

Dynamical Model of a Vehicle with Omni Wheels: Improved and Generalized Contact Tracking Algorithm

Ivan Kosenko¹ Sergey Stepanov² Kirill Gerasimov³ Alexey Rachkov⁴

¹Department of Theoretical Mechanics, Moscow Aviation Institute, Russia, kosenko@ccas.ru

²Department of Mechanics, Dorodnitsyn Computing Center of RAS, Russia, stepsj@ccas.ru

³Department of Theoretical Mechanics and Mechatronics, Lomonosov Moscow State University, Russia, kiriger@gmail.com

⁴Department of Theoretical Mechanics, Moscow Aviation Institute, Russia, alexey-rachkov@yandex.ru

Abstract

A model of the multibody dynamics for an omni wheel assuming embedded in a frame of wider dynamical environment of the whole vehicle is under development and verification. Modelica base classes developed earlier for the multibody applications with contacts involving friction are used. Generalization has been performed for the model of contact tracking algorithm between roller and horizontal floor. Generalization includes non-zero angle between the roller axis of rotation and plane of the omni wheel. Contact tracking algorithm is implemented in two cases: (a) implicit and (b) explicit.

Models for these cases (a) and (b) are currently “embedded” into the omni vehicle model earlier verified. For simplicity we analyze a multibody system comprising the wheel plus set of rollers being mounted along its circumference. A remainder of the vehicle is replaced by the wrench properly arranged in a way such that the wheel keeps its vertical orientation permanently. The performed computations have shown that two algorithms of the contact tracking generate completely identical dynamics of the whole multibody system.

Keywords: omni wheel; contact tracking; unilateral constraint; angled rollers; model of friction

1 Introduction

A construct of the omni vehicle (Ilon, 1975) dynamical model has been presented in (Kosenko and Gerasimov, 2014), see also papers (Kálmán, 2013; Tobolár et al., 2009). Simplified model for roller mounting on the wheel disk has been considered there: the roller axis of rotation assumed to be in the disk, or equivalently the angle between this axis and the wheel plane, denote it by ψ , is equal to zero. We will call this angular parameter of the model the angle of the preliminary roller rotation (pre-rotation) about the wheel radius intersecting the roller axis of rotation at its central point.

Omni wheel for this case is shown in Figure 1. There one can see the lateral view, fragment (a), of the wheel being equipped by four axisymmetrical rollers, each having a shape of the circular spindle. These rollers have been enumerated by their numbers. Each roller is connected to the wheel by a joint which axis coincides with the roller axis of rotation. These latter axes both are orthogonal to the wheel radius exiting from the central point O and passing through the roller central point. So it is possible for the wheel to have a free rolling in direction perpendicular to its plane. Corresponding contacting curve with respect to the wheel coordinate system, being a circle in the case shown, has a coloured highlighting. This curve has a circular shape provided the wheel plane keeps its vertical orientation. Front view of the omni wheel is shown in fragment (b).

For the case of $\psi = 0$ being shown in Figure 1 a roller outer profile, generatrix, along its axis of rotation has evidently a circular shape, see Figure 1, fragment (a), again. This shape provides smooth transfer from one roller to another while the motion occurs. Evidently if $\psi \neq 0$ then it is not the case. Thus, the contact tracking algorithm for the case of $\psi = 0$ implemented in (Kosenko and Gerasimov, 2014) turned out to be simple enough. In the case of $\psi > 0$ it becomes visibly complicated. And its implementation on Modelica language is the main goal of this paper.

Other details of Figure 1 are the following: R is the omni wheel radius, R_1 is the distance between the wheel central point O and the roller central point, α is the half roller angular length from the viewpoint O . Unit vectors $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ of the base being connected with the wheel are shown in their initial positions.

In engineering applications one may encounter frequently a situation with $\psi > 0$. We proposed in (Kosenko and Gerasimov, 2014) fast algorithm for tracking a contact provided the omni wheel keeps vertical orientation of its plane (in frame of the whole vehicle construct). Thus the task for building up the contact tracking algorithm also for the case of $\psi > 0$ is of interest. This task

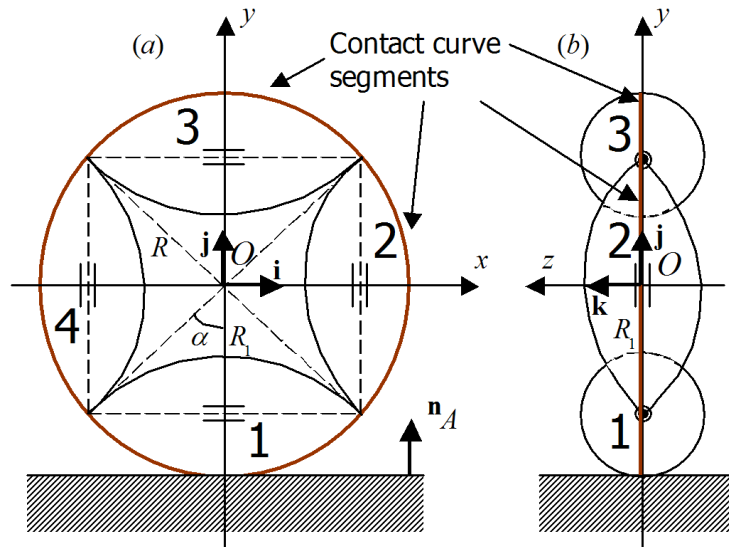


Figure 1. The omni wheel vertically aligned: (a) lateral view; (b) front view.

has been completed in this paper. To reach this goal we accept the working model of a virtual testbench consisting of one wheel equipped by rollers along its rim. One can see easily that this simplification has mostly methodical nature and does not prevent us from integrating all the construct back into the whole vehicle having generally several omni wheels previously analyzed (Kosenko and Gerasimov, 2014).

So let us consider an omni wheel, see Figure 1 its lateral and front views with four rollers, which is able to keep vertical orientation of its plane. We will see later how to arrange an implementation of such a servo-constraint. Note in addition, that in the case of $\psi > 0$ a generatrix of the roller outer surface will not be a segment of the circle anymore. It is represented by a more complicated curve. Moreover, point break of contact on the roller surface does not correspond to the surface tip for the case of $\psi > 0$ as it took place for the simple case of $\psi = 0$. To arrange correct simulation on event of the contact exchange between rollers one has to truncate the roller surface properly.

2 Model of the Omni Wheel Dynamics

Vehicle equipped by omni wheels might be replaced by a wrench consisting of force and torque in the multibody, rigid, representation. The force supposed to act at the wheel center. Thus approximately we can analyze the omni wheel dynamics with the wrench applied instead of a remainder of the vehicle.

Moreover, the vehicle, or a separated wheel, performs in our example motion on the horizontal floor for simplicity. Thus, the wheel being embedded into the vehicle in the simplest case should be aligned vertically. To ex-

press such an alignment analytically we can connect with the wheel the base $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ originating from the wheel center. Both unit vectors \mathbf{i}, \mathbf{j} lie in the wheel plane, and unit vector \mathbf{k} is normal to it. Thus the vertical alignment of the wheel is equivalent to horizontal alignment for the vector \mathbf{k} . Analytical condition for this is

$$\mathbf{k} \cdot \mathbf{n}_A = 0,$$

where unit vector \mathbf{n}_A is vertical, or normal to the floor. In other words, let $T \in SO(3)$ be the matrix of transformation from base $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ to the inertial absolute coordinate system. Then components of vector \mathbf{k} are exactly the components of the matrix $T = (t_{ij})_{i,j=1}^3$ third column. Thus one can express condition of the wheel vertical alignment in the form

$$t_{23} = 0.$$

This latter equation shows that the omni wheel multibody system undergoes the geometrical servo constraint. It is easy to see that this constraint may be implemented via control effort, rotating torque \mathbf{M} directed such as to prevent rotation of the wheel plane w. r. t. horizontal line belonging to this plane.

For details of the torque vector \mathbf{M} computation note that this vector has to be directed along horizontal line passing through the wheel center and belonging to its plane. Directing unit vector \mathbf{l} for this line has to satisfy the equation

$$\mathbf{l} = \mathbf{k} \times \mathbf{n}_A / |\mathbf{k} \times \mathbf{n}_A|.$$

Hence

$$\mathbf{M} = \lambda \mathbf{l}$$

and the multiplier λ is simply a value of torque balancing the wheel vertical orientation. In the wheel model torque \mathbf{M} has to be added to other torques applied to the wheel

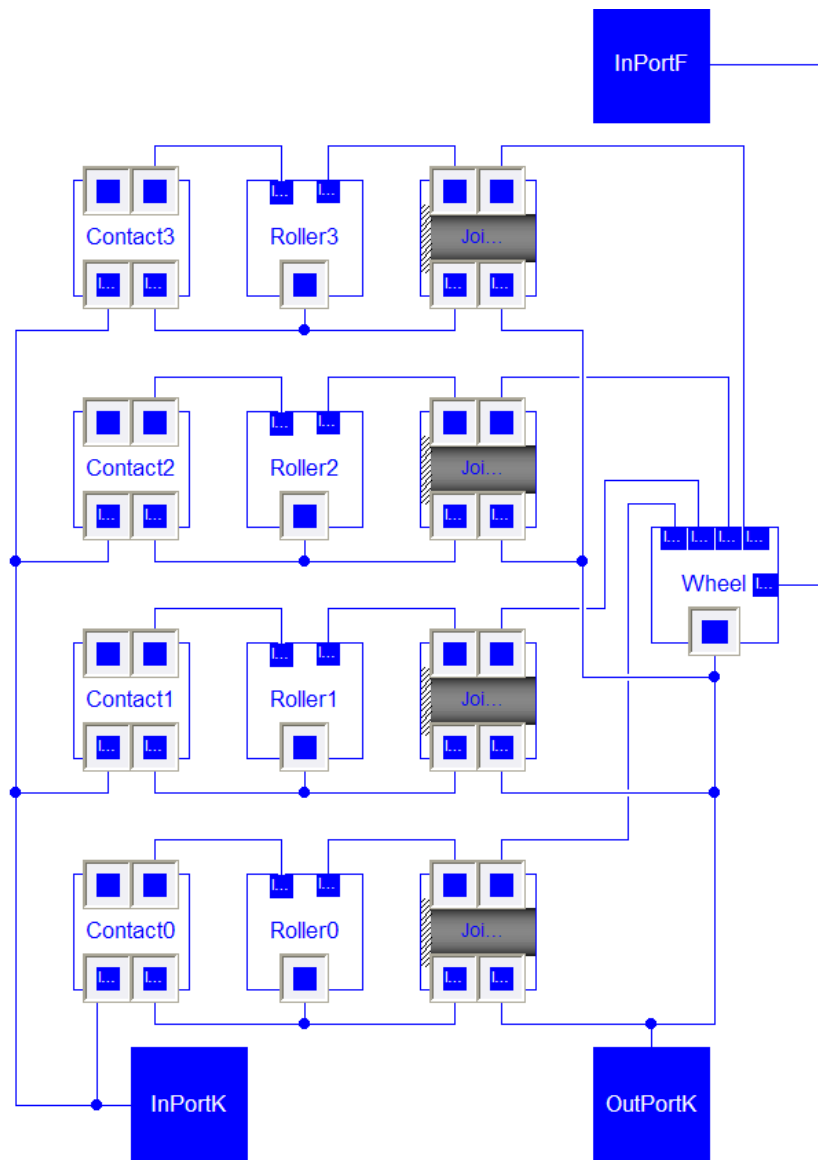


Figure 2. The omni wheel dynamics visual model.

under simulation. The value λ is exactly the Lagrange multiplier corresponding to the servo-constraint above.

It is easy to see the servo-constraint plays here a role of the virtual testbench for investigating the omni wheel dynamics. The remainder of the whole vehicle model is replaced simplistically by the wrench being applied to the wheel. The whole omni wheel dynamics visual model is seen Figure 2

As one can detect here the model of the omni wheel multibody system has been implemented using original multibody dynamics class library developed previously (Kosenko, 2005; Kosenko et al., 2006). One can use this library independently or with help of the known Modelica Standard Multibody class library or with any other Modelica library. The better way being recommended for such use is the following one. Firstly, one can implement mechanical subsystems of the whole sys-

tem under implementation. For instance, mechanisms having tree structure are modeled in a better way using Modelica Standard Multibody Library while mechanical subsystems including unilateral constraints with friction are better implemented using the aforementioned library of classes. Secondly, the only issue remained is to implement proper interfaces using models of ports mapping corresponding signals being transferred from one subsystems to another.

3 Implicit Contact Tracking Algorithm

We will assume in the further course that the wheel plane keeps its vertical orientation permanently. We have to introduce auxiliary orthonormal bases: $b_1 = \{\mathbf{i}_1, \mathbf{j}_1, \mathbf{k}_1\}$

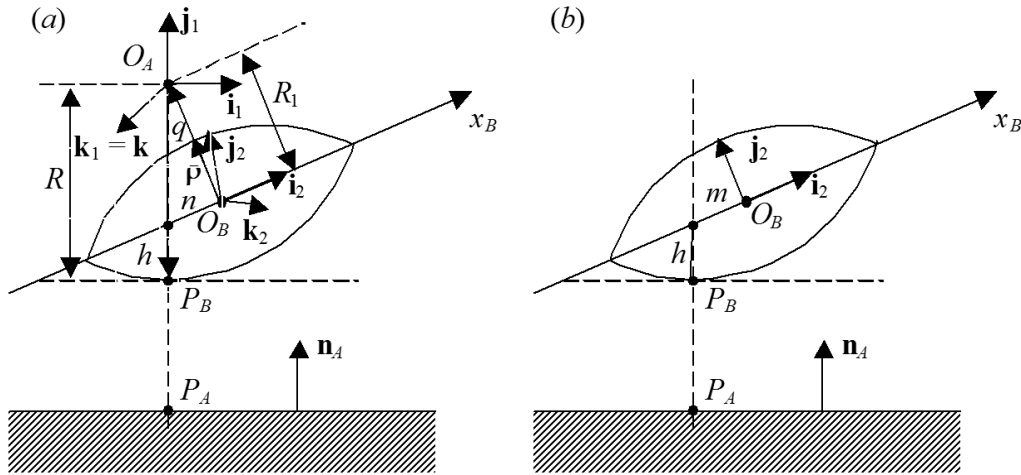


Figure 3. Contact tracking scheme: (a) lateral view of the omni wheel with a roller has been rotated about line $O_A O_B$ by the angle ψ , (b) lateral view of the individual roller.

and $b_2 = \{\mathbf{i}_2, \mathbf{j}_2, \mathbf{k}_2\}$. Intermediate base b_2 characterises partially position and orientation of the roller, while the base b_1 relates to the omni wheel.

The base b_2 coordinate system has its origin O_B at the roller central point. The unit vector \mathbf{i}_2 is directed along the roller axis of rotation, see Figure 3, fragment (a). The unit vector \mathbf{j}_2 is directed orthogonally to \mathbf{i}_2 and lies simultaneously in the vertical plane. The third unit vector \mathbf{k}_2 of the base b_2 is defined in a natural way as

$$\mathbf{k}_2 = \mathbf{i}_2 \times \mathbf{j}_2.$$

Remind here that all unit vectors are computed w. r. t. given fixed (absolute) coordinate system. We assume that positions and orientations are known for all bodies belonging to the multibody system for any instant $t \in [t_0, t_1]$ of simulation process. Therefore, we have

$$\mathbf{i}_2 = T_B \cdot (1, 0, 0)^T, \quad \boldsymbol{\rho} = (\mathbf{r}_{O_A} - \mathbf{r}_{O_B}) / |\mathbf{r}_{O_A} - \mathbf{r}_{O_B}|,$$

where T_B is the roller current orientation matrix.

Origin of the base b_1 coordinate system is located at the point O_A ($= O$ in Figure 1) of the wheel center. The unit vector \mathbf{i}_1 is oriented horizontally and belongs to the wheel plane. The unit vector \mathbf{k}_1 is orthogonal to the wheel plane and is identical to one of the wheel connected base vectors. We assume that using a controller the vector \mathbf{k}_1 permanently maintains its horizontal state. Supposing vector $\mathbf{k}_1(t)$ known we also have $\mathbf{j}_1(t) = (0, 1, 0)^T$ and $\mathbf{i}_1(t) = \mathbf{j}_1(t) \times \mathbf{k}_1(t)$.

Consider now relations providing base b_2 construction. Unit vector \mathbf{i}_2 has been built above. During roller and the floor contact the vector $\mathbf{i}_2(t)$ can not become vertical. Moreover, if the roller distortion takes place, its angle of rotation $\psi > 0$ about axis $O_A O_B$ is fixed non-zero, then the condition $\mathbf{i}_2 \neq (0, 1, 0)^T$ is fulfilled permanently. So we can assume that the condition

$$\mathbf{c} = \mathbf{i}_2 \times (0, 1, 0)^T \neq \mathbf{0}$$

is also fulfilled.

Thus, we can define $\mathbf{k}_2 = \mathbf{c} / |\mathbf{c}|$. And after this we can set $\mathbf{j}_2 = \mathbf{k}_2 \times \mathbf{i}_2$. Geometrical constraints, conditions of orthogonality to be exact, play important role in the omni wheel kinematics

$$\boldsymbol{\rho} \cdot \mathbf{i}_2 = 0, \quad \boldsymbol{\rho} \cdot \mathbf{k}_1 = 0.$$

These equations actually apply to computing the unit vector $\boldsymbol{\rho}$ and we have their differential versions

$$\frac{d}{dt} \boldsymbol{\rho} \cdot \mathbf{i}_2 + \boldsymbol{\rho} \cdot \frac{d}{dt} \mathbf{i}_2 = 0, \quad \frac{d}{dt} \boldsymbol{\rho} \cdot \mathbf{k}_1 + \boldsymbol{\rho} \cdot \frac{d}{dt} \mathbf{k}_1 = 0.$$

The value $c_\beta = \cos \beta = \mathbf{i}_2 \cdot (0, 1, 0)^T$ of cosine for the angle β of the roller axis inclination to vertical $(0, 1, 0)^T$ plays also an important role in the contact tracking algorithm. If current value of the variable c_β is less than some limiting parameter $c_{\beta \max}$, and simultaneously if an altitude of the point O_B defining position of the roller center is less than value R of the wheel radius then the contact takes place. Otherwise no contact occurs.

Note here that in order to arrange the unilateral constraint in the multibody system dynamics model the developer usually has to implement anything like hybrid automata construct. In our omni wheel model, on the contrary, this is not the case. It turned out sufficient to implement “simple” “if” construct to switch states “contact” and “no contact” for each individual roller, and simultaneously to advance forward “contact” state from one roller to its neighbour. The whole picture looks like from time to time neighbouring rollers mutually exchange their states. One can find details of the unilateral constraint implementation in (Kosenko and Gerasimov, 2014). Merely note that “if”-alternatives are the following: (a) “contact” state corresponds to zero-valued relative acceleration of two contacting surfaces at the point of contact, (b) “no contact” branch corresponds to

the zero-valued reaction mutual for both bodies at contact. All this is according to the Signorini rule. “if”-condition depends on the roller orientation variables.

Essential role in all these computations plays a contact tracking algorithm. Generally, its implementation reduces to computation of the contact point/patch which enables computing forces at contact. Usually, one considers contact of two surfaces participating in rigid/elastic interaction of two massive bodies. As a rule, such algorithms are pretty expensive and noticeably slow the whole simulation process. Fortunately, in case of omni wheels we found here the simplest way to make this computation as fast as possible using “elementary” geometric considerations.

We can also easily see from the Figure 3 that the point P_B of contact between roller and floor is obtained using formula

$$\mathbf{r}_{P_B} = \mathbf{r}_{O_B} + R_1 \boldsymbol{\rho} - R \mathbf{j}_1 + \mu \mathbf{k}_1,$$

where the scalar μ is to be computed. Here the value R_1 is the distance between points O_A and O_B . The scalar μ can be computed if we multiply the last equation by \mathbf{k}_2 using dot-product. Thus we have

$$\mu = [R \mathbf{j}_1 \cdot \mathbf{k}_2 - R_1 \boldsymbol{\rho} \cdot \mathbf{k}_2] / \mathbf{k}_1 \cdot \mathbf{k}_2$$

since the vector $\mathbf{r}_{P_B} - \mathbf{r}_{O_B}$ lies in vertical section of axisymmetrical surface of the roller, and the vector \mathbf{k}_2 by construction is orthogonal to this section. As a result the position \mathbf{r}_{P_B} of the contact point P_B is uniquely computed.

4 Explicit Contact Tracking Algorithm

Yet another way to obtain current position \mathbf{r}_{P_B} of the contact point P_B , or more accurately: the roller point closest to the floor, is an application of the following chain of equations. This chain is simply understood from geometrical scheme shown in Figure 3, (a) and (b).

$$\mathbf{r}_{P_B} = \mathbf{r}_{O_B} - m \mathbf{i}_2 - h \mathbf{j}_1,$$

where $m = R_1 \sin q / \cos q / \cos \psi$, $h = R - R_1 / \cos q$, q is the current value for angle of deviation of the vector $\boldsymbol{\rho}$ from direction of the vector \mathbf{j}_1 . So we have

$$\cos q = \boldsymbol{\rho} \cdot \mathbf{n}_A, \quad \sin q = (\mathbf{n}_A \times \boldsymbol{\rho}) \cdot \mathbf{k}_1.$$

Here we give explanations of some details of Figure 3. Fragment (a) corresponds to the lateral projection of the wheel and likewise the distorted roller projection. This latter object is shown here in a general position. Furthermore, P_B is the current contact point between the roller and the horizontal floor, n is a projection of the roller axial line segment onto the wheel plane. We can see easily that this projection is computed by the formula $n = m \cos \psi$ because the roller axis is turned about $O_A O_B$

by the angle ψ , see fragment (b) for the roller axial vertical lateral section. Thus, we have to pass two straight line segments from the roller center O_B to reach the point P_B : (a) the segment of the roller axis of length m ; (b) the segment down the vertical of length h . As we already mentioned above all variables needed are computed through known variables using explicit formulae.

In case of $\psi > 0$, distortion exists, for both implicit and explicit algorithms not all the length of the roller surface generatrix is necessarily in contact. So really we have to cut tips of rollers to provide regular simulation process. Length of the tip to be cut we can obtain for instance empirically or compute it explicitly. Indeed, one can easily see from Figure 3 that the real roller length should be computed by the formula

$$L = 2R \sin \alpha / \cos \psi.$$

“Ideal” switching of contact takes place in this case: exactly at the instant of contact loss for current roller a contact immediately arises for the “next” roller in direction of the wheel rolling.

5 The Wheel Model Classes Hierarchy

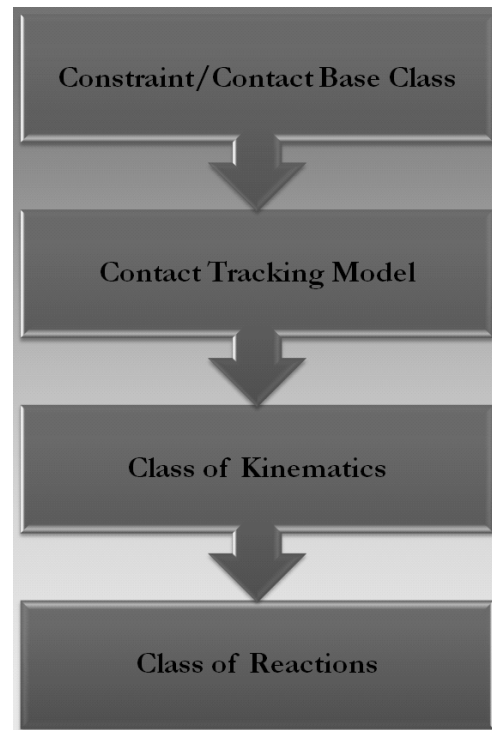


Figure 4. Contact model by stages of inheritance.

Model of the omni wheel testbench virtual prototype is a container class including the following objects instantiated: (a) disk of the wheel; (b) objects of rollers

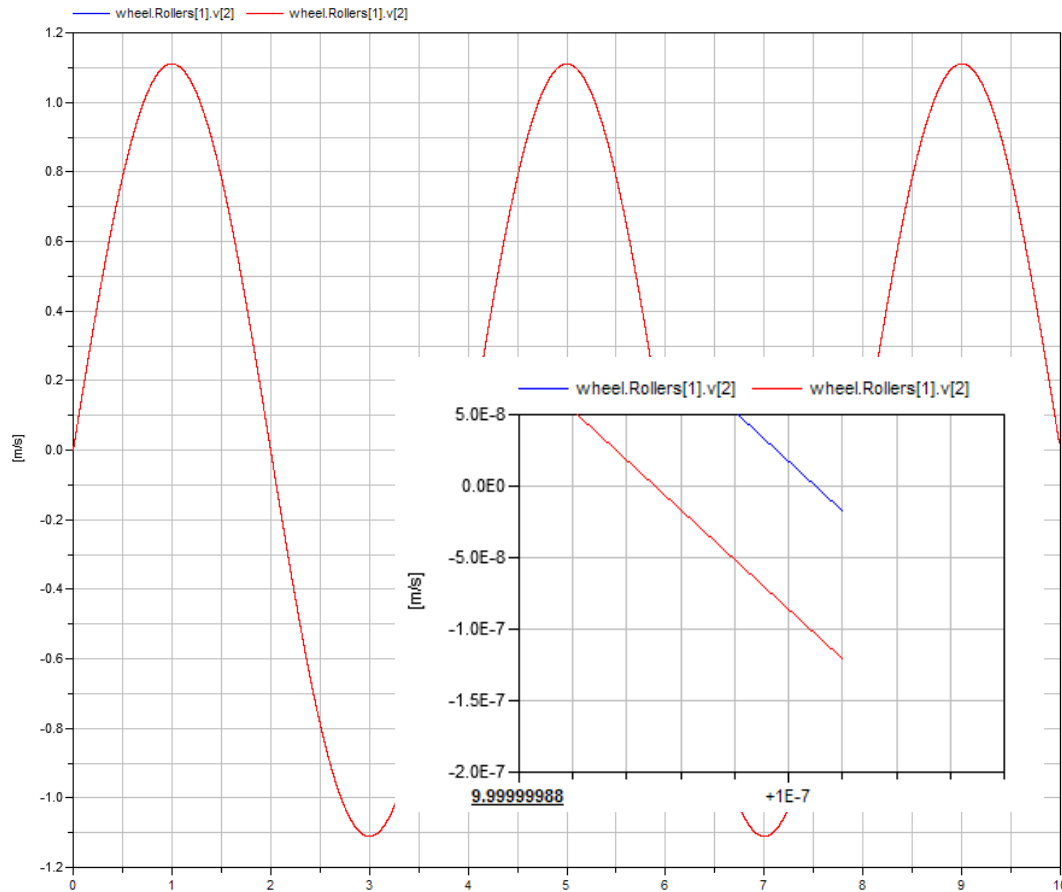


Figure 5. Comparison of dynamics for the roller No. 1 central point, its velocity y-coordinate, for cases of: explicit (blue curve) and implicit (red curve) algorithm of contact tracking.

mounted along the wheel rim; (c) objects of joints connecting rollers and the wheel disk; (d) objects of contacts connecting objects of rollers and the object of the horizontal floor surface; (e) model of base body as a horizontal floor.

Let us analyse in more details a structure of contact model. This model has many similarities with contact models previously considered (Kosenko, 2005). Nevertheless important differences exist. One of them mentioned above with regard to organization of the contact class using simple and efficient construct (Kosenko and Gerasimov, 2014). Note that in case of $\psi > 0$ the point of contact creates a curve with discontinuities at instances of rollers changes. However, this circumstance does not prevent the process of regular simulation.

Finally, we apply rigid point contact model as part of the simplest omni wheel model. For this we use the base class for constraint/contact models having only equations of Newton's third law as a behavioral section (Kosenko et al., 2006).

We use class of the contact tracking model on the second stage of inheritance, see Figure 4. Cases of this class organization have been analysed above. Coordinates of nearest points P_A and P_B at contact for each pair (floor, roller) are computed as a result for this class functional-

ity.

Class for computing all kinematical characteristics at contact needed “works” in case of contact existence on the next stage of inheritance. On the third stage class for computing the reactions at contact is “turned on”. Reactions are the following: (a) normal reaction; (b) tangent force of friction; (c) torque of reactions (zero in the current consideration though it is not difficult to compute torque for several contact models).

To verify an approach for building up the models under analysis we compare the omni wheel dynamics in cases of implicit and explicit algorithms. The wheel performs free motion (combining rotation and sliding) with the only restriction: keep vertical alignment of the wheel disk.

Roller No. 1 central point, its mass center, altitude was analyzed and verified. More accurately we examine y-coordinate of the point velocity. Both models turned out almost identical: in the worst case we have a divergence: in accelerations of order 10^{-8} , in velocities of order 10^{-7} , in position of order 10^{-6} over the time span being equal to 10 units of time. Results of simulation for velocities are shown in Figure 5. Other divergencies for the roller No. 1 central point acceleration and position at time = 10 units are shown in Figure 6 and 7 respectively.

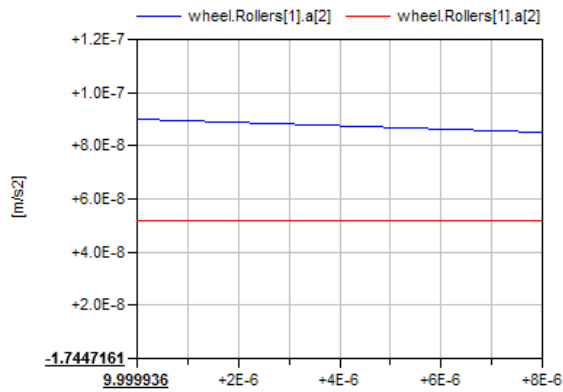


Figure 6. Divergence for y-components of acceleration.

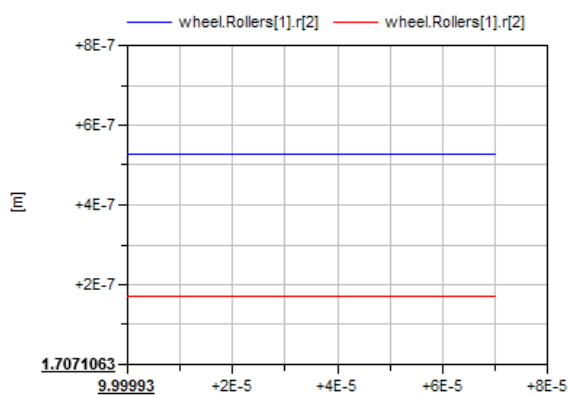


Figure 7. Divergence for y-components of position.

As expected the model with explicit contact tracking algorithm is faster approximately in 1.5 times.

6 Conclusions

The following effects were found as a results of new contact tracking algorithms applying to the omni wheel multibody system:

1. Two contact tracking algorithms were proposed: implicit and explicit. As expected the second algorithm turned out to be faster almost in 1.5 times. Both algorithms are simple (and efficient) even in simpler case of rollers without any distortion.
2. In case of distorted rollers contact curve becomes discontinuous at instants of rollers change. But simulation process maintains its regularity.
3. Both algorithms generate identical dynamics.
4. Process of the contact model design using technology of “vertical separation” outlined above has an

evident motivation and allows a simple generalization both for the normal force computation and for the tangent friction force model.

7 Acknowledgement

The investigation was performed under financial support provided by RSF, project 14-21-00068.

References

- B. E. Ilon. Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base. Technical report, US Patents and Trademarks office, Patent 3,876,255, 1975.
- V. Kálmán. Controlled braking for omnidirectional wheels. *International Journal of Control Science and Engineering*, 3(2):48–57, 2013.
- I. Kosenko and K. Gerasimov. Implementation of the omni vehicle dynamics on Modelica. In *Proceedings of the 10th International Modelica Conference*, pages 311–322, March 2014.
- I. I. Kosenko. Implementation of unilateral multibody dynamics on Modelica. In *Proceedings of the 4th International Modelica Conference*, pages 13–23, March 2005.
- I. I. Kosenko, M. S. Loginova, Ya. P. Obratsov, and M. S. Stavrovskaya. Multibody systems dynamics: Modelica implementation and bond graph representation. In *Proceedings of the 5th International Modelica Conference*, pages 213–223, September 2006.
- J. Tobolár, F. Herrmann, and T. Bünte. Object-oriented modelling and control of vehicles with omni-directional wheels. In *Computational Mechanics 2009*, November 2009.