Object-Oriented Decomposition of Tire Characteristics based on semi-empirical Models

Markus Andres
Vorarlberg Univ. of Appl. Sc.
Austria
Markus.Andres@students.fhv.at

Dirk Zimmer
ETH Zürich
Switzerland
DZimmer@Inf.ETHZ.CH

François E. Cellier
ETH Zürich
Switzerland
FCellier@Inf.ETHZ.CH

Abstract

This article introduces a new and freely available Modelica library, called *Wheels and Tires*, for modeling wheels and tires. The contained models are intended to be used in vehicle simulations, where computational performance is a major concern. Semi-empirical single contact point models are well suited for this kind of applications and are therefore applied in the presented library.

The *Wheels and Tires* library provides a tool to quickly build custom tire models, and allows a convenient customization of existing models. This is achieved by a modular and expandable design system utilizing well established models. In addition, a set of ready-made models is provided to allow a quick insight in the used modeling structure and to enable a direct application in vehicle models. The final version of the library will be published as a free library via the Modelica website as well as the website of F. E. Cellier.

Keywords: object-oriented tire modeling; object-oriented tyre modelling; semi-empirical tire model; tire decomposition

1 Introduction

During the last decades a fairly large number of tire models of varying levels of complexity suiting basically differing fields of application have been developed. These range from simple non-slipping tires to very complex FEA (finite element analysis) models for performance prediction [6].

The library developed is intended to be used in simulations that cover entire vehicles, therefore computational effort is an important issue. Hence, the selection of the appropriate level of detail for the used models is essential for the overall simulation performance. Semi-empirical single contact point models provide a very good trade-off between accuracy and computational effort. Such models are based on physical considerations, like those emerging from multibody dynamics. These physical aspects get enhanced with empirical formulas representing measurement results that cover e.g. friction and slip characteristics. Two of these semi-empirical models are commonly accepted and widely used. These are TMeasy by G. Rill [11] and the magic-formula model by H. B. Pacejka [10]. However, both are often implemented in a flat and mainly unstructured fashion, which makes them difficult to understand and maintain. Customizing these models for particular situations or expanding them in order to cover new aspects of tires can be cumbersome and is often error-prone.

A paper by D. Zimmer and M. Otter [14] builds on the previously mentioned models and demonstrates how models of varying levels of complexity can be integrated within the object-oriented framework of Modelica. However, the object orientation in these models limits itself primarily to their external interfaces. The models themselves continue to be mostly flat. For instance the most complex tire model created, defines approximately 200 equations [14] and is a good example showing the difficulties that arise from the common flat structure.

Another example for a quite flat structure can be found in the freely available but outdated Vehicle Dynamics library [2]. There, a wheel-base model gets extended with friction models of [11] and [10], but not much further effort was spent regarding object-orientation.

In [1], a tire model is modularized in hub, belt and road elements. A further enhancement is made in [5] by redesigning the model's structure as well as enabling uneven road surfaces and losing contact to the ground due to enhanced vertical dynamics. This modularization is well defined, but still the different aspects of friction are summarized in the *Tyre-Road* class

DOI: 10.3384/ecp09430029

and can not be customized easily. Moreover the libraries presented in [14], [1] and [5] are not freely available.

The newly developed library takes the object-orientation even further than in [5]. Therefore the focus of this research effort concerns itself less with modeling new tire properties, but more with an improved organization of existing knowledge. This will enable future modelers to conveniently customize the models to their own purposes.

2 Basic Considerations

2.1 A Closer Look at Tires

In motion dynamics of vehicles the forces exerted by the tire-road contact are of major importance. This section is intended to provide basic knowledge about tires buildup. For a more detailed information the reader is referred to [8], [9], [10] or [11].

The modern tire is a complex construction resulting from clashing requirements. They basically have to carry the vertical load, transmit forces to accelerate (and slow down) the vehicle and generate cornering forces to guide the vehicle through curves securely. This has to be fulfilled under a large variety of environmental conditions with a long life time ensured. The rolling resistance has to be as small as possible, with damping and acoustic properties suiting modern demands. As one can imagine there is no optimal solution to this problem resulting in a large variety of different tires for varying demands.

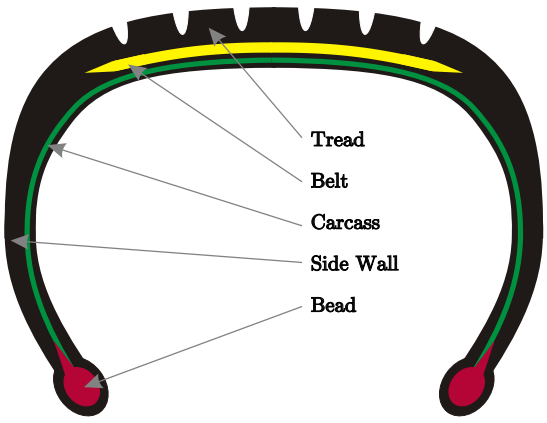

Figure 1: Basic structure of a tire.

A quite basic design example for a tire is depicted in Figure 1. For today's passenger cars steel-belt tires are used exclusively, which differ in construction only marginally, when treated from such a basic point of view. On the inside of the tire a coating (not depicted in Figure 1) inherits the function of the tube, preventing the over-pressurized air to leak to the outside. The *Bead* is usually built of steel wire with synthetic rubber components, ensuring a tight fit of the tire on the rim, allowing a reliable operation under difficult conditions e.g. when driving over a curbstone. The rubber elements building the sidewall strongly affect the vertical dynamics of the tire and are important when it comes to handling precision and stability. The *carcass* is the element absorbing the tension from the inflation pressure. Therefore, it has to be protected from damage, which is ensured by the *side wall*. The *tread* is responsible for the force generation by establishing a reliable contact to road and is therefore a very central element of the tire. Its composition is a major factor when it comes to the frictional properties of the tire. The *tread* is reinforced by the steel *belt* that enhances mileage and reduces the rolling resistance. Overall a mixture of more than 20 rubber composites form the tire, which makes them quite difficult to describe as well as enabling the tire engineer to adjust the tire properties to different needs.

2.2 Definition of Coordinate Systems

Figures 2 to 5 are intended to present basic assumptions made. Figure 2 shows the two coordinate systems used to describe the orientation of a wheel. The contact point's coordinate system is described by e_{Long}, e_{Lat} and e_N, whereas e_{Long}, e_{Axis} and e_{Plane}[1] form the rim's system. Figure 4 shows the standardized

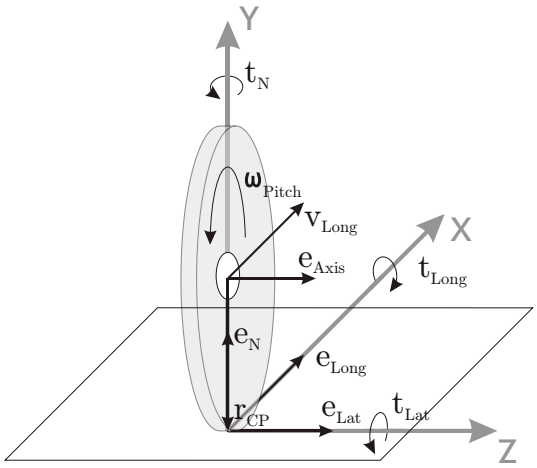

Figure 2: The unitary vectors of the tire without a lean angle and the vector r_{CP} pointing from the rim's center to the contact point.

[1] Pointing in the opposite direction of e_N in Figure 2 and shown in Figure 3 and 4.

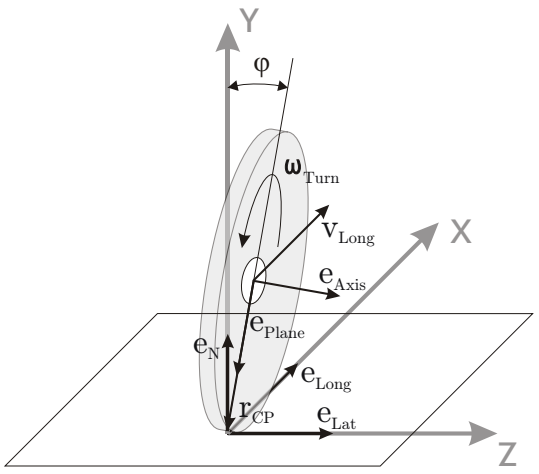

Figure 3: The unitary vectors of the tire with a lean angle φ of 10°.

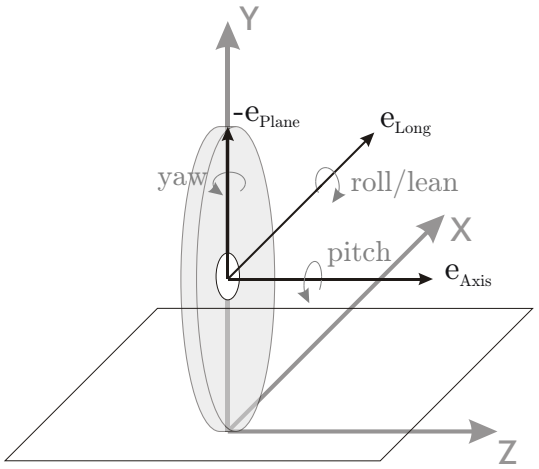

Figure 4: Standardized names of the angles and angular velocities.

names of the angels and their corresponding angular velocities.

An enlarged view of the contact point is depicted in Figure 5. It shows important properties like the translational velocity of the contact point in longitudinal (v_{Long}) and lateral (v_{Lat}) directions, the overall velocity v and the slip angle α. The sliding velocity in longitudinal $v_{SlipLong}$ direction is calculated by $(\omega \times r_{CP}) \cdot e_{Long} + v_{Long}$.

3 Object-oriented Tire Modeling

Section 3.1 is intended to demonstrate the considerations that lead to the actual structure of the tire model. Afterwards Section 3.2 explains the resulting structure from the modeler's point of view. Sections 3.3 to 3.10 shallowly introduce the classes forming the tire.

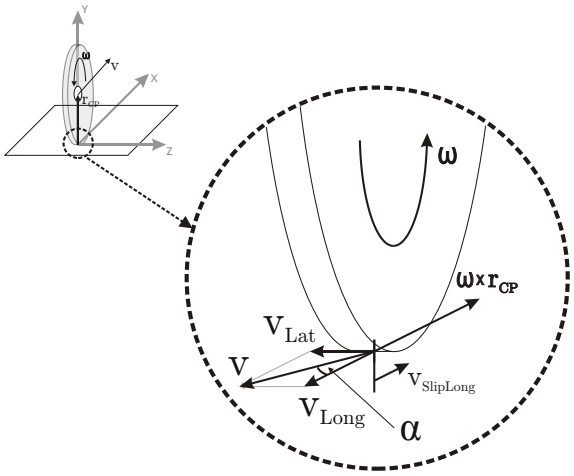

Figure 5: An enlarged view of the contact point with sliding velocities.

3.1 Decomposition into Objects

Thinking about wheels, the first division of the model is quite obvious, as there are two physical components: the rim and the tire. The rim does not need to be split up into further objects, as its properties can be modeled in a quite simple fashion in a semi-empirical tire model. This and the main properties of tires regarding modeling are shown in Figure 6.

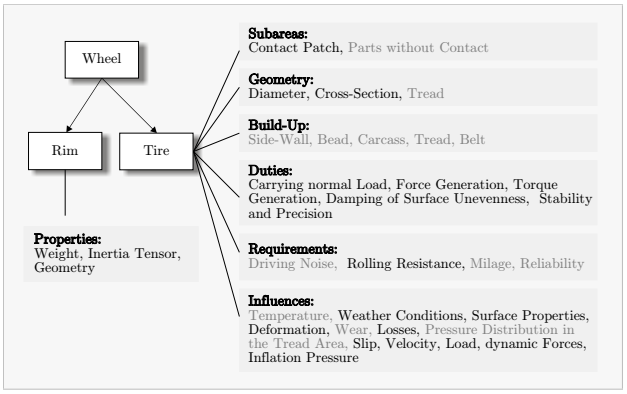

Figure 6: Composition of wheels and properties of tires. Properties depicted in gray are neglected in the presented tire model.

The tire does require a much closer look concerning its properties. Modeling every single of the properties shown in Figure 6 by a class of its own is an infeasible task, suggesting a certain grouping of properties. Due to the semi-empirical single contact point model one constraint is fixed. There has to be a model describing the contact point. It is named the *Contact Physics* model.

One of the most challenging tasks when modeling a tire, is to calculate the forces the tire excites in dif-

ferent driving situations. There are several different models trying to describe the relations resulting in the forces that act on the contact point. Hence a decision was made to create a class that gathers the different effects that are responsible for the generation of forces and torques acting on the contact point. Due to its origin it is called the *Friction* class, resulting in Figure 7.

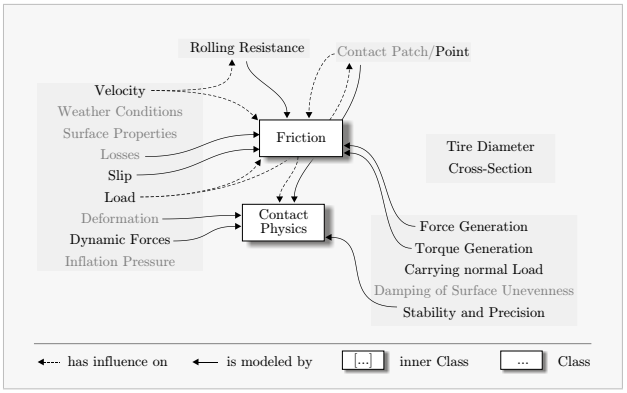

Figure 7: Properties from Figure 6 shown in relation to the corresponding *Tire Component* classes which are closely related with the semi-empirical contact point model. Properties depicted in gray are considerably simplified in the model.

Still, absolutely basic properties of the tire are not yet part of the model as shown in Figure 7. The probably most obvious are the *Tire Diameter* and the *Cross-Section*. These properties basically define the geometry of the tire, which shall be changeable conveniently in the final tire model, allowing basically different geometric properties of the tire. Therefore a *Geometry* class is introduced, defining the positional relation between the tire hub and the contact point and some other properties.

The next very basic duty the tire has to fulfill, is the *Carrying of normal Load* resulting in a normal force. Due to the changing requirements to these aspects, the effects are described in a class named *Vertical Dynamics*.

Until now, all possible tire models would be totally rigid. To enable a certain deformation of the tire, the *Belt Dynamics* class is introduced. It allows a flexibility of the still rigid tire, defined by the *Geometry* class, related to the tire's hub.

The wheel is now modularized into six classes including the *Rim* class which is not depicted in Figure 8, as it was shown in Figure 6. Section 3.2 explains the implemented version of these classes from the modeler's point of view. It also explains why the final model of the tire contains seven classes, introducing a *Center to Contact Point* class.

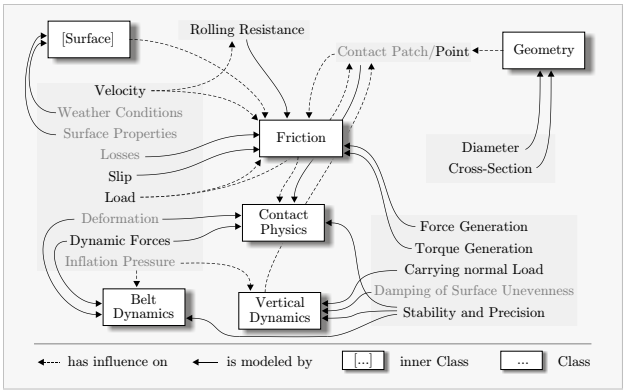

Figure 8: Final decomposition of the tire.

Still there is one class found in Figure 8 that has not been introduced until now. This is justified as it is an *inner* model and does not form a part of the actual tire model. It realizes uneven surfaces and allows a position-depending friction coefficient. This model is described in Section 5.

3.2 The Tire Model's Structure

The tire is split up into seven objects shown in Figure 9. It consists of objects modeling

- *Vertical Dynamics*,
- *Friction*,
- *Geometry*,
- *Contact Physics*,
- the translation from *Center to Contact Point*,
- *Belt Dynamics* and
- the *Rim*.

The single classes are described in the following sections. Here the relations between the classes shall be illustrated.

The connection to the superordinate vehicle model is established by the three-dimensional frame named *Tire Hub* depicted in Figure 9. The *Tire Hub*, a standard element of the *MultiBondLib* [13], is rigidly connected to the model of the *Rim*. The rim's frame connected to the *Tire Hub* therefore models the center-point of the rim. The "output" of the *Rim* again models the center-point of the *Rim*, and is connected to the *Belt Dynamics* model. In this model the relative movement between the rigid belt and the rim can be described. The "output" of the *Belt Dynamics* is again

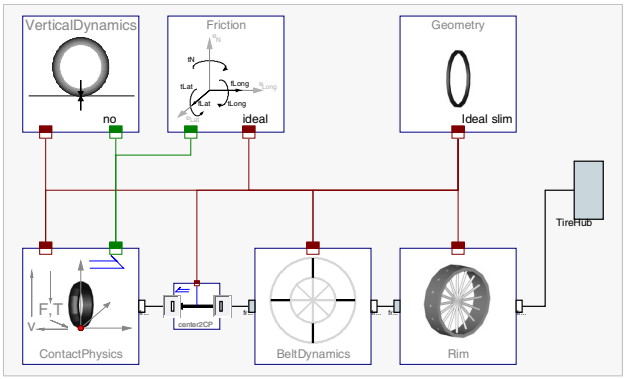

Figure 9: Model of the ideal tire showing the seven used objects and the communication structure on the top level. The *Tire Bus* is colored red whereas the *Contact Point Connector* is green.

positioned at the center-point of the belt. Therefore an element is needed to realize the translation from this point to the contact point. As this cannot be modeled by a standard element, the *Center To Contact Point* model has been created. It connects the *Contact Physics* model that applies forces and torques to the contact point. This lower part of the model represents the mechanically connected part of the model. The upper three classes are not directly connected by a mechanical connector, although the *Contact Point Connector* implies kind of a mechanical connection. The *Vertical Dynamics* class determines if and how the tire is able to lift from the ground and how it responds to normal load. The *Friction* class determines the longitudinal and lateral forces as well as the torques that act on the contact point. Finally, the *Geometry* class determines the unitary vectors shown in Figure 2 and the position of the contact point depending on the actual geometry, position and orientation of the tire.

The visualization is implemented in the corresponding object, e.g. the rim is visualized in the *Rim* object and the tire is visualized in the *Geometry* class. This makes it possible to recognize basic changes to the models in the animation directly.

All of the featured classes that model a certain type of effect are extended from the corresponding base class, ensuring that the necessary output is calculated. This guarantees also that all models stay exchangeable not depending on the implementation of the class. In this way, the user can add new classes or adapt existing ones being sure that the other elements stay unchanged and remain exchangeable.

3.3 Rim Class

The *Rim Class* is a rather simple model, as the rim can be modeled ideally just consisting of a body that has a mass and an inertia tensor.

3.4 Friction

For the sake of object oriented modeling, the different modeled effects are put into subclasses (see Figure 10 as an example) that are of varying complexity. The

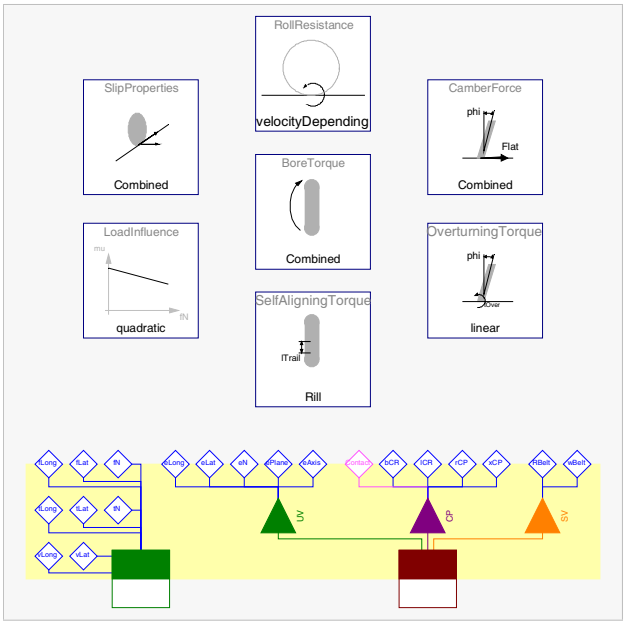

Figure 10: Model of the advanced friction.

communication between the classes is established using *inner/outer* statements. Therefore no connections between the sub-models are visible in Figure 10.

To use a certain combination of frictional effects in a tire model, a class gathering these effects has to be created as shown in Figure 10. This class can then e.g. replace the *Ideal Friction* depicted in Figure 9. In the library three different combinations of *Friction* classes are composed, forming an *Ideal Friction*, one simple *Dry Friction With Rolling Resistance* but without e.g. *Bore Torque* or *Camber Force* and one *Advanced Friction* that is shown in Figure 10. New frictional models can be created easily, which is the reason why only three frictional classes were included in the library.

3.5 Geometry

All geometric classes have two main tasks to fulfill. The first is to determine the contact point properties including the vector *rCP* that points from the center of the rim to the contact point and the penetration depth.

Secondly the unit vectors shown in Figure 2 get computed by the *Geometry* class. To enable that functionality it utilizes the *Surface*'s functions *get_eN* and *get_elevation* establishing the connection from the tire to the surface.

Three different *Geometry* classes are provided for ideally *Slim*, *Circular* tires with cross-section being modeled as a semi-circle, and a "*Belt*" profile with side walls and a sector of a circle modeling the tread area.

3.6 Contact Physics

The *Contact Physics* model applies forces and torques on the contact point, as well as measuring its velocities. All these physical quantities are connected to the models determining frictional and vertical dynamic properties via the *Contact Point Connector*. Measuring and setting of speeds and forces has to be done in an a-casual way as ideal models set velocities rather then apply forces.

The second property determined by the *Contact Physics* class is the dynamic behavior of the contact point. It can introduce flexibility of the contact point in longitudinal and lateral direction.

3.7 Center to Contact Point

The *Center To Contact Point* class is a model without a nice physical interpretation. It is kind of a *Fixed Translation* element known from the *Modelica Standard Library*. The model is built using multi-bond graphs and therefore derived from the *Fixed Translation* in the *MultiBondLib* [13]. It describes the connection between the contact point and the belt's center point.

3.8 Vertical Dynamics

The models in the *Vertical Dynamics* package determine the behavior of the tire normal to the surface. The normal force is related to the penetration depth computed in the *Geometry* class. The *Vertical Dynamics* models can either set the penetration depth to a certain value, or use it as a base for the calculation of the normal force f_N.

There are basically three different approaches to model vertical dynamics in the library. One is to not allow any elevation from the ground or penetration into it introducing a holonomic constraint. The second utilizes an *ElevationGap* model that compensates forces of a attached 1D mechanical model when the tire lifts from the ground. It is a derivation from the *ElastoGap* model found in the *BondLib* [7]. It makes the vertical dynamics easily changeable by a modification of the 1D mechanical system. The third is a derivation of the *ElastoGap* model of the *Modelica Standard Library 3.0*. It overcomes the sticking effect of the *ElevationGap* but is harder to adapt to different dynamics.

3.9 Belt Dynamics

The models described in this section allow the tire to have a certain flexibility. This is realized by connecting an ideal (virtual) belt to the ideal rim in a flexible fashion. All models except the *Rigid Belt* model add a considerable amount of complexity to the simulation. Different models are provided to realize the dynamic behavior. One allows translation of the belt in longitudinal and lateral direction, another models a rotational degree of freedom around the axis of rotation, both defining the dynamics by 1D mechanical elements. The last model is the most complex with the dynamics defined by four ideally stiff translational elements for rim and belt respectively, connected by 3D spring damper systems.

3.10 Communication Structure

The tire's objects compute a bunch of different variables, some of which are used in different objects as well. The *Tire Bus*, with a connector as shown in Figure 11, gathers the variables used in most of the objects. Additionally a division into records of similar variables ensures a better overview and allows a more convenient graphical connection. For each of these sets of variables separate inputs and outputs have been created. This makes it possible to show, in which objects variables are computed or just used (see Figure 11). The second communication structure is the

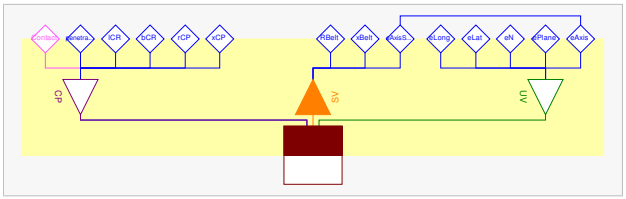

Figure 11: Separation of the *Tire Bus* connector (lower quadratic element) within the *Geometry Class* in an input for *Sensor Values* (SV) and outputs for *Contact Properties* (CP) and *Unit Vectors* (UV). The rhombuses define protected variables used in the model.

connector that contains the velocities as well as the forces and torques that act on the contact point. It

is named *Contact Point Connector*. This connector is implemented in an a-causal manner due to the requirements of this connection. It can be found connecting *Contact Physics*, *Vertical Dynamics* and *Friction*. Both bus connectors are illustrated in Figure 9.

4 Provided Tire Models

All tires are basically built up like the ideal tire in Figure 9. In the library eleven ready-made tires are provided for a quick application and a better understanding of the model structure. Four models of slim tires include three rigid versions and one with a dynamically behaving contact point. Two tires with semi-circular cross section are provided, one rigid and the other having rotational dynamics. The so-called belted tires feature three rigid models differing in frictional behavior. The other two belted tires behave dynamically.

5 Environment

The *Environment* in the current version of the library is limited to even or uneven but slowly changing surfaces, which is a limitation arising from the single contact point model. Basically the method to be implemented has to be able to compute elevation (the *y* coordinate shown in Figure 12) and the normal vector on the surface. There are many significantly different approaches, whereas the one chosen is rather simple but still conveniently configurable.

5.1 Surface Base

The *Surface Base* (Section 5.1) class ensures compatibility with future enhancements. This partial inner model defines the functions necessary to calculate the behavior of the tire. These include the following functions.

- *get_eN_Base* – returning the normal vector of the surface with the actual *x* and *z* coordinates as inputs.
- *get_elevation_Base* – returns the y coordinate of the surface (*x* and *z* are inputs again).
- *get_mu_Base* – returns the frictional coefficient at the *x* and *z* coordinates of the contact point.

5.2 Surface Class

The implemented version of the *Surface* in the *Wheels and Tires* library is based on the method shown in [4]. It is an interpolation in a unit square based on four y values and the eight corresponding partial derivatives. A sketch of the unit square is shown in Figure 12. To enable the user to define more complex

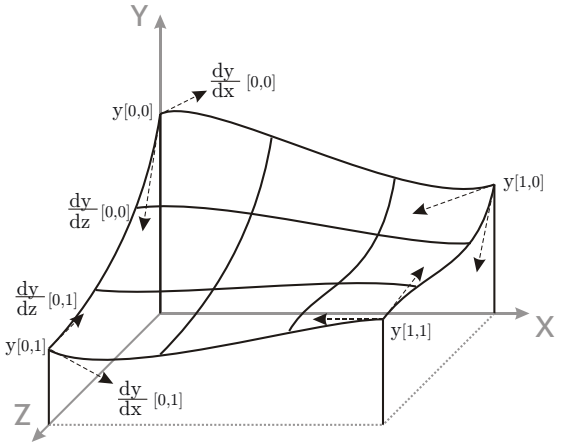

Figure 12: The basis for the interpolation used to find the elevation *y* and normal vector in one square.

surfaces an arbitrary amount of interpolated elements can be combined to one surface of definable size. The only fact limiting the amount of rectangles is simulation time, especially the time consumed for compilation that rises quickly with growing surfaces.

6 Simulation Results

The following sections are intended to give an impression of what the simulation results look like. Therefore, results from the *Test Bench* package and from the *Example* package are depicted.

6.1 Test Bench

Figures 13 to 16 show results from different models in the *Test Bench* package. The models are used to test basic functionalities of the tire and the surface classes.

6.2 Examples

Six examples are included in the library demonstrating the application of the models in vehicles. Five of the Examples are two-wheeled (single-track) vehicles

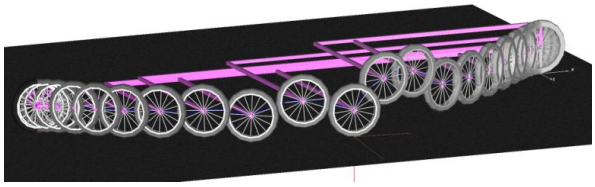

Figure 13: Tire elevating from the ground while driving a curve.

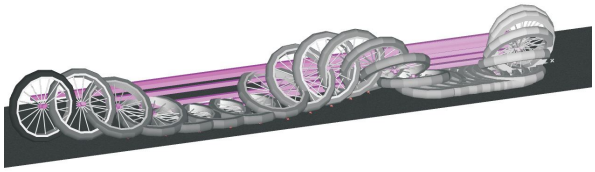

Figure 14: Tire's overturning torque *Test Bench* result.

that are provided by [12] and copied to the *Used Models* in the *Examples* package of the *Wheels and Tires* library. Three of the five examples are unsprung bicycle models composed of rigid elements exclusively. The other two models are sprung motorcycles including front and rear suspension. For both vehicles, models are provided to analyze the uncontrolled stability, e.g. with different tire properties for geometry and friction as shown in Figures 17 and 18.

The other two models are "nice to show" models with the bicycle being accelerated by a strong torque to make the front wheel lift from the ground as shown in Figure 19 and the motorcycle jumping over a gap as depicted in Figure 20.

The last example is a very basic unsprung model of a four-wheeled vehicle with a predefined steering angle profile and driving torques acting on the rear tires. An impulse in the driving torque makes the vehicle-model slide after a certain simulation time, with the

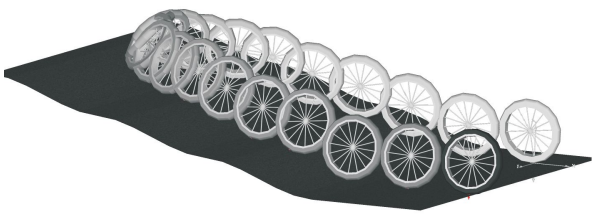

Figure 15: A tire rolling up an uneven surface with an initial lean angle.

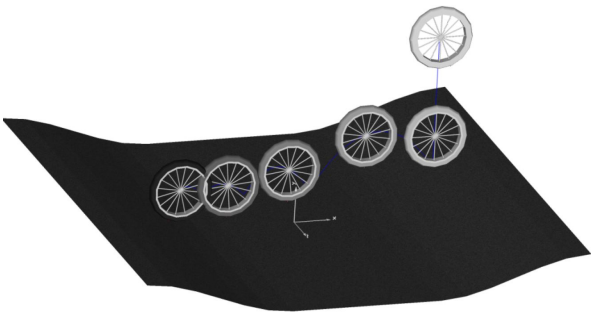

Figure 16: Tire dropping on an uneven surface.

Figure 17: Two similar bicycles with non-slipping tires differing in their geometric properties. The upper bicycle is equipped with ideally slim tires, whereas the lower one driving a narrowing curve has a belted tire with a width of 2cm.

model "reacting" by a full breaking, which makes the wheels slide to a standstill of the vehicle.

Regarding the simulation speeds of the examples it can be stated, that the bicycle models take about half of the simulated time for the computation of the result. The motorcycle jumping over the gap takes about 250s of calculation time for 16s of actual simulation. The reason for that is the more complex structure of the motorcycle in comparison with the bicycle. The again undamped four-wheeled models compute the results in a little less time than the simulation time. All models used non-dynamic tires either with *slipping* or the *advanced* friction models. The simulations have

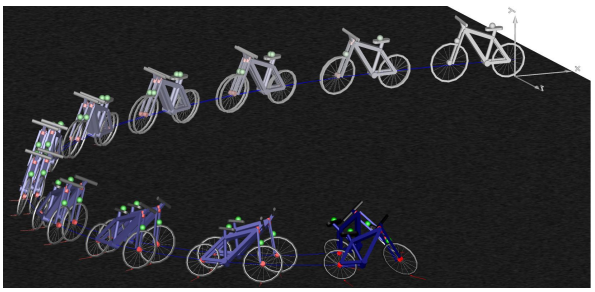

Figure 18: Two similar bicycles with slipping tires differing in their frictional properties. The one driving the inner circle is equipped with the *Advanced Friction* class, whereas the outer bicycle is equipped with the *Dry Fricition with Constant Roll Resistance* class.

Figure 20: A motorcycle jumping over a gap.

Figure 19: A bicycle with non-ideal tires accelerated by a torque on the rear tire, with the front tire lifting from the ground.

been carried out on a Intel Core2Duo clocked at 2GHz and equipped with 4GB RAM, computing 250 output intervals per second. It has to be mentioned that no big effort was spent regarding the optimization of simulation speed of the overall models. A more suitable combination of state variables would most likely lead to a considerable enhancement in speed.

7 Library Structure

This section introduces the top level packages that are contained in the *Wheels and Tires* library and are depicted in Figure 21.

The *Tire Components* package covers the classes that the *Tires* are built of. The contained sub-packages are described in Sections 3.3 to 3.10. These classes form the ready-made tires contained in the *Tires* package as one example shows for the *Ideal Tire* as depicted in Figure 9.

The *Environment* package contains the *Surface Base* as well as a possible implementation for even and uneven surfaces. It is described in Section 5. The *Test Bench* and the *Example* Package are top-level packages as well. They are provided to test the tire models' functionality and get an insight into the usage of the models. A short description can be found in Sections 6.1 and 6.2 respectively. Finally the *Visualization* package gathers a few models used to enable the animation of all necessary parts of the library.

8 Conclusion

To sum up, the library enables a quick and convenient building of easily customizable tire models. There are some further enhancements possible but the structure of the tire models should persist as it is well expandable, fulfilling the major requirements that were demanded at the beginning of the work. Still the lack of comparison to real applications and measurement data is considered to be a drawback as some minor malfunctions could probably not be identified.

For the use of the models in real-time applications a further optimization of the computational efficiency would be desirable to ensure quick enough simulation. Furthermore a prediction of the influences that different objects have on the computational effort of the overall tire would be of advantage.

A more detailed description of the library can be found in the corresponding master's thesis [3].

References

[1] J. Andreasson and J. Jarlmark. Modularised tyre modelling in modelica. In *Proceedings of the Second International Modelica Conference, Oberpfaffenhofen, Germany*, pages 267–274, 2002.

[2] Johan Andreasson. Vehicledynamics library. In *Proceedings of the Third International Modelica Conference, Linköping, Sweden*, pages 11–18, 2003.

[3] Markus Andres. Object-oriented modeling of wheels and tires in dymola/modelica. Master's thesis, Vorarlberg University of Applied Sciences, 2009.

[4] G. Aumann and K. Spitzmüller. *Computerorientierte Geometrie*. BI-Wiss.-Verl, 1993.

[5] Mats Beckmann and Johan Andreasson. Wheel model library for use in vehicle dynamic studies. In *Proceedings of the third Modelica Conference, Linköping, Sweden*, pages 385–392, 2003.

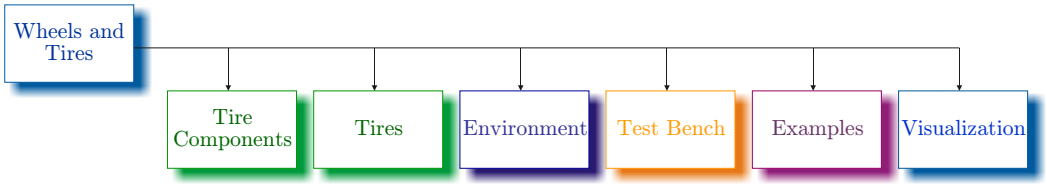

Figure 21: The library's top level packages.

[6] P. Bohara, A. Saha, P. Ghosh, and M. Roopak. Tyre perfomance prediction through fea. Hasetri - Hari Shankar Singhania Elasotmer and Tyre Research Institute, 2008.

[7] F. E. Cellier and À. Netbot. The modelica bond-graph library. In *Proceedings of the 4th International Modelica Conference, Hamburg*, pages 57–65, 2005.

[8] John C. Dixon. *Tires, Suspension and Handling*. Cambridge University Press, 1996. Second Edition.

[9] Günter Leister. *Fahrzeugreifen und Fahrwerkentwicklung*. Vieweg + Teubner, 2009. 1. Auflage.

[10] Hans B. Pacejka. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, 2006. Second Edition.

[11] Georg Rill. *Simulation von Kraftfahrzeugen*. Vieweg-Verlag, 2007. genehmigter Nachdruck.

[12] Thomas Schmitt. Modeling of a motorcycle in dymola/modelica. Master's thesis, Vorarlberg University of Applied Sciences, 2009.

[13] Dirk Zimmer. A modelica library for multibond graphs and its application in 3d-mechanics. Master's thesis, ETH Zürich, 2006.

[14] Dirk Zimmer and Martin Otter. Real-time models for wheels and tires in an object-oriented modeling framework. Accepted for publication in Vehicle Dynamics, 2009.

