

DYNAMICS OF MULTIBODY SYSTEMS SUBJECT TO UNILATERAL MOTION CONSTRAINTS USING A NEW NUMERICAL ALGORITHM FOR CONTACT DETECTION

Antonios Pournaras¹, Fotios Karaoulanis¹, and Sotirios Natsiavas¹

¹Department of Mechanical Engineering
Aristotle University
54 124 Thessaloniki, Greece

e-mail: anpourn@gmail.com, fotios.karaoulanis@gmail.com, natsiava@auth.gr

Keywords: Collision Detection, Unilateral Contact, Friction, Multibody Dynamics, Computational Methods.

Abstract. *In this work, some new results are presented on the dynamics of multibody mechanical systems involving contact and friction. The main contribution refers to the development of a new, systematic and accurate method for detecting contact among the components of the system. For simple geometries, this task is achieved by employing analytical means. For systems with components involving complex geometric shapes a more involved numerical methodology is developed. In both cases, once contact is detected, the common normal vector and the penetration depth is also calculated, leading to determination of the contact force and moment arising between the contacting bodies. This information is then passed to a solver, providing the full dynamic response of the system. The validity and numerical efficiency of the methodology developed is first demonstrated by considering a number of examples with relatively small geometric complexity but large traditional value and interesting dynamic response. Some new results are obtained on the dynamics of these systems. Finally, the same methodology is also tested and found to give more realistic results in more complicated mechanical applications.*

1. INTRODUCTION

Accurate and efficient prediction of dynamics of a multibody system subject to unilateral constraints is necessary in a large number of modern engineering applications [1-4]. This prediction is based heavily on suitable algorithms, leading to a fast and accurate prediction of the contact points and the forces developed there during motion. As a result, there is a continuous effort to improve the available techniques in both the contact detection and the prediction of the overall dynamics, as reflected by the vast amount of the literature devoted to the subject. The present research work belongs to this general category of investigations.

In general, the contact detection methods rely heavily on the type of geometric modeling. There are two major categories of geometric models employed currently, the polygonal and the non-polygonal. According to the first category, a solid body is divided into polygon soups (the solid is represented by a set of connected triangles) and conventional polyhedra (the geometry is represented with a polyhedron). These polyhedra may be convex or non-convex. In the category of the non-polygonal models, the body is divided into constructive solid geometry, implicit surfaces and parametric surfaces. Using the constructive solid geometry method, one can create objects from basic primitives such as spheres, blocks, cones and others, by combining them with set theoretical operations. The contact detection methods developed in the present research work use conventional convex polyhedra and basic primitives. These methods have found wide application in many diverse areas such as robotic simulations and computer games [5-8].

In particular, a two step approach has been developed in this work for the contact detection problem, referring to the computational problem of detecting intersection of two or more rigid bodies. In brief, a broad phase detection is considered first, which is typically a computationally efficient operation that yields pairs of objects that may intersect. This involves either the bounding volume approach (more specifically the axis aligned bounding boxes (AABB) for the cases of simple primitives) or the more elaborate approach of an AABB tree for the cases where a triamesh construction of a surface is involved. Next, a narrow phase detection is considered, where the individual pairs of bodies identified in the previous step as potentially intersecting are checked in a more elaborate manner. This includes simple analytical relations (as in the case of sphere to sphere or sphere to box) or more complex approaches for the general convex polytope cases. In the latter case, intersection is tested using the Gilbert-Johnson-Keerthi's algorithm, which relies on support functions to measure the shortest directed distance for the two bodies of interest by creating and iteratively updating a simplex inside the Minkowski difference. As soon as contact is detected, the corresponding contact response is resolved. This includes evaluation of the penetration depth and the contact normal as well as identification of the contact manifold. For the non trivial cases, the penetration depth and the contact normal are resolved by employing the expanding polytope algorithm. According to this algorithm, a polytope is created inside the Minkowski difference and is iteratively expanded until an edge of the Minkowski difference is hit, thereby yielding the penetration depth and the contact normal vector. Finally, for the non trivial cases a clipping algorithm is used, where the two objects are clipped in the direction of the contact normal generating the contact manifold. The geometric center of the contact manifold is assumed to be the contact point.

In the present paper, a collection of contact detection methods between bodies of specific shape are presented. This includes a two part combination of canonical shapes, like a sphere, an orthogonal parallelepiped, an ellipsoid, a conical frustum and a triangle. The algorithms used are analytical in some cases and numerical in others. In all cases, the algorithm comprises of three parts. At first, it is determined whether there is contact between two bodies. Then, the contact point, the common normal vector and the penetration depth are calculated. Finally, the contact force and moment are applied on both bodies taking into account the penetration depth,

the relative velocities of the bodies and the contact point. The last part of the algorithm is the same for all the above cases. The first two parts are different for each case and they are presented thoroughly in Sections 2 and 3 below.

Employing the aforementioned contact detection algorithm, a number of characteristic example problems was solved by using existing numerical codes for predicting the dynamic response [9, 10]. The first set of examples chosen are of academic interest and well-known in previous literature [4, 11-17]. This includes the rattleback, the Euler disk, the sphere on a turntable, the tippe-top and the woodpecker toy. A selected set of numerical results are first presented for these systems, throwing some new light into special features of their dynamics, which were not examined before. In addition, results for a more involved mechanical system [18-20], taken from an industrial application, are also obtained by using the same methodology and presented.

The organization of this paper is as follows. First, in Section 2, some simple cases of contact detection are briefly summarized. In particular, these cases refer to contact of bodies with a canonical geometric shape, which can be solved in an analytical manner. Then, in Section 3, a general contact detection methodology is presented, referring to bodies with arbitrary geometrical shapes. Next, in Section 4 some numerical results are presented for two selected mechanical systems. Finally, the most important findings of this work are summarized in the last section.

2. CONTACT DETECTION BETWEEN BODIES OF CANONICAL SHAPE

In the present section, a collection of results on contact detection between bodies with a specific canonical shape are presented. These shape combinations include the following: sphere to sphere, sphere to orthogonal parallelepiped (or simply box), sphere to ellipsoid, sphere to conical frustum and sphere to triangle.

2.1 Sphere to Sphere

This case is trivial. Let \mathbf{c}_1 and \mathbf{c}_2 be the position vectors of the two spheres' centers, while r_1 and r_2 are the corresponding radii. Then, the two spheres intersect when the distance between their centers is less than or equal to the sum of the radii. This means that $\|\mathbf{c}_1 - \mathbf{c}_2\| \leq r_1 + r_2$. In addition, the penetration depth is $d = (r_1 + r_2) - \|\mathbf{c}_1 - \mathbf{c}_2\|$, as shown in Fig. 1. Moreover, the contact point \mathbf{x} is defined by the intersection of the line that connects the centers and an arbitrarily chosen sphere. The common normal vector \mathbf{n} defines the line of impact and its direction is defined from the contact point to the sphere's center which was arbitrarily chosen.

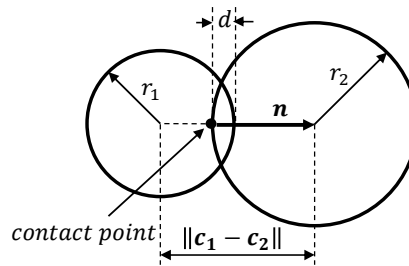


Fig. 1. Geometry of sphere to sphere contact.

2.2 Sphere to box

The intersection test is done by computing the point in the parallelepiped (or box, for brevity) which is closest to the sphere. Let the box be centered at the origin and aligned with the axes,

as shown in Fig. 2. The dimensions of the box in the x , y and z axes are $2n_1$, $2n_2$ and $2n_3$, respectively. The sphere's center is at $\mathbf{c} = (c_1, c_2, c_3)$. The point in the box closest to the sphere's center is given by

$$\mathbf{x} = (\text{clamp}(c_1, -n_1, n_1), \text{clamp}(c_2, -n_2, n_2), \text{clamp}(c_3, -n_3, n_3)) \quad (1)$$

where

$$\text{clamp}(c, a, b) = \begin{cases} a, & \text{if } c < a \\ b, & \text{if } c > b \\ c, & \text{otherwise} \end{cases} \quad (2)$$

The sphere intersects the box if the point \mathbf{x} is contained in the sphere $\|\mathbf{x} - \mathbf{c}\| \leq r$, where r is the sphere's radius. The contact point is \mathbf{x} . The penetration depth is given by $d = r - \|\mathbf{x} - \mathbf{c}\|$. The common normal vector \mathbf{n} is a vector starting from the contact point \mathbf{x} and ending at the sphere's center \mathbf{c} .

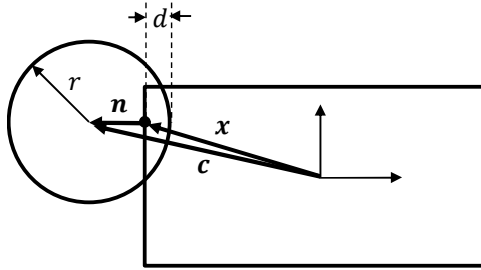


Fig. 2. Geometry of sphere to orthogonal parallelepiped contact.

2.3 Sphere to ellipsoid

For testing the intersection of a sphere with radius R and an ellipsoid with semi-principal axes a , b and c , the point of the ellipsoid closest to the sphere is calculated. Let the axes' origin be centered at the ellipsoid center, as depicted in Fig. 3. Also, let $\mathbf{p} = (p_x, p_y, p_z)$ be the sphere's center and $\mathbf{r} = (x, y, z)$ be the point in the ellipsoid closest to the sphere. In order to determine this point, the following constrained minimization problem needs to be solved

$$\min (p_x - x)^2 + (p_y - y)^2 + (p_z - z)^2 \quad (3)$$

such that

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 = 0. \quad (4)$$

For this, a Lagrange multiplier λ and a function $L(x, y, z, \lambda)$ are introduced by

$$L(x, y, z, \lambda) = (p_x - x)^2 + (p_y - y)^2 + (p_z - z)^2 + \lambda \left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 \right). \quad (5)$$

Then, the minimum of the above function is calculated by solving the following equations

$$\frac{\partial L}{\partial x} = 0, \frac{\partial L}{\partial y} = 0, \frac{\partial L}{\partial z} = 0, \frac{\partial L}{\partial \lambda} = 0 \quad (6)$$

The resulting set of nonlinear algebraic equations is solved with the Newton–Raphson method.

The sphere intersects the ellipsoid if the point \mathbf{r} is contained in the sphere $\|\mathbf{r}-\mathbf{p}\| \leq R$. The contact point is \mathbf{r} . The penetration depth is given by $d = R - \|\mathbf{r}-\mathbf{p}\|$. The common normal \mathbf{n} vector is a vector starting from the contact point \mathbf{r} and ending to the sphere's center \mathbf{p} .

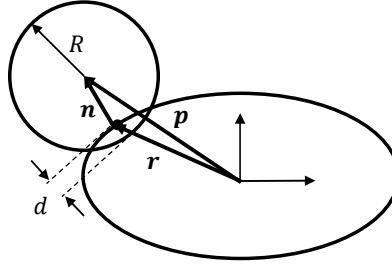


Fig. 3. Geometry of sphere to ellipsoid contact.

2.4 Sphere to conical frustum

This case leads to an axisymmetric problem with axis being the frustum's axis, as shown in Fig. 4. By taking the plane which is defined by the frustum's axis and the sphere's center, the problem becomes equivalent to determining the intersection between a trapezoid and a circle. The coordinate system origin is placed at the center of the bottom face of the frustum. For testing the intersection, a point \mathbf{x} in the trapezoid which is closest to the sphere is calculated. The space outside the trapezoid is divided into five sections and according to which section the sphere's center \mathbf{p} belongs to, a different analytical method of calculating the closest point is applied. Then, the sphere intersects the frustum if the point \mathbf{x} is contained within the sphere $\|\mathbf{x}-\mathbf{c}\| \leq r$, where r is the sphere's radius. The contact point is \mathbf{x} . The penetration depth is given by $d = r - \|\mathbf{x}-\mathbf{c}\|$. The common normal vector \mathbf{n} is a vector starting from the contact point \mathbf{x} and ending to the sphere's center \mathbf{c} .

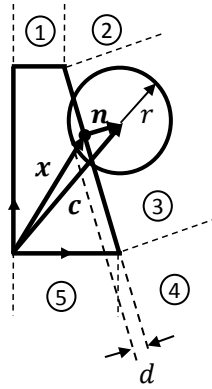


Fig. 4. Geometry of sphere to conical frustum contact.

2.5 Sphere to triangle

The intersection test is done by computing the point \mathbf{x} in the triangle which is closest to the sphere. Let the coordinate system origin be placed at the center of the sphere, as shown in Fig. 5. Let \mathbf{p}_0 , \mathbf{p}_1 and \mathbf{p}_2 be the vertices of the triangle. The sphere's center is projected on the triangle's plane. This plane is divided into seven sections and according to which section the sphere's projected center belongs to, a different analytical method of calculating the closest point is applied. Then, the sphere intersects the triangle if the point \mathbf{x} is contained in the sphere

$\|\mathbf{x}\| \leq r$, where r is the sphere's radius. The contact point is \mathbf{x} . The penetration depth is given by $d = r - \|\mathbf{x}\|$. The common normal vector \mathbf{n} is a vector starting from the contact point \mathbf{x} and ending to the sphere's center.

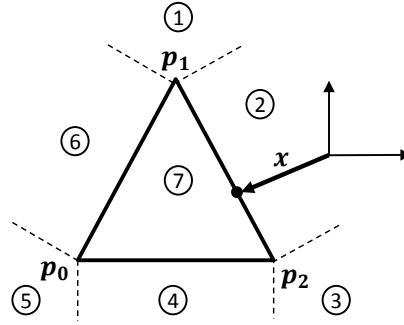


Fig. 5. Geometry of sphere to triangle contact.

3. CONTACT DETECTION BETWEEN BODIES OF GENERAL SHAPE

For the general case of two convex polytopes, the contact detection problem is resolved by following the algorithmic treatment presented in this section. The computational problem is assumed to be defined given a pair of convex shapes and the spatial relation between them. The above treatment is currently employed for cases including the rectangular cuboid (or simply box), the conical frustum and the ellipsoid, giving a total of six different contact combinations. The proposed approach is split into the following three steps.

1. The *intersection testing*, which refers to a Boolean query that is true if and only if the intersection of two shapes is nonempty [8]. The intersection is tested within this work, using the Gilbert-Johnson-Keerthi's algorithm [5, 6].
2. The *evaluation of the penetration depth and the contact normal*. As soon as the intersection testing query yields true, the corresponding penetration depth and the contact normal are evaluated using the Expanding Polytope Algorithm (EPA) [5]. The penetration depth and the contact normal can then be used for the estimation of the corresponding contact force/moment vector.
3. The generation of the contact manifold, i.e., the set of contacting points [6], which can be used for the contact force/moment vector application. An implementation based on clipping is proposed for all cases except those employing an ellipsoid. For the latter, the geometric properties of the ellipsoid are exploited.

3.1 Intersection testing

The intersection is tested using the Gilbert-Johnson-Keerthi's (GJK) algorithm. GJK is an iterative method, initially proposed for computing the distance between complex objects in three-dimensional space [5]. It can, however, be also employed to solve different proximity queries including ones that are of interest in the collision detection problem. GJK is quite popular in this field, mainly because it is very versatile (only a set of vertices describing the convex polytope can be used) and is extremely efficient [6].

Overall, GJK relies on support functions to measure the shortest directed distance between the two convex polytopes of interest, say A and B . It does so, by creating and iteratively updating a simplex inside the Minkowski difference set $A-B$, with $A-B = \{\mathbf{a}-\mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\}$. A simplex is defined as the convex hull of an affinely independent set of vertices. Up to four vertices can be employed, so a simplex can be a single point, a line, a triangle or a tetrahedron.

In each iteration, the updated simplex lies closer to the origin than the previous one. When the origin of the Minkowski difference is included in the simplex, the two shapes are determined to intersect.

```

function EPA:
    D ← arbitrary vector
    v ← A.Support(d) - B.Support(-d)
    W ← {v}
    do:
        w ← A.Support(v) - B.Support(-v)
        if v·w > 0:
            return false
        W ← ProjectOrigin(W ∪ v)
    until |W| = 4 or ||v|| < tol
    return true
    
```

Table 1. The basic steps of the GJK algorithm

In the above, `Support` provides the support mapping

$$S_A(\mathbf{v}) = \{\mathbf{p}: \max\{\mathbf{v} \cdot \mathbf{a}: \mathbf{a} \in A = \mathbf{v} \cdot \mathbf{p}, \mathbf{p} \in A\}\}.$$

Given a direction \mathbf{v} this returns the points \mathbf{p} in the shape A that would be the first to come in contact with a plane with normal \mathbf{v} that moves from infinitely far away towards the center of A . Furthermore, `ProjectOrigin` returns the point on a point (obvious solution), line or triangle which is the closest to the origin.

3.2 Penetration depth and contact normal

The penetration depth and the contact normal are found using the Expanding Polytope Algorithm (EPA). EPA is an accurate algorithm used for computing the penetration depth between a pair of shapes A and B , having a non-empty intersection. Similar to the GJK, EPA employs only the support mappings of the corresponding shapes, further contributing to the versatility of the overall approach.

Starting point for the EPA is a polytope that contains the origin of the Minkowski difference. The simplex provided by GJK is used to generate the polytope required. The GJK may consist of only a point, a line or a triangle. In this case, proper initialization of the polytope is performed, following the approach described in [6].

As soon as a valid simplex is build inside the Minkowski difference set, then this is iteratively expanded into a convex polytope. The algorithm proceeds in each step so that the face closest to the origin is the one that is expanded. The iterations stop when an edge of the Minkowski difference is recovered during expansion (within some numerical accuracy), thereby yielding the penetration depth (and the corresponding normal vector), as equal to the distance from the origin to this face.

```

function EPA:
    P ← A convex polytope containing the origin
    Q ← ∅
    for each face f in P:
        e ← construct(f)
        if e.closest_internal():
            Q.push(e)
    
```

```

 $\mu \leftarrow \text{infinity}$ 
do:
     $e = Q.\text{pop}()$ 
    if not  $e.\text{obsolete}$ :
         $v \leftarrow e.v$ 
         $w \leftarrow A.\text{Support}(v) - B.\text{Support}(-v)$ 
         $\mu \leftarrow \min(\mu, (v \cdot w)^2 / ||v||^2)$ 
         $\text{close\_enough} \leftarrow \mu \leq (1 + \text{tol})^2 ||v||^2$ 
        if not  $\text{close\_enough}$ :
             $e.\text{obsolete} = \text{true}$ 
             $E \leftarrow \emptyset$ 
            for  $i \leftarrow 0, 1, 2$ :
                 $\text{sihlouette}(e.\text{adj}_i, e.j_i, w, E)$ 
            for  $se, i \in E$ :
                 $ne \leftarrow \text{construct}(se.y_{i \oplus 1}, se.y_i, w)$ 
                if  $ne.\text{affinely\_dependent}$ :
                    return  $||v||$ 
                if  $ne.\text{closest\_internal}()$ 
                    and  $||v||^2 \leq |ne \cdot v| \leq \mu$ :
                         $Q.\text{push}(ne)$ 
until  $\text{close\_enough}$  or  $Q.\text{empty}()$  or  $||Q.\text{pop}().v||^2 > \mu$ 
return  $||v||$ 
    
```

Table 2. The algorithm in [6]

In the above Q is a priority list and μ is an upper bound for the squared penetration depth. Entry is a structure for the polytope face, that holds the vertices (y), the adjacent entries (adj), the indices of the corresponding adjoining edges (j) and obsolete which is a boolean that identifies if this face has become obsolete during a previous iteration. Furthermore \oplus denotes the addition modulo 3.

Once an entry e that contains the point closest to the origin, point w is calculated as described above. The front side of entry e can always be seen from the w , so that e is marked obsolete . Then, the polytope is recursively traversed starting with each entry adjacent to e . All the entries that can be seen from e are further marked as obsolete . This procedure is provided by function sihlouette , which is based on a recursive flood filling algorithm.

```

function  $\text{sihlouette}(e, i, w, E)$ :
    if not  $e.\text{obsolete}$ :
        if  $e.v \cdot w < ||e.v||^2$ :
             $E \leftarrow \{E, (e, i)\}$ 
        else:
             $e.\text{obsolete} \leftarrow \text{true}$ 
             $\text{sihlouette}(e.\text{adj}_{i \oplus 1}, e.j_{i \oplus 1}, w, E)$ 
             $\text{sihlouette}(e.\text{adj}_{i \oplus 2}, e.j_{i \oplus 2}, w, E)$ 
    
```

 Table 3. Pseudocode for function sihlouette

3.3 Contact manifold generation

The contact manifold, which usually is further reduced to a single point where the contact force is to be applied, is generated using a clipping algorithm. For this case, the faces of the shapes A and B should be explicitly defined. This is a major difference from the algorithmic

treatment described so far, that required only the corresponding support mappings. Furthermore, this procedure may introduce accuracy issues when the shape is not defined from flat faces only. This is the reason why the contact point in the ellipsoid is not generated by this method, but by exploiting its geometric property of the normal uniqueness.

As a first step, the reference face is identified, which is the face normal to the contact vector (or its opposite), as returned by the EPA. An incident face is then identified on the other shape, which is the one that is closer to the opposite of the vector used for the reference face. Then, a set of clippings is performed between the incident face and the faces that are adjacent to the reference one. Each clipping reduces the size of the contact manifold, which is initially set equal to the incident face. A final clipping is performed in the remaining area by the reference face.

```

function clipping:
    reference ← A.find_reference(v)
    incidence ← B.find_incidence(v)
    if not reference:
        reference ← B.find_reference(-v)
        incidence ← A.find_incidence(-v)
    manifold ← incidence.vertices()
    for f ← reference.faces():
        manifold ← manifold.clip(f)
    manifold ← manifold.translate(|v|)
    manifold ← manifold.clip(reference)
    
```

Table 4. Pseudocode for the clipping function

The contact point can then be assumed as the geometrical center of the contact manifold.

4. NUMERICAL RESULTS

Employing the aforementioned contact detection algorithms, a number of characteristic example problems was solved by using an existing numerical codes for predicting the dynamic response [9]. The first example chosen is of academic interest and is known in literature as the woodpecker toy [15-17]. In addition, results for a more complex mechanical system, involving ball bearings, were also obtained by using the same methodology [18-20].

4.1 Results for the woodpecker toy

The woodpecker toy shown in Fig. 6 consists of a pole fixed to the ground, a sleeve, a torsional spring and a woodpecker. The sleeve and the woodpecker are connected with a revolute joint and a torsional spring. The hole in the sleeve is slightly larger than the diameter of the pole. The toy's response is heavily affected by friction and impact phenomena. The woodpecker moves down due to gravity with a pitching motion and impacts the pole while the sleeve acts as a jamming mechanism with the presence of friction. The system oscillates and may end up in a stable limit cycle.

The first scientific studies on the dynamics of this toy, based on application of non-smooth and contact mechanics theory, was presented in [15-17]. The model is planar and consists of three rigid bodies: the woodpecker, the sleeve and the pole. The sleeve's lateral movement is small so it is neglected. Another major assumption is that both the woodpecker and the sleeve rotate with small angle, so the gap functions are calculated with linearized kinematics.

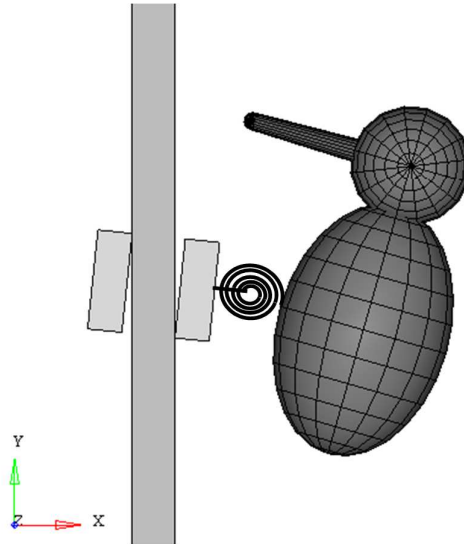


Fig. 6. The woodpecker toy.

In the present paper, the model is created using basic solids. Specifically, the pole is represented by an orthogonal parallelepiped, the sleeve by two orthogonal parallelepipeds, the body of the woodpecker by an ellipsoid, its head by a sphere, its beak by a conical frustum and a sphere at the tip (see Fig. 6). There exist also a revolute joint and a torsional spring between the sleeve and the woodpecker. Contact is taken into account between the beak's tip and the pole (sphere to box) and between the sleeve and the pole (box to box). Values for the geometry, masses, moments of inertia and contact parameters are taken from [16]. Here, the model is also planar. However, the sleeve's lateral movement is taken into account, although it does not make any significant difference. The major difference with [16] is that the rotations of both the sleeve and the woodpecker are not assumed to be small, hence the kinematics is nonlinear. The above assumptions are critical and play a significant role to the system's response. By examining the results of [5], it can be seen that the woodpecker's and sleeve's extremum angles are 30 and 6 degrees respectively. The woodpecker's rotation is far too large to be linearized, whereas the sleeve's angle just passes the limit.

Comparing the results obtained here with those presented in [16] it is concluded that the linearization based on small rotations influences the results especially in the woodpecker's angle where the upper and lower limits are slightly greater and smaller respectively (see Fig. 7a). Regarding the sleeve's angle there are no significant differences because for such an angle range the linearization is a good approximation (Fig. 7d). Similar behavior is evident concerning the angular velocities. Examining these quantities, it is evident that a slight variation appears at the upper and lower limits for both the woodpecker and the sleeve (Figs. 7c and 7d).

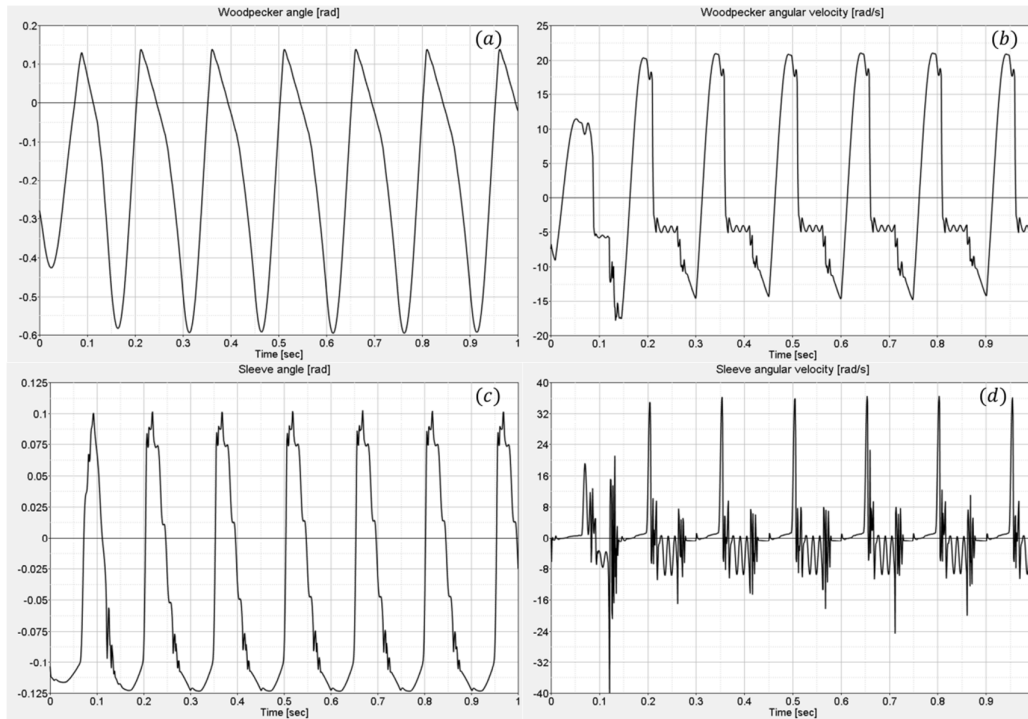


Fig. 7. Numerical results for the woodpecker toy. Time history of the woodpecker: (a) rocking angle, (b) angular speed, (c) sleeve angle and (d) sleeve angular speed.

4.2 Results for a mechanism with a rolling bearing

Dynamics of ball bearings has been studied from many different angles of view, including lubrication, wear, vibration and sound. In some cases, a multibody model was used. The most preferred modeling assumptions are the following: no slipping of the balls during rolling on the surface (the ball motion is defined by a kinematic constraint), no friction between the balls and the bearing's rings, planar motion and negligible mass of the balls [18, 19]. To investigate whether these assumptions are reasonable, development and consideration of more realistic models that remove the above assumptions is needed.

In the present paper, a model for a ball bearing (SKF BB1B420205) is first developed. Then, this is used in a crank-rod-slider mechanism to represent the revolute joint between the crank and the rod, as shown in Fig. 8a. Regarding the ball bearing modeling, the balls are modeled by eight spheres, while the inner and outer rings and the cage are modeled by triangle meshed solids (see Fig. 8b). Moreover, contact is enabled between each sphere and both rings, while sphere to sphere and inner to outer ring contact is not taken into account. Although contact between the cage and the balls could also be studied, it was chosen not to do it here because the loads arising between the cage and the balls are not significant. Another reason for this is that the model becomes much more complicated, resulting in additional computer simulation time with no significant gain. To prevent the balls from coming into contact, eight inline constraints are added between each ball's center and a specific coordinate system on the cage (see Fig. 8c). Concerning other major connections, the cage is connected to the inner ring's center with a revolute joint, the outer ring is fixed to the crank, while the inner ring is fixed to the rod. Values for the geometry, masses, moments of inertia and contact parameters are taken from [19]. The evaluation of the contact forces is based on Herzan contact deformation theory that

takes into account the geometry and the material properties. The methodology for calculating the stiffness values for the contact is described thoroughly in [20].

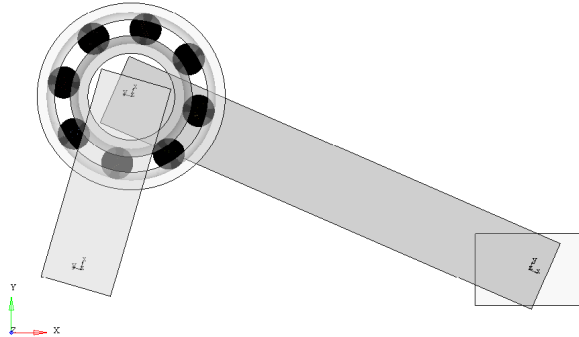


Fig. 8a. Mechanism



Fig. 8b. Ball bearing model

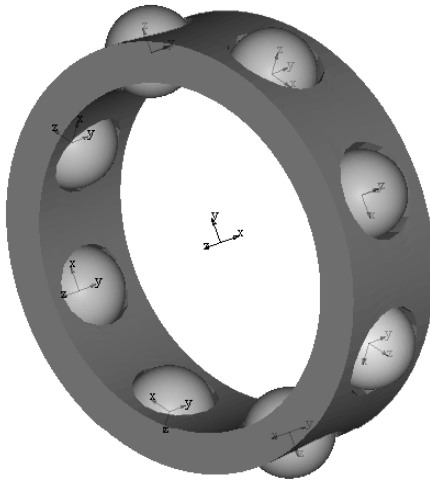


Fig. 8c. Cage model

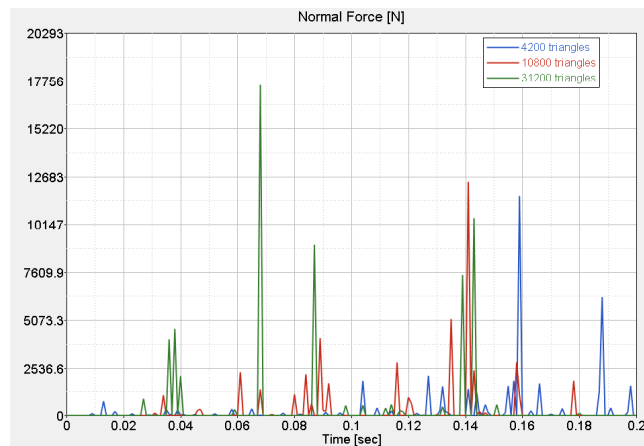


Fig. 8d. Normal force on a ball

By implementing the contact detection algorithm between sphere and triangle presented in Section 2, the mechanical model developed was simulated, assuming that the crank rotates with a constant angular speed of 300 rpm. In Fig. 8d is depicted the normal force acting from the outer ring to a ball for three different meshes. Although the maximum force is congruent with that presented in [19], the time history is completely different. This is believed to be due to the lack of friction in the model employed in [19], where the balls movement is induced by imposing a kinematic constraint, taking into account that the balls roll without sliding. In reality, this is not true and causes the balls to be at a different position.

5. SYNOPSIS

A new contact detection methodology was presented, leading to an accurate and numerically efficient determination of the dynamic response of a class of multibody engineering systems involving contacts and friction. The methodology developed allows solution to general

mechanical problems, without having to formulate the equations of motion and the gap functions in each case, separately. Furthermore, the new algorithm takes into account large relative rotations between two solid bodies and does not suffer from restrictions caused by kinematic linearization. In particular, the contact detection algorithm between a sphere and a triangle allows to create more detailed and realistic models for ball bearings or any machine element involving contact between a sphere and a general surface (e.g., constant velocity joint). Furthermore, it allows incorporation of friction using either Amontons–Coulomb or even more sophisticated friction laws. Therefore, it is expected that this tool can prove useful to engineers who want to study various phenomena arising in complex mechanical systems with components involving contact and friction. Some of these advantages were illustrated in this work by presenting results for the classical woodpecker toy and a more complex systems involving ball bearings.

REFERENCES

- [1] M. Geradin, A. Cardona, *Flexible Multibody Dynamics: A Finite Element Approach*, John Wiley & Sons, New York, 2001.
- [2] O.A. Bauchau, *Flexible Multibody Dynamics*, Springer Science+Business Media B.V., London, 2011.
- [3] B. Brogliato, *Nonsmooth Mechanics* (2nd edition), Springer-Verlag, London, 1999.
- [4] C. Glocker, *Set-Valued Force Laws, Dynamics of Non-Smooth Systems*, Springer, Berlin, 2001.
- [5] E. Gilbert, D. Johnson, S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space, *IEEE Journal of Robotics and Automation* 4 (1988) 193–203.
- [6] G. van den Bergen, *Collision Detection in Interactive 3D Environments*, Morgan Kaufmann Publishers, San Francisco, 2004.
- [7] C. Ericson, *Real-Time Collision Detection*, Morgan Kaufmann Publishers, San Francisco, 2005.
- [8] L. Olvång, *Real-Time Collision Detection with Implicit Objects*, Department of Information Technology, Uppsala University, Sweden, 2010.
- [9] MotionSolve v11.0, User Guide, Altair Engineering Inc., Irvine, California, USA.
- [10] MSC ADAMS, User Guide, MSC Software Corporation, California, USA.
- [11] C. Studer, *Numerics of Unilateral Contacts and Friction*, Springer-Verlag, Berlin, 2009.
- [12] R.I. Leine, C. Glocker, A set-valued force law for spatial Coulomb–Contensou friction, *European Journal of Mechanics A/Solids* 22 (2003) 193–216.
- [13] K. Weltner, Stable circular orbits of freely moving balls on rotating discs, *Am. J. Phys.* 47 (1979), 984–986.
- [14] F. Pfeiffer, C. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley-VCH, Weinheim, 2004.
- [15] C. Glocker, C. Studer, Formulation and preparation for numerical evaluation of linear complementarity systems in dynamics, *Multibody System Dynamics* 13 (2005) 447–463.
- [16] R.I. Leine, D.H. van Campen, C. Glocker, Nonlinear dynamics and modeling of various wooden toys with impact and friction, *Journal of Vibration and Control* 9 (2003) 25–78.
- [17] P. Kessler, O. M. O’ Reilly, The ringing of Euler’s disk. *Regural and Chaotic Dynamics* 7 (2002) 49–60.
- [18] V.N. Patel, N. Tandon, R.K. Pandey, A dynamic model for vibration studies of deep groove ball bearings considering single and multiple defects in races. *Journal of Tribology* 132 (2010) 1193–1201.

- [19] L.X. Xu, Y.H. Yang, Y.G. Li, C.N. Li, S.Y. Wang, Modeling and analysis of planar multibody systems containing deep groove ball bearing with clearance, *Mechanism and Machine Theory* 56 (2012) 69-88.
- [20] T.A. Harris, *Rolling Bearing Analysis*, Wiley, New York, 2001.