VI International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2015
B. Schrefler, E. Oñate and M. Papadrakakis(Eds)

CONTACT TYPES HIERARCHY AND ITS OBJECT-ORIENTED IMPLEMENTATION

IVAN I. KOSENKO*, KIRILL V. GERASIMOV† AND MIKHAIL E. STAVROVSKIY‡

*Moscow State Technical University of Radio Engineering, Electronics, and Automation
78 Vernadsky Avenue, 119454, Moscow, Russia
e-mail: kosenko@mirea.ru, web page: https://fcyb.mirea.ru/kafedri/internal/tm.html

†Faculty of Mechanics and Mathematics, Lomonosov Moscow State University
1 Leninskiye Gory, Main Building, GSP-1, 119991, Moscow, Russia
e-mail: kiriger@gmail.com

‡STAN Group
1 Vadkovsky lane, 127994, Moscow, Russia
e-mail: stavrov@list.ru, web page: http://www.stan-group.com/

Key words: Object-Oriented Modeling; Mechanical Constraint; Contact Model Template; Omni Wheel; Contact Tracking.

Abstract. Technology of the object-oriented implementation for the multibody dynamics models is the key feature when developing the corresponding computer structures. We are based on an approach originating from concepts explained earlier. Following the guidelines outlined there one can develop the family of the constraint abstractions being adapted to any type of the machinery applications and relatively easily implement corresponding family of Modelica models. One also can reorder these classes hierarchically using sequences of the behaviour inheritance. Solutions concerning contact problems and corresponding examples are under consideration.

1 INTRODUCTION

Development of a computer model for the multibody system (MBS) dynamics is connected usually with use of a unified technology of any type for constructing models in an efficient way. Object-oriented languages [1] are known to resolve such a problem successively step by step using their natural features. On the other hand one of the natural way for representing the MBS dynamics is the so-called multiport representation of the models initially based on the bond graph application [2]. This latter approach is based on the idea of energy interchange, and substantially on energy conservation for physically interconnected subsystems of any engineering type. Consider in the sequel a technology for constructing a model of MBS dynamics with constraints of any specific type in a unified way. Note that the unilateral constraints, like ones of mechanical contact may also be included in the further consideration process.

A lot of methods to describe the structure of the MBS using different graph approaches is known, see for instance [3]. 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 (w. r. t.) 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 $G = (\mathcal{B}, \mathcal{C}, \mathcal{I})$. Here $\mathcal{I} \subset \mathcal{C} \times \mathcal{B}$ 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 BASE CLASSES FOR THE MULTIBODY DYNAMICS MODELS

It is clear that consideration of the graph G provides only a simplest structural information insufficient for the MBS dynamical 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, reactions, 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 in a natural way can be 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 laws 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 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 reaction 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. Thus computer model of the particular constraint is represented by the communication network.

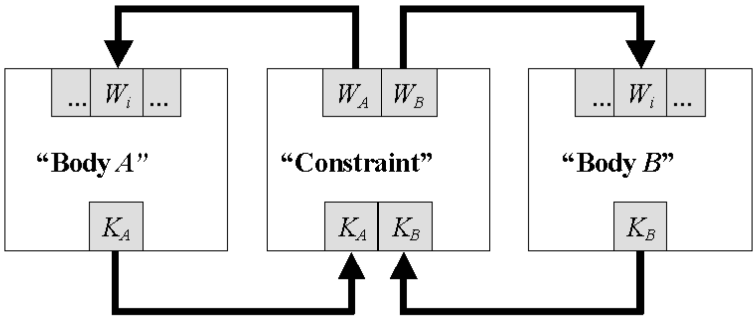

Figure 1: Architecture of constraint

KinematicPort is to be used to transfer the data of 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 $W_1, \ldots, W_s$ 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 apply now base classes from above for simulating the dynamics of MBSs with bilateral constraints [4]. Application of the components for the unilateral constraints [5, 6] we will see later.

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 need not 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 is implemented inside the *RigidBody* class by the known formula. 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 way described in [4].

Remind that according to our technology of the constraint construction [4] 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 $\mathbf{F}_A + \mathbf{F}_B = \mathbf{0}$, $\mathbf{M}_A + \mathbf{M}_B = \mathbf{0}$ 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 the last equations in different classes-inheritors corresponding to these particular types of constraints.

3 JOINT MODEL AS ONE OF A BILATERAL CONSTRAINT

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 [5] 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 (**??**) 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 under construction here let us represent the motion of the body B as a compound one including the body A convective 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 an axis of the joint in the body A; (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 always to keep geometric axes fixed in each of the bodies in coincidence.

First of all one has to compute the radii vectors of the points fixed in the bodies w. r. t. inertial system $\mathbf{R}_\alpha = \mathbf{r}_{O_\alpha} + T_\alpha \mathbf{r}_\alpha$ ($\alpha = A, B$), where [5] $\mathbf{r}_{O_\alpha}$ 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 $\mathbf{v}_{Ba} = \mathbf{v}_{Be} + \mathbf{v}_{Br}$, $\mathbf{v}_{Ba} = \mathbf{v}_{O_B} + [\boldsymbol{\omega}_B, T_B\mathbf{r}_B]$, $\mathbf{v}_{Be} = \mathbf{v}_{O_A} + [\boldsymbol{\omega}_A, \mathbf{R}_B - \mathbf{r}_{O_A}]$, 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.

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

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

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, $\mathbf{v}_{Br}$ 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 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 $\boldsymbol{\omega}_B = \boldsymbol{\omega}_A + \boldsymbol{\omega}_r$, $\boldsymbol{\varepsilon}_B = \boldsymbol{\varepsilon}_A + [\boldsymbol{\omega}_A, \boldsymbol{\omega}_r] + \boldsymbol{\varepsilon}_r$, $\boldsymbol{\varepsilon}_r = \lambda\mathbf{n}_{Ai}$, 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).

4 CONTACT AS A PARTICULAR CASE OF MECHANICAL INTERCONNECTION

For simplicity we suppose here objects of unilateral constraint implement model of mechanical contact without impacts. Though in general according to the nature of unilateral constraint one can describe it using fundamental state variable which at any time instant can have one of three values: “Flight”, “Sliding”, “Rolling”. Values enumerated have a sense transparent enough. State “Flight” means the constraint at a current time instant is in a disconnected condition i. e. bodies in fact are not connected and can perform free relative flying. As state variable has one of values “Sliding” or “Rolling” then bodies supposed to be in a contact. The difference is that the first state permits the relative slipping of the bodies but the second one does not.

The dynamics of a rigid body translatory–rotary motion was outlined above. However the mechanical constraint model representation undergoes an essential changes in compare with the bilateral case. We use the so called complementarity rules [7] as a base for the unified description of the unilateral constraint. Taking into account complementarity rules one can see easily that any constraint always is defined by the three scalar equations. To derive these equations first consider local geometry of the problem, see Figure 2.

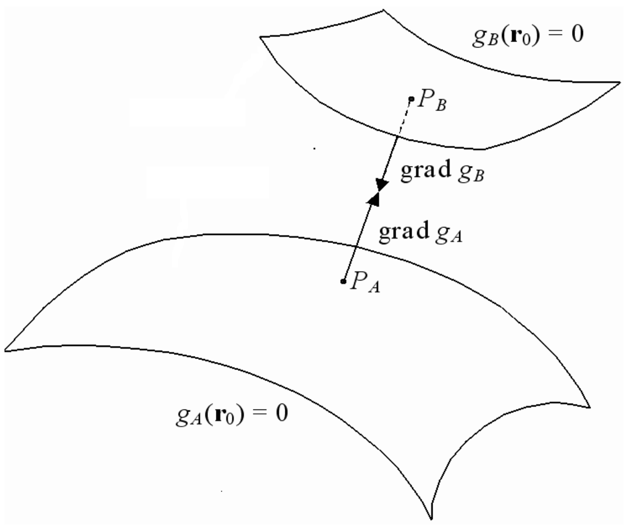

Figure 2: Area of Constraint

Outer surfaces supposed to be defined with respect to principal central axes of corresponding bodies by the equations $f_\alpha(\mathbf{r}_\alpha) = 0$ $(\alpha = A, B)$. Then in inertial frame of reference for the whole multibody system these equations will take the form $g_\alpha(\mathbf{r}_0) = f_\alpha\left[T^*_\alpha(\mathbf{r}_0 - \mathbf{r}_{O_\alpha})\right] = 0$ $(\alpha = A, B)$, with $\mathbf{r}_{O_A}$, $\mathbf{r}_{O_B}$ being a vectors of masscenters positions O_A, O_B for the bodies A and B, and T_A, T_B mean an orthogonal matrices for current bodies orientations. An asterisk denotes conjugating what equivalent to inverting of matrix for the case of orthogonality. Thus the functions $g_A(\mathbf{r}_0)$, $g_B(\mathbf{r}_0)$ depend upon the time indirectly through the variables $\mathbf{r}_A$, $\mathbf{r}_B$, T_A, T_B.

Constraint object of our model is to compute at each current instant positions of the points P_A and P_B which are the nearest ones for interacting bodies A and B. By virtue of above assumptions such points are to be evaluated in a unique way. Denote the radii vectors of these points with respect to inertial frame of reference by $\mathbf{r}_{P_A}$, $\mathbf{r}_{P_B}$. Then using simple geometric considerations for the coordinates of the cited vectors one can derive the following system of algebraic equations

$$\begin{array}{rclcrcl} \operatorname{grad} g_A(\mathbf{r}_{P_A}) & = & \lambda \cdot \operatorname{grad} g_B(\mathbf{r}_{P_B}), & & g_A(\mathbf{r}_{P_A}) & = & 0, \\ \mathbf{r}_{P_A} - \mathbf{r}_{P_B} & = & \mu \cdot \operatorname{grad} g_B(\mathbf{r}_{P_B}), & & g_B(\mathbf{r}_{P_B}) & = & 0. \end{array} \tag{1}$$

One can verify easily that the gradients can be computed by formulae $\operatorname{grad} g_\alpha(\mathbf{r}_{P_\alpha}) = T_\alpha \operatorname{grad} f_\alpha [T_\alpha^* (\mathbf{r}_{P_\alpha} - \mathbf{r}_{O_\alpha})]$, where $\alpha = A, B$. It easy to see the system (1) consists of eight scalar equations and has eight scalar unknowns: x_{P_A}, y_{P_A}, z_{P_A}, x_{P_B}, y_{P_B}, z_{P_B}, λ, μ. Variables λ, μ are an auxiliary ones. The equations (1) are in use either without or with a presence of the contact of bodies A, B. In a latter case the equation $\mu = 0$ instead of one of the surfaces equations is in use.

According to computational experience it is more reliable and convenient to use the equations of constraints in a differential form instead of those ones in the algebraic form (1). Such an approach is used frequently also when analyzing the properties of mechanical systems.

Normal vector $\mathbf{n}_A = \operatorname{grad} g_A / |\operatorname{grad} g_A|$ will play an important role in the further course. Normal for an outer surface of the body A is chosen here for definiteness. One can use the vector $\mathbf{n}_B$ as well.

Let us perform now a unified description of the unilateral constraint using kinematic and/or force equations. Denote by $\mathbf{F}_A$ the force acting on the body A from the body B. And by $\mathbf{F}_B$ denote the force acting on the body B from one of A vice versa. Each force mentioned acts at the point P_α, $\alpha = A, B$. In addition, let us introduce auxiliary notations $F_{An} = (\mathbf{F}_A, \mathbf{n}_A)$, $\mathbf{F}_{A\tau} = \mathbf{F}_A - F_{An}\mathbf{n}_A$, $\mathbf{v}_r = \mathbf{v}_{P_A} - \mathbf{v}_{P_B}$, $v_{rn} = (\mathbf{v}_r, \mathbf{n}_A)$, $\mathbf{v}_{r\tau} = \mathbf{v}_r - v_{rn}\mathbf{n}_A$.

If the bodies are disconnected and the constraint is in a state “Flight” then the force of reaction is equal to zero. Thus we have three scalar equations. For unifying the system of constraint equations further and for taking into account arbitrary directions of the normal $\mathbf{n}_A$ let us introduce auxiliary scalar variable κ in a way such that

$$F_{An} = 0, \quad \mathbf{F}_{A\tau} - \kappa \mathbf{n}_A = \mathbf{0}.$$

Actually one has obtained the system of four equation with four unknown variables F_{Ax}, F_{Ay}, F_{Az}, κ.

In case of bodies contact the condition $F_{An} = 0$ is substituted by the kinematic one $v_{An} = 0$. States “Sliding” and “Rolling” differ one from another using conditions in a tangent plane. Implementation of the Coulomb friction model supposed for the simplicity. Then one can obtain the vector force equation in the tangent plane

$$\mathbf{F}_{A\tau} - d \cdot F_{An}\mathbf{v}_{r\tau} / |\mathbf{v}_{r\tau}| - \kappa \mathbf{n}_A = \mathbf{0}, \tag{2}$$

where d is the coeffitient of friction.

For rolling the tangent velocity has to be zero

$$\mathbf{v}_{A\tau} - \kappa \mathbf{n}_A = \mathbf{0}.$$

In the case of sliding the model equation (2) "works" properly if the relative velocity isn't very small. However the problem of regularization for the equation of constraint (2) arises at the instance of transition from "Rolling" to "Sliding". It turns out that one can apply here the known approximation for Coulomb's friction using regularized expression for the tangent force

$$\mathbf{F}_{A\tau} - \kappa \mathbf{n}_A = d \begin{cases} F_{An}\mathbf{v}_{r\tau} / |\mathbf{v}_{r\tau}| & \text{as} \quad |\mathbf{v}_{r\tau}| > \delta, \\ F_{An}\mathbf{v}_{r\tau}/\delta & \text{as} \quad |\mathbf{v}_{r\tau}| \leq \delta, \end{cases}$$

where one supposes that $\delta \ll 1$.

It is known [8] that in this case the solution of the regularized problem remains close to the solution of the original one on asymptotically large time intervals. Implementation and further simulation show this closeness holds with the very high degree of accuracy. Such an approach resolves completely the problem of modeling for accurate transitions between states of "Sliding" and "Rolling".

Thus properties of the frictional contact as: (a) contact tracking algorithm; (b) contact velocities computation; (c) contact forces computation; (d) contact surfaces particular properties are combined in an object-oriented manner forming lines of inheritance in a natural way.

5 CLASS PARAMETRIZATION IN CASE OF COMPLIANT CONTACT

Implementation of the mechanical contact model with compliance assumes much more possibilities for the contact properties than in the rigid case. According to experience while developing the models for elastic contacting of rigid bodies interactions in the multibody dynamics a flexibility provided by an object-oriented approach can be used to utilize a wide variety of different properties concerning a contact of solids. The properties are mainly of the following categories:

(a) geometric properties for surfaces in vicinity of the contact patch (gradients of the functions defining surfaces, their Hesse matrices);

(b) a model to compute the contact area dimensions and normal elastic force;

(c) model for the normal viscous force of resistance;

(d) model for the tangent forces along the plane of the contact area.

A submodel of the geometry properties is to describe analytically algebraic surfaces of the structure complex enough. To implement the normal force computation one can choose from at least two approaches: the Hertz model and its volumetric modification. Force of viscous resistance also can be modeled in several different ways: linear, non-linear, etc. In the models for tangent forces one can adopt either "simplest" approaches based

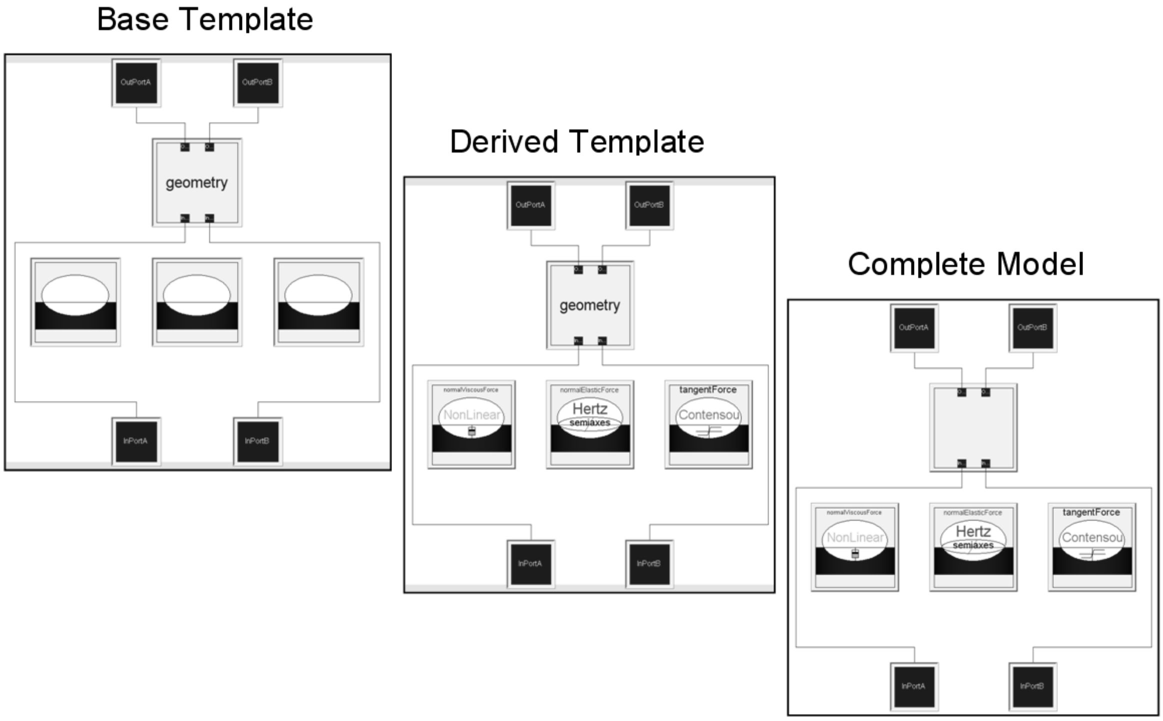

Figure 3: The model of mechanical contact by stages of inheritance.

on the Amontons–Coulomb friction or more complex ones represented by the Contensou–Erismann, and other models.

While developing a mechanical contact model architecture we used the base class *Constraint* described above as a starting point to construct its inheritor *ContactConstraintTemplate* being simultaneously a base class of new family of models to simulate mechanical contacts. Really this class is a base template represented as a container having four "sockets" to instantiate there the specific parameter classes of four types enumerated above, see its visual model in Figure 3 at a top left corner.

To develop complete model one can move along different ways. Class parametrization implemented in Modelica language for example is the facility in line to apply to the problem under description. In our case we have four class parameters corresponding to the submodel categories enumerated above. An example to construct specific contact interaction model see in Figure 3. The example includes two stages of inheritance:

1. to derive a template with the forces models, namely: the Hertz model for normal force, non-linear viscous force, the Contensou–Erismann model for the dry friction forces (to "fill in" three sockets in the middle of the base template visual model, see the derived template visual model at a central position of the Figure 3);

2. to complete the whole construct one should define a specific geometry submodel for the surfaces in contact (to "seal" the socket for geometry properties, see the complete visual model at a bottom right corner of the Figure 3).

On all the stages of inheritance the templates considered have an internal information interconnections between the submodels to be instantiated. These interconnections are implemented via the set of equations hidden behind the visual models and can vary for different models requiring different variables for the algorithms to compute normal and tangent forces of the complete model. So the whole picture remind us known construct of a card with the sockets and the interconnection wiring in its internal layers as a base template, and a chips to be instantiated in the sockets as a models of four types from above. With one exclusion: we have the derived template playing a role of additional card with its own additional wiring servicing already instantiated models "covering the card" of the base template.

One can remark finally an approach under presentation allows us to create and to change fast enough different types of an elastic contact models while developing the multibody dynamics systems simulators.

6 EXAMPLE OF THE OMNI VEHICLE DYNAMICAL MODEL

Investigation of omni vehicle dynamical properties is sufficiently popular topic in field of the multibody dynamics [9, 10, 11, 12]. The omni vehicle is one having omni wheels, wheels equipped by rollers along the rim. Simplified, idealized models having contacting rollers as an infinitely small discrete elements are known. Thus one has a resulting nonholonomic constraint being "uniformly distributed" over the wheel rim.

Our goal here is to develop a technique for building up a dynamical prototype for the "real" model of the omni vehicle explicitly involving dynamics of physical rollers. Here we rely upon the "simple" 3D multibody dynamics library classes from above which was utilized previously in several examples of the multibody systems dynamics [13]. Simultaneously this library enables us to create complex dynamical models including unilateral constraints of different nature.

We will pay here main attention to the process of the omni vehicle dynamical model development. Once again for simplicity we apply for the mechanical contact model the rigid point-contact one. This model has three structural levels of complexity: level of the roller; level of the omni-wheel; one of the whole vehicle, see Figure 4).

All the classes used in the vehicle model were mentioned above. In addition, one has to note that the contact tracking algoritm is possible to be arranged extremely simple and effective for the case of the wheel vertically aligned.

In the model of Figure 4 one has a vehicle with three omni-wheels. Each wheel has four rollers. Computational experiments were performed for different numbers of rollers. In addition, computational comparisons were performed for almost limit cases of simplified models with rollers inertial properties almost vanishing. The verification process was completed successfully.

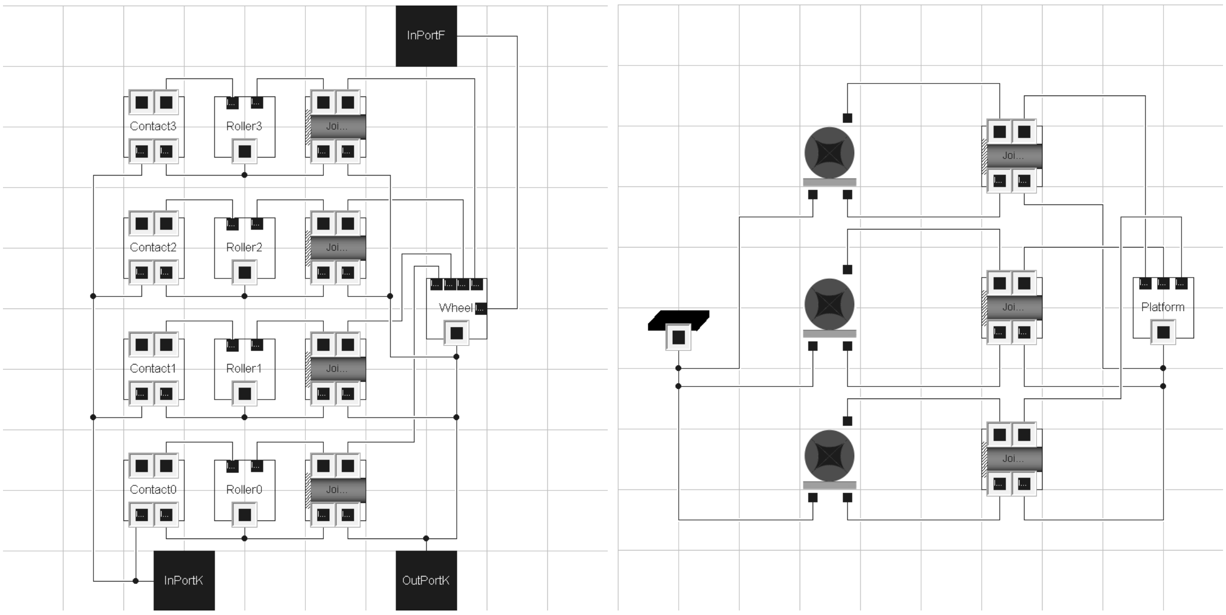

Figure 4: Left: The omni wheel visual model. Right: The omni vehicle visual model.

7 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.

-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 effective manner.

-The bond graph theory guidelines are useful for the MBS model building process to create consistent resulting DAE system.

-Introducing the compliance into the model may be useful and effective preserving the principal properties of the MBS like anholonomity etc.

-The most natural and effective way to implement computer model of the mechanical contact is use of the templates with the object-oriented class parametrization.

-There exist a possibility for smooth impactless switching between rollers at contact upon rolling of omni wheel.

-Efficient and simplified contact tracking algorithm was implemented.

-Influence of friction model on dynamics of the omni vehicle was analyzed.

8 ACKNOWLEDGEMENT

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

REFERENCES

[1] Fritzson, P. *Principles of object–oriented modeling and simulation with Modelica 2.1.* IEEE Press. (2004).

[2] Paynter, H. M. *Analysis and design of engineering systems.* The M. I. T. Press. (1961).

[3] Wittenburg, J. *Dynamics of systems of rigid bodies.* B. G. Teubner. (1977).

[4] Kossenko, I. I., 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) 299309.

[5] 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) 1323.

[6] Kosenko, I. 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) 8pp.

[7] Pfeiffer, F. Unilateral multibody dynamics. *Meccanica.* (1999) **34**:437–451.

[8] Novozhilov, I. V. *Fractional Analysis : Methods of Motion Decomposition*, Birkhauser. (1997).

[9] Campion, G., Bastin, G., d'Andréa-Novel, B. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. *IEEE Transactions on Robotics and Automation*, (1996) **12**:47–62.

[10] Kálmán, V. Controlled Braking for Omnidirectional Wheels. *International Journal of Control Science and Engineering*, (2013) **3**:48–57.

[11] Tobolár, J., Herrmann, F., Bünte, T. Object-oriented modelling and control of vehicles with omni-directional wheels. *Computational Mechanics 2009.* Hrad Nectiny, Czech Republic, November 9–11. (2009).

[12] Zobova, A. A., Tatarinov, Ya. V. The Dynamics of an Omni-Mobile Vehicle. *Journal of Applied Mathematics and Mechanics*, (2009) **73**:8–15.

[13] Kosenko, I. I., Loginova, M. S., Obraztsov, Ya. P., Stavrovskaya, M. S. Multibody Systems Dynamics: Modelica Implementation and Bond Graph Representation. *Proceedings of the 5th International Modelica Conference*, arsenal research, Vienna, Austria, September 4–5. (2006) 213–223.