14th World Congress on Computational Mechanics (WCCM)
ECCOMAS Congress 2020)
Virtual Congress: 11-–15 January 2021
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

HOW ONE CAN ARRANGE THE MULTIBODY SYSTEM DYNAMICS COMPUTER MODEL

Ivan I. Kosenko[1]

[1] Moscow Aviation Institute (National Research University), Volokolamskoe shosse, 4, Moscow, 125993, Russia, kosenkoii@gmail.com

Key words: Graph Representations, Visual Model, Quaternion Algebra, Contact Models

Abstract. The symplest computer models for mechanical systems can be organized using Cauchy normal form. Further complicating the mechanical model is able to bring us to application the implicit functions of next complexity level. Process of the multibody system computer models development is of special difficulty.

There exist different ways for organizing such a models. Mainly these ways are reduced to the model transformation to the form of differential-algebraic equations (DAEs). These latter ones correspond to Lagrange equations of the first kind. Note that usually differential equations of DAEs mentioned correspond to dynamical and kinematical equations of mechanics, while the algebraic equations are generated by constraints.

Computational experience makes it possible to classify objects of the multibody system dynamics [1]. Such a model includes two classes of objects. They correspond to notions of “body” and “constraint”. Let us also remark that these two classes of objects define the structure of the undirected graph such that “bodies” play a role of the graph vertices, while “constraints” play the role of edges.

There exists yet another graph interpretation using the bi-chromatic bipartite graph. In this case both bodies and constraints are interpreted as vertices. Objects of bodies compose a partition and are coloured by one colour while objects of constraints compose the complement partition of the whole graph and are coloured by another colour. Edges connecting vertices of partitions for the graph are arranged in a way such that for any vertex of constraint there exist exactly two vertices of class “body” thus implementing participation in the constraint. Two ways for the multibody system dynamics computer model graph composition mentioned above define ways for constructing the visual model of such a system thus defining ports interconnection structure.

Different cases of the multibody system dynamics computer model implementation were analysed as an examples. Models under construction are the following ones: (a) Rattleback; (b) Snakeboard; (c) Skateboard; (d) Tippe-Top; (e) Ball Bearing; (f) Spur Involute Gear; (g) Omni–Vehicle.

1 INTRODUCTION

When developing a computer model of the multibody system, MBS, it is interesting to have a unified technology to construct the models in an efficient way. It turns out Modelica language provides a tools to resolve such a problem successively step by step using its natural approaches Refs. [2, 3]. One of them is connected tightly with the so-called multiport representation of the models initially based on the bond graph use Ref. [4]. These latter ones in turn based on the idea of energy interaction Ref. [5], and substantially on energy conservation for physically interconnected subsystems of any engineering type.

Moreover, Modelica introduces the notions similar to ones of the bond graph theory, but in a way more natural for the usual engineering approaches with forces, interfaces, parameters, equations etc. Ref. [6]. Consider in the sequel a technology to construct a model of MBS dynamics with constraints of any specific type in a unified way. Note that the unilateral constraints can also be included in the further consideration process Ref. [7].

A lot of methods to describe the structure of the MBS using different graph approaches is known, see for instance Ref. [8]. Consider the MBS consisting of $m+1$ bodies $B_0,\ldots,B_m$. Represent it as a set $\mathcal{B} = \{B_0,\ldots,B_m\}$. Here B_0 is assumed to be a base body. We suppose B_0 to be connected with an inertial frame of reference, or to have a known motion with respect to the inertial frame of reference. For example one can imagine the base body as a rotating platform, or as a vehicle performing its motion according to a given law. For definiteness and simplicity we suppose in the sequel all state variables describing the rigid bodies motion always refer to one fixed inertial coordinate system connected to the base body by default.

Some bodies are considered as connected by mechanical constraints. Suppose all constraints compose the set $\mathcal{C} = \{C_1,\ldots,C_n\}$. We include in our considerations constraints of the following types: holonomic/nonholonomic, scleronomic/rheonomic.

Thus one can uniquely represent a structure of the MBS via a undirected graph $G = (\mathcal{B},\mathcal{C},I)$. Here $I \subset \mathcal{C}\times\mathcal{B}$ is an incidence relation setting in a correspondence the vertex incident to every edge $C_i \in \mathcal{C}$ of the graph. According to physical reasons it is easy to see that for any mechanical constraint C_i there exist exactly two bodies $B_k, B_l \in \mathcal{B}$ connected by this constraint.

2 AN APPROACH TO CONSTRUCT THE LIBRARY OF CLASSES

2.1 General description

It is clear that consideration of the graph G does not provide a structural information sufficient for the MBS dynamics description. Indeed, in addition to the force interaction represented usually by wrenches between bodies B_k, B_l through the constraint C_i there exist kinematic conditions specific for different kinds of constraints. Wrenches themselves can be represented in turn by constraint forces and constraint torques couples. These forces and couples are connected by virtue of Newton's third law of dynamics.

Thus if the system of ODEs for translatory-rotary motion can be associated with the object of a model corresponding to rigid body, then the system of the algebraic equations can be naturally associated with the object of a model corresponding to constraint. Note that according to above consideration the set of algebraic equations comprises relations for constraint wrenches, and kinematic relations depending on the certain type of constraints.

Thus all the "population" of any MBS model is reduced to objects of two classes: *RigidBody* (objects $B_0,\ldots,B_m$), *Constraint* (objects $C_1,\ldots,C_n$). According to this approach simulation of the whole system behavior reduces to permanent information interaction between the objects of two considered types. Within the frame of Newton's plus Euler's models of dynamics one can construct the MBS as a communicative network for this interaction. In this case the objects of bodies "feel" the action of other ones

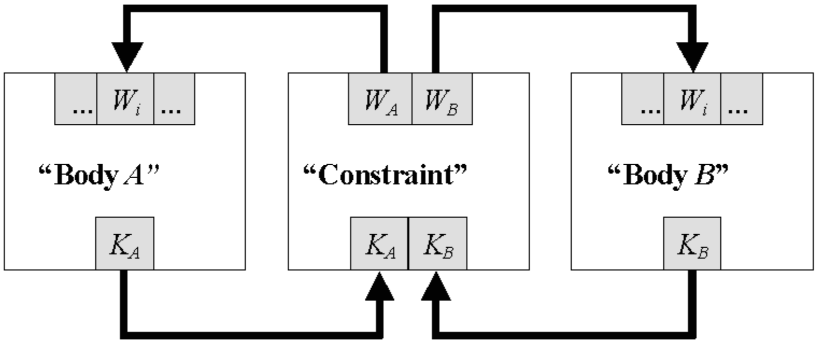

Figure 1: Architecture of constraint

through corresponding objects of constraints.

Physical interactions are conducted in models due to objects splitted also in two classes of ports: *WrenchPort*, *KinematicPort*. The first one is to be used to transfer wrench. In addition, *WrenchPort* has to be used for transferring the information about current location of the point constraint force acts upon.

In our idealized model the force interaction between bodies supposed exactly at a geometric point. Its coordinates are fed outside constraint object through *WrenchPort* permanently in time.

Now it is possible to describe an architecture of information interactions within the particular constraint C_i corresponding to an individual edge of graph G, see Fig. 1.

KinematicPort is to be used to transfer data of the rigid body kinematics: configuration (position of center of mass, orientation), velocity (velocity of the center of mass, angular rate), and acceleration (acceleration of the center of mass, angular acceleration) containing in particular information about twist. When getting force information through ports $W_1, \dots, W_s$ from the incident objects of class *Constraint* the object of class *RigidBody* simultaneously generates, due to an integrator, kinematic information being fed outside through the port K. On the other hand every object of class *Constraint* gets kinematic data from the objects corresponding to bodies connected by the constraint under consideration through its two "input" ports K_A, K_B. Simultaneously using the system of algebraic equations this object generates information concerning wrenches, and transmits the data to "output" ports W_A, W_B for the further transfer to objects of bodies under constraint.

For simplicity and clearness we will apply the component library to simulate the dynamics of MBSs with bilateral constraints. Application of the components for the unilateral constraints Refs. [9, 7] doesn't change anything in principle. The only difference is that dynamics of moving bodies becomes more complicated. For example in the latter case a vehicle under simulation get an ability to bounce over the uneven surface it rolls on. In addition, its wheels can slip while moving. Thus in frame of the current paper we suppose that nonholonomic constraints are implemented as exact ones, without any slip or separation with respect to (w. r. t.) the surface.

In superclass *RigidBody* dynamics of rigid body is described here by means of Newton's differential equations for the body mass center, and by Euler's differential equations for the rotary motion. Note that to be able to have an invariant description of the rotary motion one can use an excellent tool: quaternion algebra $\mathbf{H}$. In this case we "lift" the configuration manifold from $SO(3)$ to $S^3 \subset \mathbf{H}$ and then implement dynamics of rotation in flat space $\mathbf{H} \cong \mathbf{R}^4$ taking into account that S^3 is an invariant manifold of the rotary dynamics redefined on $\mathbf{H}$. In this way we have only one flat chart $\mathbf{H}$ for the underlying due to double covering configuration manifold $SO(3)$, and needn't in any special choices of the configuration angles or anything like that.

The double covering $S^3 \to SO(3)$, $\mathbf{q} \mapsto T$ mentioned implemented inside the *RigidBody* class by the known formula

$$T = \frac{1}{|\mathbf{q}|^2} \cdot \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix},$$

where q_0, q_1, q_2, q_3 are the Euclidean coordinates of the quaternion $\mathbf{q}$ and $|\mathbf{q}|$ is its Euclidean norm. The rotation matrix T is fed outside the object through the *KinematicPort* permanently in time. The Euler equations are constructed using quaternion algebra in a well known way.

Remind that according to our technology of the constraint construction two connected bodies are identified by convention with the letters A and B fixed for each body. All kinematic and dynamic variables and parameters concerned one of the bodies are equipped with the corresponding letter as a subscript.

All objects of the class *Constraint* must have classes-inheritors as subtypes of a corresponding superclass. According to Newton's third law this superclass must contain the equations of the form

$$\mathbf{F}_A + \mathbf{F}_B = \mathbf{0}, \quad \mathbf{M}_A + \mathbf{M}_B = \mathbf{0}. \tag{1}$$

in its behavioral section. Here arrays $\mathbf{F}_A$, $\mathbf{M}_A$ and $\mathbf{F}_B$, $\mathbf{M}_B$ represent constraint forces and torques "acting in directions" of bodies A and B correspondingly. Kinematic equations for different types of constraints are to be added to Eqs. (1) in different classes-inheritors corresponding to these particular types of constraints.

2.2 Implementation of the joint constraint

Class *Joint* plays a key role in the future model of a vehicle we will build. Moreover, this class is important while development the dynamics for machines of different categories. *Joint* is a model derived from the base class *Constraint*. Remind Ref. [9] that in order to make a complete definition of the constraint object behavior for the case of rigid bodies one has to compose a system of twelve algebraic equations w. r. t. twelve coordinates of vectors $\mathbf{F}_A$, $\mathbf{M}_A$, $\mathbf{F}_B$, $\mathbf{M}_B$ constituting the wrenches acting upon the connected bodies.

First six Eqs. (1) always present in the base model *Constraint* due to Newton's third law. For definiteness suppose these six equations are used to express six components of $\mathbf{F}_B$, $\mathbf{M}_B$ depending on $\mathbf{F}_A$, $\mathbf{M}_A$. Thus six components of $\mathbf{F}_A$, $\mathbf{M}_A$ remain as unknowns. To determine them each constraint of rigid bodies need in six additional independent algebraic equations. These equations may include components of force and torque directly, or be derived from the kinematic relations corresponding to specific type of the constraint.

In the case of the joint constraint being investigated here let us represent the motion of the body B as a compound one consisting of the body A transporting motion w. r. t. an inertial frame of reference, and a relative motion w. r. t. the body A. An absolute motion is one of the body B w. r. t. the inertial system.

Define the joint constraint with help of the following parameters: (a) a unit vector $\mathbf{n}_A$ defining in the body A an axis of the joint; (b) a vector $\mathbf{r}_A$ fixed in the body A and defining a point which constantly stays on the axis of the joint; (c) a vector $\mathbf{r}_B$ fixed in the body B and defining a point which also constantly stays on the axis of the joint. The main task of the base joint class is to keep always in coincidence the geometric axes fixed in each of the bodies.

First of all one has to compute the radii vectors of the points fixed in the bodies w. r. t. inertial system

$$\mathbf{R}_\alpha = \mathbf{r}_{O_\alpha} + T_\alpha \mathbf{r}_\alpha \quad (\alpha = A, B),$$

where Ref. [9] $\mathbf{r}_{O_\alpha}$ is the position of the α-th body center of mass, T_α is its current matrix of rotation. The joint axis has the following components

$$\mathbf{n}_{Ai} = T_A \mathbf{n}_A$$

in the inertial frame of reference. According to the equation for relative velocity for the marked point of the body B defined by the position $\mathbf{R}_B$ we have

$$\mathbf{v}_{Ba} = \mathbf{v}_{Be} + \mathbf{v}_{Br}, \quad \mathbf{v}_{Ba} = \mathbf{v}_{O_B} + [\boldsymbol{\omega}_B, T_B \mathbf{r}_B], \quad \mathbf{v}_{Be} = \mathbf{v}_{O_A} + [\boldsymbol{\omega}_A, \mathbf{R}_B - \mathbf{r}_{O_A}], \tag{2}$$

where $\mathbf{v}_{Ba}$, $\mathbf{v}_{Be}$, $\mathbf{v}_{Br}$ are an absolute, transporting, and relative velocities of the body B marked point, $\boldsymbol{\omega}_A$, $\boldsymbol{\omega}_B$ are the bodies angular velocities.

Furthermore, according to the computational experience of the dynamical problems simulation the pre-compiler work is more regular if the kinematic equations are expressed directly through accelerations. Indeed, otherwise the compiler tries to perform the formal differentiation of equations for the velocities when reducing an index of the total DAE system. Frequently this leads to the problems either during time of translation or when running the model.

Thus using the known Euler formulae for the rigid body kinematics and the Coriolis and Rivals theorems we obtain an equations for the relative linear acceleration in the form

$$\begin{array}{ll} \mathbf{a}_{Ba} = \mathbf{a}_{O_B} + [\boldsymbol{\varepsilon}_B, T_B \mathbf{r}_B] + [\boldsymbol{\omega}_B, [\boldsymbol{\omega}_B, T_B \mathbf{r}_B]], & \mathbf{a}_{Ba} = \mathbf{a}_{Be} + 2[\boldsymbol{\omega}_A, \mathbf{v}_{Br}] + \mathbf{a}_{Br}, \\ \mathbf{a}_{Be} = \mathbf{a}_{O_A} + [\boldsymbol{\varepsilon}_A, \mathbf{R}_B - \mathbf{r}_{O_A}] + [\boldsymbol{\omega}_A, [\boldsymbol{\omega}_A, \mathbf{R}_B - \mathbf{r}_{O_A}]], & \mathbf{a}_{Br} = \mu \mathbf{n}_{Ai}, \end{array} \tag{3}$$

where $\mathbf{a}_{Ba}$, $\mathbf{a}_{Be}$, $\mathbf{a}_{Br}$ are an absolute, transporting, and relative accelerations of the body B marked point, $\boldsymbol{\varepsilon}_A$, $\boldsymbol{\varepsilon}_B$ are the bodies angular accelerations, $\mathbf{v}_{Br}$ is a relative velocity of the body B marked point, $\boldsymbol{\omega}_A$, $\boldsymbol{\omega}_B$ are the bodies angular velocities.

We also need in an analytic representation of the conditions that the only projections of the bodies angular velocities and accelerations having a differences are ones onto the joint axis. Corresponding equations have the form

$$\boldsymbol{\omega}_B = \boldsymbol{\omega}_A + \boldsymbol{\omega}_r, \quad \boldsymbol{\varepsilon}_B = \boldsymbol{\varepsilon}_A + [\boldsymbol{\omega}_A, \boldsymbol{\omega}_r] + \boldsymbol{\varepsilon}_r, \quad \boldsymbol{\varepsilon}_r = \lambda \mathbf{n}_{Ai}, \tag{4}$$

where $\boldsymbol{\omega}_r$, $\boldsymbol{\varepsilon}_r$ are the relative angular velocities and accelerations.

Besides the kinematic scalars μ, λ we will need in their reciprocal values $F = (\mathbf{F}_A, \mathbf{n}_{Ai})$, $M = (\mathbf{M}_A, \mathbf{n}_{Ai})$ correspondingly. Note that the class described above is a partial one (doesn't yet complete the constraint definition) and can be used to produce any imaginable model of the joint type constraint. To obtain a complete description of the joint model one has to add to the behavioral section exactly two equations.

One of them is to define one of the values μ, F (translatory case). Other equation is intended to compute one of the values λ, M (rotary case).

Consider several examples of the classes derived from the *Joint* model for the several particular types of joints. The model *FixedIdealJoint* is defined by the equations

$$\mu = 0, \quad M = 0$$

and prevents the relative motion along the joint axis but allows free rotation about it. It is exactly a revolute joint without any control for the rotary motion. The model *FreeIdealJoint* is defined by the equations

$$F = 0, \quad M = 0$$

permitting free translation along and free rotation about the joint axis. Class *SpringIdealJoint* is described by the equations

$$F = c\nu + d\dot{\nu}, \quad M = 0, \quad \ddot{\nu} = \mu$$

with the initial data $\nu(t_0) = 0$, $\dot{\nu}(t_0) = 0$ for the relative translatory position ν provides a visco-elastic compliance with the stiffness c and damping d. The rotary motion remains free. This model is useful to simulate almost rigid constraints to avoid the potential problems with so-called statically undefinable systems of forces acting upon the ideal rigid bodies.

The model *FixedControlledJoint* with the behavior defined by the equations

$$\mu = 0, \quad M = f(t, \varphi, \dot{\varphi}), \quad \ddot{\varphi} = \lambda \tag{5}$$

provides the rotating torque as a control effort with the prescribed control function $f(t, \varphi, \dot{\varphi})$. Initial data $\varphi(t_0) = \varphi_0$, $\dot{\varphi}(t_0) = \dot{\varphi}_0$ are prepared according to the initial data concerning the joint. This type of joint corresponds to the *Revolute* joint constraint of Modelica Standard Library from the *ModelicaAdditions* package. Such a joint can be driven by the electromotor.

The model *FreeSlideJoint* defined by the equations

$$F = 0, \quad \lambda = 0$$

provides free, without any resistance, relative sliding along the joint axis without any rotation about it. As one can see this is a prismatic type of joint.

We can reformulate the *FixedControlledJoint* model creating the model *FixedServoJoint* in a following useful way

$$\mu = 0, \quad \lambda = f(t, \varphi, \dot{\varphi}), \quad \ddot{\varphi} = \lambda$$

thus composing a kinematic restricting constraint, so-called servoconstraint. The function $f(t, \varphi, \dot{\varphi})$ supposed as a prescribed one. Initial data for the angle φ of the relative rotation are prepared in the same way as for Eqs. (5). It is clear one can create a lot of other different combinations of equations to construct the joint constraints needed in engineering applications.

It is clear one can create a lot of other different combinations of equations to construct the joint constraints of any desirable type. The only issue one has to take into account when constructing the kinematic pair that if there exists an obstacle for the relative motion eliminating one DOF then it causes an additional scalar kinematic equation lifted to the level of accelerations. Reciprocally if the corresponding motion is possible then the developer has to include the scalar equation instead imposing the condition on the matching generalized effort, force or torque what applicable.

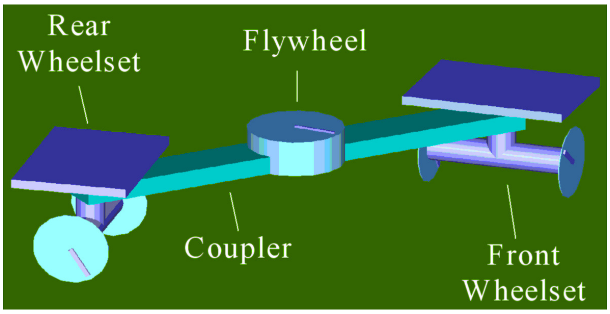

Figure 2: The snakeboard

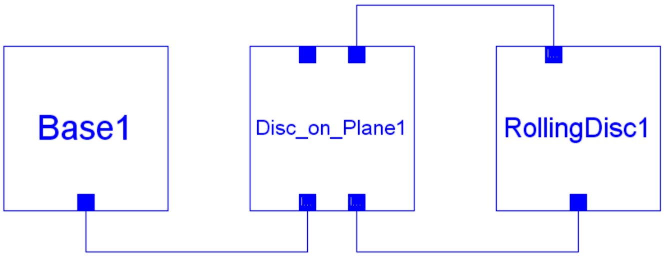

Figure 3: Visual model of the rolling disc

3 EXAMPLE OF A VEHICLE

The snakeboard Ref. [10], see Fig. 2, represents a four wheeled vehicle moving in field of gravity on a horizontal surface due to the servocontrol of a relative rotation of wheelsets and a flywheel located at the midpoint of the coupler and having a vertical axis of rotation. The flywheel simulates a torso of the snakeboard rider.

We will construct the model hierarchically step by step verifying and integrating the parts into an assembly units. Ideal mechanical system of the snakeboard has three degrees of freedom (DOF). But we will add new DOFs on some stages of modeling either to make the model more physically oriented or to apply any procedures of regularization.

3.1 Dynamics of the rolling disc

This problem is a classic one of dynamics Ref. [11] and has a visual representation depicted in Fig. 3.

Disc, the Body B, supposed an axisymmetric rigid body which is able to roll on the another body, horizontal surface, only by the curve fixed in the Body B. In our case this curve supposed a circle relocated in the plane $z_B = 0$ of the Body B coordinate system $O_B x_B y_B z_B$ and has the fixed radius R, see Fig. 4. In the current paper we assume that the nonholonomic constraints are implemented in an accurate sense as bilateral constraints.

The horizontal plane, Body A, is defined by its normal unit vector such that radius vector $\mathbf{r}_P = \{x_P, y_P, z_P\}$ of the contact point P has to satisfy an equation of the horizontal plane $(\mathbf{r}_P, \mathbf{n}_A) = 0$.

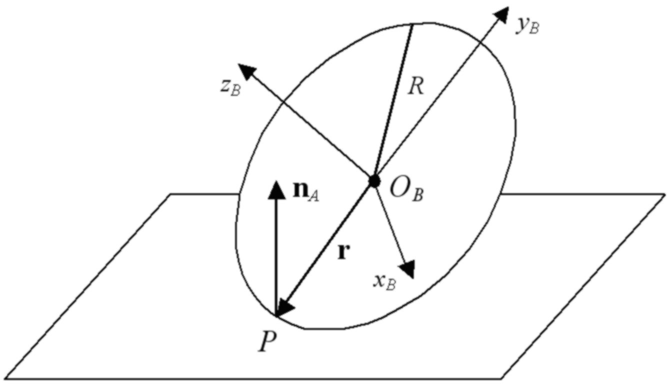

Figure 4: Rolling disc

Further denoting the Body B current orientation matrix by T_B and by $\mathbf{r}_{O_B}$ its center of mass position vector we obtain the system of three equations $T_B\mathbf{r} = \mathbf{r}_P - \mathbf{r}_{O_B}$ defining the dependence between the vector $\mathbf{r}_P$ and the vector $\mathbf{r}$ of the contact point position in the Body B coordinate system.

On the other hand the vector $\boldsymbol{\tau}$ tangent to the circle at the contact point can be expressed in the disc coordinates as $\boldsymbol{\tau} = \{-y_B, x_B, 0\}$ because the vectors $\boldsymbol{\tau}$ and $\mathbf{r} = \{x_B, y_B, z_B\}$ are to be mutually orthogonal and to be situated in the disc plane permanently. In addition, in inertial system the path vector $T_B\boldsymbol{\tau}$ has to lie in the horizontal plane. Then also holds the condition $(\mathbf{n}_A, T_B\boldsymbol{\tau}) = 0$.

The system of six Eqs. from above together compose the one w. r. t. six variables x_P, y_P, z_P, x_B, y_B, z_B and implements in a simple and effective way the model *Disc_on_Base* derived from the class *Roll* of the relative rolling definition.

Verification of the model outlined above was based on the comparison of its simulation results with ones obtained for the corresponding classic problem defined by the system of ODEs Ref. [11] expressed w. r. t. the Body B rotating system. These ODEs depend on the following variables: $\mathbf{M}$ is the vector of the disc angular momentum computed w. r. t. the contact point, $\boldsymbol{\omega}$ is its angular velocity, $\mathbf{r}$ is the vector already mentioned above, $\boldsymbol{\gamma}$ is the unit vector $\mathbf{n}_A$ but expressed w. r. t. the Body B system.

The simulations showed a high degree of accordance between the two above models of the rolling disc dynamics. Errors increase inevitably and for the vectors $\boldsymbol{\omega}$, $\mathbf{M}$, $\boldsymbol{\gamma}$ errors for components are of the order 10^{-7} over the time interval of the several hundreds units.

3.2 Model of the wheelset

This model plays an important role when constructing the simplest vehicle models. It is assembled using the considered model of the rolling disc. Visual model of the wheelset is depicted in Fig. 5. Application of the model *FixedIdealJoint* for joints connecting the wheels and a rod of the wheelset axis is impossible due to the uncertainty for forces acting along this axis. If the contact points with a floor supposed without slipping then introduction of the compliance in the joints is a natural way to avoid the degeneracy mentioned. Making this we add two DOFs to the mechanical system of the wheelset. One else additional DOF has the rod rotating independently about its, and of the wheelset, axis. Compliances are implemented by the model *SpringIdealJoint*.

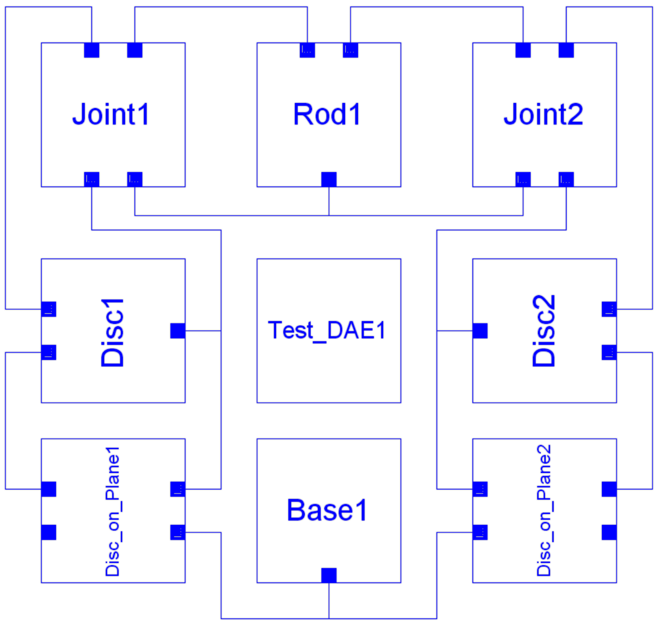

Figure 5: Visual model of the isolated wheelset

To verify the wheelset model built the proper analytical system of DAEs was applied. this system is written w. r. t. moving coordinate system connected with the wheelset in an evident way. This system of coordinates performs a transporting motion tracing the motion of the rod which plays a role of the wheelset axis shaft.

The DAE system consists of twelve equations w. r. t. twelve unknowns. Computational experiments show a high degree of concordance between our “physically oriented” model of the wheelset and the ideal model described above if the parameters of stiffness c and damping d in the joint objects of class *SpringIdealJoint* are large enough. Namely, in simulations we have used the values $c = 1000$, $d = 5000$.

3.3 Model of the vehicle

Let us construct at last a complete model of the snakeboard. Its visual representation see in Fig. 6. Similar to the wheelset case we have here a static indeterminacy along the coupler axis if one supposed a rigid body. To avoid this degeneration we splitted it into two equal parts and connected them via viscoelastic joint, with an axis along the coupler, using the model *SpringIdealJoint* with the stiffness and damping large enough for the longitudinal compliance of the snakeboard.

To perform a comparison with the known results Ref. [10] three servoconstraints were introduced to the model. These servoconstraints imitate the control of the robot-snakeborder and are implemented by the *FixedServoJoint* class which defines a relative rotation of the bodies by the prescribed angle. To be more precise in the class mentioned the control is given by a law of the relative acceleration with a proper initial values of the angle and the angular velocity.

Servoconstraints are mounted at the joints between the coupler and the wheelsets, and between the flywheel and, for definiteness, the left part of the coupler. The joints mentioned correspond to the objects

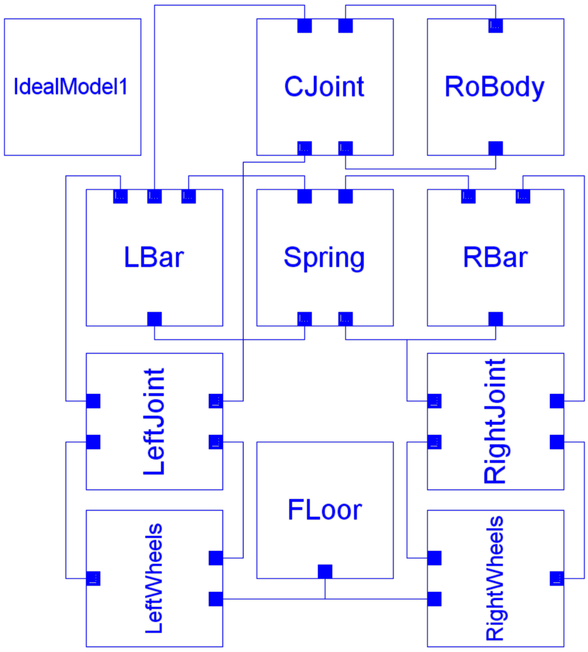

Figure 6: Visual model of the snakeboard

LeftJoint, *RightJoint*, and *CJoint* in Fig. 6. All three servoconstraints can be described by the equations

$$\varphi_f = a_f \sin(\omega_f t + \beta_f), \quad \varphi_b = a_b \sin(\omega_b t + \beta_b), \quad \psi = a_\psi \sin\left(\omega_\psi t + \beta_\psi\right),$$

where φ_f, φ_b are the angles of the front and rear (back) wheelsets relative to the coupler rotation correspondingly, ψ is the angle of the flywheel rotation w. r. t. the coupler, to be more exact relative to its left (rear) part, the object *LBar* in Fig. 6, a_f, a_b, a_ψ are the corresponding amplitudes of libration, ω_f, ω_b, ω_ψ are their frequencies, and β_f, β_b, β_ψ are their initial phases.

According to Ref. [10] three types of the snakeboard gait were under verification:

1. "drive": $a_b = -a_f$, $\omega_f = \omega_b = \omega_\psi$;
2. "rotate": $a_b = -a_f$, $2\omega_f = 2\omega_b = \omega_\psi$;
3. "parking": $a_b = -a_f$, $3\omega_f = 3\omega_b = 2\omega_\psi$.

The simulation results showed a full coincidence of the gait types for our regularized model and the idealized model of the paper Ref. [10]. In Ref. [10] for the ideal model when deriving the DAEs of the snakeboard motion for simplicity of the model the wheels rotary motion and the wheelsets translatory motion weren't taken into account. In such a sense from the dynamical point of view our model is more complete.

If we introduce a small parameter playing a role of the scaling multiplier for the inertia moments and masses for the motion types neglected in Ref. [10] then if its value is small enough, 10^{-7} for our simulations, the motions compared become practically indistinguishable.

A set of different laws of the snakeboard control performed by the robot-snakeborder generating the masscenter trajectories like astroid, cycloid, eight, 3-rose, 4-rose is presented in Ref. [12].

Considering balance of energy in the total model one can remark here that servodrives applied between the coupler on one side and the flywheel and wheelsets on the other one are implemented correspondingly in the objects *CJoint*, *LeftJoint*, *RightJoint* of the class *FixedServoJoint*. Such kinematic constraints are known in the bond graph theory to be able to inject into the system any amount of energy needed to hold the desired motion. On the other hand energy loses due to the resistance elements encapsulated in the objects *Spring*, *LeftWheels.Joint1*, *LeftWheels.Joint2*, *RightWheels.Joint1*, *RightWheels.Joint2* of the class *SpringIdealJoint*. Remind that all motions supposed here as a relative ones of the body B w. r. t. body A in each of the constraint objects considered.

4 CONCLUSIONS

- The process of the models development and debugging becomes fairly easy and simple if one uses physically-oriented approach for the MBS dynamics simulation. Besides it is useful if we want to investigate dynamic properties of the system without a tedious work to construct and analyze the full nonlinear system of ODEs/DAEs, especially if we want to make it in a fast manner.
- An acausal modeling accelerates the model development releasing an engineer from the problem of causality assignment if s/he takes into account some requirements like complementarity rules.
- An object-oriented representation makes it possible to develop the constraints models adopted to the specific types of the bodies interconnections in a fast and efficient manner.
- Introducing the compliance into the model may be useful and efficient preserving the principal properties of the MBS like anholonomity etc.

5 ACKNOWLEDGEMENTS

This work has been prepared at MAI and FRC CSC.

REFERENCES

[1] Kosenko, I. I., Stepanov, S. Y. and Gerasimov, K. V. Contact tracking algorithms in case of the omni–directional wheel rolling on the horizontal surface. *Multibody System Dynamics* (2019) **45**:273–292.

[2] *Modelica — a unified object-oriented language for physical systems modeling. Tutorial.* Modelica Association, 2000.

[3] *Modelica — a unified object-oriented language for physical systems modeling. Language specification.* Modelica Association, 2004.

[4] D. C. Karnopp, D. L. Margolis, R. C. Rosenberg. *System dynamics: modeling and simulation of mechatronic systems. — 4th ed.* John Wiley & Sons, Hoboken, New Jersey, 2006.

[5] D. C. Karnopp. The energetic structure of multibody dynamic systems. *Journal of the Franklin Institute*, **306**, 165–181, 1978.

[6] P. Fritzson. *Principles of object–oriented modeling and simulation with Modelica 2.1.* IEEE Press, Piscataway, New Jersey, 2004.

[7] I. Kosenko. Implementation of unilateral constraint model for multibody systems dynamics on Modelica language. Proceedings of ACMD2006, The Third Asian Conference on Multibody Dynamics 2006, Institute of Industrial Science, The University of Tokyo, Tokyo, Japan, August 1–4, 2006.

[8] J. Wittenburg. *Dynamics of Multibody Systems, second edition.* Springer-Verlag, 2008.

[9] I. I. Kossenko. Implementation of unilateral multibody dynamics on Modelica. Proceedings of the 4th International Modelica Conference, Hamburg University of Technology, Hamburg–Harburg, Germany, March 7–8, 2005.

[10] A. D. Lewis, J. P. Ostrowski, R. M. Murray, and J. W. Burdick. Nonholonomic mechanics and locomotion: the snakeboard example. Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, May 1994.

[11] A. V. Borisov, I. S. Mamaev, A. A. Kilin. Dynamics of rolling disk. *Regular and Chaotic Dynamics*, **8**, 201–212, 2003.

[12] Yu. F. Golubev. *Motion design for a robot-snakeborder.* Keldysh Institute of Applied Mathematics, the Russian Academy of Science, Preprint No. 65, Moscow, 2004.