WRAPPING MULTI-BOND GRAPHS: A STRUCTURED APPROACH TO MODELING COMPLEX MULTI-BODY DYNAMICS

François E. Cellier and Dirk Zimmer
Institute of Computational Science
ETH Zurich
CH-8092 Zurich, Switzerland
E-mail: {FCellier,DZimmer}@Inf.ETHZ.CH

KEYWORDS
Bond Graph, Multi-bond Graph, 3D Modeling, Model Wrapping, Object-oriented Modeling.

ABSTRACT

Bond graphs have established themselves as a reliable tool for modeling physical systems. Yet, they are highly abstract due to their domain independence. Wrapping techniques allow the modeler to preserve the better of two worlds: the flexibility and reliability of bond graphs on the one hand, and the intuitive appeal and familiarity offered by a domain-specific modeling methodology on the other. The talk introduces a new multi-bond graph library for Dymola that includes a partial re-implementation of Dymola's standard multi-body systems library using wrapped multi-bond graphs.

INTRODUCTION

In the early days of modeling and simulation (M&S), the systems that scientists and engineers were dealing with were so simple that the focus of M&S research was primarily on simulation. It didn't matter, what modeling formalisms were being used, as any formalism was good enough for the task at hand.

In later years, "fancy" (in terms of then available technology) M&S environments, like ACSL, were used to describe simple systems, where modeling methodology really didn't matter much, whereas models of more complex systems, e.g. aircraft or missiles, were consistently coded in low-level languages, like Fortran or C, because the more advanced M&S environments of those days were incapable of producing sufficiently efficient simulation run-time code.

In parallel, specialized codes were developed to capture models of particular application domains, e.g. Spice for electronic circuits or Adams for multi-body systems. These codes enabled the modeler to describe systems within the given domain efficiently and effectively, while guaranteeing an optimized execution speed of the resulting simulation code, but these tools were limited to a specific domain only. Especially, mechatronic systems that reach into multiple energy domains could not be handled using such tools.

M&S environments that enable modelers to capture arbitrarily complex physical systems in an object-oriented fashion, yet generate simulation run-time code that is as efficient as if not more efficient than the best manually coded spaghetti Fortran or C programs of the past in terms of execution time are of a more recent vintage. One such environment is *Dymola* (Dynasim 2006).

The Dymola M&S environment consists essentially of four separate programs. At the top layer, Dymola offers a *graphical user interface (GUI)* that enables the modeler to assign icons to component models and store these component models with their graphical representations in model libraries. More complex models can be graphically composed in a diagram window by dragging and dropping models from libraries into the diagram window and interconnecting them graphically on the screen. New icons can then be assigned to the composed models, enabling modelers to create hierarchically composed models. The Dymola modeling paradigm is thus closed under composition.

At the next lower level, Dymola offers a *model compiler* that extracts the equations from the individual component models, performs significant symbolic preprocessing on the resulting set of equations, e.g. to automatically reduce the perturbation index of a structurally constrained model. At the end, the model compiler generates a simulation run-time program coded in C.

At the next lower level, Dymola offers a simulation run-time environment that contains an appropriate set of numerical integration algorithms for simulating the previously generated simulation code.

Finally, Dymola offers a graphical postprocessor for viewing and animating simulation results.

THE BOND GRAPH LIBRARY

Bond graphs are a graphical modeling technique, enabling a modeler to describe physical systems stretching over multiple energy domains in a unified framework (Karnopp et al. 2006). Hence bond graphs are ideally suited for modeling mechatronic systems.

Bond graphs model the power flow through a physical system. Since the concepts of power and energy are domain independent, bond graphs can be used to model systems from any domain that subscribes to the concept of energy conservation, i.e., all physical domains.

Bond graphs are object oriented, since bond graphs of subsystems can be connected to each other topologically to form a correct model of a composed

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

system, and because it is possible to lump detailed bond graphs of subsystems together to form new bond graph elements that are hierarchically composed (Cellier 1990). Hence it should be possible to implement the bond graph modeling paradigm within Dymola.

A first version of a bond graph library for Dymola was released in 1991 (Cellier 1991). However at that time, Dymola did not offer a GUI yet. Consequently, that version of the bond graph library was purely alphanumerical. A fully graphical version of the bond graph library was released in 2003 (Cellier and McBride 2003, Cellier and Nebot 2005).

THE MULTI-BOND GRAPH LIBRARY

Although bond graphs can be used to describe any and all physical systems, they are not equally convenient for all energy domains. For example, bond graphs of mechanical multi-body systems operating in three-dimensional space will be difficult to compose and even harder to read, because each independently moving body has six degrees of freedom, as it can translate in three directions and rotate around three axes. Yet, the equations governing these six motions are essentially the same.

Consequently, it makes sense to offer a vectorial version of a bond graph:

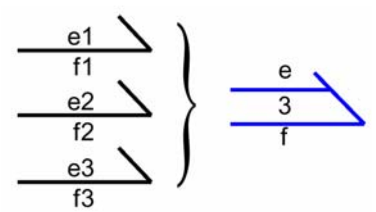


Figure 1: Grouping Individual Bonds to Multi-bonds

These vector bonds are called multi-bonds. Figure 1 shows a multi-bond of length three, as it might, for example, be used to describe a multi-body operating in a two-dimensional space.

Each regular bond carries two variables, an effort variable, *e*, and a flow variable, *f*. The power flowing through the bond is the product of effort and flow:

$$P = e \cdot f \qquad (1)$$

In the multi-bond version, effort and flow are vectors, and the multiplication operator denotes the inner product of these two vectors.

Clearly, the multi-bond graph library (Zimmer 2006) can also be used to describe regular bond graphs. To this end, the user simply needs to employ vectors of length 1. The default length of all vector bonds can be set by parameter assignment in the "world model." Yet, the regular bond graph library is still being offered, as multi-bonds are unnecessarily bulky for describing regular bonds, and as hardly any of the examples provided with the regular bond graph library have been copied over to the multi-bond graph library.

Multi-bond graphs are well suited for describing simple mechanical multi-body systems. For example, let us look at the multi-bond graph representation of a planar pendulum:

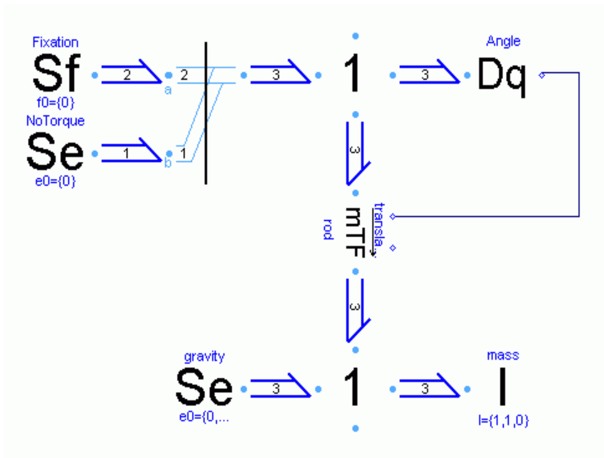


Figure 2: Multibond Graph Model of Planar Pendulum

The pendulum consists of a two-dimensional revolute joint and a mass-less bar translating the motion of the joint to the mass connected to the end of the bar.

In a mechanical bond graph, the effort variables represent forces and torques, whereas the flow variables represent velocities or angular velocities. The product of either force times velocity or torque times angular velocity represents mechanical power.

The joint itself does not move in a translation. Its linear velocity is zero, and consequently, we need a vector source of flow, *Sf*, of dimension two. The joint is free to rotate, i.e., it doesn't experience any torque. Hence we need a source of effort, *Se*, of dimension one. The two vectors are merged to a single vector of dimension three, whereby the first two components represent the linear translations in *x* and *y* directions, whereas the third component represents the rotation around the *z* axis.

The mass-less bar that converts the motion of the joint to the motion of the mass is represented by a (multi-port) transformer, *TF*. The transformation matrix is modulated by the angle of the revolute joint, which is measured by a sensor element, *Dq*.

The mass itself is represented by the 1-junction. In a 1-junction, the flow variables are equal, whereas the effort variables add up to zero. Hence the 1-junction represents the d'Alembert principle applied to the mass.

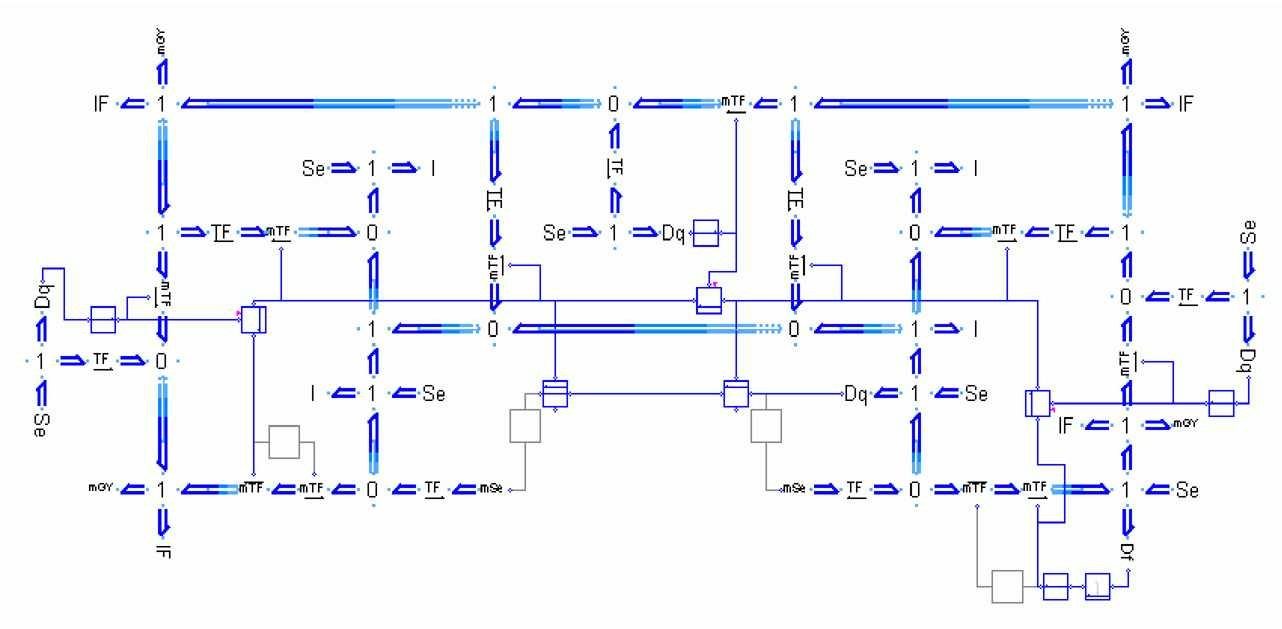


Figure 3: Multi-bond Graph Model of a Bicycle

The forces acting on the mass are the inertial force, *I*, and the gravitational force, which can be represented by another source of effort, *Se*, pulling in the negative *y* direction.

The notation may look unfamiliar at first, but with a bit of experience, it becomes easily readable and understandable.

Let us now proceed with modeling a more complex multi-body system: a bicycle consisting of a frame, two wheels, the handlebars, and a driver. The multi-bond graph model is shown in Figure 3. A similar model had been presented in the Ph.D. dissertation of Bos (Bos 1986), although at that time, the graphical representation was drawn by hand and translated manually into corresponding equations. In contrast, our own model represents a perfectly executable code.

Let us refrain from trying to explain how this model works. The model is clearly too big to fit easily on a single screen. Furthermore, the sheer generality of the bond graph approach to modeling is also its downfall. In order to be general, bond graphs cannot conveniently be made specific as well. In a bicycle, we can easily identify objects, such as wheels and handlebars, but not effort sources or modulated transformers. Bond graphs offer a low-level interface, that is more readable than an equation-based interface, but not readable enough for modeling complex systems.

THE STANDARD MULTI-BODY LIBRARY

Dymola offers a standard multi-body systems (MBS) library, developed at the German Aerospace Center in Oberpfaffenhofen (Otter et al. 2003). Using this library, multi-body systems can be easily and conveniently composed out of blocks that carry an intuitive meaning.

Figure 4 shows a six degree of freedom (DoF) robot arm.

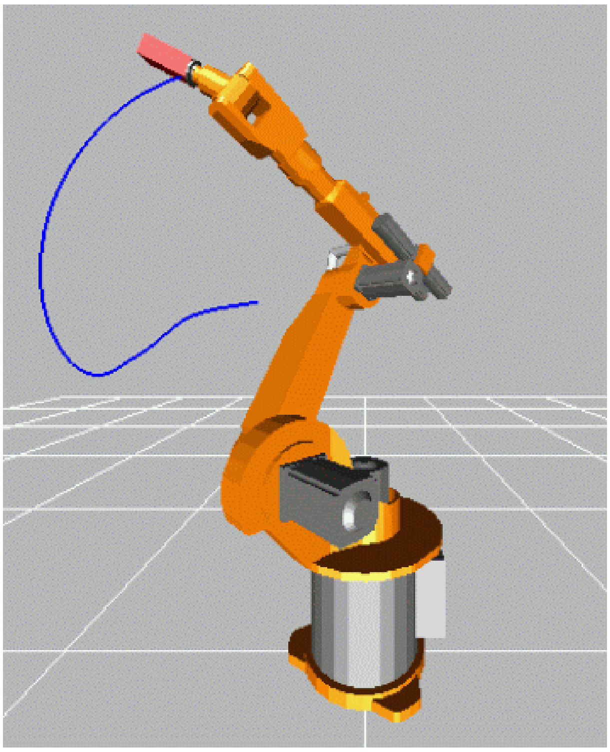


Figure 4: Six degree of freedom robot arm

The robot arm exhibits seven bodies that are connected by six revolute joints. Each of the joints is controlled by a controller. Together they determine the motion of the robot arm.

The corresponding Dymola model is shown in Figure 5:

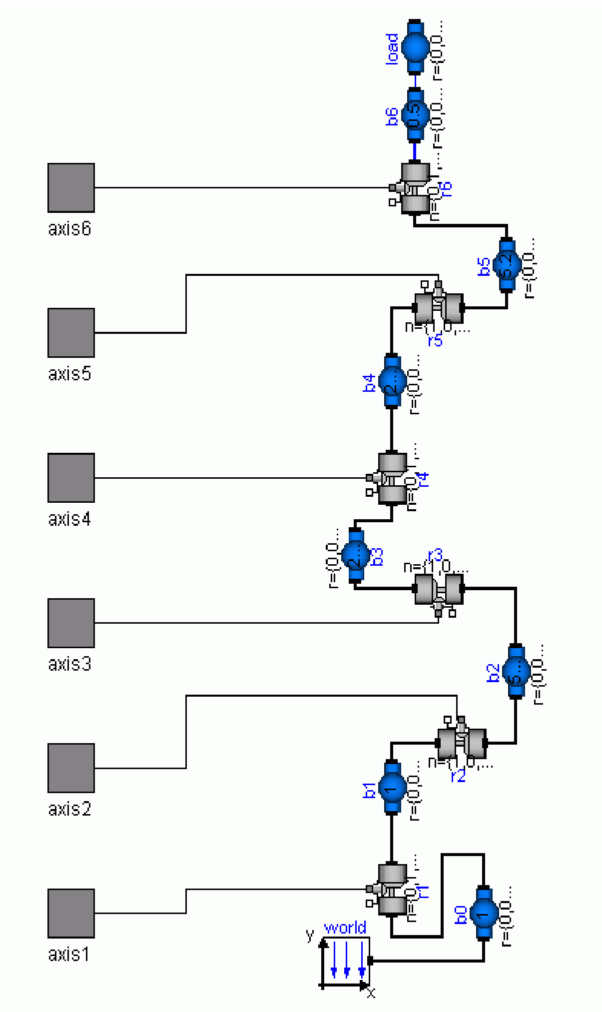


Figure 5: Dymola model of a six DoF robot arm

The model is perfectly understandable. The lower-most body, i.e., the base, is connected to the inertial system, which in the MBS library also assumes the role of the world model. It determines the world coordinate system, defines the gravity field, and sets up default animation parameters.
The model represents an abstracted version of the system topology, and is easily understandable. The MBS library is easy to use. It can even be used by modelers without any deeper understanding of MBS dynamics.
The occasional modeler will, however, be in deep problems, whenever and as soon as a model is not simulating correctly. It will be an almost hopeless undertaking to try figuring out what went wrong.
The reason is that the step from the component models of Figure 5 to the next lower hierarchical level in the model hierarchy is huge. Bodies and joints are modeled in terms of matrix equations directly, which are difficult to understand. In order to obtain efficiently executing simulation code, suitable coordinate transformations are taking place inside the code that make the code even more cryptic.

WRAPPING BOND-GRAPH MODELS

The previously introduced multi-bond graph library contains a modified MBS library that, from the outside, looks very similar to the MBS library offered as part of the standard Dymola installation.
Let us revisit the bicycle example to demonstrate, how the modified MBS library works. Figure 6 depicts the bicycle model coded in the modified MBS library.

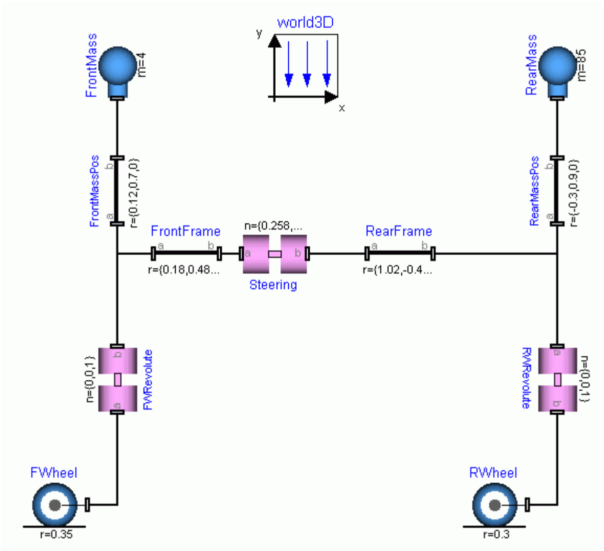


Figure 6: Dymola model of a bicycle

The model is perfectly understandable. At the right bottom of the graph, the rear wheel is depicted. It is connected to the frame of the bicycle by a revolute joint. At a certain distance from the center of the rear wheel sits the driver, who, together with the rear part of the frame, weighs 85 kg. Also at a fixed distance from the center of the rear wheel are the handlebars. They are connected to the frame by a second revolute joint, and have a mass of 4 kg. Finally, a third revolute joint connects the front wheel to the handlebars.
Let us examine the model of the rear wheel. It is shown in Figure 7:

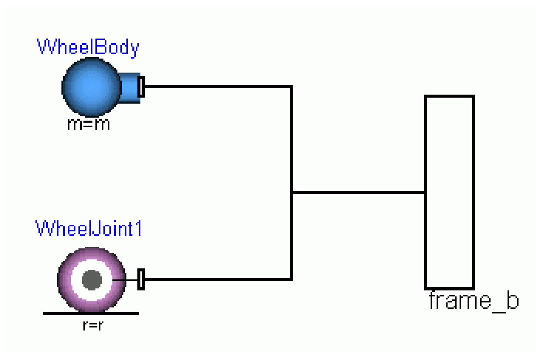


Figure 7: Dymola model of a wheel

The model consists of the inertia of the wheel together with a joint connecting the wheel to the road.
The overall bicycle model contains a closed kinematic loop from the road through the rear wheel, the frame, and the front wheel back to the road. Closed kinematic loops cause problems, because they introduce additional constraints, thereby reducing the number of degrees of freedom of the model.
In older versions of the MBS library, the modeler had to manually break closed kinematic loops by introducing so-called cut joints (Otter 2000). Cut joints are regular joints that, however, do not define integrators connecting the accelerations with the velocities and with the positions, thereby avoiding the creation of redundant equations.
In the mean time, algorithms were built into both the standard and the modified MBS libraries that are capable of automatically breaking most kinematic loops (Otter et al. 2003).
What is the advantage of the modified MBS library over the standard one? To answer that question, let us examine the model of the wheel joint. It is shown in Figure 8.

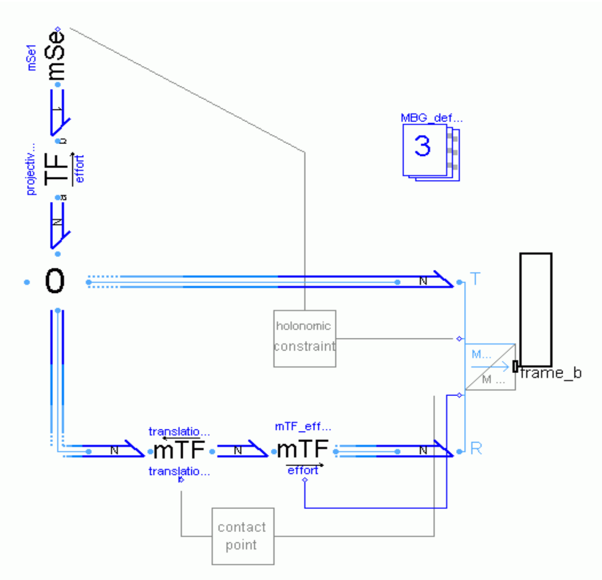


Figure 8: Multi-bond graph model of a wheel joint

The internal description of the wheel joint is a multi-bond graph. The corresponding model of the standard vehicle dynamics library (Andreasson 2003) would have shown a rather unholy mess of matrix equations instead.
Although the multi-bond graph may require some explanation, use of the multi-bond graph library has enabled us to subdivide the step from the wheel model down to the equation model by introducing an additional graphical layer in between the two.
Multi-bond graphs have been wrapped inside most of the MBS component models of the modified MBS library with the purpose of making these models better understandable and more easily maintainable.

Let us analyze the wrapper model that converts the bondgraphic connectors to mechanical connectors and vice-versa. It is shown in Figure 9.

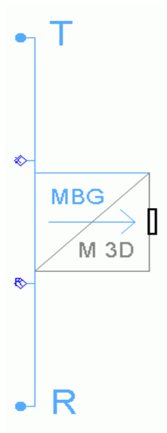


Figure 9: Wrapper icon of the modified MBS library

In the modified MBS library, the three-dimensional mechanical bond vectors of length six are subdivided into two subvectors of length three each, one used to describe the translational motions, the other used for the rotational motions.
The reason for this separation is simple. We prefer to resolve translational motions in the inertial frame, whereas rotational motions are resolved in body-fixed coordinates. This minimizes the number of coordinate transformations needed in the description of three-dimensional mechanical systems.
The bond-graphic connectors use thus either forces or torques as effort variables, and either velocities or angular velocities as flow variables. The standard MBS library, on the other hand, uses positions and angles as potential (effort) variables, and forces and torques as flow variables.
In a mechanical system, it is important to transmit the positional variables between neighboring bodies, as they allow the formulation of holonomic constraints, i.e., constraints that prevent bodies from transgressing each other.
In order to be compatible with the bond graph methodology, the mechanical connectors of the modified MBS library have been augmented by the translational velocity vector[1], i.e., the connectors of the standard and modified MBS libraries are incompatible with each other, and component models from the two libraries cannot be arbitrarily mixed.
On the bond graph side, the positions and angles are made available as two additional connectors that enable the formulation of holonomic constraints on the bond graph.
Figure 10 shows the internal description of the wrapper model. This model is formulated at the equation level.

[1] The rotational velocity vector is contained in the connectors of the standard MBS library as well.

```
model MBG2Mech "Links bond graph with mechanic connector"

equation
  MultiBondConTrans.e*MultiBondConTrans.d = frame_b.f;
  MultiBondConRot.e*MultiBondConRot.d = frame_b.t;

  MultiBondConTrans.f = frame_b.P.v;
  MultiBondConRot.f = frame_b.P.w;

  x = frame_b.P.x;
  R = frame_b.P.R;
end MBG2Mech;
```

Figure 10: Wrapper model of the modified MBS library

The translational effort, *e*, multiplied by the directional variable, *d*, which assumes a value of –1 at the beginning of a bond and a value of +1 at the end of a bond, is set equal to the mechanical force, *f*. Similarly, the rotational effort multiplied by the directional variable is set equal to the mechanical torque, *t*. The translational flow vector, *f*, is set equal to the mechanical velocity vector, *v*, and the rotational flow vector is set equal to the mechanical angular velocity vector, *w*. Finally, the mechanical position vector, *x*, and the mechanical angular position vector, *R*, are made available as *x* and *R* through separate connectors also on the bond-graphic side.

We are now ready to discuss the multi-bond graph model of Figure 8. The 0-junction represents the position of the center of the wheel. The rotation of the wheel results in a translation at the contact point of the wheel with the road. The translation at the contact point is calculated from the rotation by means of a transformer. The second transformer further to the left in Figure 8 converts the contact point back to the center of the wheel.

The position of the center of the wheel is thus determined twice, yet the two values must obviously be the same. If there were only one wheel, there wouldn't be a problem. The bicyclist moves the wheel, i.e., causes a rotation, which in turn can then be used to compute the translation of the bicycle forward. Yet, since there are two wheels, we face a closed kinematic loop. This generates surplus equations that need to be removed again. The MBS library is supposed to take care of this automatically.

Yet, there is a second problem. The weight of the bicycle would make the bicycle sink into the road. Yet, this cannot be. The distance of the center of the wheel to the contact point with the road must always be equal to the radius of the wheel. Hence there is a holonomic constraint. The holonomic constraint is satisfied by a reaction force that compensates for the force that wants to drive the bicycle into the road.

Bond graphs have notoriously a hard time with holonomic constraints, as they don't operate on positions at all. They only deal with forces and velocities. Consequently, rather than formulating a holonomic constraint, we calculate the reaction force that keeps the bicycle on the road. This is done using the effort source, *Se*, at the top of Figure 7.

GRAPHICAL VS. EQUATION MODELING

Evidently, the bottom layer component models of any system description must be coded using equations. The graphical models can only serve to describe the topology of a system, whereas the basic physical properties must be captured using equations.

Using wrapped multi-bond graphs, we were able to ban the equations almost entirely down to the level of the bond-graphic components, i.e., the transformers, resistors, capacitors, inductors, etc. These models can be created once and for all, and they are flexible enough to capture the basic properties of essentially all physical systems.

At the next higher level in the modeling hierarchy, i.e., the level, where bodies, joints, and force elements are being described, there is relatively little need for additional equations. Almost all of these elements can be mapped onto a corresponding multi-bond graph, which enhances both the readability and the maintainability of these models.

Additional equations are needed to describe the geometric properties of bodies for the purpose of animation. The geometric model of the bicycle is depicted in Figure 11.

Figure 11: Geometric model of the bicycle

Although it would be possible to design the geometric model graphically using a CAD tool, this is not, how the geometric bicycle model was created. Instead, the model was coded by means of equations associated with the four body models, i.e., the two wheels, the rear frame, and the handlebars.

SUMMARY

In this paper, we have shown that wrapped multi-bond graphs offer a means to minimize the need for equation modeling in the description of complex mechanical multi-body systems. The equation models are forced down to the level of the bond-graphic component models. These models are small, and therefore easily maintainable. The level of the mechanical component models can thus already be described by graphical techniques, i.e., in the form of relatively small and compact multi-bond graphs that can be more easily debugged and maintained than the equation models used in Dymola's standard multi-body systems library.

The modified MBS library forms an integral part of the multi-bond graph library. Beside from replicating component models of the MBS library, the multi-bond graph library also offers a separate set of component models for planar mechanics, as well as a set of models for describing mechanical systems undergoing collisions (impacts).

REFERENCES

Andreasson, J. 2003. "Vehicle Dynamics Library." *Proc. 3rd International Modelica Conference*, Linköping, Sweden, 11-18.

Bos, A.M. 1986. *Modelling Multibody Systems in Terms of Multibond Graphs with Applications to a Motorcycle*. PhD Dissertation, University of Twente, The Netherlands.

Cellier, F.E. 1990. "Hierarchical Non-linear Bond Graphs: A Unified Methodology for Modeling Complex Physical Systems." Keynote presentation. *Proc. SCS European Simulation Multi-Conference,* Nuremberg, Germany, 1-13.

Cellier, F.E. 1991. *Continuous System Modeling*. Springer-Verlag, New York, 755p.

Cellier, F.E. and R.T. McBride 2003. "Object-oriented Modeling of Complex Physical Systems Using the Dymola Bond-graph Library." *Proc. ICBGM'03, 6th SCS Intl. Conf. on Bond Graph Modeling and Simulation*, Orlando, Florida, 157-162.

Cellier, F.E. and A. Nebot 2005. "The Modelica Bond Graph Library." *Proc. 4th International Modelica Conference,* Hamburg, Germany, Vol.1, 57-65.

Dynasim AB. 2006. *Dymola User Manual.* Version 6.0, Research Park Ideon, SE-223 70 Lund, Sweden.

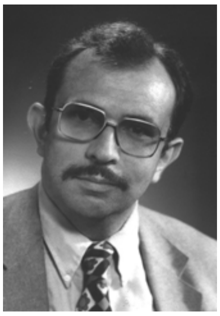
Karnopp, D.C., D.L. Margolis and R.C. Rosenberg. 2006. *System Dynamics: Modeling and Simulation of Mechatronic Systems, 4th Edition*. John Wiley & Sons, New York, 576p.

Otter, M. 2000. "Objektorientierte Modellierung physikalischer Systeme, Teil 17: Modellierung mechanischer Systeme." *Automatisierungstechnik*, **48**(12), A65-A68.

Otter, M., H. Elmqvist, and S.E. Mattsson. 2003. "The New Modelica MultiBody Library." *Proc. 3rd International Modelica Conference*, Linköping, Sweden, 311-330.

Zimmer, D. 2006. *A Modelica Library for MultiBond Graphs and its Application in 3D-Mechanics*. MS Thesis, ETH Zurich, Switzerland.

AUTHOR BIOGRAPHIES



François E. Cellier received his BS degree in electrical engineering in 1972, his MS degree in automatic control in 1973, and his PhD degree in technical sciences in 1979, all from the Swiss Federal Institute of Technology (ETH) Zurich. Dr. Cellier worked at the University of Arizona as professor of Electrical and Computer Engineering from 1984 until 2005. He recently returned to his home country of Switzerland. Dr. Cellier's main scientific interests concern modeling and simulation methodologies, and the design of advanced software systems for simulation, computer aided modeling, and computer-aided design. Dr. Cellier has authored or co-authored more than 200 technical publications, and he has edited several books. He published a textbook on Continuous System Modeling in 1991 and a second textbook on Continuous System Simulation in 2006, both with Springer-Verlag, New York. He served as general chair or program chair of many international conferences, and serves currently as president of the Society for Modeling and Simulation International.

Dirk Zimmer received his MS degree in computer science from the Swiss Federal Institute of Technology (ETH) Zurich in 2006. He gained additional experience in Modelica and in the field of modeling mechanical systems during an internship at the German Aerospace Center DLR 2005. Dirk Zimmer is currently pursuing a PhD degree with a dissertation related to computer simulation and modeling under the guidance of Profs. François E. Cellier and Walter Gander. His current research interests focus on the simulation and modeling of physical systems with a dynamically changing structure.