the boxes intersect each other.

In a 3D space, two boxes A and B must be tested for fifteen axes, in order to determine if one of them is a *separating axis*. Those axes are the six

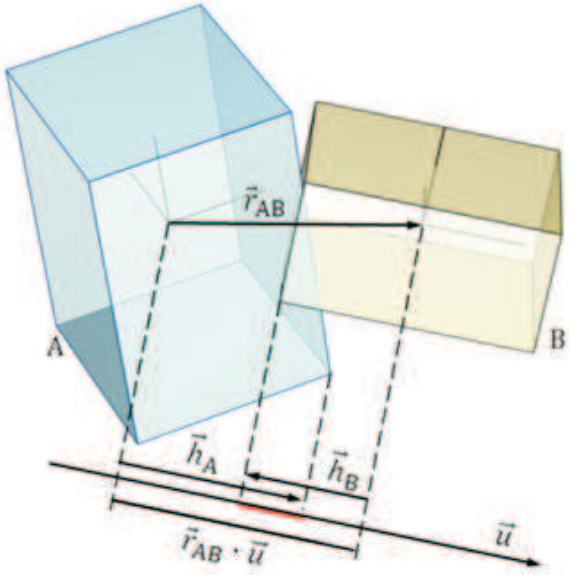


Fig. 3. OBB approximation of the contact pair for contact detection

principal coordinate axes of A's and B's local reference frames and the nine axes obtained by the cross products of a coordinate axis of the box A and a coordinate axis of the box B. If none of the tests is satisfied, the objects are in contact. For facility purposes, the box B's axes, as well as the distance between boxes $\mathbf{r}_{AB}$ should be written as a combination of the axes of box A, i.e, B's axes (Eq. (5)) and $\mathbf{r}_{AB}$ (Eq.(6)) are written in the local reference frame of box A. The rotation matrix $\mathbf{T}_{B/A}$ of box B with relation to box A is obtained using Eq. (7), where the orientation of both boxes is known from the transformation matrices $\mathbf{T}_{A/G}$ and $\mathbf{T}_{B/G}$ regarding to the global reference frame.

$$\mathbf{r}_B = \mathbf{T}^T_{B/A}\mathbf{r}_A \tag{5}$$

$$r_{AB} = T^T_{B/A}r_{AB} \tag{6}$$

$$\mathbf{T}_{B/A} = \mathbf{T}_{B/G} \cdot \mathbf{T}^T_{A/G} = \begin{bmatrix} t'_{11} & t'_{12} & t'_{13} \\ t'_{21} & t'_{22} & t'_{23} \\ t'_{31} & t'_{32} & t'_{33} \end{bmatrix} \tag{7}$$

The method of separating axes tests is described in more detail in [3,4]. Table 1 sum-marises the quantities that must be determined for the SAT. When one separating axis is found the remaining ones are not calculated.

3. Superquadrics and their geometric properties

Superquadrics are a family of geometric shapes that includes superellipsoids, superhyperboloids of one or two pieces and supertoroids (Fig. 4). The differences in this shapes are obtained with minor changes in the superquadric equations.

Table 1.

SAT test quantities for OBB contact detection

axis $\mathbf{u}$	extent $\boldsymbol{h}_{\mathbf{A}}$	extent $\boldsymbol{h}_{\mathbf{B}}$	$\lvert\mathbf{r}_{\mathbf{AB}} \cdot \mathbf{u}\rvert$
$\mathbf{x}_{\mathbf{A}}$	a_1	$b_1\lvert t'_{11}\rvert + b_2\lvert t'_{12}\rvert + b_3\lvert t'_{13}\rvert$	$\lvert x_{\mathrm{AB}}\rvert$
$\mathbf{y}_{\mathbf{A}}$	a_2	$b_1\lvert t'_{21}\rvert + b_2\lvert t'_{22}\rvert + b_3\lvert t'_{23}\rvert$	$\lvert y_{\mathrm{AB}}\rvert$
$\mathbf{z}_{\mathbf{A}}$	a_3	$b_1\lvert t'_{31}\rvert + b_2\lvert t'_{32}\rvert + b_3\lvert t'_{33}\rvert$	$\lvert z_{\mathrm{AB}}\rvert$
$\mathbf{x}_{\mathbf{B}}$	$a_1\lvert t'_{11}\rvert + a_2\lvert t'_{21}\rvert + a_3\lvert t'_{31}\rvert$	b_1	$\lvert x_{\mathrm{AB}}t'_{11} + y_{\mathrm{AB}}t'_{21} + z_{\mathrm{AB}}t'_{31}\rvert$
$\mathbf{y}_{\mathbf{B}}$	$a_1\lvert t'_{12}\rvert + a_2\lvert t'_{22}\rvert + a_3\lvert t'_{32}\rvert$	b_2	$\lvert x_{\mathrm{AB}}t'_{12} + y_{\mathrm{AB}}t'_{22} + z_{\mathrm{AB}}t'_{32}\rvert$
$\mathbf{z}_{\mathbf{B}}$	$a_1\lvert t'_{13}\rvert + a_2\lvert t'_{23}\rvert + a_3\lvert t'_{33}\rvert$	b_3	$\lvert x_{\mathrm{AB}}t'_{13} + y_{\mathrm{AB}}t'_{23} + z_{\mathrm{AB}}t'_{33}\rvert$
$\mathbf{x}_{\mathbf{A}} \times \mathbf{x}_{\mathbf{B}}$	$a_2\lvert t'_{31}\rvert + a_3\lvert t'_{32}\rvert$	$b_2\lvert t'_{13}\rvert + b_3\lvert t'_{12}\rvert$	$\lvert z_{\mathrm{AB}}t'_{21} - y_{\mathrm{AB}}t'_{31}\rvert$
$\mathbf{x}_{\mathbf{A}} \times \mathbf{y}_{\mathbf{B}}$	$a_2\lvert t'_{32}\rvert + a_3\lvert t'_{22}\rvert$	$b_1\lvert t'_{13}\rvert + b_3\lvert t'_{11}\rvert$	$\lvert z_{\mathrm{AB}}t'_{22} - y_{\mathrm{AB}}t'_{32}\rvert$
$\mathbf{x}_{\mathbf{A}} \times \mathbf{z}_{\mathbf{B}}$	$a_2\lvert t'_{33}\rvert + a_3\lvert t'_{23}\rvert$	$b_1\lvert t'_{12}\rvert + b_2\lvert t'_{11}\rvert$	$\lvert z_{\mathrm{AB}}t'_{23} - y_{\mathrm{AB}}t'_{33}\rvert$
$\mathbf{y}_{\mathbf{A}} \times \mathbf{x}_{\mathbf{B}}$	$a_1\lvert t'_{31}\rvert + a_3\lvert t'_{11}\rvert$	$b_2\lvert t'_{23}\rvert + b_3\lvert t'_{22}\rvert$	$\lvert x_{\mathrm{AB}}t'_{31} - z_{\mathrm{AB}}t'_{11}\rvert$
$\mathbf{y}_{\mathbf{A}} \times \mathbf{y}_{\mathbf{B}}$	$a_1\lvert t'_{32}\rvert + a_3\lvert t'_{12}\rvert$	$b_1\lvert t'_{23}\rvert + b_3\lvert t'_{21}\rvert$	$\lvert x_{\mathrm{AB}}t'_{32} - z_{\mathrm{AB}}t'_{12}\rvert$
$\mathbf{y}_{\mathbf{A}} \times \mathbf{z}_{\mathbf{B}}$	$a_1\lvert t'_{33}\rvert + a_3\lvert t'_{13}\rvert$	$b_1\lvert t'_{22}\rvert + b_2\lvert t'_{21}\rvert$	$\lvert x_{\mathrm{AB}}t'_{33} - z_{\mathrm{AB}}t'_{13}\rvert$
$\mathbf{z}_{\mathbf{A}} \times \mathbf{x}_{\mathbf{B}}$	$a_1\lvert t'_{21}\rvert + a_2\lvert t'_{11}\rvert$	$b_2\lvert t'_{33}\rvert + b_3\lvert t'_{32}\rvert$	$\lvert y_{\mathrm{AB}}t'_{11} - x_{\mathrm{AB}}t'_{21}\rvert$
$\mathbf{z}_{\mathbf{A}} \times \mathbf{y}_{\mathbf{B}}$	$a_1\lvert t'_{22}\rvert + a_2\lvert t'_{12}\rvert$	$b_1\lvert t'_{33}\rvert + b_3\lvert t'_{31}\rvert$	$\lvert y_{\mathrm{AB}}t'_{12} - x_{\mathrm{AB}}t'_{22}\rvert$
$\mathbf{z}_{\mathbf{A}} \times \mathbf{z}_{\mathbf{B}}$	$a_1\lvert t'_{23}\rvert + a_2\lvert t'_{13}\rvert$	$b_1\lvert t'_{32}\rvert + b_2\lvert t'_{31}\rvert$	$\lvert y_{\mathrm{AB}}t'_{13} - x_{\mathrm{AB}}t'_{23}\rvert$

Usually, a superquadric surface is defined as a spherical product of two parametric 2D curves, for instance, superellipses, resulting in a parametric three-dimensional shape. Some mathematical manipulation of this parametric equation leads to the superquadric implicit equation as defined by Eq. (8), in this case a superellipsoid, where parameters a_1, a_2 and a_3 represents the surface semi-axes and ε_1 and ε_2 are the indices that determine the surface shape. The parameter ε_1 determines the shape of the superquadric cross section in a perpendicular plane to (x, y) and containing z, while ε_2 determines the shape of the cross section parallel to (x, y). Figure 5 depicts the influence of ε_1 and ε_2 on the shape of superquadrics.

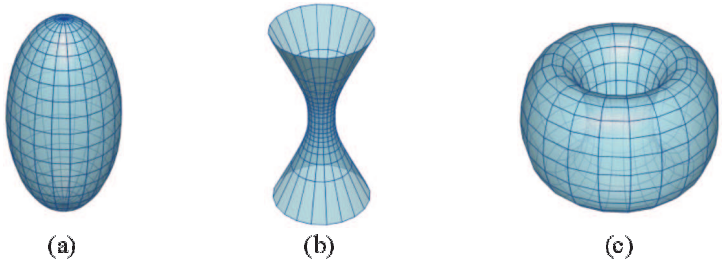


Fig. 4. Superquadric shapes: a) superellipsoid; b) superhyperboloid of one piece and c) supertoroid

$$h(x,y,z) = 0 \Leftrightarrow \left(\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1 \tag{8}$$

The left-hand side of Eq. (8) is called the inside-outside function F, because its value provides a way to determine if a certain point (x, y, z) lies inside or outside the surface,; if $F < 1$, then point (x, y, z) is inside the surface if $F > 1$, then (x, y, z) is outside the superellipsoid, every point (x, y, z) that verify $F = 1$ lies in the superquadric surface.

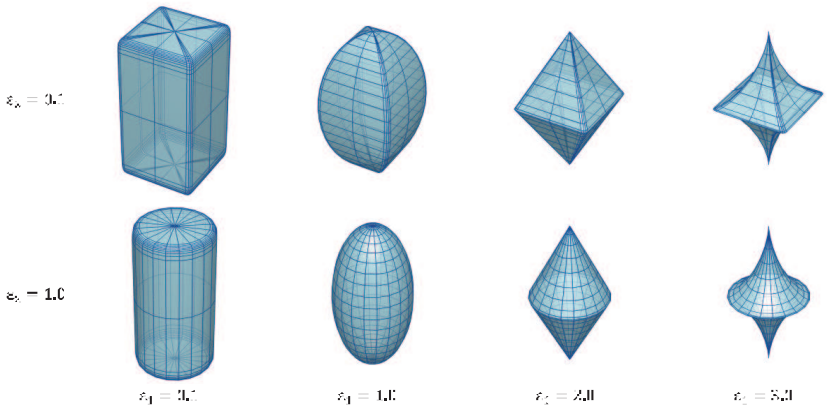


Fig. 5. ε_1 and ε_2 influence on the shape of superquadrics (superellipsoids)

The normal of an implicit superquadric equation (Eq. (8)) is its gradient or its partial derivative. Knowing this, one can calculate the surface normal vector as

$$\mathbf{n}(x,y,z) = \nabla h(x,y,z) = \left(\frac{\partial h}{\partial x}\frac{\partial h}{\partial y}\frac{\partial h}{\partial z}\right) = (n_x n_y n_z) \tag{9}$$

where the normal vector components are

$$\mathrm{n}_x=\frac{2}{\varepsilon_1 a_1}\left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}-1}\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}-1} \tag{10}$$

$$\mathrm{n}_y=\frac{2}{\varepsilon_1 a_1}\left[\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}}+\left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right]^{\frac{\varepsilon_2}{\varepsilon_1}-1}\left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}-1} \tag{11}$$

$$\mathrm{n}_z=\frac{2}{\varepsilon_1 a_3}\left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}-1} \tag{12}$$

Superellipsoids are described in their local frame, in order to determine the distance or the interference among a pair of objects it is necessary to transform them to a global frame. This is done by applying the usual rotation and translation operations. Eq. (13) represents the transformation matrix **T**, where the rotation matrix is a function of the Euler parameters e_0, e_1, e_2 and e_3 [5], and the translation is given by the vector $[p_X, p_Y, p_Z]^T$, that represents the local frame position of each surface in the global frame.

$$\mathrm{T}=\begin{bmatrix} t_{11} & t_{12} & t_{13} & p_X \\ t_{21} & t_{22} & t_{23} & p_Y \\ t_{31} & t_{32} & t_{33} & p_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}=2\begin{bmatrix} e_0^2+e_1^2-\frac{1}{2} & e_1e_2-e_0e_3 & e_1e_3+e_0e_2 & p_X \\ e_1e_2+e_0e_3 & e_0^2+e_2^2-\frac{1}{2} & e_2e_3-e_0e_1 & p_Y \\ e_1e_3-e_0e_2 & e_2e_3+e_0e_1 & e_0^2+e_3^2-\frac{1}{2} & p_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Considering the terms A_i given by Eq. (14), one can describe the implicit superquadrics equations in the system global coordinates.

$$\mathrm{A}_i=\frac{t_{1i}(X-p_X)+t_{2i}(Y-p_Y)+t_{3i}(Z-p_Z)}{a_i} \tag{14}$$

with i=1,2,3, implicit Eq. (8) can be written as

$$h(X,Y,Z)=0 \Leftrightarrow \left((\mathrm{A}_1)^{\frac{2}{\varepsilon_2}}+(\mathrm{A}_2)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}}+(\mathrm{A}_3)^{\frac{2}{\varepsilon_1}}-1=0 \tag{15}$$

and Eq. (10) to Eq. (12) which represent the normal vector components as

$$\mathrm{n}_x=\frac{2}{\varepsilon_1}\left[\left[(\mathrm{A}_1)^{\frac{2}{\varepsilon_2}}+(\mathrm{A}_2)^{\frac{2}{\varepsilon_2}}\right]^{\frac{2}{\varepsilon_1}-1}\left[\frac{t_{11}}{a_1}(\mathrm{A}_1)^{\frac{2}{\varepsilon_2}-1}+\frac{t_{12}}{a_2}(\mathrm{A}_2)^{\frac{2}{\varepsilon_2}-1}\right]+\frac{t_{13}}{a_3}(\mathrm{A}_3)^{\frac{2}{\varepsilon_1}-1}\right] \tag{16}$$

$$\mathrm{n}_y=\frac{2}{\varepsilon_1}\left[\left[(\mathrm{A}_1)^{\frac{2}{\varepsilon_2}}+(\mathrm{A}_2)^{\frac{2}{\varepsilon_2}}\right]^{\frac{2}{\varepsilon_1}-1}\left[\frac{t_{21}}{a_1}(\mathrm{A}_1)^{\frac{2}{\varepsilon_2}-1}+\frac{t_{22}}{a_2}(\mathrm{A}_2)^{\frac{2}{\varepsilon_2}-1}\right]+\frac{t_{23}}{a_3}(\mathrm{A}_3)^{\frac{2}{\varepsilon_1}-1}\right] \tag{17}$$

$$n_z=\frac{2}{\varepsilon_1}\left[\left[(A_1)^{\frac{2}{\varepsilon_2}}+(A_2)^{\frac{2}{\varepsilon_2}}\right]^{\frac{2}{\varepsilon_1}-1}\left[\frac{t_{31}}{a_1}(A_1)^{\frac{2}{\varepsilon_2}-1}+\frac{t_{32}}{a_2}(A_2)^{\frac{2}{\varepsilon_2}-1}\right]+\frac{t_{33}}{a_3}(A_3)^{\frac{2}{\varepsilon_1}-1}\right] \quad (18)$$

4. Optimization problem

Two different optimization techniques for contact detection between surfaces modelled by implicit superquadric equations are used. Contact detection is treated here as problem of convex nonlinear optimization, in what concerns to the objective function and to the equality constraints.

4.1. Objective Function

The objective of the algorithm deals with the calculation of the distance among the surfaces A and B. It is considered the square distance for facility purposes. This being the case, the objective function is described as the distance between the points $\mathbf{r}_A = (X_A, Y_A, Z_A)$ and $\mathbf{r}_B = (X_B, Y_B, Z_B)$ from surfaces A and B

$$\|\mathbf{r}_{AB}\|^2 = \|\mathbf{r}_B - \mathbf{r}_A\|^2 \Leftrightarrow \|\mathbf{r}_{AB}\|^2 = (X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2 \quad (19)$$

When the algorithm converges to r_{AB}, it is calculated the contact condition in order to determine if the value of the objective function correspond to the minimum distance or the maximum penetration among A and B. This contact condition is calculated from the sign of the projection of the normal vector of surface A on the distance vector $\mathbf{r}_{AB}$, in other words if

$$\begin{aligned} &\mathbf{n}_A \cdot \mathbf{r}_{AB} \leq 0 \Rightarrow \text{contact detected with maximum penetration} \\ &\mathbf{n}_A \cdot \mathbf{r}_{AB} > 0 \Rightarrow \text{minimum distance between A and B} \end{aligned} \quad (20)$$

In Fig. 6 it is notice that although the objective function converges for the same value r_{AB}, there is a difference between the two depicted cases. Figure 6a represents a case where the minimum distance r_{AB} separates surfaces A and B, in the case of Fig. 6b, there is a in-terference between A and B, where r_{AB} represents the maximum penetration among them.

4.2. Equality Constraints

During optimization, contact points should lie on the superquadric surfaces. If so and in the foreground, there are defined two equality constraints, formulated from the implicit equations of surfaces A and B, in other words, using Eq. (15) one obtains

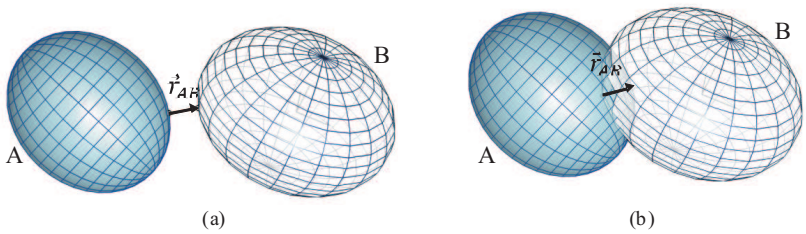


Fig. 6. Possible positions for surfaces A and B, for the same value r_{AB}: a) a separation among A and B and b) an interference between both surfaces

$$h_1 = h_A(X_A, Y_A, Z_A) = 0 \Leftrightarrow \left((A_{1A})^{\frac{2}{\varepsilon_2}} + (A_{2A})^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + (A_{3A})^{\frac{2}{\varepsilon_1}} - 1 = 0 \qquad (21)$$

$$h_2 = h_B(X_B, Y_B, Z_B) = 0 \Leftrightarrow \left((A_{1B})^{\frac{2}{\varepsilon_2}} + (A_{2B})^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + (A_{3B})^{\frac{2}{\varepsilon_1}} - 1 = 0 \qquad (22)$$

The constraints given by Eq. (21) and Eq. (22) do not guarantee convergence for the minimum distance (Fig. 7a), moreover in an interference situation the algorithm evolutes for a null distance (Fig. 7b), which makes mandatory the implementation of two additional constraints.

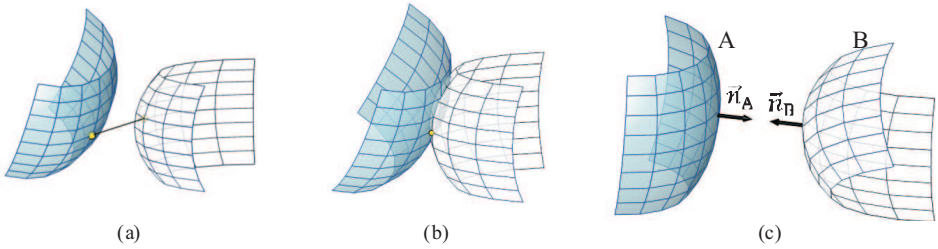


Fig. 7. Equality constraints: a,b) to ensure that the contact points are located on the surfaces and; c) Parallelism between the normal vectors of surfaces A and B

The minimum distance r_{AB} between two surfaces is the one that makes the normal vectors of the superquadrics parallel to each other (Fig. 7c), quite as well as in the case where interference exists, where r_{AB} represents the maximum penetration between both objects, i.e., the contact points correspond to the maximum elastic deformation.

This additional constraint $\mathbf{h}_3$ is given by the cross product (Eq. (23)) between the surfaces normal vectors $\mathbf{n}_A$ and $\mathbf{n}_B$ that represents their alignment condition [13], in other words

$$\mathbf{h}_3 = \mathbf{n}_A \times \mathbf{n}_B = 0 \tag{23}$$

The cross product can be separated in to its scalar components that results in three additional constraint equations, as

$$h_{3_X} = 0 \Leftrightarrow n_{Y_A} n_{Z_B} - n_{Z_A} n_{Y_B} = 0 \tag{24}$$

$$h_{3_Y} = 0 \Leftrightarrow n_{Z_A} n_{X_B} - n_{X_A} n_{Z_B} = 0 \tag{25}$$

$$h_{3_Z} = 0 \Leftrightarrow n_{X_A} n_{Y_B} - n_{Y_A} n_{X_B} = 0 \tag{26}$$

It can be noticed that h3 could lead to some numerical problems, one way of avoiding this is to replace it by the inner product, as depicted by Eq. (27) and (28).

$$\mathbf{n}_A \cdot \mathbf{n}_B = -1 \tag{27}$$

$$h_3 = -1 \Leftrightarrow n_{X_A} n_{X_B} + n_{Y_A} n_{Y_B} n_{Z_A} n_{Z_B} + 1 = 0 \tag{28}$$

In this way, h_3 uses one equation condition so it is numerically more stable, although revealing the disadvantage of using a more weak formulation (Eq. (28)), instead of a strong one (Eq. (24-26)), leading to non-convergence, in some cases.

To ensure that the method converges to the minimum distance (or the maximum penetration) between both surfaces new geometric equations must be set. This is done by imposing that the distance vector $\boldsymbol{r}_{AB}$ and the normal vectors are parallel. The required conditions use Eq. (29) or Eq. (30), respectively using $\mathbf{n}_A$ or $\mathbf{n}_B$.

$$\mathbf{n}_A \cdot \mathbf{r}_{AB} = 1 \Leftrightarrow h_4 = \frac{n_{X_A}(X_B - X_A)}{\|r_{AB}\|} + \frac{n_{Y_A}(Y_B - Y_A)}{\|r_{AB}\|} + \frac{n_{Z_A}(Z_B - Z_A)}{\|r_{AB}\|} - 1 = 0 \tag{29}$$

$$\mathbf{n}_B \cdot \mathbf{r}_{AB} = 1 \Leftrightarrow h_5 = \frac{n_{X_B}(X_B - X_A)}{\|r_{AB}\|} + \frac{n_{Y_B}(Y_B - Y_A)}{\|r_{AB}\|} + \frac{n_{Z_B}(Z_B - Z_A)}{\|r_{AB}\|} - 1 = 0 \tag{30}$$

4.3. Problem Formulation

Once defined the objective function and the constraint equations, the optimization problem can formulate as

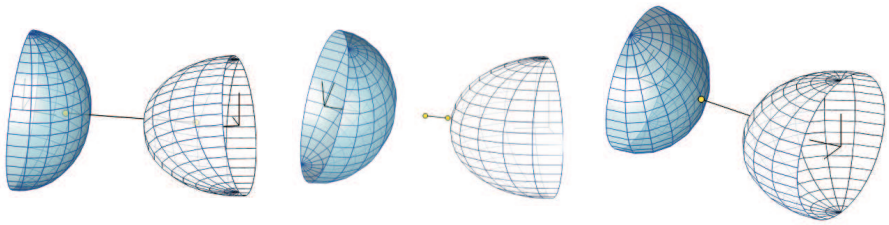
$$\begin{aligned} &\text{minimize} && \|\mathbf{r}_B - \mathbf{r}_A\|^2 \\ &\text{subject to :} && h_A(X_A, Y_A, Z_A) = 0 \\ & && h_B(X_B, Y_B, Z_B) = 0 \\ & && \mathbf{n}_A \times \mathbf{n}_B = 0 \text{or} \quad \mathbf{n}_A \cdot \mathbf{n}_B = -1 \\ & && \mathbf{n}_A \cdot \mathbf{r}_{AB} = 1 \text{or} \quad \mathbf{n}_B \cdot \mathbf{r}_{AB} = -1 \end{aligned} \tag{31}$$

The numerical solution of the optimization problem can be obtained using a mathematical programming approach. This task is not easy, in terms of computational efforts and solution stability. There are several sets of numerical libraries, available on the market, to solve this optimization problem, with different methods and algorithms each one more appropriate to a specific problem. The discussion of the appropriate optimization algorithm is outside the scope of this work, but the requirement that a stable convergence to the solution is necessary.

4.4. Initial Estimate

The optimization algorithms used, need an initial estimate for the design variables. This estimate lies in the vector that connects both surfaces origins, deviate from each origin a certain percentage given by the factor δ_A or δ_B respectively. Initial estimate for surfaces A and B are written as

$$\begin{Bmatrix} X_A \\ Y_A \\ Z_A \end{Bmatrix} = \begin{Bmatrix} p_{X_A} + \delta_A(p_{X_B} - p_{X_A}) \\ p_{Y_A} + \delta_A(p_{Y_B} - p_{Y_A}) \\ p_{Z_A} + \delta_A(p_{Z_B} - p_{Z_A}) \end{Bmatrix} \text{and} \begin{Bmatrix} X_B \\ Y_B \\ Z_B \end{Bmatrix} = \begin{Bmatrix} p_{X_B} - \delta_B(p_{X_B} - p_{X_A}) \\ p_{Y_B} - \delta_B(p_{Y_B} - p_{Y_A}) \\ p_{Z_B} - \delta_B(p_{Z_B} - p_{Z_A}) \end{Bmatrix} \tag{32}$$



(a) $\ddot{a}_A = \ddot{a}_B = 0.20$ (b) $\ddot{a}_A = \ddot{a}_B = 0.45$ (c) $\ddot{a}_A$ and $\ddot{a}_B$ over the surfaces

Fig. 8. Initial estimate for the design variables

Figure 8 shows the initial estimate, with fixed value for δ. Note that the initial estimate depends on the factor δ, as well as the surfaces position.

Considering two different configurations, where the distance between the origins of A and B for the second configuration doubles the first one, the distance between the initial estimate and the origin of the corresponding surface will also be the double for the second configuration. Other disadvantage of this initial estimate is that the optimization method starts with the equality constraints from Eq. (21) and Eq. (22) violated. In order to eliminate the dependency of the superquadrics position, and to start with non-violated constraints, one minimizes two nonlinear equations that ensure that the initial estimate lie on the surfaces A and B, [14]. Such nonlinear equations are obtained by substituting Eq. (32) in Eq. (15), resulting in an equation dependent on δ (Eq. (33)), where the terms R_i are given by Eq. (34).

$$\left((R_1)^{\frac{2}{\varepsilon_2}}+(R_2)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}}+(R_3)^{\frac{2}{\varepsilon_1}}-1=0 \tag{33}$$

$$R_i=\frac{t_{1i}\delta(p_{X_B}-p_{X_A})+t_{2i}\delta(p_{Y_B}-p_{Y_A})+t_{3i}\delta(p_{Z_B}-p_{Z_A})}{a_i} \tag{34}$$

Solving Eq. (33) for δ, it is obtained the values for δ_A and δ_B, substituting those again in Eq. (32) the initial estimate that lie on the surfaces are determined.

5. Results

The constraint optimization problem is solved using two different methods, the first one is a nonlinear Interior Point method [7-10], coded in MATLAB, using the function *fmincon*. The second method is a Sequential Quadratic Programming (SQP), coded in FORTRAN, using the nonlinear libraries ADS and DOT, [11,12].

The results are obtained on a personal computer showing the simplicity of the implementation. In this section, one presents several cases that represent the capability of the method in determining the minimum distance or the maximum penetration among both surperquadrics. Superquadric properties and their position are summarized in Tab. 2. The results of these cases are presented in Tab. 3, where it can be verified the similarities and discrepancies of the methods.

The results gathered in Tab. 3 and in Fig. 9 show that both methods are suitable for determine the interpenetration among surfaces modeled by superquadrics. Note the similarity between the objective function values. In the time column, the value inside brackets is the time that the method takes to determine δ_A and δ_B.

Table 2.

Superquadric properties and position. (note that i=1,2,3 and j=4,5,6)

case #	$(p_X, p_Y, p_Z)_A$ $(p_X, p_Y, p_Z)_B$	$(a_i)_A$ $(a_j)_B$	$(2/\varepsilon_i)_A$ $(2/\varepsilon_j)_B$	$(e_i)_A$ $(e_i)_B$
1	(0.0 , 0.0 , 0.0) (4.0 , 0.0 , 0.0)	1.0 , 1.0 , 1.0 ; 1.0 , 1.0 , 1.0	2.0 , 2.0 ; 2.0 , 2.0	(0.0 , 0.0 , 0.0) (0.0 , 0.0 , 0.0)
2	(4.0 , 0.0 , 0.0) (5.8 , 0.0 , 0.0)	1.0 , 1.0 , 1.0 ; 1.0 , 1.0 , 1.0	2.0 , 2.0 ; 2.0 , 2.0	(0.0 , 0.0 , 0.0) (0.0 , 0.0 , 0.0)
3	(0.0 , 0.0 , 0.0) (0.0 , 0.0 , 4.0)	1.0 , 1.0 , 1.0 ; 1.2 , 1.4 , 1.8	2.0 , 2.0 ; 2.6 , 3.1	(0.0 , 0.0 , 0.0) (0.0 , 0.0 , 0.0)
4	(0.0 , 0.0 , 4.0) (2.0 , 1.5 , 3.0)	1.2 , 1.4 , 1.8 ; 1.5 , 1.0 , 1.0	2.6 , 3.1 ; 4.0 , 2.2	(0.0 , 0.0 , 0.0) (0.0 , 0.4 , 0.25)
5	(4.0 , 0.0 , 0.0) (2.0 , 1.5 , 3.0)	1.0 , 1.0 , 1.0 ; 1.5 , 1.0 , 1.0	2.0 , 2.0 ; 4.0 , 2.2	(0.0 , 0.0 , 0.0) (0.0 , 0.4 , 0.25)
6	(0.0 , 0.0 , 0.0) (2.0 , 1.5 , 3.0)	1.0 , 1.0 , 1.0 ; 1.5 , 1.0 , 1.0	2.0 , 2.0 ; 4.0 , 2.2	(0.0 , 0.0 , 0.0) (0.0 , 0.4 , 0.25)
7	(-1.0 , 2.0 , 1.0) (0.5 , -1.5 , 3.3)	1.0 , 1.25 , 1.6 ; 2.0 , 1.1 , 0.6	4.0 , 2.0 ; 2.2 , 4.5	(-0.1,-0.20,0.14) (-0.4,0.4,-0.05)
8	(1.0, 3.2 , 1.0) (1.8 , 1.5 , 1.3)	1.0 , 1.25 , 1.6 ; 2.0 , 1.1 , 0.6	4.0 , 2.2 ; 3.2 , 2.5	(0.3, 0.0, 0.0) (0.4,0.4,-0.05)

Table 3.

Results for cases 1 to 6 with IP and SQP methods

case #	δ_A / δ_B	method	$(x,y,z)_A$ $(x,y,z)_B$	distance r_{AB}		time [ms]	iterations	fun. eval.
1	0.2501	PI	(1.0 , 0.0 , 0.0) (3.0 , 0.0 , 0.0)	2.00	distance	534 (362)	0	13
	0.2501	SQP	(1.0 , 0.0 , 0.0) (3.0 , 0.0 , 0.0)	2.00	distance	20 (362)	8	69
2	0.5556	PI	(5.0 , 0.0 , 0.0) (4.8 , 0.0 , 0.0)	0.20	contact	514 (386)	0	13
	0.5556	SQP	(5.0 , 0.0 , 0.0) (4.8 , 0.0 , 0.0)	0.20	contact	0 (386)	1	7
3	0.2501	PI	(0.0 , 0.0 , 1.0) (0.0 , 0.0 , 2.2)	1.20	distance	508 (392)	0	13
	0.4501	SQP	(0.0 , 0.0 , 1.0) (0.03 , -0.01 , 2.2)	1.20	distance	10 (392)	3	28
4	0.5479	PI	(1.04 , 0.95 , 3.41) (0.71 , 0.76 , 3.46)	0.381	contact	779 (383)	11	95
	0.5816	SQP	(1.01 , 1.03 , 4.39) (2.18 , 0.96 , 3.27)	0.381	contact	20 (383)	15	128
5	0.2561	PI	(3.45 , 0.49 , 0.67) (2.56 , 1.27 , 1.74)	1.590	distance	738 (378)	8	71
	0.3128	SQP	(3.38 , 0.38 , 0.71) (2.51 , 1.16 , 1.78)	1.587	distance	10 (378)	4	36
6	0.2561	PI	(0.60 , 0.34 , 0.72) (1.69 , 0.95 , 2.05)	1.821	distance	749 (379)	8	73
	0.2662	SQP	(0.59 , 0.34, 0.73) (1.68 , 0.96 , 2.05)	1.821	distance	20 (379)	8	69

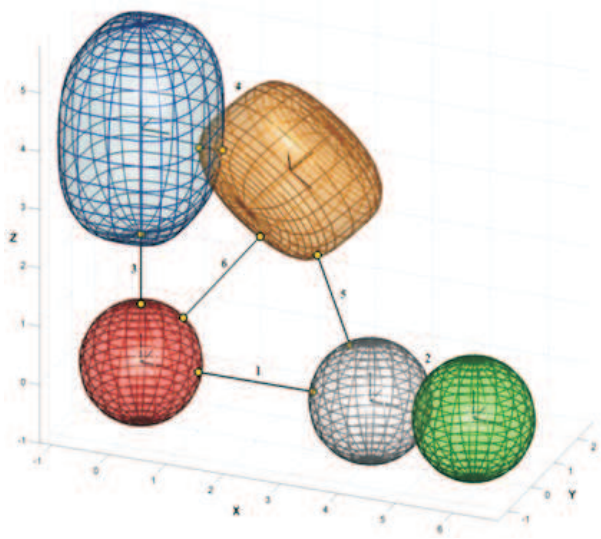


Fig. 9. Contact detection results for cases 1 to 6

IP method reveals more efficiency in the number of iterations and function evaluations, on the other hand, SQP method shows that it takes much less time to converge.

With the cases 7 and 8 from Tab. 2, the intention is to confront several different constraint sets, in both studied situations, superquadrics separated and with some interference. Consequently, there were performed eight optimizations for each case:

a) Superquadrics surfaces constraints only. These constraints are applied in all simulations.
b) Same as (a) plus the parallelism constraint written in terms of the cross product of the normal vectors of the surfaces.
c) Same as (b) plus the parallelism constraint written in terms of the dot product of the normal vector of surface A and the distance vector $\mathbf{r}_{AB}$.
d) Same as (b) plus the parallelism constraint written in terms of the dot product of the normal vector of surface B and the distance vector $\mathbf{r}_{AB}$.
e) Same as (a) plus the parallelism constraint written in terms of the dot product of the normal vectors of the surfaces.
f) Same as (e) plus the parallelism constraint written in terms of the dot product of the normal vector of surface A and the distance vector $\mathbf{r}_{AB}$.
g) Same as (e) plus the parallelism constraint written in terms of the dot product of the normal vector of surface B and the distance vector $\mathbf{r}_{AB}$.
h) Same as (a) plus the parallelism constraints written in terms of the dot product of the normal vectors of surfaces A and B and the distance vector $\mathbf{r}_{AB}$.

Figure 10 points out the superquadrics properties, as well as the positions and orientations of the surfaces for cases 7 and 8. The results for case 7 are gathered in Fig. 11 and in Tab. 4.

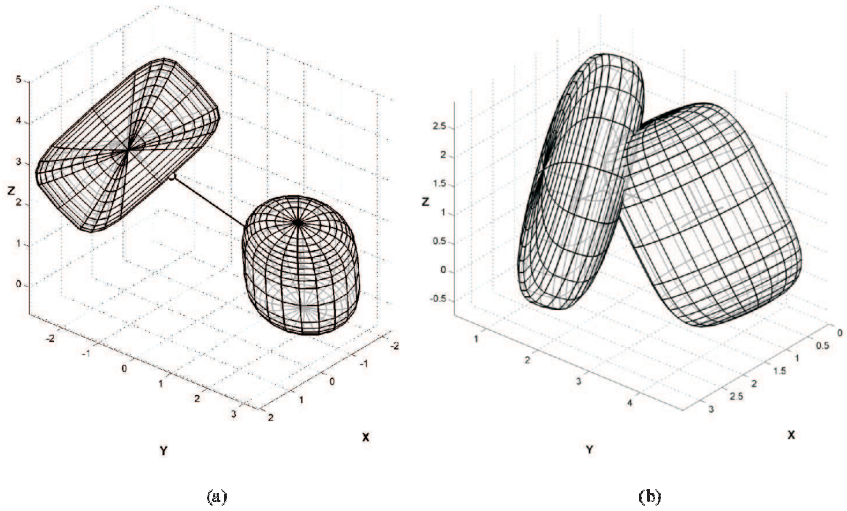


Fig. 10. Superquadric properties and position and initial estimate for: a) case 7 and b) case 8

The simulations shows that minimum distance calculated for case 7 are not dependent of the type of constraints, the method converge for the minimum distance for all constraint sets. One can only notice, in case 7.c, a significantly higher number of iterations and function evaluations, and consequently, in the computational time.

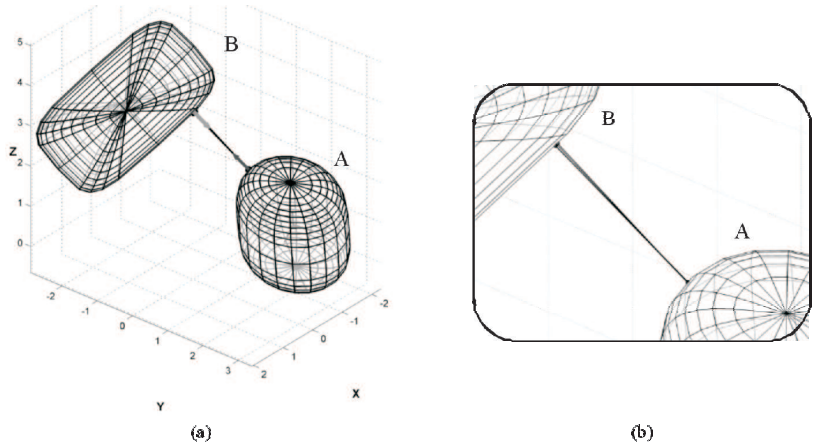


Fig. 11. Minimum distance for case 7. a) solutions; b) detail

Table 4.

Influence of the constraint set used (case 7)

case 7.	const. Eq.(i)	$(x,y,z)_A$	$(x,y,z)_B$	distance $\mathbf{r}_{AB}$	time [ms]	iterations	fun. eval.
a	21,22	(-0.4988 , 1.168 , 2.278)	(-0.3599, -0.2923, 3.223)	1.745	1215	12	106
b	21,22,24-26	(-0.526, 1.147, 2.248)	(-0.361, -0.279, 3.246)	1.748	1162	4	43
c	21,22,24-26,29	(-0.4822, 1.179, 2.292)	(-0.3692, -0.2959, 3.219)	1.746	2047	101	775
d	21,22,24-26,30	(-0.5161, 1.152, 2.255)	(-0.373, -0.276, 3.251)	1.747	1400	27	222
e	21,22,28	(-0.4976, 1.168, 2.278)	(-0.3597, -0.2911, 3.225)	1.745	1186	9	80
f	21,22,28,29	(-0.4925, 1.167, 2.276)	(-0.3613, -0.2921, 3.224)	1.745	1178	7	63
g	21,22,28,30	(-0.4995, 1.168, 2.279)	(-0.3584, -0.2893, 3.228)	1.745	1187	7	67
h	21,22,29,30	(-0.5057, 1.165, 2.274)	(-0.3593, -0.2914, 3.225)	1.745	1211	8	71

Results for case 8 are presented in Fig. 12, Tab. 5 summarizes those results. In case 8, only the constraint sets 8.b and 8.e converged to the value of penetration among the superquadric surfaces.

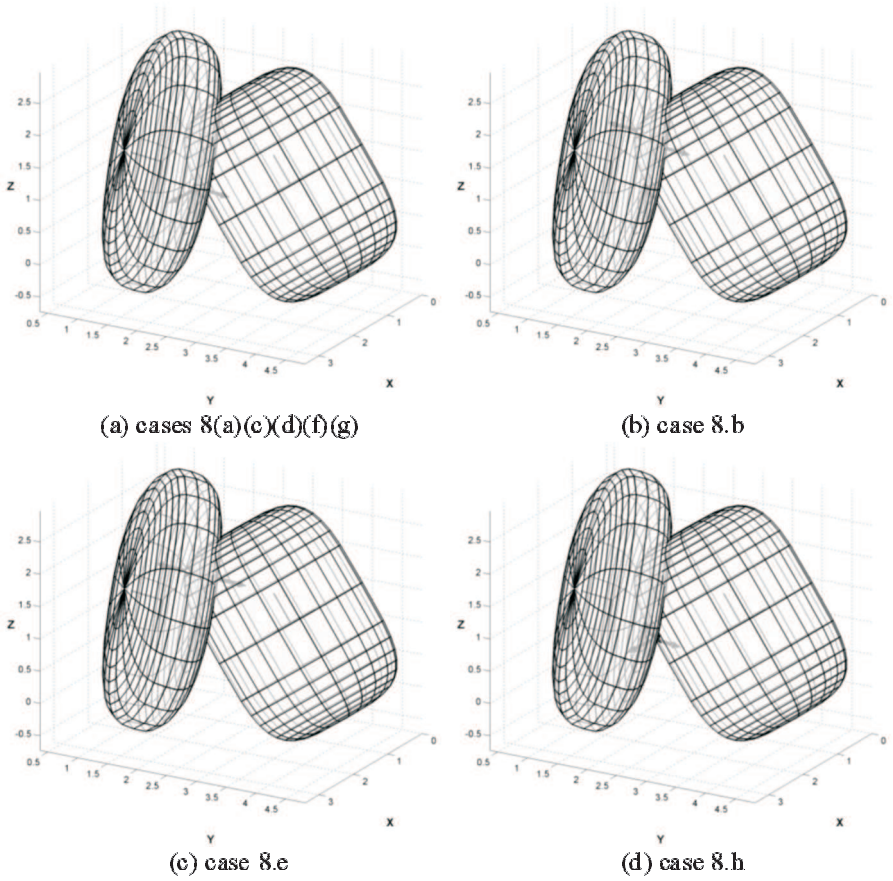


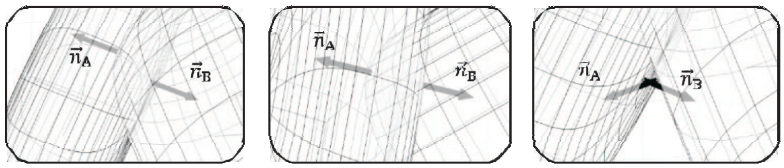
Fig. 12. Interference results for case 8

Cases 8(a,c,d,f,g,h) lead to contact points located over the surfaces but with no penetration. These results indicate that one should only use the constraint sets that do not restrict too much the algorithm, by other words, the constraints given by Eq. (21) and Eq. (22) with the parallelism between the normal vectors give by Eq. (24-26) or Eq. (28). Even so, the obtained results using the Eq. (28) present better behaviour, as it converges with better alignment between normal vectors A and B.

Table 5.

Influence of the constraint set used (case 8)

case 8.	const. Eq.(i)	$(x, y, z)_A$	$(x, y, z)_B$	penetr. $\mathbf{r}_{AB}$	time [ms]	iterations	fun. eval.
a	21,22	(1.613, 2.001, 0.9268)	(1.613, 2.001, 0.9268)	0.0003 (distance)	49	6	56
b	21,22,24-26	(1.636, 1.864, 1.786)	(1.280, 1.980, 1.464)	0.4938 (contact)	69	9	77
c	21,22,24-26,29	(1.548, 1.961, 0.9468)	(1.559, 1.944, 0.9316)	0.0253 (distance)	280	41	333
d	21,22,24-26,30	(1.723, 2.158, 1.059)	(1.724, 2.156, 1.06)	0.0025 (distance)	550	89	688
e	21,22,28	(1.603, 1.817, 1.741)	(1.362, 2.119, 1.619)	0.4046 (contact)	103	14	123
f	21,22,28,29	(1.776, 2.185, 1.003)	(1.778, 2.183, 1.003)	0.0035 (distance)	171	17	160
g	21,22,28,30	(1.772, 2.187, 1.009)	(1.775, 2.182, 1.011)	0.0065 (distance)	148	18	174
h	21,22,29,30	(1.739, 2.214, 0.8419)	(1.823, 2.140, 0.8434)	0.1127 (distance)	169	30	256



(a) case 8.b (b) case 8.e (c) case 8.h

Fig. 13. Detailed interference results for case 8

Figure 13 shows that although both constraint sets converge with non-violated constraints, the dot product obtains better results for the determination of the penetration between the surfaces, notice Fig. 13b, where the normal vectors are perfectly aligned. Moreover, Fig. 13a shows that the normal vectors are parallel but not collinear, showing that using the cross product leads to slightly worse results.

6. Conclusions

A methodology for contact detection and quantification between superquadric surfaces is developed in this work, by using their implicit equations.

The formulation uses the distance between the surfaces as the objective function. The constraints are introduced in the nonlinear optimization problem by means of the implicit equations of the superquadrics, which assures that the contact points lie over it. Other equality constraints use parallelism conditions between the surfaces normal vectors and the distance vector itself. The design variables are the coordinates of the contact points or the points that minimizes the distance between surfaces A and B.

Results show that the optimization technique suits for the accurate contact detection between objects modelled by implicit superquadric equations, either with the Interior Point or SQP methods. The initial estimate calculated by the minimization of two nonlinear equations reduces the number of iterations and function calls but increase computational time.

Acknowledgements

The authors gratefully acknowledge the valuable discussions had with Profs. Miguel Silva, João Folgado, Rui Ruben and José Aguilar. The authors also thank the support granted by the Portuguese Foundation for Science and Technology (FCT), with the financing of the scholarship BD/ 26013/2005.

Manuscript received by Editorial Board, December 15, 2009;
final version, April 15, 2010.

REFERENCES

[1] A. Jaklic, A. Leonardis and F. Solina, Segmentation and Recovery of Superquadrics, Kluwer, ISBN 0-7923-6601-8, 2000.
[2] A.H: Barr, Superquadrics and angle-preserving transformations, IEEE Computer Graphics and Applications, Vol. 1(1), pp. 11-23, 1981.
[3] D. Eberly, 3D Game Engine Design – A practical approach to real-time Computer Graphics, Morgan Kaufmann, 2000.
[4] G. Bergen, Collision Detection in Interactive 3D Environments, Morgan Kaufmann, 2004.
[5] P.E. Nikravesh, Computer Aided Analysis of Mechanical Systems, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[6] R. Portal and J. Dias, Multibody Dynamics Methodologies for Road Accidents. 8th World Congress on Computational Mechanics (WCCM8) e 5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008). B.A. Schrefler e U. Perego (eds.). Venice, Italy, June 30 – July 5, 2008.
[7] S. Akella, N. Chakraborty, J. Peng and J. Mitchell, Proximity Queries between Convex Objects: An Interior Point Approach for Implicit Surfaces, IEEE Transactions On Robotics, Vol. 24, No. 1, pp. 211-220, 2008.
[8] R.H. Byrd, J.C. Gilbert and J. Nocedal, A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming, Mathematical Programming, Vol. 89, No. 1, pp. 149-185, 2000.
[9] T.F. Coleman and Y. Li, An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds, SIAM Journal on Optimization, Vol. 6, pp. 418-445, 1996.
[10] R.A. Waltz, J.L. Morales, J. Nocedal and D. Orban, An interior algorithm for nonlinear optimization that combines line search and trust region steps, Mathematical Programming, Vol 107, No. 3, pp. 391-408, 2006.
[11] G.N. Vanderplaats, ADS – A Fortran Program for Automated Design Synthesis-2.01. EDO, Inc., St. Barbara, California, 1987.
[12] G.N. Vanderplaats, DOT – Design Optimization Tools, Version 3.0, VMA Engineering, Colorado Springs, 1992.
[13] J. Pombo, J. Ambrósio and M. Silva, A new wheel-rail contact model for railway dynamics, Vehicle System Dynamic, Vol. 45, No.2, pp. 165-189, 2007.
[14] M.J.D. Powell, A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations, Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Ch.7, 1970.

Detekcja kontaktu pomiędzy wypukłymi powierzchniami kawadratowymi

Streszczenie

W artykule przedstawiono metodologię wykrywania kontaktu między wypukłymi powierzchniami typu kwadryki (superkwadratowych) opisanych równaniami uwikłanymi. Po niewielkich modyfikacjach równań, można modelować uogólnione elipsoidy, hiperboloidy jedno- lub dwupowłokowe oraz toroidy. Opisana metodologia ma być zaimplementowana w programie do symulacji układów wieloczłonowych i służyć do modelowania wzajemnej penetracji układów mechanicznych, w szczególności do symulacji kolizji pojazdów mechanicznych z innymi użytkownikami dróg, takimi jak samochody, motocykle i piesi.

Zagadnienie detekcji kontaktu między dwoma ciałami jest formułowane jako wypukły, nieliniowy problem optymalizacyjny z ograniczeniami, który jest rozwiązywany przy użyciu dwu metod: metody punktu wewnętrznego (Interior Point, IP) oraz metody sekwencyjnego programowania

kwadratowego (Sequential Quadratic Programming, SQP), dostępnych odpowiednio w środowiskach MATLAB i FORTRAN. Funkcja celu podlegająca minimalizacji jest odległością pomiędzy obiema powierzchniami. Ograniczeniami optymalizacji są uwikłane równania powierzchni drugiego stopnia oraz operacje pomiędzy ich wektorami do nich normalnymi i odległością. Zmiennymi decyzyjnymi są punkty kontaktu lub punkty wyznaczające minimalną odległość pomiędzy powierzchniami. Efektywność obliczeniową można poprawić stosując metodę objętości otaczających (Bounding Volumes) we wstępnych krokach detekcji kontaktu. Najpierw aproksymuje się geometrię ciał za pomocą kul, a następnie zorientowanych prostopadłościanów otaczających (Oriented Bounding Boxes, OBB). Wyniki pokazują, że taka technika optymalizacji zapewnia dokładną detekcję kontaktu pomiędzy obiektami modelowanymi przy użyciu uwikłanych równań powierzchni drugiego stopnia.