Multibody System Dynamics **10:** 239–262, 2003.
© 2003 *Kluwer Academic Publishers. Printed in the Netherlands.*

Sliding Contact between Freeform Surfaces

A. TASORA and P. RIGHETTINI
Politecnico di Milano, Dipartimento di Ingegneria Elettrotecnica, P.zza L. da Vinci 32, I-20133 Milano, Italy; E-mail: {tasora,righettini}@mech.polimi.it

(Received: 31 July 2001; accepted in revised form: 28 April 2003)

Abstract. This paper deals with the sliding-contact constraint equations describing the relative motion of two freeform surfaces, assuming that the surfaces can have arbitrary curvature in three-dimensional space. The sliding-contact equations are developed either for the non-penetration condition and for the surface-tangency condition. Both are differentiated twice in time in order to allow a straightforward application to dynamic and kinematic multibody simulation within the context of an augmented Lagrangian approach. This formulation represents the contact constraint by means of a *sliding tangent plane*, hence exploiting the advantageous optimizations of the so called *lock formulation*.

Key words: multibody systems, surfaces, contact, collision.

1. Introduction

Modern multibody software may be requested to perform simulations involving the contact between arbitrarily-shaped surfaces: these are the higher-pair joints which are extensively used in applied mechanics. Cam-follower mechanisms are notable examples of such joints, where the contact does not happen between cylindrical or prismatic surfaces, as in lower-pair joints, but happens instead along a line or a point.

Several methods for lower-pair joints (cylindrical joints, revolute joints, etc.) have been proposed and studied in multibody dynamics literature, but not so many methods have been discussed for a general-purpose approach to the problem of the contact between freeform surfaces.

Unfortunately, the analytical description of the kinematics of higher-pair joints may require complex formalisms, especially whenever the contact happens between two freeform surfaces in three-dimensional space (Figure 1). This may be the case of the contact between wheel and railway, or in spatial cams.

The kinematics of two rigid bodies subject to sliding contact is complicated by the fact that the curvature of their surfaces is liable of mutual accelerations; moreover these curvatures could be non-uniform as in the example of cams [1].

This problem has been already investigated by some researchers, for example Balling [2] recently suggested a method which fits well into whatever multibody formalism which is based on joint-coordinates: in that paper the contact point

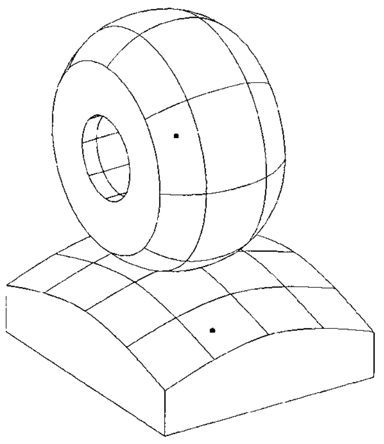

Figure 1. Example of contact between free-form rigid bodies.

becomes a joint of the kinematical chain of rigid bodies, and its coordinates are the four *u,v* parameters of the two surfaces.

Instead, our method is rather targeted at multibody software based on the Lagrangian approach, where the constraints are added by means of Lagrange multipliers, and the coordinates of the equations are the natural coordinates of the rigid bodies. This, of course, implies that a fast and efficient formalism must be defined in order to compute the contact constraint equations as well as their derivatives and their Jacobians (which will be used to solve the DAE differential-algebraic system, as explained in [3]).

Given that a fast and efficient way to handle the 'point on a flat plane' basic constraint has already been developed and tested within the framework of the *lock formulation* [4], we managed to extend its capabilities to the case of contact between surfaces.

In fact, we can represent the contact constraint by introducing an auxiliary tangent plane which moves between the two bodies as they slide. If one manages to compute the exact alignment and position of the tangent plane as function of body states during the motion, the sliding-contact constraint can be expressed by means of a simple 'point on a flat plane' constraint, where the flat reference plane belongs to one of the two bodies, and the point belongs to the other body. Note that both the point and the plane must have specific relative motions respect to their bodies, because the plane must stay tangent to the shifting contact point, and these kinematical contributions can be easily applied to the 'point on plane' constraint as described in the *lock formulation* (where arbitrary speed/accelerations can be freely imposed to the references used to describe the plane and the sliding point).

2. The Sliding Plane Approach

Let us consider two rigid bodies $O1$ and $O2$ being in contact, under the simplificative hypothesis of existence and non-singularity of the point of contact (multi-point contacts and degenerate situations of the type surface-surface or line-surface are not taken into consideration)

Note that, for the moment, the hypothesis of unilateral constraint is not imperative, therefore we will deal with bilateral contact for sake of simplicity. Here we will not discuss the problems of non-smooth dynamics and impacts, which are investigated for example in [5] or [6].

It can be shown that the contact is geometrically correct if two conditions are satisfied at once: the two surfaces must have a point in common, and the tangent planes in that point must be the same.

Say $\mathbf{P}_{O1}$ is the point of contact on surface of body $O1$, and $\mathbf{S}_{O2}$ is the point of contact on surface of body $O2$, both vectors being expressed in absolute coordinate system.

The first constraint equation implies that $\mathbf{P}_{O1}$ and $\mathbf{S}_{O2}$ must coincide in space, that is:

$$\mathbf{C}_{ps} = \mathbf{P}_{O1} - \mathbf{S}_{O2} = \mathbf{0}. \tag{1}$$

Now, say $\mathbf{n}_{O1}$ is the unit-length normal to the surface of body $O1$ at the point of contact $\mathbf{P}_{O1}$, and $\mathbf{n}_{O2}$ is the unit-length normal to the surface of body $O2$ on the point of contact $\mathbf{S}_{O2}$. Both vectors are expressed in the same coordinate system.

This leads us to the second constraint equation, which requires that the two surfaces must be tangent at the contact point, hence the normals must be aligned:

$$\mathbf{C}_{n} = \mathbf{n}_{O1} + \mathbf{n}_{O2} = \mathbf{0}. \tag{2}$$

We assume that the position of point $\mathbf{P}_{O1}$ on surface of body $O1$ can be expressed (at least, locally) as a function of two curvilinear coordinates u_{O1}, v_{O1}, and the same for point $\mathbf{P}_{O1}$, whose position on surface can be a function of two curvilinear coordinates u_{O2}, v_{O2}. Hence, satisfying Equations (1) and (2) implies a system of nonlinear equations

$$\mathbf{C}_{ps,n} = \mathbf{C}(q_{O1}, q_{O2}, u_{O1}, v_{O1}, u_{O2}, v_{O2}, t) = \mathbf{0}. \tag{3}$$

which must be solved either for the positions of the two bodies (the coordinates q_{O1}, q_{O2}), either for the auxiliary variables $u_{O1}, v_{O1}, u_{O2}, v_{O2}$.

Note that one of the three scalar constraints of Equation (2) is redundant (because unit norm of normals is implied, $\| \mathbf{n}_{O1} \| = \| \mathbf{n}_{O2} \| = 1$), therefore the complete system of constraints equation (3) has $3 + (3 - 1) = 5$ scalar equations. Meanwhile, four auxiliary variables $u_{O1}, v_{O1}, u_{O2}, v_{O2}$ were added: hence the contact effectively subtracts $5 - 4 = 1$ degree of freedom from the mechanical system, an intuitive result which is also confirmed by many authors dealing with classical mechanical problems [7].

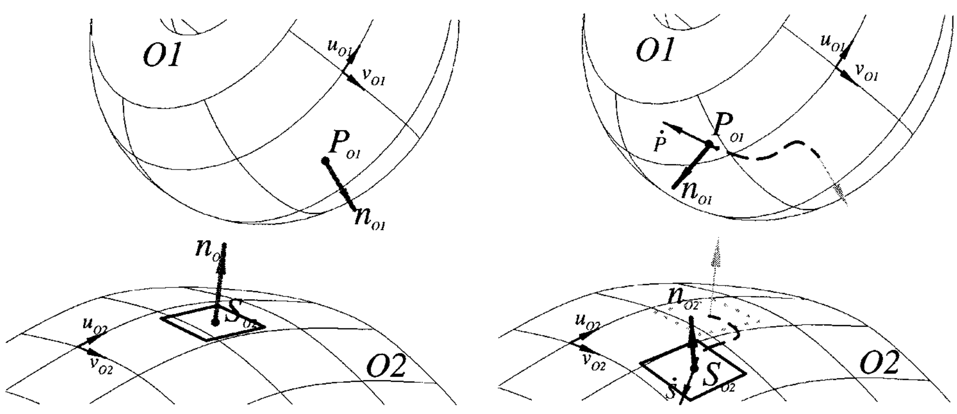

Figure 2. The bodies are taken apart to show the contact point on body $O1$ and the sliding plane on body $O2$. Contact happens for coincident $\mathbf{P}_{O1}$ and $\mathbf{S}_{O2}$, and for aligned normals $\mathbf{n}_{O2}$ and $\mathbf{n}_{O1}$. Contact point $\mathbf{P}_{O1}$ and the sliding plane must move on their surfaces during relative body motion, so that $\mathbf{P}_{O1}$ and $\mathbf{S}_{O2}$ share the same speed and position.

The introduction of four auxiliary variables in the state vector of our system, as well as the description of the contact by way of the five-dimensional equation (3), of course adds unwanted complication into our multibody formalism and may have a negative impact on the performance of the simulation code.

Therefore one may want to reduce the system to a more handy formulation, where only a single scalar constraint equation is added, and the four auxiliary variables can be recovered afterward as dependent variables (i.e. only rigid body coordinates are introduced in state system, while u_{O1}, v_{O1}, u_{O2}, v_{O2} variables and their derivatives are computed separately).

An effective way to accomplish this task may be represented by the *sliding plane* approach, which we discuss in this paper. Such method introduces a ‘point on plane’ constraint between the two contacting bodies, which is responsible of reducing the degrees of freedom of the system by one unit. During the multibody simulation, the position of the reference plane is continuously moved tangentially to the surface of a body (as well as the reference point continuously moves on the surface of the other), thus updating the auxiliary variables u_{O1}, v_{O1}, u_{O2}, v_{O2} and their derivatives as dependent variables (Figure 2).

Since there is no need to add the variables u_{O1}, v_{O1}, u_{O2}, v_{O2} in system’s state vector, the solution of kinematic and dynamic problems is somewhat simple: it just requires the implementation of a holonomic constraint of the type ‘point on a plane’, where the position/speed/acceleration of both the reference point and reference plane can be imposed. This is easily achieved, for example, through the *lock formulation* approach, formulated in [4] and briefly discussed in the next section.

Furthermore, as a positive side effect of this approach, the orthogonal contact force is effortlessly recovered from the Lagrangian multiplier of the ‘point on plane’ constraint.

However, special attention must be paid to the problem of computing the rheonomic terms which are required by the *lock formulation*, as they will be responsible of the acceleration effects caused by surface curvature. In other words, one must know not only the position but also the speed of the contact point as a function of body states, in order to set proper values for position/speed/acceleration of both the reference point and reference plane. Section 4 will deal with this problem.

3. Basic Point-Plane Constraint via 'Lock Formulation'

The so called *lock formulation* relies heavily on quaternion algebra and offers a compact yet efficient way to implement the derivations of constraint equations, where most common holonomic and rheonomic constraints can be inherited from a single formalism. Moreover, the Jacobians are obtained analytically, and this has a positive impact on the performance of multibody simulations based on the Lagrangian approach.

Let us consider two generic rigid bodies $O1$ and $O2$, both with two auxiliary coordinate systems P_{O1} and S_{O2} (the so called 'markers') whose body-relative positions and body-relative rotations can be constant or imposed via time-functions (Figure 3).

One can impose a translation constraint on the relative position of P_{O1} respect to the coordinate system of S_{O2}: this is the 'positional' constraint. Also, one can impose a rotation constraint on the relative rotation of P_{O1} respect to S_{O2}, in the coordinate system of S_{O2}, and this is the 'rotational' constraint.

If needed, both the positional and the rotational constraints can be expressed with time-dependent functions, as well as the relative positions and relative alignments of markers respect to their rigid bodies.

The effect of these positional and rotational constraints between P_{O1} and S_{O2} is a kind of welding between the two bodies, hence the name *lock formulation*. This is expressed by six scalar equations. However, if one suppresses one or more scalar conditions, the constraint gets some degrees of freedom and turns into specific holonomic constraints.

For example, a spherical joint is obtained by suppressing all the three scalar components of the rotational constraints, and a cylindrical joint is obtained by suppressing, for example, the Z positional component and the Z rotational component. In the same way we can obtain lot of other holonomic constraints, for instance the prismatic guide, the point-on-line condition, the point-on-plane condition (used extensively in this article), the Cardano joint, the revolute joint, the parallelism condition, etc. Also, by setting adequate time-dependant functions in the rheonomic terms of the equations, one can get whatever kind of actuators, motors, motion laws, imposed trajectories, etc.

A simplified and compact version of the *lock formulation* is described below, as a quick reference, but advanced details and implementation issues are described extensively in [4].

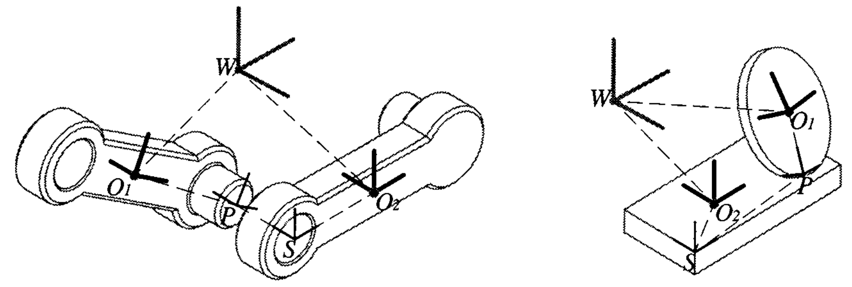

Figure 3. Reference frames which are used to build constraints with the *lock formulation* method (two examples).

Let us introduce the following notation:

- $[\Lambda_r] = [\Lambda_r(\mathbf{q}_{\theta_r})]$ is a generic rotation matrix, function of a quaternion $\mathbf{q}_\theta$,
- $\mathbf{q}_{xO1}$, $\mathbf{q}_{xO2}$, $\mathbf{q}_{\theta O1}$, $\mathbf{q}_{\theta O2}$, are the position-coordinates and rotation-coordinates of bodies $O1$ and $O2$, where rotations are expressed as unit quaternions $\mathbf{q}_\theta$,
- $\mathbf{q}_{xP}$, $\mathbf{q}_{xS}$, $\mathbf{q}_{\theta P}$, $\mathbf{q}_{\theta S}$ are the coordinates (positions and rotations) of markers P_{O1} and S_{O2} respect to their bodies, $O1$ and $O2$, and may be user-imposed functions of time. Example: absolute origin of P_{O1} is $\mathbf{P}_{O1} = \mathbf{q}_{xO1} + [\Lambda_{O1}]\mathbf{P}_{O1,O1}$, where body-relative position can be set by means of a function $\mathbf{P}_{O1,O1} = \mathbf{q}_{xP}(t)$,
- $[Gl_{O1}]$ is a 3×4 rectangular matrix, function of the quaternion $\mathbf{q}_{\theta_{O1}}$, so that $[Gl_{O1}]\dot{\mathbf{q}}_{\theta_{O1}} = \omega_{O1}$, as described in [3],
- $[\tilde{a}]$ is a skew symmetric matrix such that $[\tilde{a}]\mathbf{b} = \mathbf{a} \wedge \mathbf{b}$,
- $\mathbf{q}_{x\Delta} = \mathbf{f}(t)$ and $\mathbf{q}_{\theta\Delta} = \mathbf{f}(t)$ are the imposed translation and rotation between P_{O1} and S_{O2}, in coordinates of S_{O2}.

Hence the positional constraint can be written as:

$$\mathbf{C}_x = \mathbf{q}_{x_{P-S,S}} - \mathbf{q}_{x_\Delta} = \mathbf{0} \tag{4}$$

$$\mathbf{C}_x = [\Lambda_S]^T[\Lambda_{O2}]^T\left(\left(\mathbf{q}_{x_{O1}} + [\Lambda_{O1}]\mathbf{q}_{x_P}\right) - \left(\mathbf{q}_{x_{O2}} + [\Lambda_{O2}]\mathbf{q}_{x_S}\right)\right) - \mathbf{q}_{x_\Delta} = \mathbf{0}. \tag{5}$$

In order to obtain the Jacobian matrix $[C_q]$ and the vector $[\mathbf{Q}_\mathbf{x}]$, which are extensively used in Lagrangian dynamics through the equation

$$\ddot{\mathbf{C}}_x = [C_q]\ddot{\mathbf{q}} - \mathbf{Q}_\mathbf{x} = \mathbf{0}; \tag{6}$$

one can differentiate Equation (5) twice. Then, after some algebraic manipulations, the analytical expression for the Jacobian of the *lock formulation* constraint (translational part) can be written in the following fashion:

$$[C_{q_x}] = \left[[C_{q_x}]_{xO1}\,[C_{q_x}]_{\theta O1}\,[C_{q_x}]_{xO2}\,[C_{q_x}]_{\theta O2}\right] \tag{7}$$

where

$$\begin{aligned}
\left[C_{q_x}\right]_{xO1} &= +[\Lambda_S]^T[\Lambda_{O2}]^T \\
\left[C_{q_x}\right]_{\theta O1} &= -[\Lambda_S]^T[\Lambda_{O2}]^T[\Lambda_{O1}][\tilde{\mathbf{q}}_{xP}][Gl_{O1}] \\
\left[C_{q_x}\right]_{xO2} &= -[\Lambda_S]^T[\Lambda_{O2}]^T \\
\left[C_{q_x}\right]_{\theta O2} &= +[\Lambda_S]^T[\Lambda_{O2}]^T[\Lambda_{O1}][\tilde{\mathbf{q}}_{xP}][Gl_{O1}] \\
&\quad + [\Lambda_S]^T\left[\widetilde{[\Lambda_{O2}]^T\mathbf{q}_{xP-S,W}}\right][Gl_{O2}]
\end{aligned} \tag{8}$$

Also, the $\mathbf{q}_x$ vector can be expressed as

$$\begin{aligned}
\mathbf{q}_x = \quad & + [\Lambda_S]^T[\Lambda_{O2}]^T\left([\Lambda_{O1}][\tilde{\omega}_{O1}][\tilde{\omega}_{O1}]\mathbf{q}_{xP} + 2[\dot{\Lambda}_{O1}]\dot{\mathbf{q}}_{xP} + [\Lambda_{O1}]\ddot{\mathbf{q}}_{xP}\right) \\
& - [\Lambda_S]^T[\Lambda_{O2}]^T\left([\Lambda_{O2}][\tilde{\omega}_{O2}][\tilde{\omega}_{O2}]\mathbf{q}_{xS} + 2[\dot{\Lambda}_{O2}]\dot{\mathbf{q}}_{xS} + [\Lambda_{O2}]\ddot{\mathbf{q}}_{xS}\right) \\
& + 2[\dot{\Lambda}_S]^T[\dot{\Lambda}_{O2}]^T\mathbf{q}_{xP-S,W} + 2[\dot{\Lambda}_S]^T[\Lambda_{O2}]^T\dot{\mathbf{q}}_{xP-S,W} \\
& + 2[\Lambda_S]^T[\dot{\Lambda}_{O2}]^T\dot{\mathbf{q}}_{xP-S,W} \\
& + [\Lambda_S]^T\left[[\Lambda_{O2}][\tilde{\omega}_{O2}][\tilde{\omega}_{O2}]\right]^T\mathbf{q}_{xP-S,W} + [\ddot{\Lambda}_S]^T[\Lambda_{O2}]^T\mathbf{q}_{xP-S,W} - \ddot{\mathbf{q}}_{x\Delta}.
\end{aligned} \tag{9}$$

On the other hand, the rotational constraint can be written in quaternion algebra as follows:

$$\mathbf{C}_\theta = \mathbf{q}_{\theta_\Delta}^{-1}\mathbf{q}_{\theta P-S,S} - \mathbf{q}_{\theta_\Re} = \mathbf{0}, \tag{10}$$

where we introduced the real quaternion $\mathbf{q}_{\theta_\Re} = [1, 0, 0, 0]^T$. However we will not enter into details for the rotational constraint, because this part of the *lock formulation* is not necessary for the sliding contact theory presented in this paper (interested readers can find the expressions for Jacobian $[C_{q_\theta}]$ and vector Q_θ in [4]).

At this point, many holonomic and rheonomic constraints can be inherited from the formulation above, simply by suppressing some scalar constraints among the three equations for the translation and/or the three equations for the rotation.

For example, the constraint 'point on plane' where the plane moves about one of the two bodies following some prescribed position/speed/acceleration, can be easily recovered from Equation (4) where only the third scalar equation is taken into consideration. Hence the reference P_{O1} (whose motion on $O1$ surface can be set via $\mathbf{q}_{xP}$, $\dot{\mathbf{q}}_{xP}$, $\ddot{\mathbf{q}}_{xP}$, $\mathbf{q}_{\theta P}$, $\dot{\mathbf{q}}_{\theta P}$, $\ddot{\mathbf{q}}_{\theta P}$) cannot move orthogonally to the XY plane described by the reference S_{O2} (whose motion on $O2$ surface can be set via the terms $\mathbf{q}_{xS}$, $\dot{\mathbf{q}}_{xS}$, $\ddot{\mathbf{q}}_{xS}$, $\mathbf{q}_{\theta S}$, $\dot{\mathbf{q}}_{\theta S}$, $\ddot{\mathbf{q}}_{\theta S}$).

4. Kinematics of Contact Plane

Since we have introduced the *lock formulation* which allows the representation of a simple point-plane constraint, as in Equations (8) and (9), now we must find the position and speed of the references which define this kind of constraint in case the plane and the point are moving on curved surfaces.

4.1. CONTACT PLANE: THE POSITION PROBLEM

In general, because of nonlinearities, the position problem is approached iteratively. With our method, two distinct correction stages are required for each iteration.

Step A1. The first stage involves a single step on Newton method for solving the typical position problem of inverse kinematics:

$$\frac{\partial \mathbf{C}_n}{\partial \mathbf{q}} \Delta \mathbf{q}_i = \left[C_q\right]_i \Delta \mathbf{q}_i = -\mathbf{C}_i,$$
$$\mathbf{q}_{i+1} = \mathbf{q}_i + \Delta \mathbf{q}_i, \qquad (11)$$

where the constraints $\mathbf{C}(\mathbf{q}, t)$ contain all joints of the mechanical system including the ‘point on plane’ constraint.

Step A2. The second operation consists in updating the the position of both point and plane of contact, using global and/or local optimization methods. Also the alignment of contact plane may change.

The two steps are repeated until convergence – if any – is reached. We experienced that this ‘two steps’ uncoupled scheme is more efficient than the direct solution of the complete coupled problem (3). In fact, as said before, there is no need to add the variables u_{O1}, v_{O1}, u_{O2}, v_{O2} in system’s state vector, and the solution of step A1 is somewhat simple: it just requires the implementation of a geometric constraint of the type ‘point on a flat plane’.

Now some considerations about step A2, involving the updating of the sliding plane position for optimal u_{O1}, v_{O1}, u_{O2}, v_{O2}.

As seen before, the contact constraint is expressed by two vectorial equations, the former saying that the contact points on both the surfaces must have the same position in space, and the latter implying that the two surfaces must be tangent (that is, the normals must be aligned):

$$\mathbf{C}_{ps} = \mathbf{P}_{O1} - \mathbf{S}_{O2} = \mathbf{0},$$
$$\mathbf{C}_n = \mathbf{n}_{O1} + \mathbf{n}_{O2} = \mathbf{0}. \qquad (12)$$

The minimization of the residuals of the equations above for optimal values $\{u_{O1}, v_{O1}, u_{O2}, v_{O2}\}$ is an highly nonlinear problem, and the solutions can be multiple or even infinite in degenerate situations (for example, if contact points are multiple, or along a line such as in the case of the contact between two parallel cylinders). In this paper we will focus our attention on the situation of a single contact. Also, we will not deal the details of the procedures which can be used to solve the nonlinear problem of finding globally the contact point(s), because this topic is extensively discussed in other scientific areas, for instance in computational geometry, computer graphics and modelling [8–10].

We just remark that collision-detection (as well as proximity-detection) is a very time-consuming processes, and efficient procedures must be set up for this task, for example, [9] one can pre-process a rough tesselated approximation of the NURBS surface, then one can perform polygon-polygon test very quickly, especially if an hierarchical tree structure of bounding boxes has been computed off-line as an optimization.

We experienced that after the 'global' rough approximation of the contact point has been obtained with such polygon-proximity test, a more precise 'local' solution can be refined by using local nonlinear programming methods, such as the gradient or Newton methods [11], which operate on the true – non-tesselated – parametric surface.

Also, we further improved the efficiency of this geometric problem by performing the global contact search with a less dense time sample than the faster local refinement. In fact the local position must be refined at each step of the simulation in order to get smooth and accurate results, but the position-jumps of the contact point (which are detected by the global method) may happen seldom. Think about a cam and a follower: after the initial contact point between cam and follower have been found once, the point may be updated during cam rotation just by using the local optimization method, whose convergency can be fast because the previous positions can be used as approximate guesses. Then, the global collision algorithm can be invoked less frequently, just to check if there are sudden jumps in contact points (for example, the cam has a hole, or the follower touches another object, etc.)

4.2. CONTACT PLANE: THE SPEED PROBLEM

We can differentiate both Equations (1) and (2) respect to time. The variables of Equation (1) are the four surface parameters $\{u_{O1}, v_{O1}, u_{O2}, v_{O2}\}$ and the coordinates of the two rigid bodies $\{\mathbf{q}_{O1}, \mathbf{q}_{O2}\}$, therefore:

$$\dot{\mathbf{C}}_{ps} = \frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{P}_{O1}(u_{O1}, v_{O1}, \mathbf{q}_{O1}) - \mathbf{S}_{O2}(u_{O2}, v_{O2}, \mathbf{q}_{O2})\right] = \mathbf{0}. \tag{13}$$

Applying the chain rule of differentiation:

$$\begin{aligned}\dot{\mathbf{C}}_{ps} &= \left(\frac{\partial \mathbf{P}_{O1}}{\partial u_{O1}}\frac{\partial u_{O1}}{\partial t} + \frac{\partial \mathbf{P}_{O1}}{\partial v_{O1}}\frac{\partial v_{O1}}{\partial t} + \frac{\partial \mathbf{P}_{O1}}{\partial \mathbf{q}_{O1}}\frac{\partial \mathbf{q}_{O1}}{\partial t} + \frac{\partial \mathbf{P}_{O1}}{\partial t}\right) \\ &\quad - \left(\frac{\partial \mathbf{S}_{O2}}{\partial u_{O2}}\frac{\partial u_{O2}}{\partial t} + \frac{\partial \mathbf{S}_{O2}}{\partial v_{O2}}\frac{\partial v_{O2}}{\partial t} + \frac{\partial \mathbf{S}_{O2}}{\partial \mathbf{q}_{O2}}\frac{\partial \mathbf{q}_{O2}}{\partial t} + \frac{\partial \mathbf{S}_{O2}}{\partial t}\right).\end{aligned} \tag{14}$$

The terms $\partial\mathbf{P}_{O1}/\partial\mathbf{q}_{O1}$ and $\partial\mathbf{S}_{O2}/\partial\mathbf{q}_{O2}$ are matrices, with three rows and six columns (given that each rigid body has six d.o.f.) and will be written $[P_{q_{O1}}]$ and $[S_{q_{O2}}]$

hereafter. Simplifying the null terms, we can write the more compact form:

$$\begin{aligned}\dot{\mathbf{C}}_{ps} &= \left(\frac{\partial \mathbf{P}_{O1}}{\partial u_{O1}}\dot{u}_{O1} + \frac{\partial \mathbf{P}_{O1}}{\partial v_{O1}}\dot{v}_{O1} + [P_{q_{O1}}]\dot{\mathbf{q}}_{O1}\right) \\ &\quad - \left(\frac{\partial \mathbf{S}_{O2}}{\partial u_{O2}}\dot{u}_{O2} + \frac{\partial \mathbf{S}_{O2}}{\partial v_{O2}}\dot{v}_{O2} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2}\right). \qquad (15)\end{aligned}$$

Given that the state of the two bodies is known, the speeds $\dot{\mathbf{q}}_{O1}$ and $\dot{\mathbf{q}}_{O2}$ are known as well, so we can collecting the unknown terms $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$ in order to obtain the following system:

$$\begin{aligned}&\left[\begin{array}{c|c|c|c} \dfrac{\partial \mathbf{P}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{P}_{O1}}{\partial v_{O1}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial u_{O2}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial v_{O2}} \end{array}\right] \begin{Bmatrix} \dot{u}_{O1} \\ \dot{v}_{O1} \\ \dot{u}_{O2} \\ \dot{v}_{O2} \end{Bmatrix} \\ &\quad = -[P_{q_{O1}}]\dot{\mathbf{q}}_{O1} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2}. \qquad (16)\end{aligned}$$

A quick glance at the system of Equation (16), which has four unknowns and three scalar equations, tells that some other equations must be written to get rid of the indetermination and finally compute the parameters $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$. In fact now we must take into consideration the above mentioned constraint on surface tangency, Equation (2), which can be differentiated as follows:

$$\dot{\mathbf{C}}_n = \frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{n}_{O1}(u_{O1}, v_{O1}, \mathbf{q}_{O1}) + \mathbf{n}_{O2}(u_{O2}, v_{O2}, \mathbf{q}_{O2})\right] = \mathbf{0}, \qquad (17)$$

that is, using the same algebraic manipulations and differentiation rules used for (14):

$$\begin{aligned}\dot{\mathbf{C}}_n &= \left(\frac{\partial \mathbf{n}_{O1}}{\partial u_{O1}}\frac{\partial u_{O1}}{\partial t} + \frac{\partial \mathbf{n}_{O1}}{\partial v_{O1}}\frac{\partial v_{O1}}{\partial t} + \frac{\partial \mathbf{n}_{O1}}{\partial \mathbf{q}_{O1}}\frac{\partial \mathbf{q}_{O1}}{\partial t} + \frac{\partial \mathbf{n}_{O1}}{\partial t}\right) \\ &\quad + \left(\frac{\partial \mathbf{n}_{O2}}{\partial u_{O2}}\frac{\partial u_{O2}}{\partial t} + \frac{\partial \mathbf{n}_{O2}}{\partial v_{O2}}\frac{\partial v_{O2}}{\partial t} + \frac{\partial \mathbf{n}_{O2}}{\partial \mathbf{q}_{O2}}\frac{\partial \mathbf{q}_{O2}}{\partial t} + \frac{\partial \mathbf{n}_{O2}}{\partial t}\right), \qquad (18)\end{aligned}$$

$$\begin{aligned}\dot{\mathbf{C}}_n &= \left(\frac{\partial \mathbf{n}_{O1}}{\partial u_{O1}}\dot{u}_{O1} + \frac{\partial \mathbf{n}_{O1}}{\partial v_{O1}}\dot{v}_{O1} + [n_{q_{O1}}]\dot{\mathbf{q}}_{O1}\right) \\ &\quad + \left(\frac{\partial \mathbf{n}_{O2}}{\partial u_{O2}}\dot{u}_{O2} + \frac{\partial \mathbf{n}_{O2}}{\partial v_{O2}}\dot{v}_{O2} + [n_{q_{O2}}]\dot{\mathbf{q}}_{O2}\right). \qquad (19)\end{aligned}$$

The following system has the same unknowns $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$ of system (16), with three scalar equations:

$$\begin{aligned}&\left[\begin{array}{c|c|c|c} \dfrac{\partial \mathbf{n}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{n}_{O1}}{\partial v_{O1}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial u_{O2}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial v_{O2}} \end{array}\right] \begin{Bmatrix} \dot{u}_{O1} \\ \dot{v}_{O1} \\ \dot{u}_{O2} \\ \dot{v}_{O2} \end{Bmatrix} \\ &\quad = -[n_{q_{O1}}]\dot{\mathbf{q}}_{O1} - [n_{q_{O2}}]\dot{\mathbf{q}}_{O2}. \qquad (20)\end{aligned}$$

We can put together the two systems (16) and (20) to get the following:

$$\begin{bmatrix} \dfrac{\partial \mathbf{P}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{P}_{O1}}{\partial v_{O1}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial u_{O2}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial v_{O2}} \\ \dfrac{\partial \mathbf{n}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{n}_{O1}}{\partial v_{O1}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial u_{O2}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial v_{O2}} \end{bmatrix} \begin{Bmatrix} \dot{u}_{O1} \\ \dot{v}_{O1} \\ \dot{u}_{O2} \\ \dot{v}_{O2} \end{Bmatrix} = \begin{Bmatrix} -[P_{q_{O1}}]\dot{\mathbf{q}}_{O1} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2} \\ -[n_{q_{O1}}]\dot{\mathbf{q}}_{O1} - [n_{q_{O2}}]\dot{\mathbf{q}}_{O2} \end{Bmatrix}. \tag{21}$$

The system above has four unknowns and six scalar equations. However two equations are redundant, to be more precise one of the first three equations and one of the last three equations can be eliminated, to obtain a 4×4 system which can be readily solved for $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$.

Let us proof this. The tangent plane in P and S is the same for both the surfaces of bodies $O1$ and $O2$ because of Equation (2), and we can build the rotation matrix $[\Lambda_n]$ which is aligned to such tangent plane. We can rewrite the system equations in that space, that is, after a coordinate projection:

$$\begin{bmatrix} [\Lambda_n]^T & [0]^T \\ [0]^T & [\Lambda_n]^T \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{P}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{P}_{O1}}{\partial v_{O1}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial u_{O2}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial v_{O2}} \\ \dfrac{\partial \mathbf{n}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{n}_{O1}}{\partial v_{O1}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial u_{O2}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial v_{O2}} \end{bmatrix} \begin{Bmatrix} \dot{u}_{O1} \\ \dot{v}_{O1} \\ \dot{u}_{O2} \\ \dot{v}_{O2} \end{Bmatrix} = \begin{bmatrix} [\Lambda_n]^T & [0]^T \\ [0]^T & [\Lambda_n]^T \end{bmatrix} \begin{Bmatrix} -[P_{q_{O1}}]\dot{\mathbf{q}}_{O1} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2} \\ -[n_{q_{O1}}]\dot{\mathbf{q}}_{O1} - [n_{q_{O2}}]\dot{\mathbf{q}}_{O2} \end{Bmatrix}. \tag{22}$$

One can easily verify that, if $[\Lambda_n]$ has been built with the Z axis parallel to the surface normals (that is, if $[\Lambda_n]^T \mathbf{n}_{O1} = \{0, 0, 1\}^T$) then the third and sixth row of system (22) have null coefficients. In other words, either vectors of Equations (13) and (17) belong to the tangent spaces of $\mathbf{P}(u, v)$ and $\mathbf{S}(u, v)$ when Equations (1) and (2) are satisfied and the states $\mathbf{q}_{O1}$, $\mathbf{q}_{O2}$, $\dot{\mathbf{q}}_{O1}$, $\dot{\mathbf{q}}_{O2}$ are coherent with the point-plane constraint equation (see later).

Therefore we can get rid of the third and sixth row of system (22) simply by using the first two rows of the 3×3 rotation matrix $[\Lambda_n]^T$ when performing the projection in tangent coordinates. This means that we can introduce the 3×2 matrix $[\lambda_{uv}]$ which is like $[\Lambda_n]$ except it does not have the third column, which represents the orthogonal versor, i.e. the surface normal. The two columns of $[\lambda_{uv}]$ are simply obtained as two generic orthogonal versors contained in the tangent plane. This lead to the following system in surface-tangent coordinates:

$$\begin{bmatrix} [\lambda_{uv}]^T & [0]^T \\ [0]^T & [\lambda_{uv}]^T \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathbf{P}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{P}_{O1}}{\partial v_{O1}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial u_{O2}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial v_{O2}} \\ \dfrac{\partial \mathbf{n}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{n}_{O1}}{\partial v_{O1}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial u_{O2}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial v_{O2}} \end{bmatrix} \begin{Bmatrix} \dot{u}_{O1} \\ \dot{v}_{O1} \\ \dot{u}_{O2} \\ \dot{v}_{O2} \end{Bmatrix}$$

$$= \begin{bmatrix} [\lambda_{uv}]^T & [0]^T \\ [0]^T & [\lambda_{uv}]^T \end{bmatrix} \begin{Bmatrix} -[P_{q_{O1}}]\dot{\mathbf{q}}_{O1} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2} \\ -[n_{q_{O1}}]\dot{\mathbf{q}}_{O1} - [n_{q_{O2}}]\dot{\mathbf{q}}_{O2} \end{Bmatrix}. \tag{23}$$

Then, the coefficient matrix of the system has four rows and four columns, and the straightforward Gauss solution scheme can be applied in order to obtain the values of $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$ as desired.

4.3. CONTACT PLANE: THE ACCELERATION PROBLEM

Now we can perform a further differentiation of Equations (13) and (17) in order to obtain the unknown accelerations of the points of contact on the two surfaces, expressed in parametric coordinates as $\{\ddot{u}_{O1}, \ddot{v}_{O1}, \ddot{u}_{O2}, \ddot{v}_{O2}\}$.

For Equation (13) we have:

$$\begin{aligned} \ddot{\mathbf{C}}_{ps} = {} & \frac{\mathrm{d}}{\mathrm{d}t}\left[\left(\frac{\partial \mathbf{P}_{O1}}{\partial u_{O1}}\dot{u}_{O1} + \frac{\partial \mathbf{P}_{O1}}{\partial v_{O1}}\dot{v}_{O1} + [P_{q_{O1}}]\dot{\mathbf{q}}_{O1}\right)\right. \\ & \left. - \left(\frac{\partial \mathbf{S}_{O2}}{\partial u_{O2}}\dot{u}_{O2} + \frac{\partial \mathbf{S}_{O2}}{\partial v_{O2}}\dot{v}_{O2} + [S_{q_{O2}}]\dot{\mathbf{q}}_{O2}\right)\right], \end{aligned} \tag{24}$$

$$\begin{aligned} \ddot{\mathbf{C}}_{ps} = {} & \frac{\partial \mathbf{P}_{O1}}{\partial u_{O1}}\ddot{u}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial u_{O1}\partial u_{O1}}\dot{u}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial u_{O1}\partial v_{O1}}\dot{u}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial u_{O1}\partial \mathbf{q}_{O1}}\dot{u}_{O1}\dot{\mathbf{q}}_{O1} \\ & + \frac{\partial \mathbf{P}_{O1}}{\partial v_{O1}}\ddot{v}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial v_{O1}\partial u_{O1}}\dot{v}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial v_{O1}\partial v_{O1}}\dot{v}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial v_{O1}\partial \mathbf{q}_{O1}}\dot{v}_{O1}\dot{\mathbf{q}}_{O1} \\ & + \frac{\partial \mathbf{P}_{O1}}{\partial \mathbf{q}_{O1}}\ddot{\mathbf{q}}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial \mathbf{q}_{O1}\partial u_{O1}}\dot{\mathbf{q}}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial \mathbf{q}_{O1}\partial v_{O1}}\dot{\mathbf{q}}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{P}_{O1}}{\partial \mathbf{q}_{O1}\partial \mathbf{q}_{O1}}\dot{\mathbf{q}}_{O1}\dot{\mathbf{q}}_{O1} \\ & - \frac{\partial \mathbf{S}_{O2}}{\partial u_{O2}}\ddot{u}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial u_{O2}\partial u_{O2}}\dot{u}_{O2}\dot{u}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial u_{O2}\partial v_{O2}}\dot{u}_{O2}\dot{v}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial u_{O2}\partial \mathbf{q}_{O2}}\dot{u}_{O2}\dot{\mathbf{q}}_{O2} \\ & - \frac{\partial \mathbf{S}_{O2}}{\partial v_{O2}}\ddot{v}_{O1} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial v_{O2}\partial u_{O2}}\dot{v}_{O2}\dot{u}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial v_{O2}\partial v_{O2}}\dot{v}_{O2}\dot{v}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial v_{O2}\partial \mathbf{q}_{O2}}\dot{v}_{O2}\dot{\mathbf{q}}_{O2} \\ & - \frac{\partial \mathbf{S}_{O2}}{\partial \mathbf{q}_{O2}}\ddot{\mathbf{q}}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial \mathbf{q}_{O2}\partial u_{O2}}\dot{\mathbf{q}}_{O2}\dot{u}_{O2} - \frac{\partial^2 \mathbf{S}_{O2}}{\partial \mathbf{q}_{O2}\partial v_{O2}}\dot{\mathbf{q}}_{O2}\dot{v}_{O2} \\ & - \frac{\partial^2 \mathbf{S}_{O2}}{\partial \mathbf{q}_{O2}\partial \mathbf{q}_{O2}}\dot{\mathbf{q}}_{O2}\dot{\mathbf{q}}_{O2}. \end{aligned} \tag{25}$$

Similarly, for Equation (17) we have:

$$\begin{aligned} \ddot{\mathbf{C}}_{n} = {} & \frac{\mathrm{d}}{\mathrm{d}t}\left[\left(\frac{\partial \mathbf{n}_{O1}}{\partial u_{O1}}\dot{u}_{O1} + \frac{\partial \mathbf{n}_{O1}}{\partial v_{O1}}\dot{v}_{O1} + [n_{q_{O1}}]\dot{\mathbf{q}}_{O1}\right)\right. \\ & \left. + \left(\frac{\partial \mathbf{n}_{O2}}{\partial u_{O2}}\dot{u}_{O2} + \frac{\partial \mathbf{n}_{O2}}{\partial v_{O2}}\dot{v}_{O2} + [n_{q_{O2}}]\dot{\mathbf{q}}_{O2}\right)\right], \end{aligned} \tag{26}$$

$$\begin{aligned}
\ddot{\mathbf{C}}_n = {} & \frac{\partial \mathbf{n}_{O1}}{\partial u_{O1}}\ddot{u}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial u_{O1}\partial u_{O1}}\dot{u}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial u_{O1}\partial v_{O1}}\dot{u}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial u_{O1}\partial \mathbf{q}_{O1}}\dot{u}_{O1}\dot{\mathbf{q}}_{O1} \\
& + \frac{\partial \mathbf{n}_{O1}}{\partial v_{O1}}\ddot{v}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial v_{O1}\partial u_{O1}}\dot{v}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial v_{O1}\partial v_{O1}}\dot{v}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial v_{O1}\partial \mathbf{q}_{O1}}\dot{v}_{O1}\dot{\mathbf{q}}_{O1} \\
& + \frac{\partial \mathbf{n}_{O1}}{\partial \mathbf{q}_{O1}}\ddot{\mathbf{q}}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial \mathbf{q}_{O1}\partial u_{O1}}\dot{\mathbf{q}}_{O1}\dot{u}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial \mathbf{q}_{O1}\partial v_{O1}}\dot{\mathbf{q}}_{O1}\dot{v}_{O1} + \frac{\partial^2 \mathbf{n}_{O1}}{\partial \mathbf{q}_{O1}\partial \mathbf{q}_{O1}}\dot{\mathbf{q}}_{O1}\dot{\mathbf{q}}_{O1} \\
& + \frac{\partial \mathbf{n}_{O2}}{\partial u_{O2}}\ddot{u}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial u_{O2}\partial u_{O2}}\dot{u}_{O2}\dot{u}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial u_{O2}\partial v_{O2}}\dot{u}_{O2}\dot{v}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial u_{O2}\partial \mathbf{q}_{O2}}\dot{u}_{O2}\dot{\mathbf{q}}_{O2} \\
& + \frac{\partial \mathbf{n}_{O2}}{\partial v_{O2}}\ddot{v}_{O1} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial v_{O2}\partial u_{O2}}\dot{v}_{O2}\dot{u}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial v_{O2}\partial v_{O2}}\dot{v}_{O2}\dot{v}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial v_{O2}\partial \mathbf{q}_{O2}}\dot{v}_{O2}\dot{\mathbf{q}}_{O2} \\
& + \frac{\partial \mathbf{n}_{O2}}{\partial \mathbf{q}_{O2}}\ddot{\mathbf{q}}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial \mathbf{q}_{O2}\partial u_{O2}}\dot{\mathbf{q}}_{O2}\dot{u}_{O2} + \frac{\partial^2 \mathbf{n}_{O2}}{\partial \mathbf{q}_{O2}\partial v_{O2}}\dot{\mathbf{q}}_{O2}\dot{v}_{O2} \\
& + \frac{\partial^2 \mathbf{n}_{O2}}{\partial \mathbf{q}_{O2}\partial \mathbf{q}_{O2}}\dot{\mathbf{q}}_{O2}\dot{\mathbf{q}}_{O2}.
\end{aligned} \tag{27}$$

For the same reasons which lead to Equations (22) and (23), the previous equations (27) and (25) can be projected in tangent coordinates (discarding the orthogonal coordinate) obtaining a linear system with four unknowns and four equations:

$$\begin{aligned}
& \begin{bmatrix} [\lambda_{uv}]^T & [0]^T \\ [0]^T & [\lambda_{uv}]^T \end{bmatrix}
\begin{bmatrix} \dfrac{\partial \mathbf{P}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{P}_{O1}}{\partial v_{O1}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial u_{O2}} & -\dfrac{\partial \mathbf{S}_{O2}}{\partial v_{O2}} \\[2ex] \dfrac{\partial \mathbf{n}_{O1}}{\partial u_{O1}} & \dfrac{\partial \mathbf{n}_{O1}}{\partial v_{O1}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial u_{O2}} & \dfrac{\partial \mathbf{n}_{O2}}{\partial v_{O2}} \end{bmatrix}
\begin{Bmatrix} \ddot{u}_{O1} \\ \ddot{v}_{O1} \\ \ddot{u}_{O2} \\ \ddot{v}_{O2} \end{Bmatrix} \\
& = \begin{bmatrix} [\lambda_{uv}]^T & [0]^T \\ [0]^T & [\lambda_{uv}]^T \end{bmatrix}
\begin{Bmatrix} \mathbf{Q}_{ps} \\ \mathbf{Q}_n \end{Bmatrix},
\end{aligned} \tag{28}$$

where $\mathbf{Q}_{ps}$ and $\mathbf{Q}_n$ are the vectors of the known terms of Equations (27) and (25).

By means of the above formulation, one can solve (23) and (28) to get, respectively, the parametric speeds and parametric accelerations $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$ and $\{\ddot{u}_{O1}, \ddot{v}_{O1}, \ddot{u}_{O2}, \ddot{v}_{O2}\}$.

In fact this works well as far as the contact-on-point does not degenerate into the contact-on-line or contact-on-surface situations, where the matrices of coefficients become ill conditioned. This happens, for example, when a shaft is inserted into a cylindric hole with exactly the same diameter. Hence singular situations must be carefully monitored and handled.

Once one has obtained $\{\dot{u}_{O1}, \dot{v}_{O1}, \dot{u}_{O2}, \dot{v}_{O2}\}$ and $\{\ddot{u}_{O1}, \ddot{v}_{O1}, \ddot{u}_{O2}, \ddot{v}_{O2}\}$, it is easy to update the equations of the *point-on-plane* holonomic constraint. As described in the introduction, such condition consists in a plane (whose position, speed and acceleration about body $O1$ are known) which constraints a point belonging to object $O2$ (also position, speed and acceleration of this point about $O2$ must be known).

Note that Equations (23) and (28) provide speeds and accelerations in parametric coordinates, while the *point-on-plane* constraint formulation needs body-relative cartesian speeds and accelerations, like $\dot{\mathbf{P}}_{O1,O1}$, $\ddot{\mathbf{P}}_{O1,O1}$, etc. It is easy, however, to compute these terms as functions of parametric speeds and accelerations, given the expression of the parametric surfaces:

$$\mathbf{P}_{O1,O1} = \mathbf{P}_{O1,O1}(u_{O1}, v_{O1}), \tag{29}$$

$$\mathbf{S}_{O2,O2} = \mathbf{S}_{O2,O2}(u_{O2}, v_{O2}), \tag{30}$$

it follows:

$$\dot{\mathbf{P}}_{O1,O1} = \dot{u}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial u_{O1}} + \dot{v}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial v_{O1}}, \tag{31}$$

$$\dot{\mathbf{S}}_{O2,O2} = \dot{u}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial u_{O2}} + \dot{v}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial v_{O2}}, \tag{32}$$

and similarly, for the accelerations:

$$\begin{aligned}\ddot{\mathbf{P}}_{O1,O1} &= \ddot{u}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial u_{O1}} + \ddot{v}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial v_{O1}} + \dot{u}_{O1}{}^2\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial u_{O1}\partial u_{O1}} \\ &\quad + \dot{v}_{O1}{}^2\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial v_{O1}\partial v_{O1}} + 2\dot{u}_{O1}\dot{v}_{O1}\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial u_{O1}\partial v_{O1}},\end{aligned} \tag{33}$$

$$\begin{aligned}\ddot{\mathbf{S}}_{O2,O2} &= \ddot{u}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial u_{O2}} + \ddot{v}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial v_{O2}} + \dot{u}_{O2}{}^2\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial u_{O2}\partial u_{O2}} \\ &\quad + \dot{v}_{O2}{}^2\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial v_{O2}\partial v_{O2}} + 2\dot{u}_{O2}\dot{v}_{O2}\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial u_{O2}\partial v_{O2}}.\end{aligned} \tag{34}$$

The partial derivatives in Equations (31), (32), (33) and (34), can be obtained by straightforward numerical differentiation of Equations (29) and (30).

Note that we can distinguish two components for the P ,S accelerations of Equations (33) and (34):

$$\begin{aligned}\ddot{\mathbf{P}}_{O1,O1} &= \ddot{\mathbf{P}}_{O1,O1\parallel} + \ddot{\mathbf{P}}_{O1,O1\perp}, \\ \ddot{\mathbf{S}}_{O2,O2} &= \ddot{\mathbf{S}}_{O2,O2}\| + \ddot{\mathbf{S}}_{O2,O2\perp}.\end{aligned} \tag{35}$$

The terms $\ddot{\mathbf{S}}_{O2,O2\parallel}$ and $\ddot{\mathbf{P}}_{O1,O1\parallel}$, depending on parametric tangential accelerations $\ddot{u}$, $\ddot{v}$ only, can be called *tangential components* since from Equations (33) and (34) it is easy to see that these vectors are always directed tangentially to the contact surfaces.

$$\ddot{\mathbf{P}}_{O1,O1\parallel} = \ddot{u}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial u_{O1}} + \ddot{v}_{O1}\frac{\partial \mathbf{P}_{O1,O1}}{\partial v_{O1}}, \tag{36}$$

$$\ddot{\mathbf{S}}_{O2,O2\parallel} = \ddot{u}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial u_{O2}} + \ddot{v}_{O2}\frac{\partial \mathbf{S}_{O2,O2}}{\partial v_{O2}}. \tag{37}$$

The terms $\ddot{\mathbf{S}}_{O2,O2\perp}$ and $\ddot{\mathbf{P}}_{O1,O1\perp}$, depending on parametric tangential speeds $\dot{u}$, $\dot{v}$ only, can be called *centripetal components* (note some analogies with tangential and centripetal accelerations for classical two-dimensional mechanics).

$$\ddot{\mathbf{P}}_{O1,O1\perp} = \dot{u}_{O1}{}^2\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial u_{O1}\partial u_{O1}} + \dot{v}_{O1}{}^2\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial v_{O1}\partial v_{O1}} + 2\dot{u}_{O1}\dot{v}_{O1}\frac{\partial^2 \mathbf{P}_{O1,O1}}{\partial u_{O1}\partial v_{O1}}, \tag{38}$$

$$\ddot{\mathbf{S}}_{O2,O2\perp} = \dot{u}_{O2}{}^2\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial u_{O2}\partial u_{O2}} + \dot{v}_{O2}{}^2\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial v_{O2}\partial v_{O2}} + 2\dot{u}_{O2}\dot{v}_{O2}\frac{\partial^2 \mathbf{S}_{O2,O2}}{\partial u_{O2}\partial v_{O2}}. \tag{39}$$

Now, one can see that the tangential terms *do no affect at all* the computation of acceleration terms of the ‘point on plane’ constraint (that is, inserting $\ddot{\mathbf{S}}_{O2,O2\parallel}$ and $\ddot{\mathbf{P}}_{O1,O1\parallel}$ in Equation (9) gives always a null vector when position constraint is already satisfied). This means that only the centripetal terms have true significance for the ‘point-on plane’ constraint.

In detail, this is very important for the practical implementation of the *sliding plane* method in a dynamical simulator: in fact the computation of the constraint vector $\mathbf{q}_x$ as in Equation (9) can take $\ddot{\mathbf{q}}_{xP} = \ddot{\mathbf{P}}_{O1,O1\perp}$ and $\ddot{\mathbf{q}}_{xS} = \ddot{\mathbf{S}}_{O2,O2\perp}$ instead of $\ddot{\mathbf{q}}_{xP} = \ddot{\mathbf{P}}_{O1,O1}$ and $\ddot{\mathbf{q}}_{xS} = \ddot{\mathbf{S}}_{O2,O2}$, with identical results. Since tangential terms equations (36) and (37) are not needed, there is no need to solve the system (28) for parametric accelerations: this has a positive impact on computation speed and, most important, on the ability to solve for unknown rigid body accelerations during dynamics.

In fact this consideration saves us from a potential tautology: the multibody dynamical solution includes the rheonomic constraint of Equation (9) in order to solve for unknown bodies’ accelerations, but the term (9) itself contains motion laws of references P,S which, among all other things, seem to be functions of body accelerations in their turn, as given in Equation (28). However, this ‘deadlock’ situation is resolved by the above mentioned consideration, that only the centripetal (speed-dependent) part of $\mathbf{P}_{O1}$, $\mathbf{S}_{O2}$ references has effect on the term equation (7) of the point-on plane constraint. That is, we do not need the *a-priori* knowledge of body accelerations in order to compute all the terms of the *sliding plane* constraint in the dynamical solution problem: speed knowledge is enough.

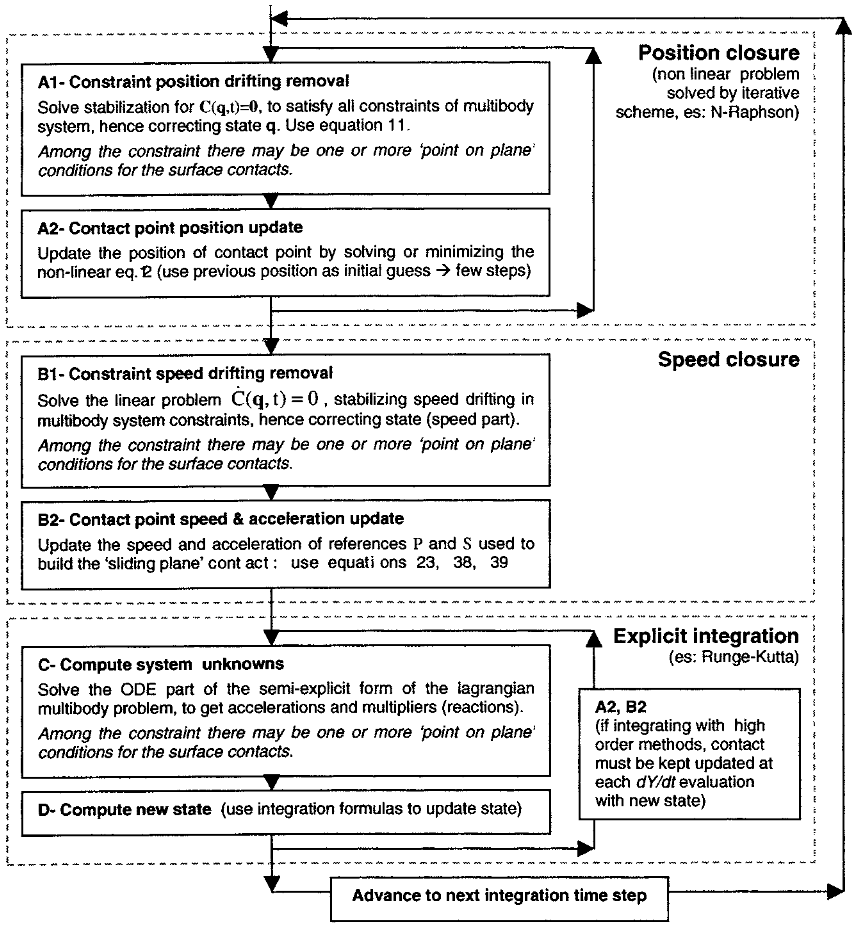
Position closure
(non linear problem solved by iterative scheme, es: N-Raphson)
A1- Constraint position drifting removal
Solve stabilization for C(q,t)=0, to satisfy all constraints of multibody system, hence correcting state q. Use equation 11.
Among the constraint there may be one or more 'point on plane' conditions for the surface contacts.
A2- Contact point position update
Update the position of contact point by solving or minimizing the non-linear eq. 12 (use previous position as initial guess → few steps)
Speed closure
B1- Constraint speed drifting removal
Solve the linear problem Ċ(q, t) = 0 , stabilizing speed drifting in multibody system constraints, hence correcting state (speed part).
Among the constraint there may be one or more 'point on plane' conditions for the surface contacts.
B2- Contact point speed & acceleration update
Update the speed and acceleration of references P and S used to build the 'sliding plane' cont act : use equati ons 23, 38, 39
Explicit integration
(es: Runge-Kutta)
C- Compute system unknowns
Solve the ODE part of the semi-explicit form of the lagrangian multibody problem, to get accelerations and multipliers (reactions).
Among the constraint there may be one or more 'point on plane' conditions for the surface contacts.
A2, B2
(if integrating with high order methods, contact must be kept updated at each dY/dt evaluation with new state)
D- Compute new state (use integration formulas to update state)
Advance to next integration time step

Figure 4. Integration scheme (simplified flowchart).

5. Implementation

An efficient approach to the solution of the DAE (differential algebraic) problem of constrained Lagrangian dynamics is discussed in [12]. The solution scheme exposed in Figure 4 relies on that method, which includes two constraint-stabilizing steps per each integration step of the underlying ODE problem. Note that, among all the constraints equations, there is always the contact constraint expressed as a 'point on plane' condition. During the iterative Newton–Raphson procedure, the problem of updating the position of the 'sliding plane' (step A2) is uncoupled from the constraint closure problem (step A1), and both are executed at each iteration one after the other. This causes a simple implementation, while convergence of the method is still good as if A1 and A2 problems were coupled.

Later, having obtained the correct orientation and position for the sliding plane constraint, the speed closure of constraints can be easily solved too (step B1).

We remark that, once positions and speeds of multibody state are correct, one can compute Equations (23), (31) and (32), as well as (38) and (39) without problems (step B2), then obtaining all the information about the speed and centripetal acceleration of the references $\mathbf{P}_{O1}$ and $\mathbf{S}_{O2}$ which are used to represent the *sliding plane* constraint.

In fact the computation of unknowns accelerations for a given state (step C) can take place only if the kinematics of $\mathbf{P}_{O1}$ and $\mathbf{S}_{O2}$ are correct in terms of both positions, speeds and accelerations: again we stress the point that only the centripetal parts of references'accelerations (depending only on parametric speeds of Equation (23)) is required in Equation (9) of the *lock formulation*, while the tangential part of acceleration (which depends from Equation (28)) has no effect at all on that constraint.

Therefore, only after step C, one may compute also Equations (28), (33) and (34) in order to get also the complete accelerations of contact points (i.e. including tangential effect), if needed.

Although we presented the *sliding plane approach* within the framework of Figure 4, it should be clear that this way of handling contacts could be implemented with minimum efforts in other integration schemes, either explicit or implicit. Given the hypothesis of previous chapters, the correctness of our approach within other integrators can be guaranteed only if, for whatever change in system state, tangency of contact surfaces is always satisfied before computing the 'acceleration effects' in step B2. This means that, for each update of the system's state performed by a generic integrator, also the position and inclination of contact plane $\mathbf{S}_{O2}$ and $\mathbf{P}_{O1}$ must be updated. In fact curvature and tangency of surfaces may change from step to step, during motion.

Summing up, steps A2 and B2 must be performed in sequence, right after each change in system state: having satisfied this requirement, the computation of unknown accelerations for the given state will be always correct, and the integration process can proceed. In general, the solution provided by the *sliding plane* approach is exact (not approximate) from a cinematical point of view: however a loss of precision can take place when the partial derivatives in Equations (23), (38), (39) cannot be recovered analytically and must be obtained via numerical differentiation.

We made the simplificative assumption that each body has a single boundary surface with continuous parametrization in (u, v), but this is seldom true when using three-dimensional models coming from CAD applications, where the boundary of each body may be built of many surface patches, each with its own (u, v) parametrization (hence the so called B-Rep, *Boundary Representation*).

Under these circumstances the computation of the linear systems (23) and (28) may become difficult: the many parametric differentiations which are needed to recover the coefficients could be numerically ill-conditioned or hard to perform [13]. For example, straightforward numerical differentiation of surfaces can be troublesome in proximity of trimmed patches, across singularities caused by neigh-

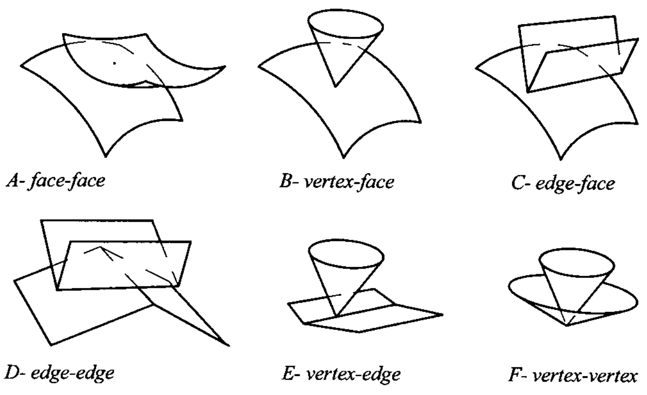

Figure 5. Contacts between topological entities of the boundary representation (B-Rep).

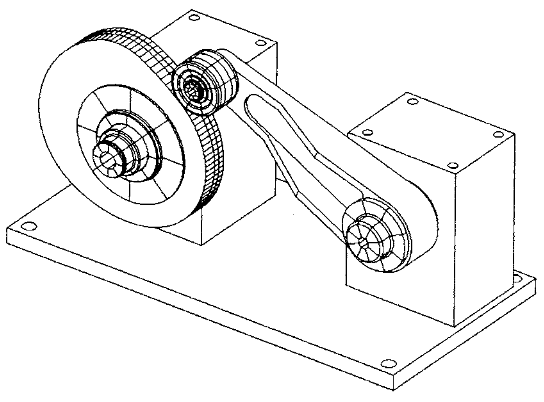

Figure 6. Cam-follower benchmark for contact between freeform surfaces.

boring faces in B-rep topology, when using Catmull-Clark limit surfaces, and so on.

Also, a robust and complete code should be able to handle the six different types of contact between all the connective elements of boundary representations (edges, faces, vertexes) as in Figure 5. Of course in this work we dealt only with the case of 'curved face on curved face', but the other five cases can be developed in a similar way.

6. Examples

6.1. CAM-FOLLOWER

To validate the model of contact, we built a simple cam-follower mechanism using the three-dimensional modelling environment of our multibody software. The cam and the follower are made of NURBS bi-parametric surfaces (Figure 6).

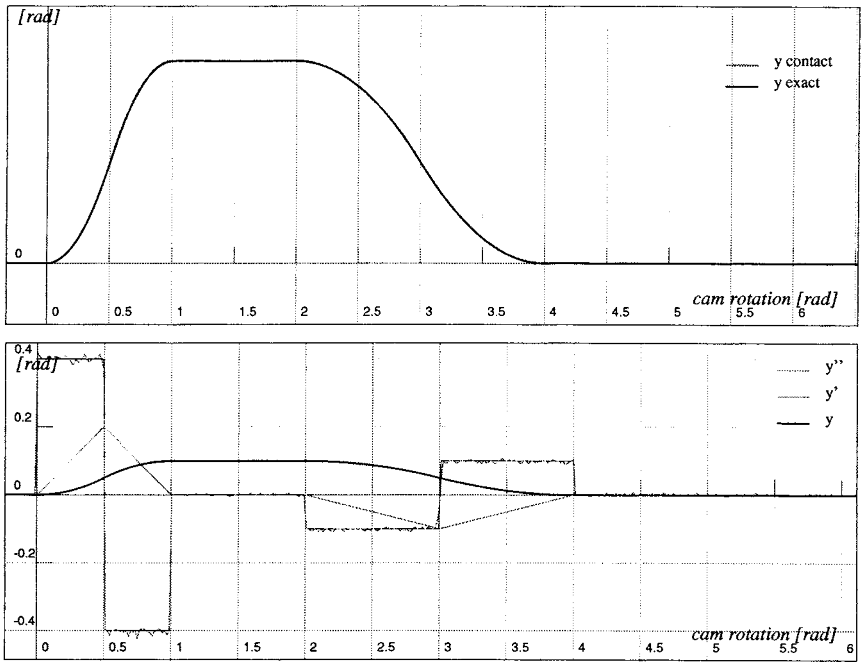

Figure 7. Comparing the two motion laws: exact (analytical original) and simulated motion of follower (moved by contact): they overlap perfectly. Also the speed profile of simulated motion coincides exactly with the analytical profile. Note some high frequency numerical noise on acceleration.

In detail, the shape of the cam has been created with a procedural modelling tool which uses the formulas in [7], where one gets the profile as a function of the motion law imposed to the follower. Therefore, using a test motion law, we built the cam surface for that motion, using 200×4 control points (the more the samples, the less the approximation).

We performed the simulation of the mechanism, and compared the resulting motion of the follower (moved only by contact) with the hypothetical 'exact' motion law that we used to build the cam. We observed little or no differences in position and speed of the follower, but sometimes a small noise can affect acceleration (Figure 7) mostly because the cam does not have an analytical shape, but it is approximated by fifth-degree Nurbs.

6.2. WHEEL ON PLANE

We can check the correctness of the *sliding plane* approach by testing it on a simple planar problem whose analytical solution is well known. This can be the case of a wheel on a plane, where the horizontal position of the spindle is fixed (Figure 8). The wheel can have a rotation q_θ, thus slipping on the plane. The contact point $\mathbf{S}_{O2}$

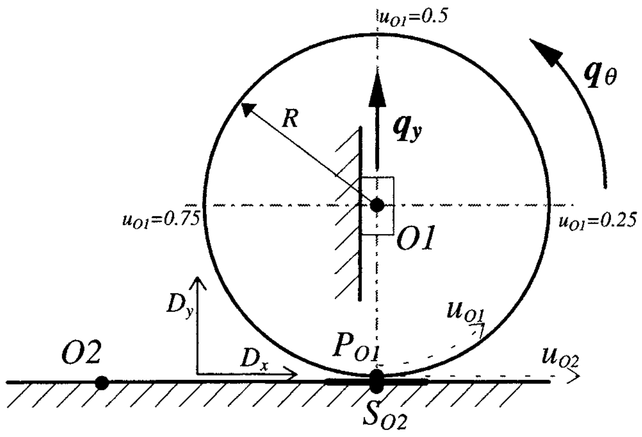

Figure 8. A contact problem in two-dimensional space: the wheel on plane.

moves on the horizontal ground, while the other point of contact $\mathbf{P}_{O1}$ belongs to the wheel and moves along its circumference.

Of course it is known, from classical mechanics, that the constraint on vertical acceleration of the spindle is simply $\ddot{q}_y = 0$. By using the *sliding plane* approach, we could consider the wheel circumference as function of a circumferential parameter u_{O1} in [0..1]. Also wheel normal would be a function of u_{O1}, then:

$$\begin{aligned}\mathbf{P}_{O1} &= \mathbf{D}_\mathbf{x}[R + R\sin(2\pi u_{O1} + q_\theta)] + \mathbf{D}_\mathbf{y}[R + q_y - R\cos(2\pi u_{O1} + q_\theta)],\\ \mathbf{n}_{O1} &= \mathbf{D}_\mathbf{x}\sin(2\pi u_{O1} + q_\theta) - \mathbf{D}_\mathbf{y}\cos(2\pi u_{O1} + q_\theta).\end{aligned} \tag{40}$$

With this simplified example, where there is only $\dot{q}_\theta$ as speed, Equation (23) becomes a simple 2×2 linear system of the type:

$$\begin{bmatrix} R2\pi & 1 \\ 2\pi & 0 \end{bmatrix}\begin{bmatrix} \dot{u}_{O1} \\ \dot{u}_{O2} \end{bmatrix} = \begin{bmatrix} -R\dot{q}_\theta \\ -\dot{q}_\theta \end{bmatrix}, \tag{41}$$

providing the solutions $\dot{u}_{O1} = -\dot{q}_\theta/2\pi$ and $\dot{u}_{O2} = 0$ as expected. Using Equation (31) we get the speed of contact point in wheel rotating reference, that is $\dot{\mathbf{P}}_{O1,O1} = -\mathbf{D}_\mathbf{x} \cdot R \cdot \dot{q}_\theta$. Also, by Equation (32), we get $\dot{\mathbf{S}}_{O1,O1} = 0$.

Now we must use Equation (38) to get the centripetal part of the acceleration of contact point P, in $O1$ reference. Performing all the derivatives and simplifications, we get:

$$\ddot{\mathbf{P}}_{O1,O1\perp} = \dot{u}_{O1}{}^2 \frac{\partial^2 \mathbf{P}_{O1,\, O1}}{\partial u_{O1} \partial u_{O1}} = \mathbf{D}_\mathbf{y} \cdot R \cdot \dot{q}_\theta^2, \tag{42}$$

which looks like the classical equation of centripetal acceleration for a point on a circular trajectory, $a_c = R \cdot \omega^2$.

Now we have all the variables that are needed for the solution of unknown accelerations. In this simple example, the *lock formulation* introduces a 'point on

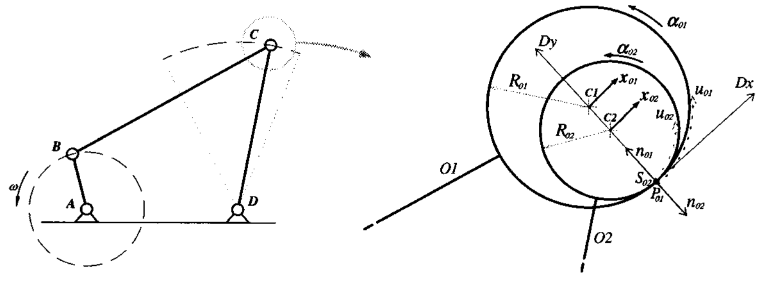

Figure 9. Four-bar linkage with clearance in a revolute pair.

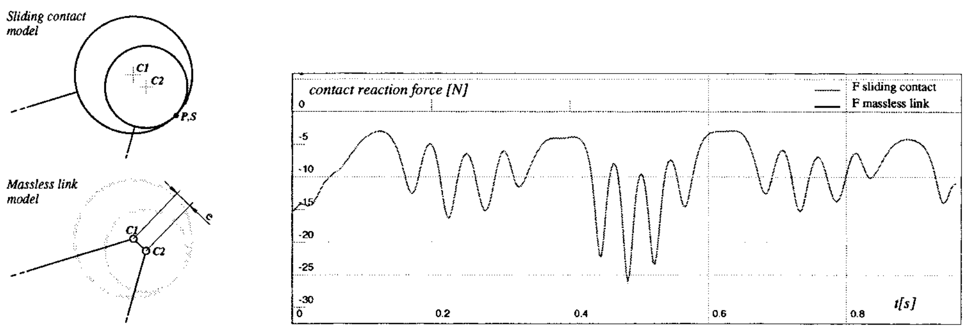

Figure 10. Clearances. Comparison between our method and the ‘massless link model’.

plane’ constraint which leads to the following linear system, with a single equation and a single unknown acceleration: $[C_q]\ddot{q_y} = Q_x$. Applying Equation (8) and simplifying, the Jacobian becomes a 1×1 matrix, with value $[C_q] = [1]$. Equation (9), after many simplifications, becomes:

$$Q_x = \dot{q_\theta}^2 R - 2\dot{q_\theta}^2 R + \dot{q_\theta}^2 R = 0 \tag{43}$$

Hence the solution of $[C_q]\ddot{q_y} = Q_x$ gives exactly $\ddot{q_y} = 0$, as expected.

6.3. CLEARANCE IN A CONTACT PAIR

Although aimed at the most general case of arbitrary three-dimensional contacts, the *sliding plane* approach can handle also the simpler two-dimensional problem of contact between hole and shaft in a revolute joint with some clearance, as in the four-bar mechanism of Figure 9.

Usually, the clearance effect is simulated by replacing the contact between the two circles with a massless link which joins the two centers (Figure 10) since this constraint has the same effect in terms of positions, speeds and accelerations [14]. We will compare our more general approach with such massless-link model.

For the sake of conciseness, we assume that either the constraint on mutual positions and speeds have been already satisfied. This means that steps A1, A2, B1 of Figure 4 are already solved, while accelerations are still to be found. Using the local coordinate system of Figure 9, and considering that speed and position constraints are already satisfied, we can write the coordinates of the two circumferences using the few mutual state coordinates which are left free, that is $\mathbf{q} = \{x_{O1}, \alpha_{O1}, x_{O2}, \alpha_{O2}\}$:

$$\begin{aligned}
\mathbf{P}_{O1} &= \mathbf{D_x}[x_{O1} + R_1 \sin(2\pi u_{O1} + \alpha_{O1})] + \mathbf{D_y}[R_1 - R_1 \cos(2\pi u_{O1} + \alpha_{O1})],\\
\mathbf{n}_{O1} &= \mathbf{D_x}[-\sin(2\pi u_{O1} + \alpha_{O1})] + \mathbf{D_y}[\cos(2\pi u_{O1} + \alpha_{O1})],\\
\mathbf{S}_{O2} &= \mathbf{D_x}[x_{O2} + R_2 \sin(2\pi u_{O2} + \alpha_{O2})] + \mathbf{D_y}[R_2 - R_2 \cos(2\pi u_{O2} + \alpha_{O2})],\\
\mathbf{n}_{O2} &= \mathbf{D_x}[\sin(2\pi u_{O2} + \alpha_{O2})] + \mathbf{D_y}[-\cos(2\pi u_{O2} + \alpha_{O2})]. \qquad (44)
\end{aligned}$$

Given that mutual speeds are already known (coming from step B1), the speed state is $\dot{\mathbf{q}} = \{\dot{x}_{O1}, \dot{\alpha}_{O1}, \dot{x}_{O2}, \dot{\alpha}_{O2}\}$ and Equation (23) becomes a simple 2×2 linear system of the type:

$$\begin{bmatrix} +R_1 2\pi & -R_2 2\pi \\ -2\pi & +2\pi \end{bmatrix} \begin{bmatrix} \dot{u}_{O1} \\ \dot{u}_{O2} \end{bmatrix} = \begin{bmatrix} -R_1\dot{\alpha}_{O1} - \dot{x}_{O1} + R_2\dot{\alpha}_{O2} + \dot{x}_{O2} \\ \dot{\alpha}_{O1} - \dot{\alpha}_{O2} \end{bmatrix}, \qquad (45)$$

providing the solutions for unknown speeds of parameters $\dot{u}_{O1}, \dot{u}_{O2}$:

$$\begin{aligned}
\dot{u}_{O1} &= \frac{1}{2\pi}\frac{\dot{x}_{O2} - \dot{x}_{O1}}{R_1 - R_2} - \frac{\dot{\alpha}_{O1}}{2\pi},\\
\dot{u}_{O2} &= \frac{1}{2\pi}\frac{\dot{x}_{O2} - \dot{x}_{O1}}{R_1 - R_2} - \frac{\dot{\alpha}_{O2}}{2\pi}. \qquad (46)
\end{aligned}$$

Given $\dot{u}_{O1}, \dot{u}_{O2}$, it is possible to use Equation (38) to get the centripetal part of the acceleration of contact point P, in $O1$ reference. Performing derivatives and simplifications, we obtain:

$$\ddot{\mathbf{P}}_{O1,O1\perp} = \mathbf{D_y} \cdot R_1 \left(\frac{(\dot{x}_{O2} - \dot{x}_{O1})^2}{(R_1 - R_2)^2} + \dot{\alpha}_{O1}{}^2 - \dot{\alpha}_{O1}\frac{(\dot{x}_{O2} - \dot{x}_{O1})}{(R_1 - R_2)} \right), \qquad (47)$$

$$\ddot{\mathbf{S}}_{O2,O2\perp} = \mathbf{D_y} \cdot R_2 \left(\frac{(\dot{x}_{O2} - \dot{x}_{O1})^2}{(R_1 - R_2)^2} + \dot{\alpha}_{O2}{}^2 - \dot{\alpha}_{O2}\frac{(\dot{x}_{O2} - \dot{x}_{O1})}{(R_1 - R_2)} \right). \qquad (48)$$

These terms must be introduced in *lock formulation* equations (8) and (9) in order to compute the Jacobian matrix $[C_q]$ and the known part Q_x of equation $[C_q]\ddot{\mathbf{q}} = Q_x$. Having $[C_q]$ and Q_x for the sliding constraint, the system can be solved for unknown accelerations with usual methods.

Note that $[C_q]$ and Q_x does not change for cases with same radial clearance $e = R_1 - R_2$. To the limit, for $R_2 = 0$ and $R_1 = e$, the relative speed in contact point is $v_{re} = \dot{x}_{O2} - \dot{x}_{O1} - R_1\dot{\alpha}_{O1}$, and Equations (47) and (48) become much simpler:

$$\ddot{\mathbf{S}}_{O2,O2\perp} = 0, \quad \ddot{\mathbf{P}}_{O1,O1\perp} = \mathbf{D_y}\left(v_{re}^2/e\right). \qquad (49)$$

It is easy to see that this corresponds to the equation introduced by a massless link with length e which joins the two centers. In fact we performed simulations using

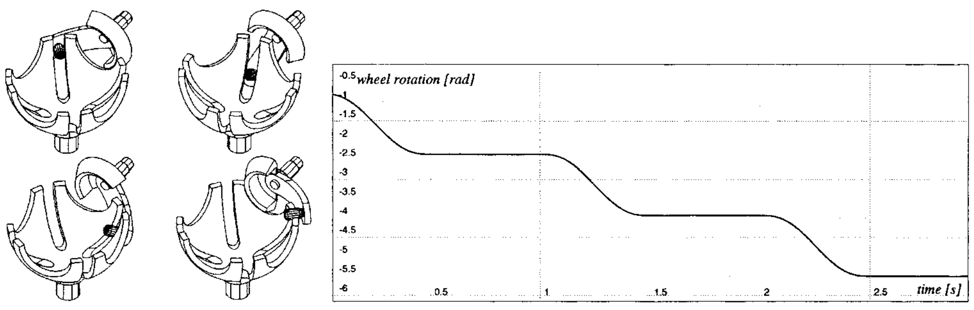

Figure 11. A contact problem in three-dimensional space: a spherical Geneva wheel.

either the massless link model and the *sliding plane* model, obtaining identical results (Figure 10).

Even if the massless link model may have a simpler formulation, we demonstrated that our approach is a more general method which can reproduce this special case, still being able to describe even more complex situations such as, for example, clearances for ovalized shafts and holes.

6.4. SPHERICAL GENEVA WHEEL

The device of Figure 11 is a *spherical Geneva wheel*, and can be used to achieve a special type of intermittent motion. The contact happens between three-dimensional surfaces, and we tested the *sliding plane* method with this complex example. Note that the simulation code must implement advanced features such as the handling of multiple contact points and monolateral constraints. The simulation provided the rotation β of the wheel as a function of crank angle α: the wheel advances with angular strokes $\beta_s = \pi/2$ and dwell period $\alpha_d = \pi$, as expected.

7. Conclusions

An approach has been proposed for the multibody simulation of sliding contact between freeform surfaces. The geometric constraint has been represented by means of a tangential plane which moves between the contact bodies, hence only a simple ‘point on plane’ constraint had to be added to the system of motion-equations. On the other side, the problem of computing the auxiliary variables of the contact constraint (position, speed and acceleration of contact point) could be solved separately, mostly for sake of better performance. The theoretical result have been implemented into our general-purpose multibody software and have been successfully tested with real world examples. Future development may embrace the application of these results to non-parametric surfaces and the problem of high-performance collision detection.

References

1. Baraff, D., ‘Curved surfaces and coherence for non-penetrating rigid body simulation’, *Computer Graphics (Proceedings ACM Siggraph’90)* **24**, 1990, 18–29.
2. Balling, C., ‘Formulation of a class of higher-pair joints in multibody systems using joint coordinates’, *Multibody System Dynamics* **3**, 1999, 21–45.
3. Shabana, A., *Dynamics of Multibody Systems*, John Wiley and Sons, New York, 1989.
4. Tasora, A. and Righettini, P., ‘Application of quaternion algebra to the efficient computation of Jacobians for holonomic-rheonomic constraints’, in *EUROMECH 404*, Lisboa, J.C. Ambrósio and W. Schiehlen (eds.), IST, 1999, 75–92.
5. Pfeiffer, F. and Glocker, C., *Multibody Dynamics with Unilateral Contacts*, John Wiley & Sons, New York, 1996.
6. Lankarani, H.M. and Pereira, M., ‘Treatment of impact with friction in multibody mechanical systems’, *Multibody System Dynamics* **6**, 2001, 203–227.
7. Magnani, P.L. and Ruggieri, G., *Meccanismi per Macchine Automatiche*, UTET, Torino, 1990.
8. Cohen, J., Lin, M.C., Manocha, D. and Ponamgi, M., ‘I-collide: An interactive and exact collision detection system for large-scale environments’, in *Proceedings of ACM Interactive 3D Graphics Conference*, ACM, 1995, 189–196.
9. Gottschalk, S., Lin, M.C. and Manocha, D., ‘OBB tree: A hierarchical structure for rapid interference detection’, *Computer Graphics (Proceedings of ACM Siggraph’96)* **30**, 1996, 171–180.
10. Krishnan, S., Pattekar, A., Lin, M. and Manocha, D., ‘Spherical shell: A higher order bounding volume for fast proximity queries’, in *Proceedings of 3rd International Workshop on Algorithmic Foundations of Robotics*, P. Agarwal (ed.), Rice University, 1998, 122–136.
11. Rao, S., *Engineering Optimization: Theory and Practice*, John Wiley & Sons, New York.
12. Tasora, A., ‘An optimized Lagrangian multiplier approach for interactive multibody simulation in kinematic and dynamical digital prototyping’, in *VII ISCSB*, F. Casolo (ed.), CLUP, Milano, 2001, 3–12.
13. Knowles, I. and Wallace, R., ‘A variational method for numerical differentiation’, *Numerische Mathematik* **70**, 1995, 91–110.
14. Furuhashi, T., Morita, M. and Matasuura, M., ‘Research on dynamics of four-bar linkages with clearances’, *Bulletin of the JSME* **21**, 1978, 1284–1305.