Generating Multibody Real-Time Models for Hardware-in-the-Loop Applications

Alexander Eichberger
INTEC GmbH

Argelsrieder Feld 13
D-82234 Wessling, Germany
Phone: +49-(0)8153-9288-0
Fax: +49-(0)8153-9288-11
E-mail: alex.eichberger@simpack.de

The paper presents an approach on how to generate multibody real-time models for the development and testing of electronical control units in hardware-in-the-loop environments. The state of the art of generating real-time models is described together with an process gap from existing off-line to on-line simulation models. SIMPACK code export functionality is introduced to close the gap and to export multibody vehicle models to real-time applications taking existing off-line models as input. A non-linear model reduction process and the export facility will be applied to a full multibody vehicle example resulting in a simulation time five times faster than real-time.

Keywords/ Hardware-in-the-loop, Model reduction, Multibody, On-line model, Real-time, Simulation

1. INTRODUCTION

Nearly in every design domain of new and modern vehicles like cars, motorcycles and trucks, simulation is of increasing importance. By the means of virtual prototypes, i.e. prototypes in software on the computer, development costs and new product development cycle times can be reduced drastically. The multibody system (MBS) simulation approach is a well known method to investigate the dynamical behaviour of new or further improved vehicle designs.

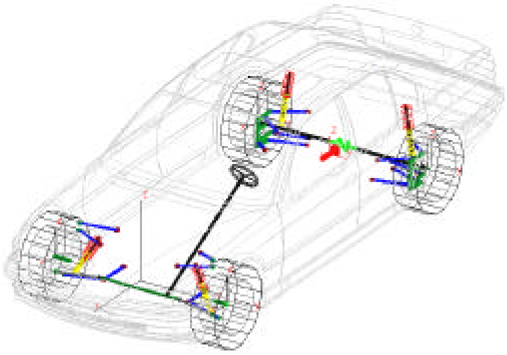

Fig. 1 MBS-model of a car

Basically a MBS-model (Fig. 1) is made up of bodies, that are linked together by idealised joints which impose reaction forces to the system and finally force elements, e.g. for shock absorbers and tyres. Traditional fields of application of functional virtual prototyping using MBS virtual models are suspension kinematics and compliant kinematics, handling performance, ride comfort, noise vibration and harshness, and durability. This CAE-processes are nearly seamlessly integrated into nowadays development and design of new vehicles (Fig. 3).

Real-time simulation is where virtual world (e.g. virtual car) and real world (e.g. electronic control units) meet each other. Typical application fields are the test of electronic control units or driving simulators with man-in-the-loop (MIL). At this point a seamless integration in the development cycle is still missing. Thus the lack of efficient real-time simulation is a considerable source where time delays and development costs are being generated.

Modern transportation vehicles assist the driver or operator with an increasing number of electronic control units (ECU). Well known are ABS (Anti-Blocking-System), ESP (Electronic-Stability-Program), ASC (Anti-Slip-Control) and many more. The ECUs more and more determine the functional behaviour of the vehicle taking the driver reactions, sensor signals and environmental information as input. Often the ECU is already existing as real world hardware, but the forthcoming or gradually further developed vehicle generation is not available in hardware. It exists as a so called functional virtual prototype (FVP) representing the mechanical dynamical behaviour and the actuators and sensors of the vehicle. To test the function of the complete vehicle, the simulation model has to be interfaced to the hardware of the ECU which leads to a hardware-in-the-loop simulation (HIL) or alternatively to a software-in-the-loop simulation (SIL) if only the software code of the ECU exists (Fig. 2). In both cases the FVP has to be processed in real-time.

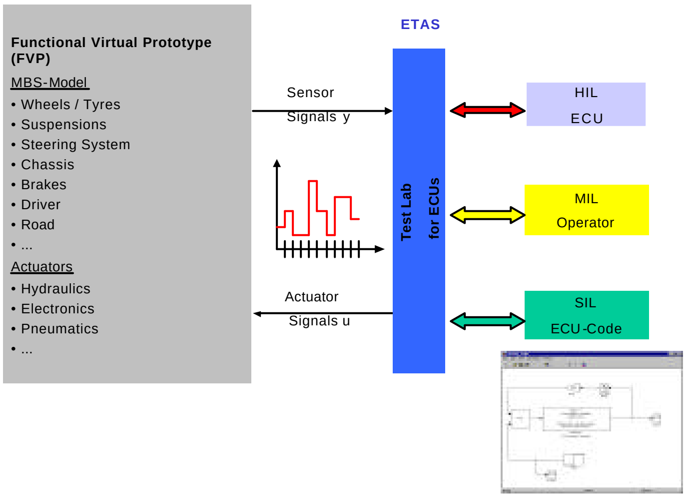

Fig. 2 Interface FVP to Hardware of Test Bed

2. STATE OF THE ART AND OPEN DEMANDS

To test electronic control units as shown in Fig. 2, the FVP is linked by sensor and actuator signals to the test environment. The dynamical behaviour of the virtual car is described by the equations of motion for the multibody model and the actuator and sensor models:

$$\dot{x} = f(x,u,t) \qquad (1)$$

The motion equations are accompanied by a set of measurement equations:

$$y = g(x,u,t) \qquad (2)$$

In Eqs. 1 and 2 x describes the state of the system, t is the system time, u is the actuator input to the simulation model and y is the sensor output. Real-time means that the equations of motion of the FVP have to be generated and numerically solved fast enough and in a fixed time frame. That means within one cycle at least one function evaluation of Eq. 1 and Eq. 2 and one numerical integration step of Eq. 1. Typical sample rates for the mechanical part are 1-2 ms, for the actuator and sensor part smaller sample rates may be necessary.

The state of the art in nowadays HIL and SIL application for numerical solvers are explicit ore semi-explicit fixed step size and fixed order methods. In fact mostly a simple one step Euler method is applied. This implies that the dynamics of the MBS-model must be described by a set of non-stiff ordinary differential equations (ODE`s). But when dealing e.g. with suspension and steering systems kinematic closed loops and/or bushings lead to stiff differential algebraic equations (DAE`s). To overcome this situation the following approaches are often used today:

- highly specialised in-house codes,
- hand-coded MBS of every new virtual prototype with simplyfied models for suspensions, steering system and chassis,
- multi-dimensional look-up tables to represent the kinematics and linear elasto kinematics,
- solver adaptation and tuning to topology and structure of each individual virtual prototype,
- co-simulation of different mechanical subsystems ,
- mixture of above methods.

Real-time models today still are a very simplified version of a functional virtual prototype. Common to the approaches above is that they lead to a high implementation and adaptation effort whenever a changing system design comes up. There exists only a very low compatibility with already existing FVP from other application areeas such as NVH, ride and durability not suited for HIL or SIL. There is a gap in the process-line from existing off-line models to the corresponding real-time model. Often a new parameterisation and data acquisition has to be done. An open issue is also to account for the effects of flexible bodies and to regard the non-linear behaviour of bushings. Specialised in-house approaches prevent a black-box model exchange between OEM and suppliers.

The open demands for a future approach to generate real-time models can be summarised to be:

- automatic generation of real-time models from existing complex FVPs using the existing parameterisation, data pool, and substructuring,
- identification environment for finding the parameters of a reduced real-time model,
- representation of non-linear kinematics and compliant kinematics of the suspension systems without using look-up tables,
- account for flexible body effects,
- real-time solvers for stiff systems and DAE`s,
- co-simulation to support a multi-processor approach,

- library with different levels of detail for e.g. subsystems like suspensions,
- black-box model exchange between OEM and supplier.

3. SIMPACK CODE EXPORT

The Code Export module, a forthcoming new member of the SIMPACK software family allows to automatically generate the symbolical form of the equations of motion from any existing MBS model thus ensuring a seamless transition from the off-line world to the real-time world (Fig. 3). Eqs. 1 and 2 are processed as follows:

STEP 1 (once): load model and transform data to order n algorithm specific co-ordinate system [1].

STEP 2 (once):

- generate algorithm for Eqs. 1 and 2 of an individual MBS in symbolical form [2],
- eliminate all if, else, do, … operations,
- perform partial linearisations,
- perform model reduction techniques,
- recursively check code for non-used statements,
- adapt dimensions to problem size,
- resolve all SIMPACK dependencies,
- export code to external simulation environment.

The equations are written out as Fortran which can be converted to C-statements and are ready to be used in different simulation environments. The exported equations are automatically tuned to the individual model and are therefore two to three times faster compared to the SIMPACK internal numerical code [3]. To further reduce simulation time model reduction techniques can be applied and performance, accuracy and stability of the real-time model can be assessed in the SIMPACK environment.

The exported model can include user written code or e.g. control code from the Matlab Real-time workshop, it can contain flexible body data and still can be parameterised to a reasonable extent. The code export closes the gap from existing off-line applications and models to real-time simulation (Fig. 3).

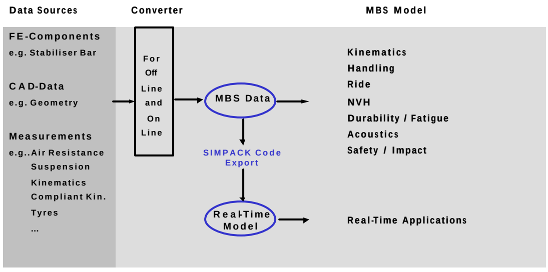

Fig. 3 From Off-line to Real-Time Model

Application fields of code export are the automatic generation of real-time models for development and testing of electronic control units, the generation of MATLAB/Simulink models, and the exchange of capsulated simulation models.

4. APPLICATION EXAMPLE

To show the potential of the new module a real-time model will be derived from a complex functional virtual prototype of a full vehicle model.

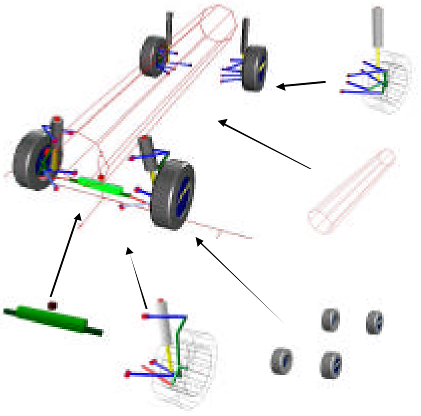

Fig.4 Full Vehicle Model

The vehicle model (Fig. 4) consists of the substructures rear axle with full compliance, front axle with full compliance, chassis, steering system and a handling tyre model. The dynamics of the car is described by 257 first order differential equations of type Eq. 1. The model consists of 59 bodies and 60 force elements. For a sinusoidal steering manoeuvre the real-time factor with a stiff ODE-solver was 1:5, i.e. for 1 second real-time 5 second CPU-time were required. The objective for this model was to achieve a real-time factor of 1:1 on a 800 MHz CPU by applying the code export facility together with a model reduction process to the existing off-line model.

The basic idea of the model reduction will be shown for the rear suspension which consists of five links attached to the wheel carrier and to the chassis by bushings.

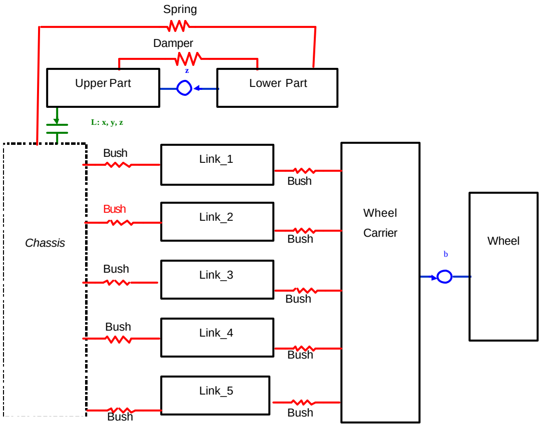

Fig. 5 Full Model of Rear Suspension

Fig. 5 shows the topology of the rear suspension. High frequencies are induced in the model due to the high stiffness values of the bushings together with the low masses of the links. In a ride or NVH simulation where the model comes from, those frequencies might be of interest. But in a handling simulation for ECU testing those frequencies are not important and only the non-linear compliant kinematics between wheel carrier and chassis are relevant. In the model reduction process of step 1 of chapter 3 the masses of the links have been neglected and quasi-static substitutions have been generated, resulting in a simplified model of the rear suspension as it is shown in Fig. 6.

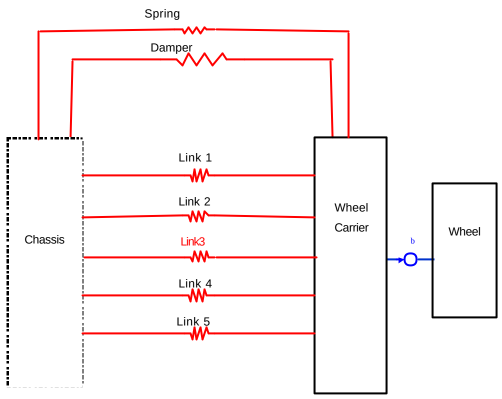

Fig. 6 Reduced Model of Rear Suspension

By applying the same reduction process to the front suspension resulted in a handling simulation model with a reduced number of 30 bodies, 54 force elements and 64 first order differential Eq. 1. For characteristic manoeuvres the simulation results of the full vehicle model and the reduced model have been compared. It turned out that in the frequency range of handling simulations the models showed identical results:

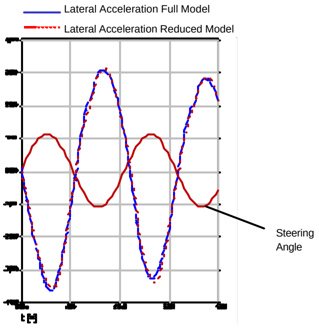

Fig. 7 Full Model versus Reduced Model

Fig. 7 shows the lateral acceleration due to the sinusoidal steering input of the full vehicle model and the reduced model. As the high frequencies could be eliminated by the reduction process an explicit Euler integration scheme with a stepsize of 1.5 ms could be applied for time integration. With the reduced model for 1 second real-time 0.5 second CPU-time could be achieved resulting in a real-time factor of 1:0.5.

5. CONCLUSION

An approach has been presented to automatically generate real-time simulation models from existing non real-time MBS models. A code export facility implemented in a commercial multibody simulation package enables to close the gap from existing off-line simulation models to on-line real-time models. The underlying symbolical equations of motion were simplified by model reduction techniques thus avoiding high frequencies in handling models but keeping the non-linear compliant kinematic effects. The code export has been applied to an existing MBS-model of a full car. The resulting simulation times could have been reduced by a factor of 10.

REFERENCES

[1] Eichberger, A. “A Transputer based Multibody System Dynamic Simulation, Part I: The Residual Algorithm – a Modified Inverse Dynamic Formulation”, Mechanics of Structures and Machines, 22(2):211-237, 1994.

[2] Schlenger, M Schäfer, B., Rulka, W., Fonsenca, R. “Realtime Simulation of Elastic Space Manipulators by MBS-Software”, 3rd eioots, estec June 1993, Noordwijk, NL

[3] Rulka, W. “SIMPACK – A Computer Programm for Simulation of Large-motion Multibody Systems”, in Schiehlen, W (Editor): Multibody Systems Handbook, (1990) Page 265-284.