# Real-Time Simulation of Elasto-kinematic Multi-body Vehicle Models

Matthijs Klomp
*Vehicle Dynamics, Volvo Car Group*

Peter Sundström
*Modelon AB*

Albin Johnsson
*Vehicle Dynamics, Volvo Car Group*

ABSTRACT: This paper presents the development of a real-time capable high fidelity multi-body vehicle dynamics model based on differential algebraic equations (DAE) that is compiled into ordinary differential equations (ODE) form using index reduction, symbolic manipulation and equation sorting. This model is shown to closely match an offline high fidelity model based on high index solvers. The real-time model directly references the model parameters used by the offline model and can be distributed in executable form according to standard interfaces. The real-time performance is achieved by applying inlining of an implicit Euler solver into the model formulation to get stable execution with fixed time steps, and by parallelizing the execution over multiple processor cores. The presented approach gives a more accurate and configurable real-time model compared to the current, lookup-table based, solution.

## 1 INTRODUCTION

Development of virtual methods to improve the efficiency of the development by reducing cost and lead-time while at the same time improving quality and robustness is of strategic relevance and importance to the vehicle industry [1].

With the increase of mechatronic chassis systems, testing of these systems using hardware-in-the-loop requires real-time simulation models. Furthermore, the introduction of high-fidelity dynamic simulators has proven useful for testing and assessment of the human-machine interaction [2]. Nevertheless, with the increasing demands for virtual testing, currently used medium fidelity models are no longer sufficient to capture higher-order phenomena such as brake judder, wheel imbalance, choppiness, etc. In [3] the conclusion was that current real-time simulation models are valid to approximately 5 Hz, requiring higher order models for trustworthy evaluation of advanced driver support systems. Also component characteristics, such as suspension hard-point locations are usually not directly available in simplified vehicle models that use look-up table representations of the suspension.

This paper presents the development of a real-time high fidelity vehicle dynamics model developed with Dymola and the Vehicle Dynamics Library (VDL) that closely matches a model based on Adams/Car that is currently used for high-fidelity off-line simulations. The real-time model directly references the model parameters used for the off-line model and can be distributed as executable models using the functional-mockup interface (FMI) standard [4], [5].

## 2 BUILDING AND PARAMETERIZING MODELS

The modeling and parameterization scheme is based on the model topology being defined independent of the parameter data. In the off-line model, the topology with bodies and joints and/or bushings is stored in templates, and the data that parameterizes the model is stored in a parameter database. Thus, the equivalent topology is built for the real-time model which then reads data directly from off-line model data files.

### 2.1 *Topology definition*

The real-time model is based on Modelica, an object oriented language for developing models based on differential algebraic equations (DAEs) [6]. When the model is compiled to executable code, the models and the connections between them are transformed into one system of DAEs that goes through symbolic manipulation, index reduction and equation sorting to generate a formulation that can be simulated using ordinary differential equation (ODE) solvers.

The chassis used in this work has a double wishbone suspension at the front and an integral rear link suspension at the rear with a transverse leaf spring. Figure 1 shows the front linkage topology viewed in Dymola.

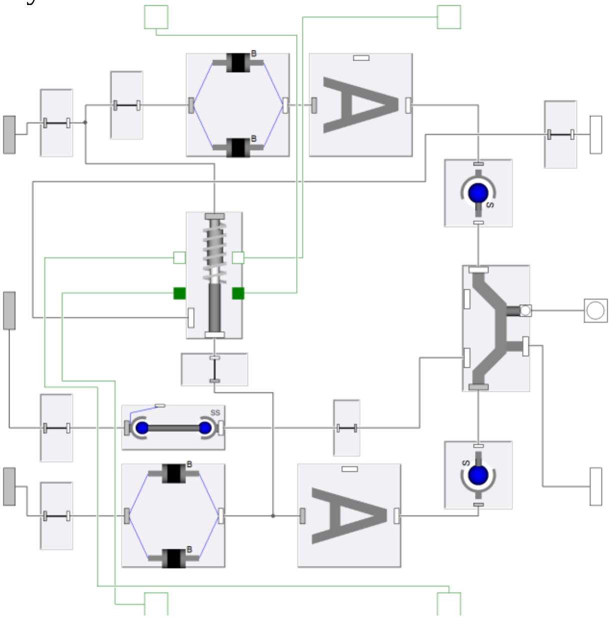

Figure 1. Front linkage layout.

The front suspension is based on existing templates in VDL while the rear suspension is a custom topology, shown in Figure 2.

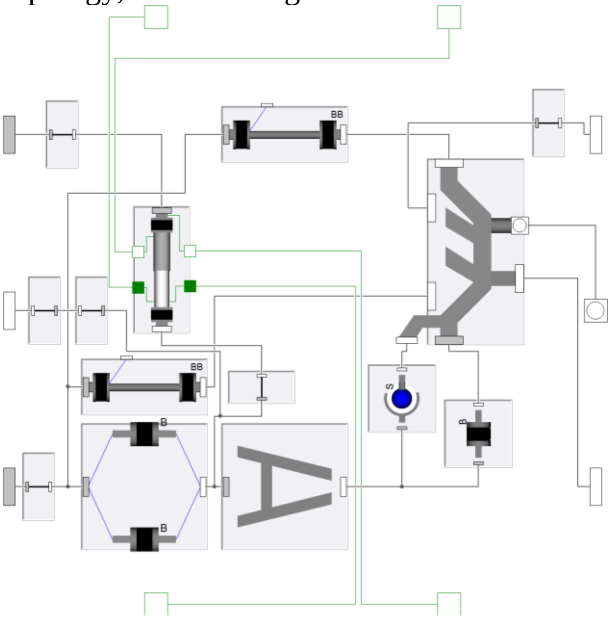

Figure 2. Rear linkage layout.

There's a lower H-arm connected with two bushings on the chassis side and a ball joint and a bushing on the upright side. Two additional link arms with two bushings each connect the upright to the chassis.

The transverse leaf spring in the rear suspension is modelled as a set of pure translational springs, shown in Figure 3. This is a simplification of a finite element (FE) based component in the offline model. The main reason for this simplification is to improve model performance as the model is meant to be used in real-time applications. There are three main spring components, one vertical stiffness between each side and the chassis as well as a roll stiffness directly attached between the two sides. In addition, there are lookup table based forces that can be defined as a function of wheel travel to incorporate the lateral and longitudinal force characteristics of the leaf spring. All these stiffnesses are tuned to the behavior of the offline model.

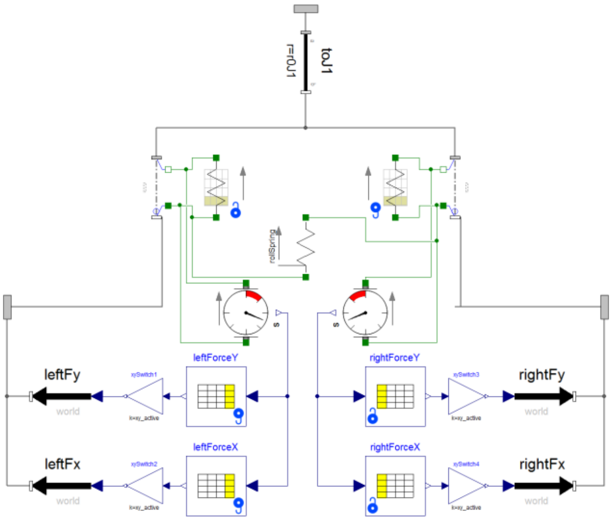

Figure 3. Transverse leaf spring model. Top attachment is to chassis, left and right are for the two linkage sides.

The front and rear suspensions are put together into a chassis model based on library templates, shown in Figure 4.

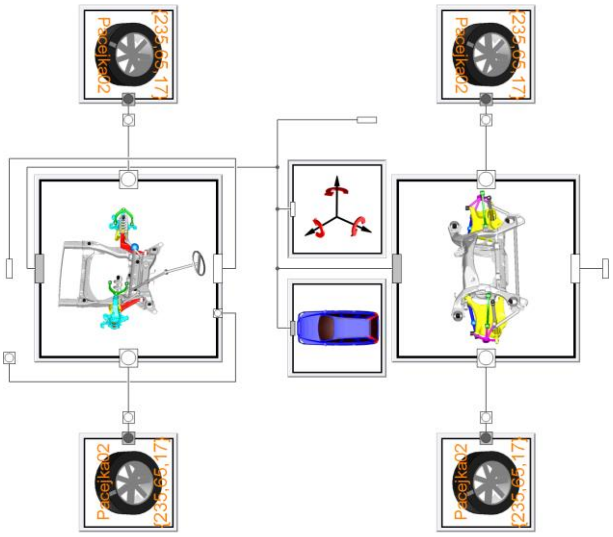

Figure 4. Model diagram of full chassis model.

A 3D visualization of the full chassis is shown in Figure 5. The visualization includes 3d models of each individual linkage component based on CAD data.

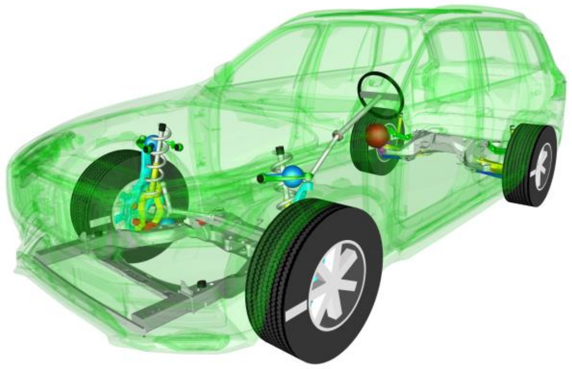

Figure 5. 3D visualization of the full chassis model.

The full chassis model has 252 continuous time states.

### 2.2 *Parameterization*

In the real-time model, data is accessed using a general-purpose data handling method called DataAccess. Several file formats can be read with this method, including TeimOrbit which is the format used in the offline model.

In general, data-aware components are responsible for reading all the data that it requires from the data file. For example, a data-aware mass part only takes a data file path and a part name as parameters and then reads the associated mass and inertia data. Similarly, a bushing takes a bushing file path as input and then reads force and torque characteristics directly from the file.

Reading model data directly from the source data files eliminates the need to manually enter parameters in the real-time model and the associated risk of user error in doing so. Furthermore, the effort to parameterize the model with the data files can be done once and then the data files can be changed to update the model, assuming the topology stays the same.

## 3 MODEL CORRELATION

Initial validation of the real-time model has been carried out against its offline counterpart in order for it to be useful. For this purpose, test rig experiments were set up with the real-time model which excites the suspension in the same way as the existing test cases used with the offline model.

The test rig model keeps the chassis side of the suspension fixed relative to ground while moving wheel pads in the vertical direction to generate motion in the wheel centers. Additional force and torque can also be applied, either in the wheel center or in the contact patch. Figure 6 shows the front suspension when it is simulated in the test rig.

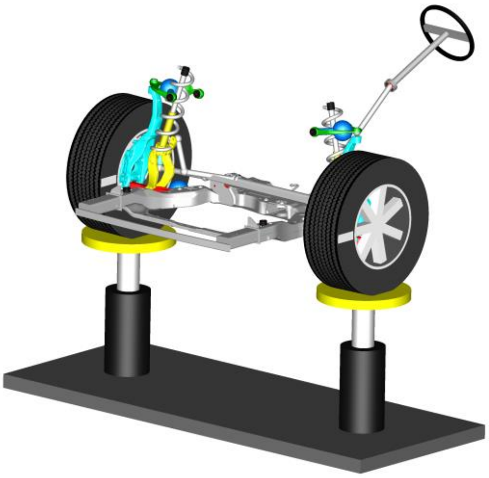

Figure 6. Front suspension in test rig.

Figure 7 and Figure 8 shows examples of the correlation result for the front suspension in a parallel wheel travel test.

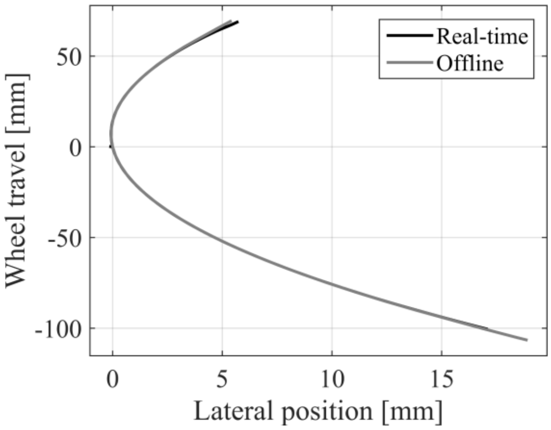

Figure 7. Front suspension in test rig.

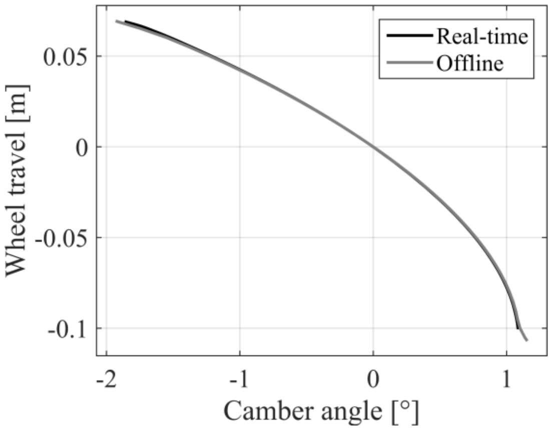

Figure 8. Front suspension in test rig.

# 4 REAL-TIME ADAPTATIONS

Besides the symbolic manipulation of the model equations, which in itself gives very efficient simulation code, there are two main techniques used to improve real-time performance of the vehicle models. Inline integration and parallelization.

## 4.1 *Inline integration*

Inline integration [7] is a method to achieve real-time performance of stiff systems. With fixed step solvers, the discretization formulae of the solver are incorporated into the equations systems of the model before doing structural analysis and symbolic manipulation to allow further performance improvements.

In this case, an implicit Euler solver is used, which gives stability in each timestep even with larger timesteps, as opposed to an explicit solver which would need small enough timesteps with regard to system dynamics to achieve stability. The implicit solver is used successfully with timesteps up to 2ms for the model in this paper. The drawback with is that this typically gives large non-linear equation systems requiring iterative solving.

## 4.2 *Parallelization*

When the above approach is applied to the chassis model it results in a single non-linear equation system of size {218}. The cost of solving non-linear equation systems increases exponentially with their size, meaning that large systems of equations needs to be avoided in order to achieve real-time performance. This can be done by introducing parallelization of the model equations. Methods for parallelizing model execution were introduced in [8].

By introducing decoupling elements between front and rear suspensions, left and right linkages and between linkages and wheels, the equation system can be divided into 13 smaller systems with sizes {30, 1, 1, 30, 1, 1, 21, 1, 1, 21, 1, 1, 21}. Two advantages are introduced by this decoupling. With the exponential increase in computational effort with equation system size, the reduction of the sizes of equation systems improve performance a great deal. Furthermore, the equation systems can now be solved in parallel, allowing the use of multi-core processors.

# 5 PERFORMANCE

The real-time model has been benchmarked with respect to performance on a regular desktop computer. This gives an indication of the turnaround times needed to execute the model. The performance is expected to improve even further when executed on a dedicated real-time platform. The accuracy of the model is also verified when using the real-time solver.

## 5.1 *Performance*

When running in Dymola on a regular Windows laptop (i7-3630QM CPU @ 2.40GHz) the model simulates each time step at 1-1.5ms turnaround time. Figure 9 shows the execution times for each time step during a double lane change maneuver. The model is stable with both 1 and 2 ms timesteps so even on a desktop computer the model can be simulated in real-time. With 2 ms timestep, each step apart from the initial ones are calculated within the duration of the timestep.

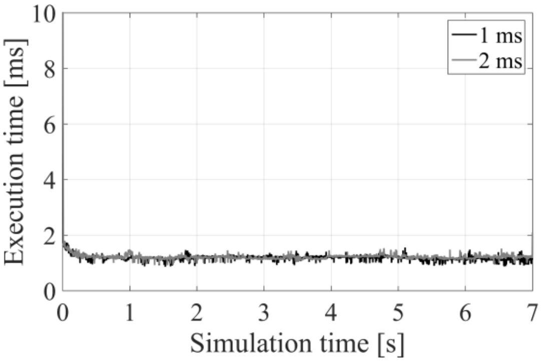

Figure 9. Real-time execution times per time step with 1ms and 2ms time steps. Each time step takes the same amount of time regardless of the length of the timestep.

Figure 10 shows histograms for the execution times, showing that the time taken for each time step is stable at 1-1.5ms.

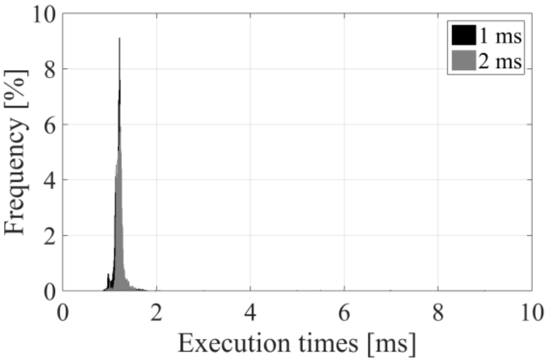

Figure 10. Real-time turnaround time per time step shown as histograms. Similar turnaround times are seen with both timesteps.

## 5.2 *Accuracy*

The use of implicit solvers can have detrimental effects on model accuracy if important model dynamics are excluded by the larger timestep. Model accuracy with the real-time solver is verified by comparing with a variable step solver in Dymola. Figure 11 show lateral acceleration output with a frequency sweep as steering wheel angle input. The variable step solver Dassl, is used as baseline and the real-time solver output is shown with 1 and 2ms timesteps.

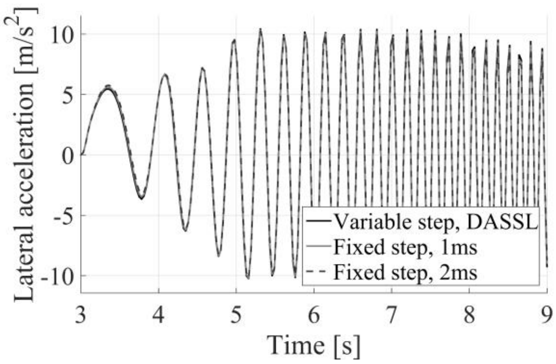

Figure 11. Solver comparison of lateral acceleration output with a frequency sweep as steering input

## 6 CONCLUSION

The DAE based model is shown to give real-time performance of high fidelity vehicle models on regular laptop computers. This is expected to improve when executed on a dedicated real-time platform.

The modeling and parameterization scheme which uses existing data files allows a shorter update cycle when adjusting model parameters on real-time platforms since lookup tables do not have to be regenerated.

The real-time model is useful for many different applications including driver-in-the-loop simulators and HIL system verification.

## 7 REFERENCES

[1] J. Rauh, "Virtual development of ride and handling characteristics for advanced passenger cars," *Vehicle System Dynamics*, vol. 40, no. 1-3, pp. 135-155, 2003.

[2] P. Goupil and G. Puyou, "A High Fidelity AIRBUS Benchmark for System Fault," in *Proceedings of the 4th European Conference for Aerospace Sciences (EUCASS)*, St. Petersburg, 2011.

[3] W. C. Prescott, G. Heirman, M. Furman, J. De Cuyper, L. Dragon, A. Lippeck and H. Brauner, "Using High-Fidelity Multibody Vehicle Models in Real-Time Simulations," in *SAE World Congress & Exhibition - SAE 2012-01-0927*, Detroit, 2012.

[4] J. Andreasson, N. Machida, M. Tsushima, J. Griffin and P. Sundström, "Deployment of high-fidelity vehicle models for accurate real-time simulation," in *Proceedings of the 1st Japanese Modelica Conference*, Tokyo, 2016.

[5] J. Andreasson, H. Elmqvist, J. Griffin, D. Henriksson, S.-E. Mattson and H. Olsson, "Real-Time Simulation of Detailed Vehicle Models," in *Proceedings of the 12th International Symposium on Advanced Vehicle Control*, Tokyo, 2014.

[6] "Modelica," 2016. [Online]. Available: www.modelica.org.

[7] H. Elmqvist, M. Otter and F. E. Cellier, "Inline Integration: A new mixed symbolic/numeric approach for solving differential-algebraic equation systems," in *Proceedings of the European Simulation Multiconference*, Prague, 1995.

[8] H. Elmqvist, S. Mattson and H. Olsson, "Parallel Model Execution on Many Cores," in *Proceedings of the 10th International Modelica Conference*, Lund, 2014.