Multibody Systems Dynamics: Modelica Implementation and Bond Graph Representation

Ivan I. Kosenko[1], Maria S. Loginova[2], Yaroslav P. Obraztsov[2], Mayya S. Stavrovskaya[1]
[1]Moscow State University of Service, Department of Engineering Mechanics
Glavnaya str. 99, Cherkizovo-1, Moscow reg., 141221, Russia
[2]Moscow State Academy of Instrument Making and Computer Science
Department of Applied Mathematics, Stromynka str. 20, Moscow, 107646, Russia

Abstract

Using an example of the snakeboard, a vehicle with four wheels and nonholonomic constraints, the process of construction and verification for the sparse dynamical models of the multibody systems is analyzed. Two approaches for the formal representation of the models: object-oriented, and bond graph based are considered. Energy based similarities between these approaches are analyzed.

A detailed description of the bond graph representation for the most general type of constraint is presented. It turned out the resulting total bond graph model of the multibody system dynamics always has exactly a canonical junction structure. This representation has a tight correspondence with our recent object-oriented implementation of the mechanical constraint architecture. As an example Modelica implementation of the joint classes family is investigated. Finally these classes are applied to construct the snakeboard dynamic model.

Keywords: vechicle; nonholonomic; disc; wheelset; snakeboard; object-oriented modeling; bondgraph; canonical junction structure; joint; servoconstraint

1 Introduction

When developing a computer model of the multibody system (MBS) dynamics 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. One of them is connected tightly with the so-called multiport representation of the models initially based on the bond graph use. These latter in turn based on the idea of energy interaction, 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. 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.

2 Constraint representation via bond graphs

Previously, when considering a unified model of the constraint, or, in a more general way, any physical interaction between two rigid/deformable bodies we defined [1, 2] two classes of the kinematic and the effort ports. These ones are the kinematic and wrench connectors. It turned out the connections of such types make it possible to construct a model of the bodies interactions based on the causality physically motivated. Namely, the constraint object imports the kinematic information accepting it from the objects of interacting bodies and reciprocally exports it in the opposite direction. Thus the constraint "computes" an efforts the bodies interact by.

On the other hand geometric formalisms to represent the MBS dynamics are known [3] which operates with the similar information objects: twists and wrenches. In our approach twist is defined by the `KinematicPort` class, and wrench obviously corresponds to our `WrenchPort` class. The representation under consideration is tightly connected with the power based approach to modeling, so-called bond graphs [4].

Indeed, let the rigid body kinematics be defined by the

twist $(\mathbf{v}, \boldsymbol{\omega})$, where $\mathbf{v}$ is the mass center velocity, and $\boldsymbol{\omega}$ is the body angular velocity. Further let all the forces acting upon the body be reduced to the wrench $(\mathbf{F}, \mathbf{M})$ with the total force $\mathbf{F}$ and the total torque $\mathbf{M}$. Thus the total power of all the forces acting on the body is computed by the known formula

$$W = (\mathbf{v}, \mathbf{F}) + (\boldsymbol{\omega}, \mathbf{M})$$

using to represent a multibond in the bond graphs simulating the MBS dynamics. We have in such the case an evident canonical duality between twists and wrenches.

Sometimes wrenches are selected as flow variables. In other cases twists play this role. For instance similarities between electricity and mechanics cause the parallelism for electric current and forces/torques in one dimensional powertrains of mechanisms. In this case we can set a correspondence between the Kirchhoff law for currents and the d'Alembert principle for external forces and forces of inertia "acting" upon the body.

In our opinion it may be interesting enough to apply an approach dual to the first one mentioned above. Such an approach is more natural in traditional classical mechanics and assumes twist for the flow variable in the multibond. In the further course we present an illustration for this approach and demonstrate its convenience to construct the mechanical constraints of different types. Moreover, object-oriented implementation may be interpreted in both above dual approaches in a symmetric ways.

Let us trace now the similarities between the bond graphs and our MBS models. Evidently the pair of classes `KinematicPort/WrenchPort` plays a role of the multiport notion, and corresponding pairs of connections in Figure 1 stand for the notion of a bond.

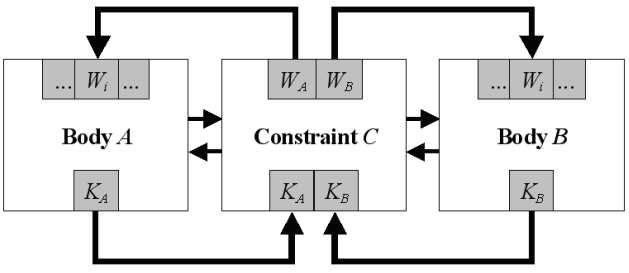

Figure 1: Architecture of Constraint

Furthermore, in this way we can associate an object of the `RigidBody` class with 1-junction, while 0-junction is associated with the object of the class `Constraint`. The relevant general bond graph representation of the constraint in any MBS may be depicted as it shown in Figure 2.

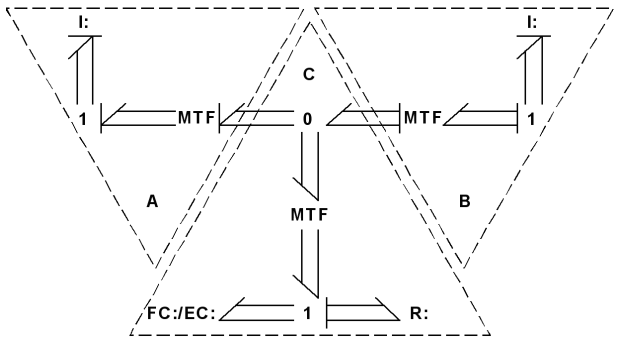

Figure 2: Architecture of Constraint: Bond Graph Representation

All multibonds here consist of the twist $(\mathbf{v}, \boldsymbol{\omega})$ signals representing the flow component, and the wrench $(\mathbf{F}, \mathbf{M})$ signals as an effort. Causality of an inertance elements arranges according to the Newton–Euler system of ODEs. Left and right transformers are to shift the twist from the mass center to the contact point according to the known Euler formula: $(\mathbf{v}, \boldsymbol{\omega}) \mapsto (\mathbf{v} + [\boldsymbol{\omega}, \mathbf{r}], \boldsymbol{\omega})$, where the vector $\mathbf{r}$ begins at the corresponding center of mass and ends at the contact point. Reciprocally the wrenches shift to the body mass center from point of the contact in a following way: $(\mathbf{F}, \mathbf{M}) \mapsto (\mathbf{F}, \mathbf{M} + [\mathbf{r}, \mathbf{F}])$. As one can see easily the transformers conserve the power.

Central transformer is responsible for the transfer to orthonormal base at the contact point with the common normal unit vector and two others being tangent ones to both contacting bodies' surfaces supposed regular enough. For definity we interpret here the case of usual contact interconnection between the bodies by their outer/inner surfaces. If the inertial coordinates of these vectors compose columns of the orthogonal rotational matrix Q then shifting from bottom to top across the transformer in Figure 2 we will have for the flow signals: $(\mathbf{v}, \boldsymbol{\omega}) \mapsto (Q\mathbf{v}, Q\boldsymbol{\omega})$. Likewise when shifting in a reverse direction we have a transformation of the efforts: $(\mathbf{F}, \mathbf{M}) \mapsto (Q^{-1}\mathbf{F}, Q^{-1}\mathbf{M})$ also conserving the power. Organization of the 0-junction depicted in Figure 2 provides a possibility to compute exactly the relative velocities at the constraint contact point.

Note that it is a usual practice to attach the inertance element to 1-junction, in particular because of its causality nature, see for example [5, 6]. Figure 2 in some degree can remind us an element of the lumped model for the flexible beam dynamics.

Causality for some multibonds inside the constraint object is defined individually for each particular scalar

bond [7] depending on the type of the constraint and is assigned finally after the whole MBS model compilation. For instance, if the constraint is of the slipping type at a contact then supposing decompositions of the relative velocities and contact forces $\mathbf{v} = \mathbf{v}_n + \mathbf{v}_\tau$, $\mathbf{F} = \mathbf{F}_n + \mathbf{F}_\tau$ we have the following flow constraint, element FC, $\mathbf{v}_n = \mathbf{0}$ representing one scalar kinematic equation for the normal relative velocity, and the effort constraint, element EC, $\mathbf{F}_\tau = \mathbf{0}$, $\mathbf{M} = \mathbf{0}$ representing two scalar equations for the tangent contact force plus three scalar equations for the contact torque. Nonzero tangent force at the contact may arise due to the resistive element, see the bottom right multibond. If we will continue to build the bond graph model for the whole MBS in a proposed way then finally we can arrive exactly to the so-called canonical junction structure [7] useful for the formal procedures of the bond graph optimal causality assignment. For this we have to add an intermediate 0-junctions for elements attached to 1-junction in the constraint component *C*, see Figure 2.

Leaving some multibonds without the causality assignment and trusting this work to compiler we apply a so-called acausal modeling [8]. On the other hand if we will act in a manner close to the real cases of constraints with the flexibility then instead of the constraint elements FC/EC, we have to use an element of the compliance with the causality uniquely determined, see Figure 3.

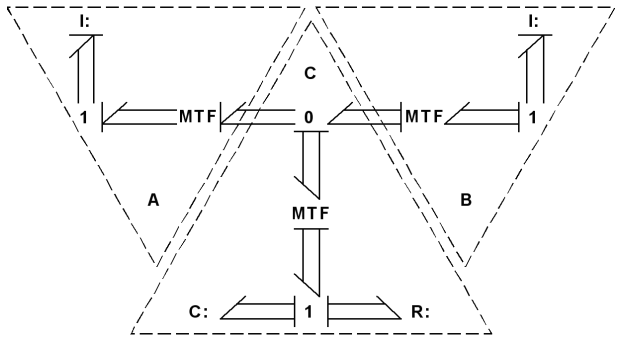

Figure 3: Bond Graph of Constraint with Compliance

Further we analyze one example of the constraint frequently occurring in engineering applications: we consider an object classification of the joint constraint.

3 Implementation of the joint constraint

For simplicity and clearness we will apply the component library to simulate the dynamics of MBSs with bilateral constraints [1]. Application of the components for the unilateral constraints [2] doesn't change anything in principle. The only difference is that dynamics of the 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 implemented exactly, without any slip or separation with respect to (w. r. t.) the surface.

Remind that according to our technology of the constraint construction [1] 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.

Class `Joint` plays a key role in the future model of a vehicle we will build. `Joint` is a model derived from the base class `Constraint`. Remind [2] 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. to 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 equations always present in the base model `Constraint` due to Newton's third law. For definity 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 can 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 complex one consisting of the body *A* convective motion w. r. t. an inertial frame of reference which is similar to the Modelica Standard MultiBody Library model `World`, and a relative motion w. r. t. the body *A*. An absolute motion is one of the body *B* w. r. t. 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 [2] $\mathbf{r}_{O_\alpha}$ 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

$$\begin{aligned} \mathbf{v}_{Ba} &= \mathbf{v}_{Be} + \mathbf{v}_{Br}, \\ \mathbf{v}_{Ba} &= \mathbf{v}_{O_B} + [\boldsymbol{\omega}_B, T_B \mathbf{r}_B], \\ \mathbf{v}_{Be} &= \mathbf{v}_{O_A} + [\boldsymbol{\omega}_A, \mathbf{R}_B - \mathbf{r}_{O_A}], \end{aligned} \tag{1}$$

where $\mathbf{v}_{Ba}$, $\mathbf{v}_{Be}$, $\mathbf{v}_{Br}$ are an absolute, convective, 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 precompiler 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 in time of translation or when running the model.
In the first case usually diagnostics of the compiler essentially helps the developer. In the second case the model has an unpredictable behavior, and only manual preliminary reduction "regularizes" the simulation process. Thus we differentiate equations (1) and obtain an equations for the relative linear acceleration in the form

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

where $\mathbf{a}_{Ba}$, $\mathbf{a}_{Be}$, $\mathbf{a}_{Br}$ are an absolute, convective, and relative accelerations of the body B marked point, $\boldsymbol{\varepsilon}_A$, $\boldsymbol{\varepsilon}_B$ are the bodies angular accelerations.
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 a form

$$\begin{aligned} \boldsymbol{\omega}_B &= \boldsymbol{\omega}_A + \boldsymbol{\omega}_r, \\ \boldsymbol{\varepsilon}_B &= \boldsymbol{\varepsilon}_A + [\boldsymbol{\omega}_A, \boldsymbol{\omega}_r] + \boldsymbol{\varepsilon}_r, \\ \boldsymbol{\varepsilon}_r &= \lambda \mathbf{n}_{Ai}, \end{aligned} \tag{3}$$

where $\boldsymbol{\omega}_r$, $\boldsymbol{\varepsilon}_r$ are the relative angular velocities and accelerations.
The Modelica code of the class `Joint` reads

```
partial model Joint
  extends Constraint;
  parameter Real[3] nA;
  parameter SI.Position[3] rA;
  parameter SI.Position[3] rB;
  SI.Position[3] RA;
  SI.Position[3] RB;
  SI.Velocity[3] vBa;
  SI.Velocity[3] vBe;
  SI.Velocity[3] vBr;
  SI.Acceleration[3] aBa;
  SI.Acceleration[3] aBe;
  SI.Acceleration[3] aBr;
  SI.AngularVelocity[3] omegar;
  SI.AngularAcceleration[3] epsilonr;
  Real[3] nAi;
  SI.Force F; // Force along axis
  SI.Torque M; // Torque about axis
  SI.Acceleration mu;
  SI.AngularAcceleration lambda;
equation
  RA = InPortA.r + InPortA.T*rA;
  RB = InPortB.r + InPortB.T*rB;
  nAi = InPortA.T*nA;
  vBa = InPortB.v +
    cross(InPortB.omega,
      InPortB.T*rB);
  vBe = InPortA.v +
    cross(InPortA.omega,
      RB - InPortA.r);
  vBa = vBe + vBr;
  aBa = InPortB.a +
    cross(InPortB.epsilon,
      InPortB.T*rB) +
    cross(InPortB.omega,
      cross(InPortB.omega,
        InPortB.T*rB));
  aBe = InPortA.a +
    cross(InPortA.epsilon,
      RB - InPortA.r) +
    cross(InPortA.omega,
      cross(InPortA.omega,
        RB - InPortA.r));
  aBa = aBe + aBr +
    2*cross(InPortA.omega, vBr);
  aBr = mu*nAi;
  omegar = InPortB.omega -
    InPortA.omega;
  epsilonr = InPortB.epsilon -
    InPortA.epsilon -
    cross(InPortA.omega, omegar);
  epsilonr = lambda*nAi;
  F = OutPortA.F*nAi;
  M = OutPortA.M*nAi;
  OutPortA.P = RA;
```

```
  OutPortB.P = RA;
end Joint;
```

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 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).

Regarding the general scheme depicted in Figure 2 we can conclude that the equations (1), (2), (3) together implement implicitly the constraint transformer to the joint local coordinate system and four scalar flow constraints forbidding relative translatory and rotary motions in the direction orthogonal to the joint axis. For derived classes only two free scalar bonds remain.

Here we encounter the known complementarity rules once more in a way similar to one described in [2]. In our context the variables in the pairs (μ, F), (λ, M) are mutually complement, where one of μ, λ is to be utilized for the flow constraint and one of F, M is used to compose the effort constraint. All the variables mentioned complete the set of constraints for the remaining yet unused joint axis creating thus two final scalar constraint elements in the bond graph of Figure 2.

Namely, the equations (2) implementing the Coriolis theorem for accelerations simultaneously implement, in an implicit manner, two scalar flow constraints, FC-elements, from the bottom left corner of the multibondgraph model in Figure 2. These flow constraints due to compiler restrictions constructed using accelerations instead of the velocities being used in a classic bond graph approach. The constraints have an obvious kinematic sense: they prevent the relative motion of the body B marked point in two directions normal to the joint axis fixed in the body A.

In addition, the equations (3) implement two other scalar flow constraints, this time for the rotary motion. These constraints forbid the relative rotation of the body B w. r. t. body A about two axes each normal to the joint axis mentioned above which is rigidly connected with the body A.

Note, that the construct of equations (2) and (3) is such that they allow the body B relative motion along and about the joint axis of the body A thus implementing the kinematic pair with two DOFs. Returning to Figure 2 of the general constraint multi-bondgraph we can conclude that the vertical multibond attached to 0-junction implements flow variables corresponding to the relative body B motion w. r. t. body A in inertial coordinates. Such a description supposes an existence of the special coordinates reference frame connected with the body A at its joint constraint marked point. The transformation to these coordinates is implemented exactly via corresponding transformer, central in the triangle block C. The transformer itself nests in formulae of equations (2) and (3).

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` described by the equations

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

with an initial data $\nu(t_0) = 0$, $\dot{\nu}(t_0) = 0$ for the relative translatory position ν provides a viscoelastic 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

$$\mu = 0, \quad M = f(t, \varphi, \dot{\varphi}), \quad \ddot{\varphi} = \lambda \tag{4}$$

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. From the bond graph viewpoint the second equation in (4) can be implemented as a combination of the source effort, compliance, and resistance elements. 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 (4). It is clear one can create a lot of other different combinations of equations to construct the joint constraints needed in engineering applications.
The derived joint classes described here are to close the system of kinematic equations (2) and (3) completing them mainly by two scalar additional equations, each playing a role of an either FC-element, like $\mu = 0$, or EC-element, like $F = 0$. Any time to be able to construct a consistent system of equations for the total model we have to follow the guidelines of the complementarity rules.
These latter correspond to the notions of the bond graph theory in a natural way. Indeed, the theory of bond graphs is based on the energy interactions. Every our multibond being an energy/power conductor reflects complementarity by its twist/wrench duality. To close the total DAE system for the model under development we have to "close" or rather to "seal" each free scalar bond in EC/FC-element of the block C in Figure 2 by the corresponding one scalar equation for flow or effort variable. Thus here we outline the main rule to compose equations for the models of constraints for MBS of any type in a consistent way when applying the object-oriented approach. In the further course we present an example for the systematic application of the rules mentioned.

4 Example of the snakeboard

The snakeboard [9], see Figure 4, 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

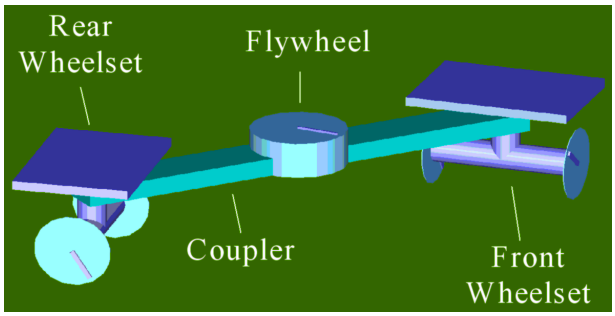

Figure 4: The Snakeboard

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.

4.1 Dynamics of the rolling disc

This problem is a classic one of dynamics [10] and has a visual representation depicted in Figure 5

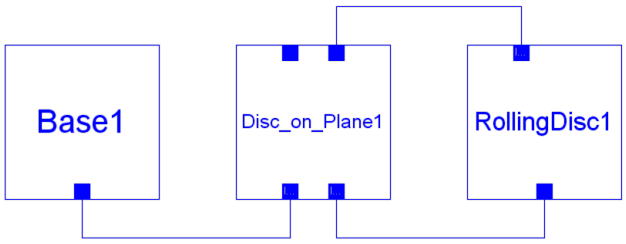

Figure 5: Visual Model of the Rolling Disc

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

$$z_B = 0 \tag{5}$$

of the Body B coordinate system $O_B x_B y_B z_B$ and has the fixed radius R, see Figure 6. 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

$$(\mathbf{r}_P, \mathbf{n}_A) = 0. \tag{6}$$

Further denoting the Body B current orientation matrix by T_B and by $\mathbf{r}_{O_B}$ its center of mass position vector we

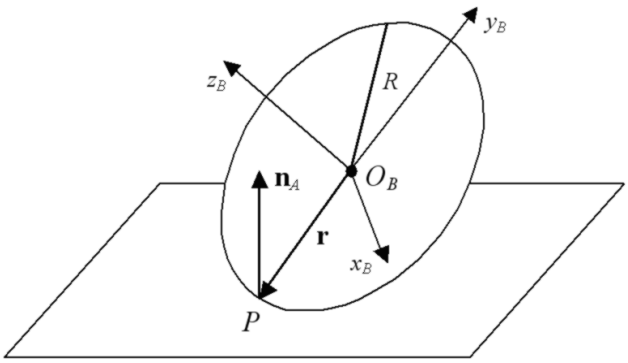

Figure 6: Rolling Disc

obtain the system of three equations

$$T_B\mathbf{r} = \mathbf{r}_P - \mathbf{r}_{O_B} \tag{7}$$

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 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 orthogonal mutually 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

$$(\mathbf{n}_A, T_B\boldsymbol{\tau}) = 0. \tag{8}$$

The system of six equations (5), (6), (7), (8) 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`[1]. 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 [10]

$$\begin{aligned} \dot{\mathbf{M}} &= [\mathbf{M}, \boldsymbol{\omega}] + m\,[\dot{\mathbf{r}}, [\boldsymbol{\omega}, \mathbf{r}]] + mg[\mathbf{r}, \boldsymbol{\gamma}], \\ \dot{\boldsymbol{\gamma}} &= [\boldsymbol{\gamma}, \boldsymbol{\omega}] \end{aligned}$$

expressed w. r. t. the Body B rotating system. Here $\mathbf{M} = I\boldsymbol{\omega} + m[\dot{\mathbf{r}}, [\boldsymbol{\omega}, \mathbf{r}]]$ is the vector of the disc angular momentum computed w. r. t. the contact point, $I = \mathrm{diag}\,(I_{xx}, I_{yy}, I_{zz})$ is the central principal inertia tensor of the disc, $\boldsymbol{\omega}$ 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 such that satisfy the relations

$$x_B = -\frac{R\gamma_x}{\sqrt{1-\gamma_z^2}}, \quad y_B = -\frac{R\gamma_y}{\sqrt{1-\gamma_z^2}}, \quad z_B = 0.$$

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 $\boldsymbol{\omega}$, $\mathbf{M}$, $\boldsymbol{\gamma}$ components are of the order 10^{-7} over the time interval of the several hundreds units.

4.2 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 depicted in Figure 7, where the Rotate and Flip commands were applied to symmetrize the diagram. Application of the model `FixedIdealJoint` for the 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`.

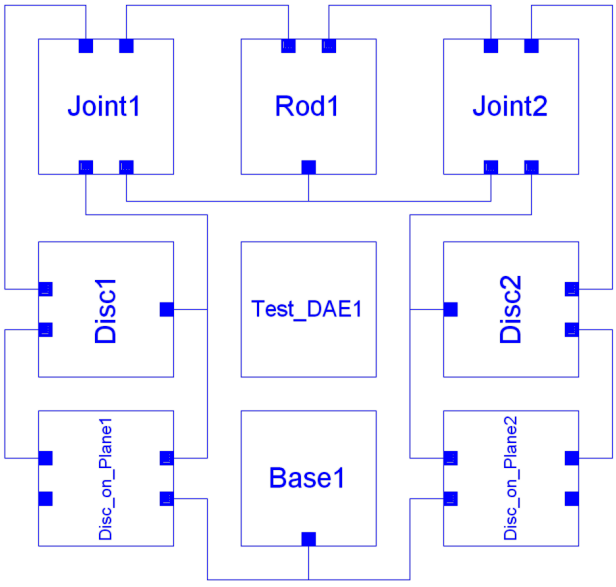

Figure 7: Visual Model of the Isolated Wheelset

To verify the wheelset model built the following system of DAEs was applied

$$\begin{gathered} ma_x = X_1 + X_2 + X_{\text{ext}}, \quad ma_y = Y_1 + Y_2 + Y_{\text{ext}}, \\ ma_z = Z + Z_{\text{ext}}, \end{gathered} \tag{9}$$

$$a_x = -\frac{R}{2}\left(\ddot{\varphi}_1 + \ddot{\varphi}_2\right), \; a_y = 0, \; a_z = \frac{R^2}{2L}\left(\dot{\varphi}_1^2 - \dot{\varphi}_2^2\right), \tag{10}$$

$$I_{dz}\ddot{\varphi}_1 = RX_1, \quad I_{dz}\ddot{\varphi}_2 = RX_2, \quad I_{rz}\ddot{\varphi}_r = 0, \qquad (11)$$

$$\dot{\varphi}\left[I_{dz}\left(\dot{\varphi}_1+\dot{\varphi}_2\right)+I_{rz}\dot{\varphi}_r\right] = \frac{L}{2}\left(Y_2-Y_1\right)-RZ+M_{\text{ext}x},$$
$$I_{rz}\ddot{\varphi} = \frac{L}{2}\left(X_1-X_2\right)+M_{\text{ext}y}, \qquad (12)$$

$$L\dot{\varphi} = R\left(\dot{\varphi}_1-\dot{\varphi}_2\right) \qquad (13)$$

which is written w. r. t. moving coordinate system connected with the wheelset according to Figure 8 in an evident way. This system of coordinates performs a convective motion tracing the motion of the rod which plays a role of the wheelset axis shaft.

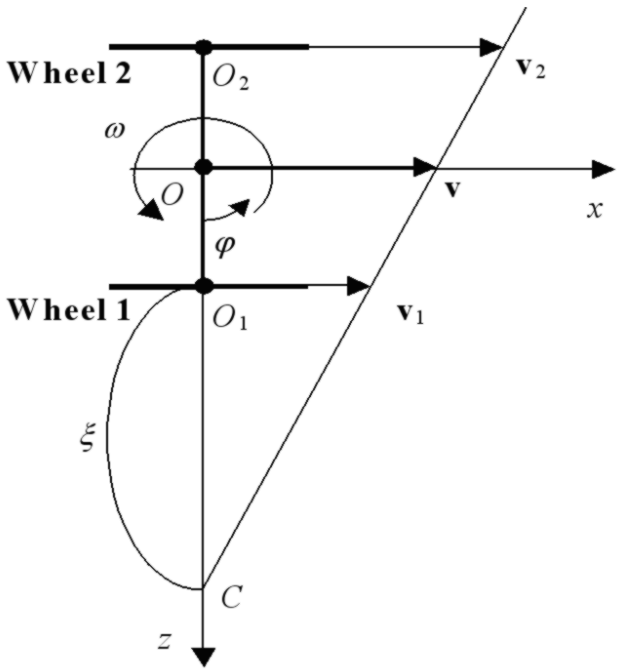

Figure 8: Top View of the Wheelset

The DAE system consists of twelve equations w. r. t. twelve unknowns: φ_1, φ_2, φ, φ_r, a_x, a_y, a_z, X_1, X_2, Y_1, Y_2, Z. Let us give a more detailed explanations to these DAEs. The subsystem of equations (9) represents the theorem for the center of mass motion of the wheelset. Here $m = 2m_d + m_r$ is the total mass of the wheelset, m_d, m_r are the masses of the wheel simulated by the disc and the rod of the wheelset axis, R is the wheels radius, L is the rod length. The variables a_x, a_z, a_y are correspondingly the tangent, normal, and binormal components of the masscenter acceleration.

The variables X_1, X_2, Y_1, Y_2 are the projections to the x, y axes of the contact forces acting to the wheels from the surface. The value $Z = Z_1 + Z_2$ is used because z-projections of the contact forces can't be computed individually for the reason of degeneration of the problem along the z-axis. This discussed above problem is resolved due to the compliance introduced for the joints.

In the kinematic equations (10) all signs are adjusted such that φ_1, φ_2 are the angles of the wheels relative rotation, $\dot{\varphi}_1$, $\dot{\varphi}_2$, $\ddot{\varphi}_1$, $\ddot{\varphi}_2$ are their relative angular velocities and angular accelerations. Point C is a center of velocities for the rigid planar convective motion of the Ozx coordinate system.

The equations (11) represent z-projections of the Euler dynamic equations for the discs and the rod considered separately. We conclude from the third equation that the rod relative angular velocity is the integral of the motion: $\dot{\varphi}_r = \text{const}$. Remind we consider rotations in the joints as an ideal, without friction, ones.

First of the equations (12) is the dynamical one for the angular momentum of the whole wheelset w. r. t. the axis Ox. The second equation is the projection of the same vector equation to the axis Oy. Further the parameters I_{dz}, I_{rz} are the moments of inertia for the wheel and rod w. r. t. the axis Oz, $I_y = 2I_{dy} + I_{ry}$ where I_{dy}, I_{ry} are the moments of inertia for the disc and shaft w. r. t. the axis Oy. The angle φ is one of the convective rotation about the Oy axis. The kinematic equation (13) is derived from a simple geometric considerations, see Figure 8.

We can add an external force $\mathbf{F}_{\text{ext}} = \{X_{\text{ext}}, Y_{\text{ext}}, Z_{\text{ext}}\}$ and rotating torque $\mathbf{M}_{\text{ext}} = \{M_{\text{ext}x}, M_{\text{ext}y}, M_{\text{ext}z}\}$ to the right hand sides of equations (9), (11), (12). Regarding the equations (11) one can distribute the torque $M_{\text{ext}z}$ between all three bodies of the wheelset in an any desirable way.

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 $c = 1000$, $d = 5000$.

4.3 Model of the vehicle

Let us construct at last a complete model of the snakeboard. Its visual representation see in Figure 9, where rotation and flipping were applied to the graphic images of the objects as it has been done for the wheelset visual model. 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.

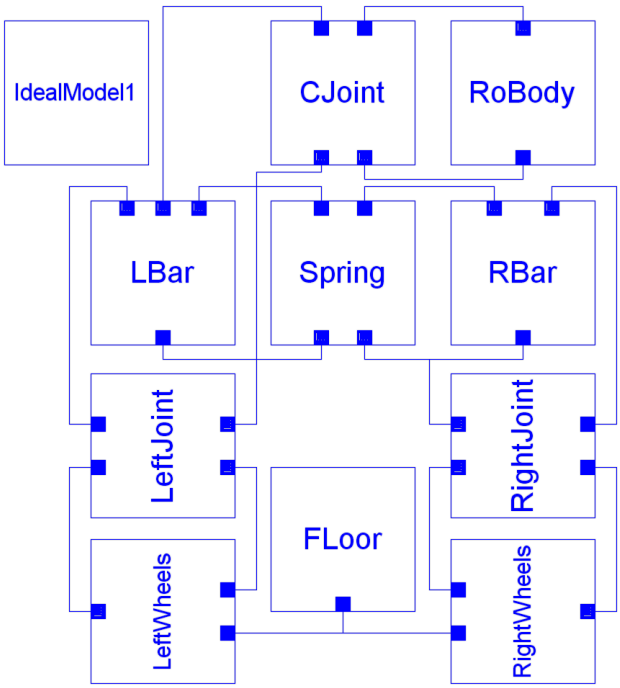

Figure 9: Visual Model of the Snakeboard

To perform a comparison with the known results [9] 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 definity, the left part of the coupler. The joints mentioned correspond to the objects `LeftJoint`, `RightJoint`, and `CJoint` in Figure 9. All three servoconstraints can be described by the equations

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

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 Figure 9, 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 [9] 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 simulations results showed a full coincidence of the gait types for our regularized model and the idealized model of the paper [9]. All types of the behavior are demonstrated in Figures 10, 11, 12, where the flywheel masscenter projections to the xz plane are presented. In [9] 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.

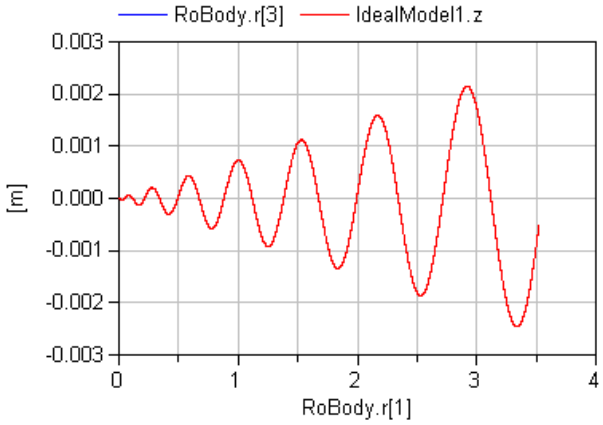

Figure 10: Masscenter Trajectory for "Drive" Gait

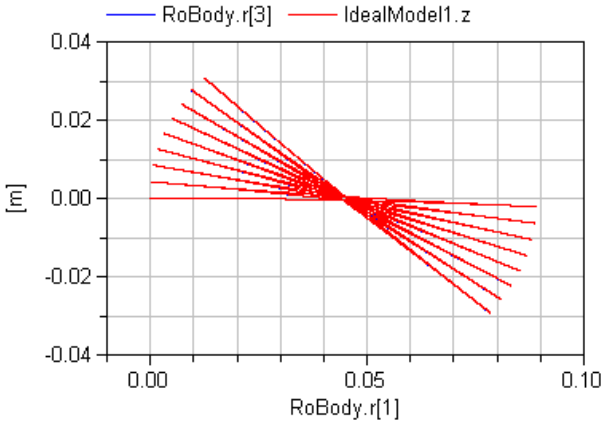

Figure 11: Masscenter Trajectory for "Rotate" Gait

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 [9] then if its value is small enough, 10^{-7} for our simulations, the motions compared become practically indistinguishable,

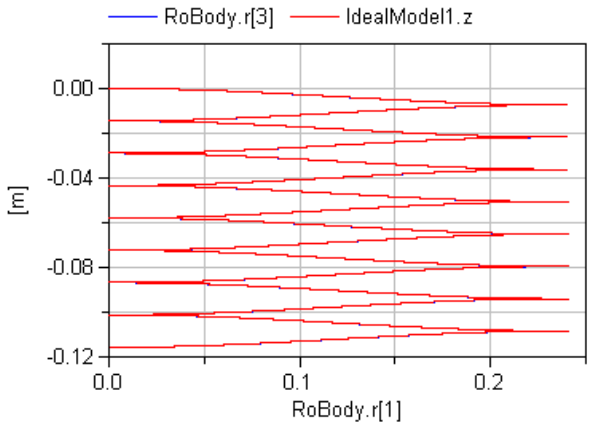

Figure 12: Masscenter Trajectory for “Parking” Gait

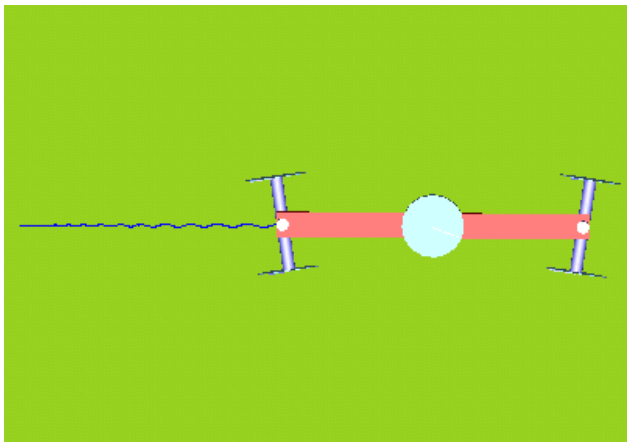

Figure 14: Animation of “Drive” Gait

see for instance the plot of the snakeboard masscenter x-components difference for the models under comparison, Figure 13, in the case of the “Rotate” gait. Animation shot in the case of the ”drive” type gait see in Figure 14.

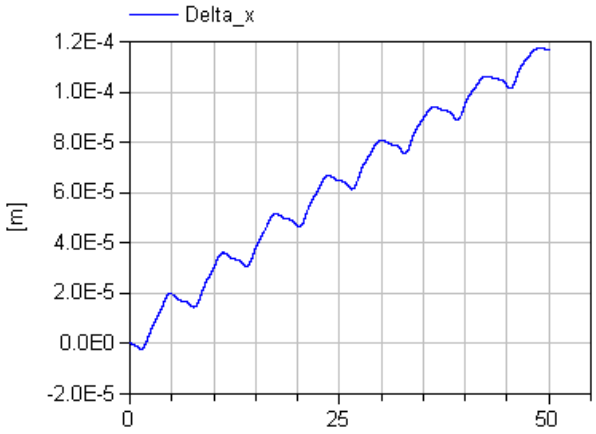

Figure 13: Closeness of Models

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 [11]. The port-controlled Hamiltonian representation of the simplified ideal snakeboard model from [9] with its bond graph implementation is investigated in [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`.
Thus the class `FixedServoJoint` implements two scalar FC-elements from the general bond graph depicted in Figure 2 for the rotary and translatory motions, while the class `SpringIdealJoint` implements one C-element, ideal elastic compliance, in combination with R-element, resistance due to viscosity, for the translatory motion plus one EC-element for the rotary motion. 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.

5 Conclusion

We can make now our brief list of conclusions in the following form:

- A unified multi-bondgraph representation of the MBS dynamics in a sufficiently simple way with the canonical junction structure is possible.

- The representation depicted in Figure 2 can be used as a guideline to construct the consistent system of DAEs in a systematic way. In other words we can say that multi-bondgraph constructs like ones of Figure 2 are to be used as a regular basis for more informal object-oriented approach.

- 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 effective manner implementing the corresponding bond graph formalisms in a more natural and informal way mainly by chains of inheritance for the behavior (equations) and properties thus gradually filling the complete multibondgraph description.
- An acausal modeling accelerates the modeling releasing a developer from the problem of causality assignment if s/he takes into account some requirements like complementarity rules.
- Introducing the compliance into the model may be useful and effective preserving the principal properties of the MBS like anholonomity etc.

Enumerate also some possible directions of the further work:

- Development of the vehicle models more complicated then considered here.
- Development of the more complicated contact models taking into account friction and a unilateral nature of the constraints.
- Account of the road uneven surfaces of different types.

6 Acknowledgement

The paper was prepared with partial support of Russian Foundation for Basic Research, projects 05–08–65470, 05–01–00454, SS–6667.2006.1.

References

[1] Kossenko, I. I., and Stavrovskaia, M. S., How One Can Simulate Dynamics of Rolling Bodies Via Dymola: Approach to Model Multibody System Dynamics Using Modelica // Proceedings of the 3rd International Modelica Conference, Linköpings universitet, Linköping, Sweden, November 3–4, 2003, pp. 299–309.

[2] Kossenko, I. I., 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, pp. 13–23.

[3] Stramigioli, S., Blankenstein, G., Duindam, V., Bruyninckx, H., Melchiorri C., Power Port Concepts in Robotics. The Geometrical–Physical Approach. Tutorial at 2003 IEEE International Conference on Robotics and Automation. — IEEE, 2003.

[4] Paynter, H. M., Analysis and Design of Engineering Systems. — The M. I. T. Press: Cambridge, Massachusetts, 1961.

[5] Cellier, F. E., Continuous System Modeling. — Springer–Verlag: New York, 1991.

[6] Mukherjee, A., Karmakar, R., Modelling and Simulation of Engineering Systems through Bondgraphs. — Alpha Science International Ltd.: 2000.

[7] Golo, G., Interconnection Structures in Port-Based Modelling: Tools for Analysis and Simulation. PhD Thesis. — University of Twente: Enschede, The Netherlands, 2002.

[8] Dymola. Dynamic Modeling Laboratory. User's Manual. Version 5.3a. — Lund: Dynasim AB, Research Park Ideon, 2004.

[9] Lewis, A. D., Ostrowski, J. P., Murray, R. M., and Burdick, J. W. Nonholonomic Mechanics and Locomotion: The Snakeboard Example // Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, May 1994, IEEE, pp. 2391–2400.

[10] Borisov, A. V., Mamaev, I. S., Kilin, A. A., Dynamics of Rolling Disk // Regular and Chaotic Dynamics, 2003, Vol. 8, No. 2, pp. 201–212.

[11] Golubev, Yu. F., Motion Design for a Robot-Snakeborder // Keldysh Institute of Applied Mathematics, the Russian Academy of Science, Preprint No. 65, 2004.

[12] Duindam, V., Blankenstein, G., Stramigioli S., Port-Based Modeling and Analysis of Snakeboard Locomotion // Sixteenth International Symposium on Mathematical Theory of Networks and Systems, Katholieke Universiteit Leuven, Belgium, July 5–9 2004.