S. Drogies, M. Bauer:

Modeling Road Vehicle Dynamics with Modelica.

Modelica Workshop 2000 Proceedings, pp. 161-168.

Modeling Road Vehicle Dynamics with Modelica

Stefan Drogies, Michael Bauer
Institute of Automatic Control
Darmstadt University of Technology
Landgraf-Georg-Straße 4
D-64283 Darmstadt
Germany
E-mail: Sdrogies@iat.tu-darmstadt.de

ABSTRACT

This paper presents a complex vehicle dynamics model which is implemented using the object oriented modeling language Modelica^{TM}. The main focus is on the multi body system (MBS) implementation of the dynamics of the chassis and wheels. The chassis contains models of an independent front suspension with anti-roll bar, as well as of a twist-beam rear axle. Furthermore, a model of the complete powertrain with automatic transmission is included. The resulting model is described in detail and validated against measured data from test drives with a notchback sedan.

Introduction

The development process of modern-time vehicles and its intelligent high-tech components is characterized by the need to minimize the time and the costs from the vision of a new automobile until its series production. Under these conditions car manufacturers and suppliers make more and more use of simulation tools to keep or even improve the current standards of vehicle safety, comfort and quality.

Modern cars can be seen as heterogeneous systems, which are composed of elements from different physical domains. For this reason a general model of a road vehicle may include components from the hydraulic, electronic, control or even thermodynamic domain. Above all, simulating vehicle dynamics requires dealing with 3D mechanics.

The development of multibody codes has made it possible for complicated models of all types of mechanical models to be produced. Tools such as ADAMS [ADA98] or SIMPACK [Kor98] are highly specialized on the simulation of multibody systems, however they lack the multi-domain capabilities: the integration of non-mechanical systems is not straightforward.

On the other hand, purpose-built codes like CARSIM [Say95] or Matlab^{TM}- and Fortran-based models usually cover most of the important aspects of a vehicle model. Unfortunately, changes of the model structure, especially of the mechanic parts, are often difficult to achieve.

For this reason we consider the use of Modelica^{TM} [Fri98], a general object-oriented language for modeling and simulation of physical systems.

The language unifies and generalizes previous object-oriented modeling languages. Compared to other commonly used simulation languages Modelica^{TM} offers three important advances: It is built on non-causal modeling with true ordinary differential and algebraic equation. Furthermore, it offers a general type system that unifies object-orientation, multiple inheritance and templates within a single class construct. Besides that, Modelica^{TM} is designed to model and to simulate systems consisting of components from different domains such as electrical circuits, drive trains, multibody systems, thermodynamical and hydraulical systems.

As requirements for the proposed vehicle dynamics model the following guidelines were defined. The model should be easy to modify and extend, intuitive to use, it should have a modular structure and contain varying levels of detail according to different simulation tasks.

Furthermore, the model is intended to have sufficient complexity to replicate the braking and handling responses of automotive vehicles through the full range of non-crash conditions.

Figure 1 shows the structure of the vehicle model with its main components powertrain including the combustion engine, car body, front and rear axle, steering, brake system, wheels and ground. Model inputs are the throttle angle, the brake pressure and the steering angle.

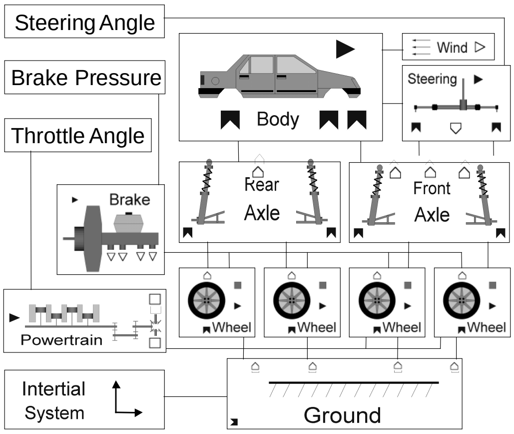

Figure 1: Top level structure of the vehicle dynamics model

With regard to the mechanical part of the vehicle it is useful to distinguish between the kinematics of the chassis, the body and the steering and the dynamics of the powertrain.

With its specialised MBS and rotational (former drivetrain) library, Modelica^{TM} favours this classifying. The wheels as an interface between the rotation of the powertrain and movement of the body are attached to both systems.

Mass	1555 kg
Wheelbase	2,6 m
Wheel track	1,4 m
Max. power	125 kW
Max. torque	227 Nm
Transmission	automatic 4-speed planetary gear with Trilok converter
Front wheel suspension	McPherson struts with A-arms and stabilizer
Rear wheel suspension	twist-beam axle
Steering	rack and pinion steering
Brake	conventional hydraulic brake with disks all around

Table 1: Technical data of the test vehicle

In the following sections the model structure is explained in detail, some annotations with regard to its realization in Modelica^{TM} are made and the resulting simulation model is validated against measured data from test drives with a notchback sedan. The technical data of the vehicle is outlined in Table 1.

Powertrain *(Powertrain)*

The class *Powertrain* contains instances of Modelica^{TM} classes representing the components of the drivetrain: combustion engine, torque converter, automatic gearbox, differential, appropriate inertias and the automatic transmission control unit (ATCU) which generates the gearshift signals. Figure 2 shows the according Modelica^{TM} component. Mechanical flanges are characterized by small square boxes, whereas signal connectors are indicated by small triangles.

In the following sections all components which are not part of the Modelica^{TM} standard library are described in more detail.

Combustion Engine (*EngineTorque*)

The characteristics of the combustion engine are described by a static neural network, which approximates the engine torque in dependency of the actual throttle angle and the engine speed. A three layer multi-layer perception network was trained using the engine data measured by an engine dynamometer test cell. In order to take the effect of ageing and wear into account, the resulting engine torque is multiplied by a throttle dependent factor.

The inertia of the engine and of the torque converter pump are taken into account in the shaft element *InertiaEnginePump*.

Torque Converter (*TorqueConverter*)

The trilok torque converter is a fluid coupling device used in automatic transmissions for its damping characteristics and torque multiplication characteristics. Torque is transferred to the turbine as a result of oil flow induced by the pump which is connected to the engine shaft. The torque converter is assumed to be always operating at its capacity and therefore can be adequately modeled by its steady state characteristics [För90]. Given the turbine speed ω_T and the pump speed ω_P, Euler's turbine equation yields

$$T_T = k_P(\omega_P, \omega_T)\cdot D_P^5 \cdot \omega_P^2 \tag{1}$$

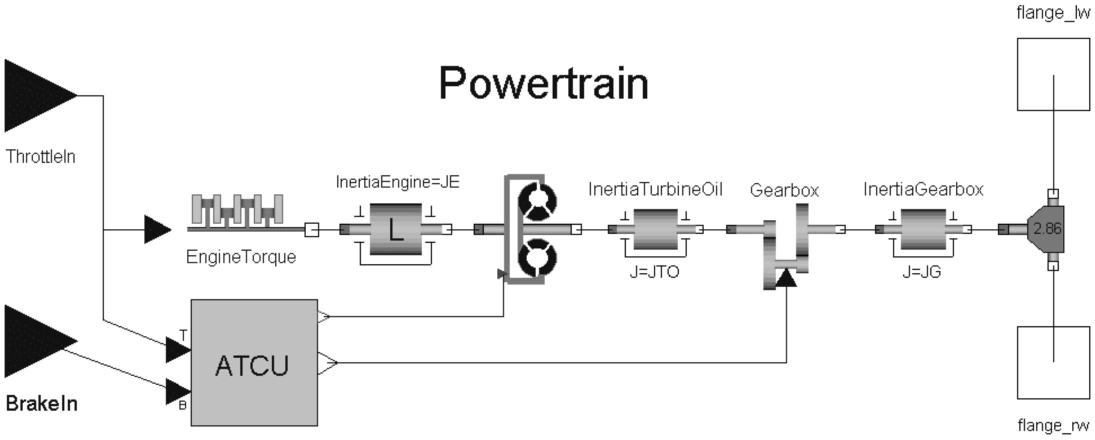

Figure 2: The class Powertrain

where k_P is a characteristic parameter of the converter that depends on the ratio between pump and turbine speed, and D_P denotes the characteristic diameter of the converter. From the resulting pump torque T_P, the turbine torque T_T is determined using

$$T_T = -\mu(\omega_P, \omega_T) \cdot T_P \tag{2}$$

where μ is the torque ratio of the converter provided by the manufacturer. The inertia of the turbine and the converter oil are taken into account in the shaft element *InertiaTurbineOil*.

The torque converter clutch (TCC) can bypass the torque converter in the fluid coupling mode of operation to provide a direct mechanical link between the engine and the drivetrain. Thus, power losses are reduced. The TCC is controlled by the ATCU and is modeled using the Modelica$^{\text{TM}}$ library class *FrictionBase*.

Automatic Gearbox (*Gearbox*)

The transmission consists of a Ravigneaux design compound planetary gear train, which provides four forward ratios. The used model assumes an ideal gearbox with fixed gear ratios. As gear shifting occurs within a finite time duration, not instantly, speed transitions are presented by low-pass filtering the gear ratio commanded by the ATCU. Furthermore, an additional delay time is applied to the gear shift as well as to the TCC signal from the ATCU, in order to better fit the dynamic behaviour of the automatic transmission. The inertia of the gearbox is taken into account in the shaft element *InertiaGearbox*.

Automatic Transmission Control Unit (*GearControlUnit*)

The ATCU determines the gear shifting operation based on the knowledge of the current gear, the throttle input, the vehicle's velocity and the status of the torque converter clutch. These inputs are applied to a shift logic map which calculates the gear ratio and the TCC signal for the automatic gearbox.

Differential (*Differential*)

An ideal conventional differential without inertia was considered according to

$$\begin{aligned} 2 \cdot \omega_{Gear} &= n_{Diff} \cdot (\omega_L + \omega_R) \\ T_L &= T_R \\ T_{Gear} \cdot n_{Diff} &= -(T_L + T_R) \end{aligned} \tag{3}$$

where n_{Diff} is the gear ratio of the differential, T_L and T_R are the torque acting on the left and right wheel shaft and ω_L and ω_R are the associated shaft speeds.

Brake System *(BrakeSystem)*

Due to the shift in dynamic forces from rear to front that accompanies vehicle braking, the braking torque applied to the front must be greater than that of the rear. Thus, a brake-pressure limiter for the rear brakes is often used. In the class *BrakeSystem* this fact is taken into account by setting the brake power distribution ratio to 70% / 30% (front / rear). Moreover, the dynamics of the hydraulic pressure build up and pressure reduction is assumed to be of first order low pass with different time constants for pressure build up an pressure drop. The brake itself is modeled in the class *Wheel*.

Multibody System

The classes *Body*, *FrontAxle*, *RearAxle*, *Steering*, *Ground* and parts of the class *Wheel* are modeled with the Modelica™ MBS library, because its use of the recursive Euler algorithm allows fast and efficient modeling with a high grade of failure immunity [Wol00]. In the following section some general remarks are made with regard to the realization of the classes in Modelica™, thereafter a more detailed description of each class follows.

Aspects of the Implementation

Many MBS vehicle simulations use only one connection from the inertial system to the car, e.g. Adamski in [Ada99]. This requires that the centers of roll and pitch are known. However, when testing constructive changes of the wheel suspension for example, this is a great inconvenience.

In this case, the vehicle's degrees of freedom (DOF) might be unintentionally decreased. In order to prevent this possibility, in our approach each wheel is connected to the ground. Without wheel suspension, this would lead to three kinematic loops.

Complex axles increase the number of loops. For example, the twist beam of the rear axle leads to an additional loop, and so does the stabilizer. Including the steering, the number of kinematic loops of the MBS is 13.

A rigid car body is assumed, and therefore any elasticities of its connections to the axle support are neglected.

Figure 3 gives an overview of the MBS, its kinematic loops and the cut elements which are used to break the kinematic loops.

The cut-components which help breaking the kinematic loops are distributed symmetrically in the chassis elements, except for the steering. Thus, the replacement of any wheel's suspension is easy to realize and a user of the model does not need to bother with kinematic loops.

All masses of an vehicle can be separated in sprung and unsprung masses. [Say95] The body and parts of the axles belong to the sprung mass. The unsprung masses are the wheels and the steered parts of the suspension. Note, that the term "unsprung" is not exactly correct because those parts are sprung by the tires yet. All sprung masses may be represented by one mass component, whereas the unsprung masses should be considered more carefully. According to the available vehicle data all masses of the vehicle are modeled using elements of the Modelica™ MBS library (e.g. *boxBodies*, *cylinderBodies* and *Masses*).

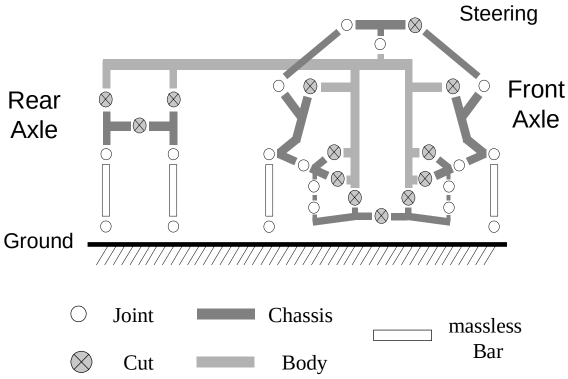

Figure 3: Overview over the MBS model

The braking of the kinematic loops is done by a six-dimensional cut spring, which allows to assign certain stiffness and damping to each of the DOF, thus permitting an exact modeling of the elasto-kinematic suspension bearings. Moreover, the stiffness of the twist-beam and the stabilizer is realized by using these cut-springs.

The non-cut bearings are modeled with *Prismatic* and *Revolute Joints* of the Modelica™ MBS library.

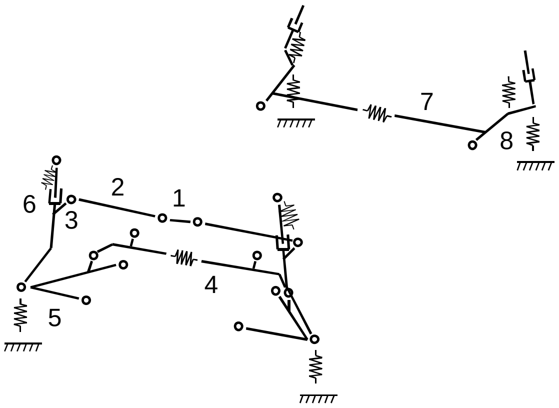

Figure 4: MBS of the test vehicle

Front Suspension *(FrontAxle)*

As shown in Figure 4 the wheels are suspended by McPherson struts (6) and A-arms (5). The A-arms of both sides are connected through an anti-roll bar (4) which is also supported by the body.

A McPherson strut differs from a common suspension strut by its ability to compensate transversal forces. This behavior is represented by a *Cylindrical Joint* with a parallel connected *Damper*, whose characteristic line is given by a *CombiTable1D*.

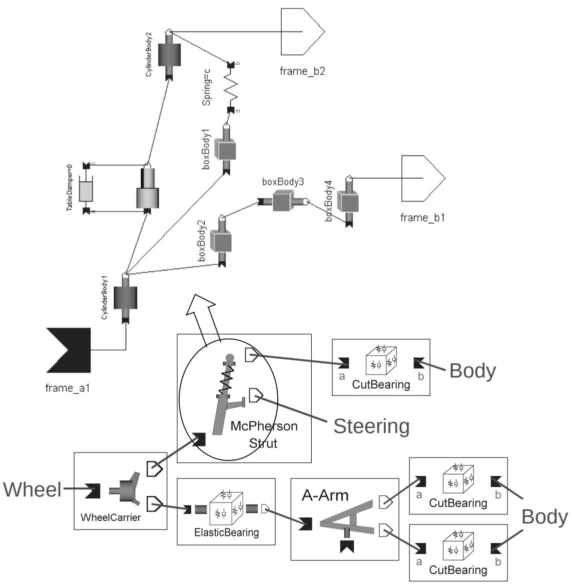

Figure 5: Model of the McPherson strut

Because the spring is inclined against the damper, it is attached in our model to the end of two bars as shown in Figure 5. Additionally, the integration of this strut into the front axle is shown there.

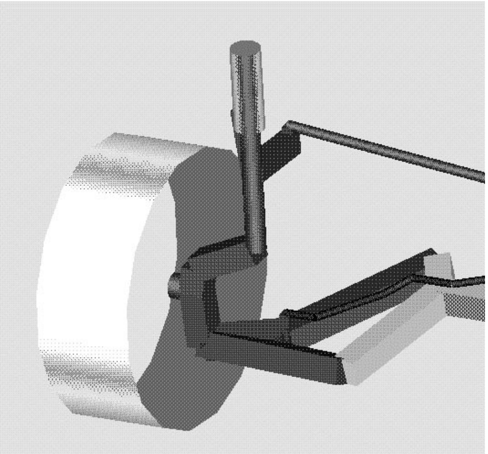

Figure 6: 3D sight of the McPherson strut

The A-arms were modeled as solid body. All bearings of the axle are realized as cut springs or classes with sprung and damped *Prismatic* and *Revolute Joints* for all six DOF . The stabilizer also got a central cut spring to reproduce the behavior of a flexible bar.

Rear Suspension *(RearAxle)*

The suspension of the rear wheels is realized as twist-beam (7) axle with longitudinal rods (8).

The twist beam of the axle is composed of two *boxBodies* which are connected by a cut spring. The stiffness around its twist-axis is set to low values, while to the other five DOF high values are assigned. For making a reciprocal deflection possible the bearings of the longitudinal links are elastic around each axis. Again springs and shock absorbers are taken from the Modelica™ MBS library.

Steering *(Steering)*

The front wheels are controlled by a rack and pinion steering. Turning of the steering wheel is resulting in displacement of the steering rack (1) which moves the steering link (2) and the steering knuckle arm (3). The effect of the power steering of the test cars is neglected, because the main focus is on the vehicle dynamics itself and not on the driving comfort. For this reason we consider only the measured steering angle and not the required torque at the steering wheel.

The modeling of the steering is carried out according to the same technique already applied to the suspension elements.

A major difference is the use of an *IdealGearR2T* class from the Modelica™ rotational sublibrary as steering gear. The DOF of the gear rack is given by a *Prismatic Joint* that can be sprung and damped for fitting stiffness and friction of the steering to measured values.

Wheel *(Wheel)*

Since the wheels transmit the drive-, brake- and side forces of the car, its description is most important for each simulation model [Kie00].

The wheel has two main tasks. First, it connects the vehicle to the ground and therefore it needs four DOF: Two *Revolute Joints* at its base, which allow that the wheel can camber and turn around its vertical axis. Furthermore, a

Prismatic Joint which enables the compression of the tire spring and the lifting of the wheel and finally a third *Revolute Joint* at the wheel center which describes its bearing.

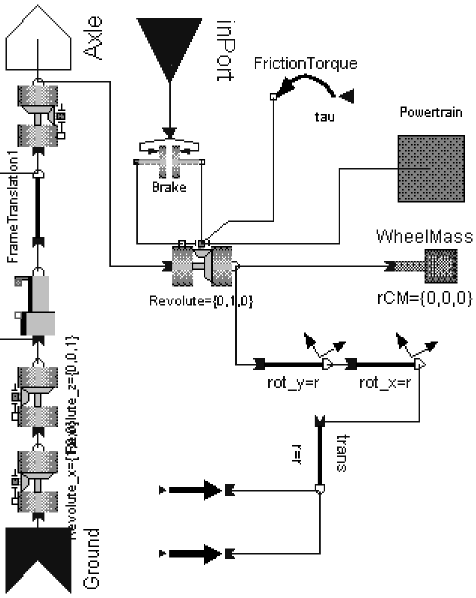

Figure 7: The class *Wheel*

The second task is to compute the angular velocity of the wheel. For this reason it has a *Revolute Joint* which is connected to the chassis, a *Mass* with inertia and a *FrameTranslation* pointing at the center of tire contact, where the tractive and side forces are acting on. In order to force this point to rest at its ground position, two *FrameRotations* compensate camber and rotation.

The brake functionality is modeled by the *Clutch* element from the Modelica^{TM} rotational library which is connected to the bearing and the axis flange of the *Revolute Joint*. Thus, a support of the braking torque at the wheel suspension can be realized. Input to the brake is the pressure of the brake fluid.

The wheel resistance force is applied as torque to the wheel using the axis flange of the *Revolute Joint*. Optionally, the drivetrain torque can act on the same flange. Note, that no distinction between driven or undriven wheel has to be made.

The angular velocity of the wheel is measured and compared with the translational speed of the wheel center. From this results slip and slip angle are computed. Using the force acting vertically at the ground contact point, the wheel friction is calculated according to Burkhardt [Bur93]. From this values the tractive forces are computed.

Ground *(Ground)*

This class serves as an interface between the inertial system and the wheels. Three *Prismatic Joints* are assigned to each wheel. Two joints allow its plain movement, whereas the third joint simulates vertical road irregularities. These road irregularities can be given by measured tables, as any analytical function or by shaping filters.

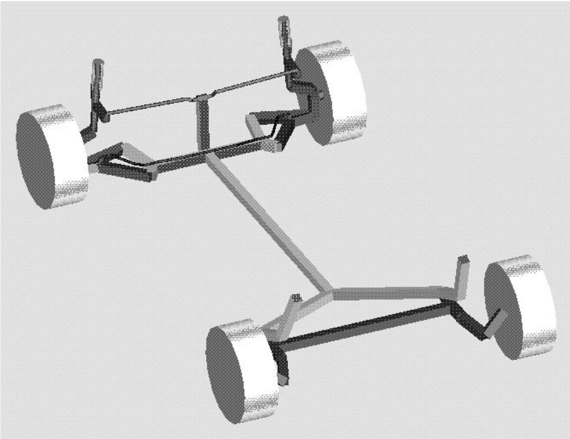

Figure 8: 3D sight of the MBS model

Examplary Results of Experimental Verification

The model was verified using the test vehicle described by table 1. Two experiments are presented. A lane changing maneuver at a vehicle speed of 14 m/s (experiment A) and a braking and accelerating maneuver (experiment B). The results given below (Figures 9-12) show the precision of the model with regard to the longitudinal dynamic of the car (experiment B).

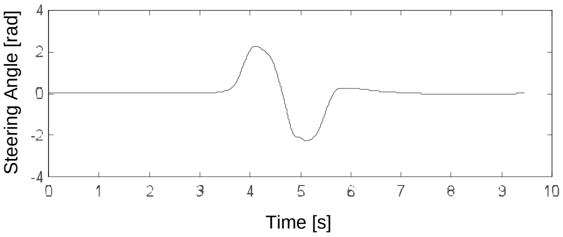

Figure 9: Input of experiment A

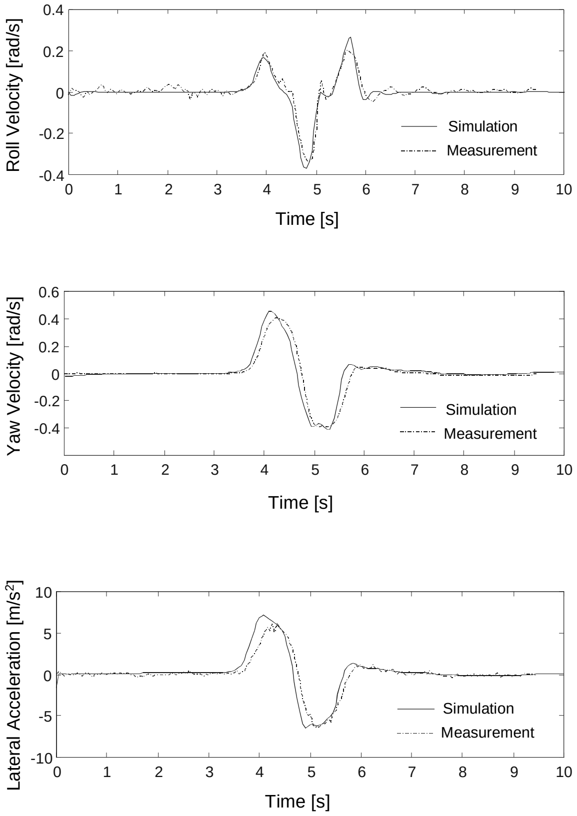

Figure 10: Results of experiment A

However, the lateral dynamics of the test vehicle (experiment A) are not as good reproduced as the longitudinal dynamics. A reason for this may rely on the fact that the parameter provided by the manufacturer of the rear axle were not complete and therefore some assumptions had to be made.

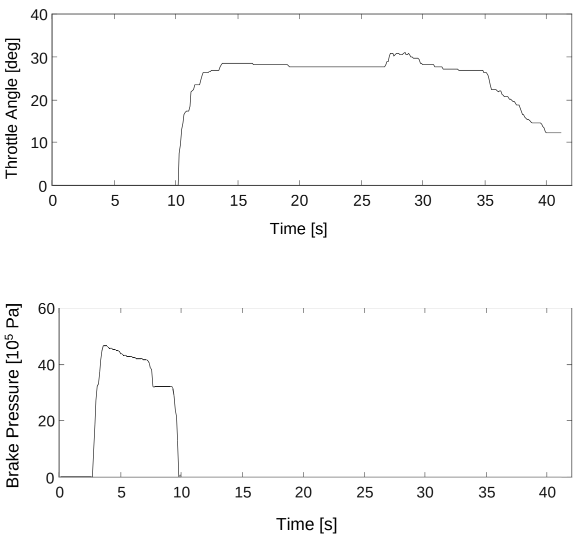

Figure 11: Inputs of experiment B

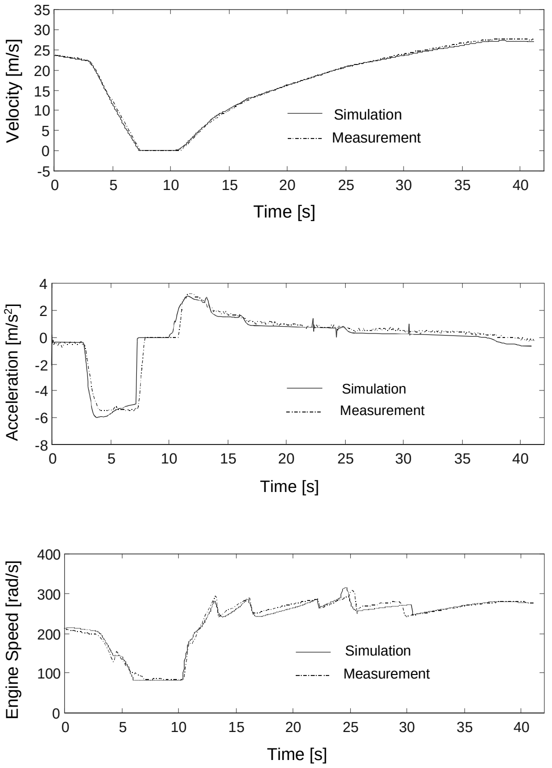

Figure 12: Results of experiment B

Conclusions and Outlook

A model of a road vehicle was presented which was modeled in Modelica^{TM}. The intention was to simulate a complete car using only a single software tool. The model should be easy to modify and extend, intuitive to use and should have a modular structure. The model showed to have sufficient complexity to replicate the braking and handling responses of a test car.

Nevertheless, some parameters still need to be fitted for a better model match.

Unfortunately the computational demand of the model is rather high, so e.g an application to. hardware-in-the-loop simulation is yet not possible. Simplifications especially of the MBS will be tested to improve this, e.g. by reducing the number of kinematic loops. This is facilitated by the ability to change any class without the need to restructure the whole model.

Due to the complexity of the model a very high number of parameters is used for its description. For this reason the organization of the parameter records should be improved, which would also provide an elegant possibility to manage a greater number of different car types.

REFERENCES

ADAMS: *Mechanical Dynamics Engineering Software*, Product Overview 1998. Mechanical Dynamics Inc., Ann Arbor, USA.

Adamski, D., Bardini, R., Bertram, T., Hörsken, C., Lange, O., Roll, U., Torlo, M., and Ward., D. 1999. "Vehicle Modeling and Simulation in the Duisburg Mechatronics Laboratory." *In A. Kecskeméthy, M. Schneider, and C. Woernle, editors, Advances in Multibody Systems and Mechatronics, pages 65-88, Institut für Mechanik und Getriebelehre,* Graz, Austria.

Burkhardt, M. 1993. *"Radschlupf-Regelsysteme."* (in German) Vogel Verlag, Köln, Germany.

Förster, H.-J., 1990, *"Automatische Fahrzeuggetriebe"* (in German), Springer Verlag, Berlin, Germany.

Fritzson, P. and V. Engelson. 1998. "Modelica – A Unified Object-Oriented Language for System Modeling and Simulation." In *Proceedings of the 12th European Conference on Object-Oriented Programming, ECOOP'98,* Brussels, Belgium.

Kiencke, U. and Nielsen, L. 2000. *"Automotive Control Systems "* Springer Verlag, Berlin, Germany.

Kortüm, W., W. Rulka, M. Spieck, 1998. *"Simulation of Mechatronic Vehicles with MBS-code SIMPACK."* Multibody Dynamics: New Techniques and Applications, London, UK.

Sayers, M. and D. Han, 1995. "*A Generic Multibody Vehicle Model for Simulating Handling and Braking*". Proc. 1995 IAVSD Symposium, Ann Arbor, USA.

Wolf, A. and Kayser, A. 2000."Component Orientated Modelling of Electrical and Mechanical Systems as Required by Component Simulation Software." In *1st IFAC-Conference on Mechatronic Systems Preprints Volume II*, Darmstadt, Germany.