High Fidelity Multibody Vehicle Dynamics Models for Driver-in-the-Loop Simulators

Mike Dempsey   Garron Fish   Juan Gabriel Delgado Beltran

Claytex Services Limited, UK, mike.dempsey@claytex.com

Abstract

Modern Driver-in-the-Loop simulators are sophisticated engineering tools that have been developed within Motorsport to support the development and optimization of race cars in Formula 1, NASCAR and Indycar. At the heart of the simulator is the vehicle model which has to accurately capture the behavior of the whole car. Modelica based vehicle models are used by many of the top teams because it enables a multi-domain vehicle model to be used in the simulators and support all the other simulation activities within the team. These technologies are now being deployed into road car applications which presents a number of additional challenges. One of the major differences is the need to include bushes within the suspension. This paper presents a number of the recent developments in Modelica based vehicle dynamics models for both Motorsport and road car applications including new suspension models with bushes, integration with tools to provide high fidelity LiDAR road data and real-time simulation of these models.

Keywords: Driver-in-the-Loop, vehicle dynamics, real-time, nonlinear bush models

1 Introduction

In the last few years driving simulators have been developed that can be used for more than basic procedural simulation such as driver training, evaluating human factors such as fatigue and stress, ergonomics and testing new man-machine-interfaces. The latest generation of systems make it possible to simulate a mathematical model of a car, over an exact replica of a road surface, with identical scenery and visual reference, with a *human* driver, in a safe, controlled, environment (Hoyle, 2014). These developments have been led by motorsport teams and organisations due to the restrictions in testing imposed by the governing bodies and the increasing complexity of the cars.

There are many technological developments that have enabled this including new software, new motion platforms, high fidelity real-time vehicle models and high precision LiDAR track data. LiDAR is an acronym for Light Detection and Ranging and typically this means that the whole track has been scanned with a laser to accurately measure the surface. In many cases the

motorsport organisations have developed their own in-house Driver-in-the-Loop (DiL) system, often integrating many different technologies coming from different suppliers (Toso, 2014). Some of these systems are being commercialised by the motorsport organisations to help them capitalise on the technological developments they have made in the development of these systems.

For Automotive OEM's the appeal of high fidelity driving simulators is that they can move the testing of new vehicle designs and parts into the virtual world and start the assessment of design decisions with professional drivers before committing to the production of a prototype. This approach also allows the design process to be accelerated because, for example, a change to a damper characteristic can be applied in seconds rather than having to wait while the mechanics strip and rebuild all 4 dampers to a new specification and refit them to the car.

This paper focuses on the recent enhancements in the vehicle models and the related interface to track data.

2 Integration with High Fidelity Road Data

2.1 Overview

rFpro have developed a tool called TerrainServer, that is capable of feeding 1cm resolution LiDAR data into a vehicle model at up to 5kHz, on standard PC hardware, enabling the vehicle model to be run in real-time. It was initially developed to support DiL simulators but it is equally capable of supporting offline simulation enabling the same track data to be used in both environments.

A Modelica library, also called TerrainServer, has been developed that provides an interface to the rFpro Terrain Server. This library enables Modelica based vehicle models to access high fidelity LiDAR data, and is compatible with the Modelica Standard Library and the Vehicle Dynamics Library.

The TerrainServer Modelica library provides a new tyre contact model, ground contact model, external functions to access rFpro Terrain Server and a new closed loop driver model. Figure 1 shows a full vehicle model, created using the Vehicle Dynamics Library enhanced with the TerrainServer interface and associated closed-loop driver model.
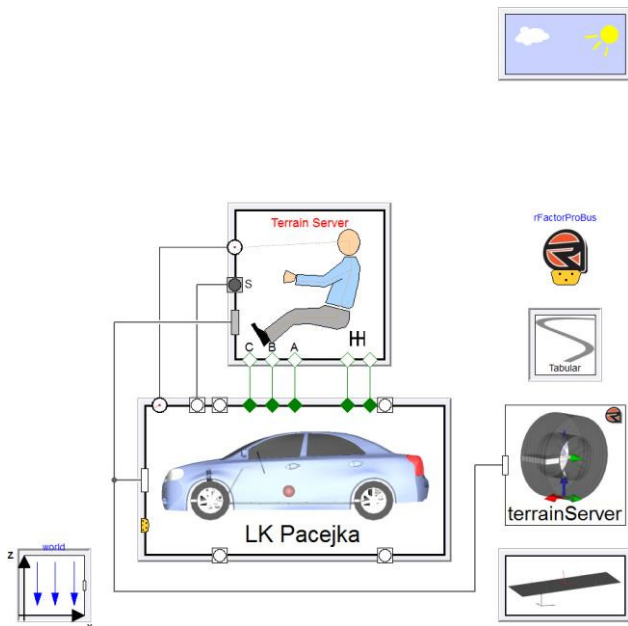
Figure 1: Vehicle model with TerrainServer interface

2.2 Tyre contact model

rFpro TerrainServer provides a number of different ways to use the LiDAR data with the most sophisticated tyre contact method referred to as a volumetric intersection sampling. Using this method the tyre is approximated as a cylinder and the resulting contact point is calculated by integrating the points within the tyre volume to return the contact patch centre and the averaged surface normal (rFactor Pro, 2014), see Figure 2. The returned data can be used in two different ways within the tyre contact model to suit different types of tyre model.

In method 1, the returned data is used to define the tyre contact point in the model. This can result in the surface normal not passing through the wheel centre a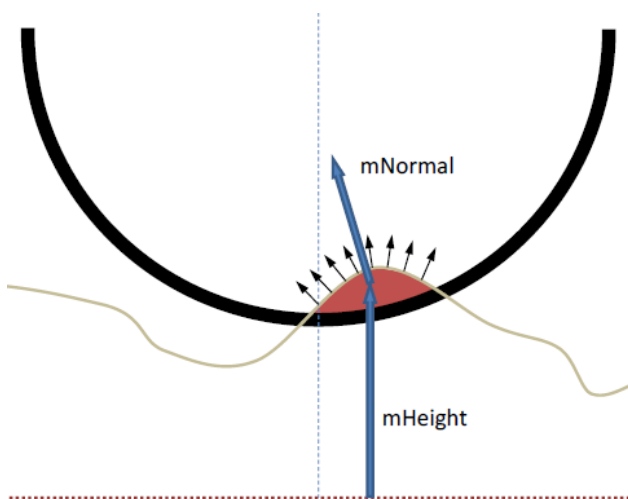nd can also induce slip velocities due to the movement of the contact patch centre as new data points enter and leave the tyre volume. As the spacing of the LiDAR data is reduced and the precision of the points improves this effect is reduced and provides an accurate contact point to the tyre model.

The problem with this approach is the interaction between the calculation of the contact point using the detailed road surface and the single point of contact tyre models typically used for handling simulation such as Pacejka. These tyre models work on the assumption that the road surface near the contact patch can be approximated by a flat plane and that the contact point lies within the tyre central plane (Pacejka 2012). This means that the smallest considered wavelength of the decomposed surface vertical profile is large with respect to the contact length and its amplitude small. The high fidelity track data used in rFpro TerrainServer provides data to the tyres that breaks this assumption, however a way to handle this has been developed.

In method 2, the returned data is used to define a plane underneath the wheel and the contact point is calculated as being the closest point to the wheel centre that lies within the ground plane and tyre central plane. This is illustrated in Figure 3 where the calculated plane and contact point are shown in green. This approach allows the assumptions in the single point of contact tyre models to remain valid, i.e. the road is treated as a flat plane underneath the tyre. This reduces the movement of the contact patch due to the entry and exit of points into the tyre volume and also ensures that the surface normal always passes through the wheel centre. The compromise in this approach is that the real surface detail available from the LiDAR data cannot be used to full effect by tyre models like the Pacejka model.
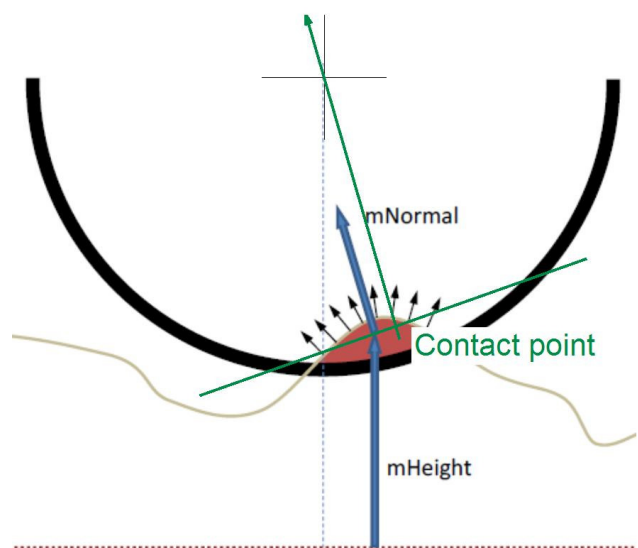

Figure 2: Calculation of the contact patch centre and average surface normal in rFpro TerrainServer

Figure 3: Calculation of the effective tyre contact point using method 2 where the rFpro data is used to define a plane underneath the tyre

3 Closed-loop driver model

3.1 Overview

A new closed-loop driver model has been developed for use with TerrainServer. The driver is a path following driver model with the trajectory defined in a new Path object that defines the driving line and speed profile to be followed. The driving line and speed profile would typically be captured from a session on the driving simulator with a professional driver, it can then be repeated and analysed offline to explore setup changes. This new driver model has several key differences to the existing model provided in the Vehicle Dynamics Library which include the way the target path is defined, how the preview points are determined and how the longitudinal tracking is implemented.

The path is defined as positions in the world coordinate system as a function of a distance along the path. A speed profile that is also a function of the distance along the path is included. The driver model looks at the path information to decide what steering, pedal and gear shift commands are necessary.

3.2 Path planning

The model uses 3 preview points: 2 are used in the longitudinal tracker and the lateral tracker can use either 1 point or average all 3. In the longitudinal tracker one point is used as the input to a PI controller to determine the throttle and brake pedal positions. This preview point should be close to the current driver position to achieve accurate tracking of the speed profile. The second point is used in a mode correction block to enable the driver model to anticipate a switch between acceleration and braking rather than waiting for the PI controller to respond after passing the transition point.

The lateral tracker calculates the angle from the vehicles current position and heading direction to the position of the preview point. From this angle it determines a steering angle that needs to be applied. Filters are used to limit the rate at which the driver can adjust the steering and pedals to keep the responses appropriate for the type of driver that is being represented.

The lateral tracking preview point is a variable distance ahead of the driver's current location and can be adjusted according to many factors including vehicle speed, lateral offset from the defined trajectory, curvature of the path and yaw velocity, see Figure 4. The preview distance and corresponding adjustments are implemented to give the driver model the ability to plan ahead and adjust the control strategy to suit the road and vehicle state just as a real driver does when driving the car.

The typical configuration of the driver model is that the preview point will move further ahead of the driver as the vehicle speed increases and this provides the basic preview distance. If the lateral tracking of the driver

model is not good, or the vehicle is unable to follow the path then the lateral offset will increase and the preview distance will also be increased. This is to avoid the driver trying to then turn too sharply when he is unable to follow the path. To avoid cutting sharp corners the preview distance is reduced as the curvature of the path increases. A large curvature value means a tight corner and to keep the lateral tracking performance within reasonable limits we make sure that the preview points are not too far ahead of the driver. The rate at which the preview point moves relative to the driver is limited to avoid large jumps in the position which would induce large, and unrealistic, changes in the steering command.
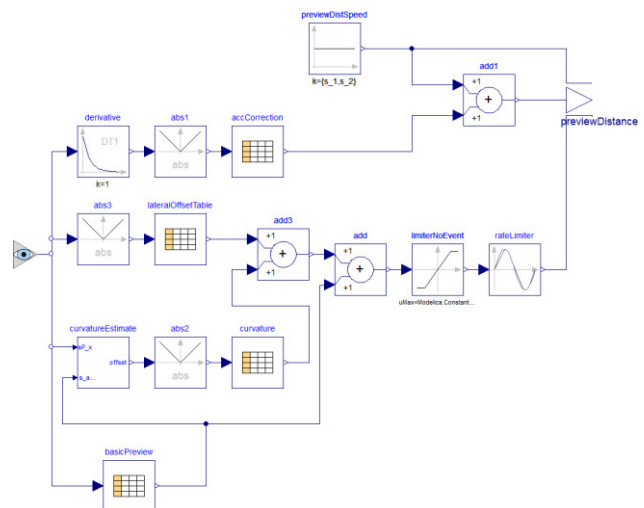

Figure 4: Preview distance calculation

3.3 Generating the target path

The path is generated by filtering the data recorded from the driving simulator to specify the minimum distance and the minimum time between points. This means that at low speeds the points that define the path will be at least the minimum distance apart but at higher speeds the points will be more spread out, for example at 50m/s with a minimum time of 0.2s the points will be 10m apart. This filtering is done to generate a smooth path for the driver model to follow.

Two ways of processing the recorded data are provided in the Modelica TerrainServer library. When using the Tabular path model the data is processed into a single continuous path that can be followed. There is also a Racing Lap path where the data is processed to extract an out lap and a single flying lap from the data. The start and end of the flying lap has to be blended together and blended with the out lap so that the flying lap can be looped allowing multiple laps to be simulated.

The driver model can also be exported and compiled as a model that can be run within the driving simulator environments. This enables an automatic driver to be used to verify the correct operation of the driving simulator platform and new vehicle models prior to a test with a human driver.

3.4 Exploring driver behavior

Using the various tuning parameters within the driver model we can explore how different driving styles influence the loads on the vehicle. For example by keeping the preview points close to the driver's position and having a high proportional gain and low integral gain in the longitudinal PI controller we can define a very aggressive driver that will work the steering wheel and pedals at a high frequency to follow the desired path very closely.

At the other extreme we can configure the driver so that the preview points are further ahead of the vehicle, and with a low proportional gain and high integral gain we get a much more relaxed driver behaviour but the path tracking performance will not be very good. For this relaxed driving profile we still need to keep 1 of the longitudinal preview points relatively close to the vehicle to ensure that he can still switch between acceleration and braking at the appropriate points and avoid large velocity overshoots which would not suit this style of driver behaviour.

Figure 5 shows a comparison of these two different driving styles on a short section of a test track. The aggressive driver has better lateral and longitudinal tracking than the relaxed driver, and achieves this through faster actuation of the pedals and steering wheel.

Through careful selection of the driver model parameters we have used this driver model in transient lapsim analysis. In these applications the driving line has been recorded from a simulator session with the professional driver, the use of the driver model then enables setup changes to be evaluated. This approach to lapsim analysis allows the full transient behaviour of the car to be considered which is not possible using quasi-static approaches.
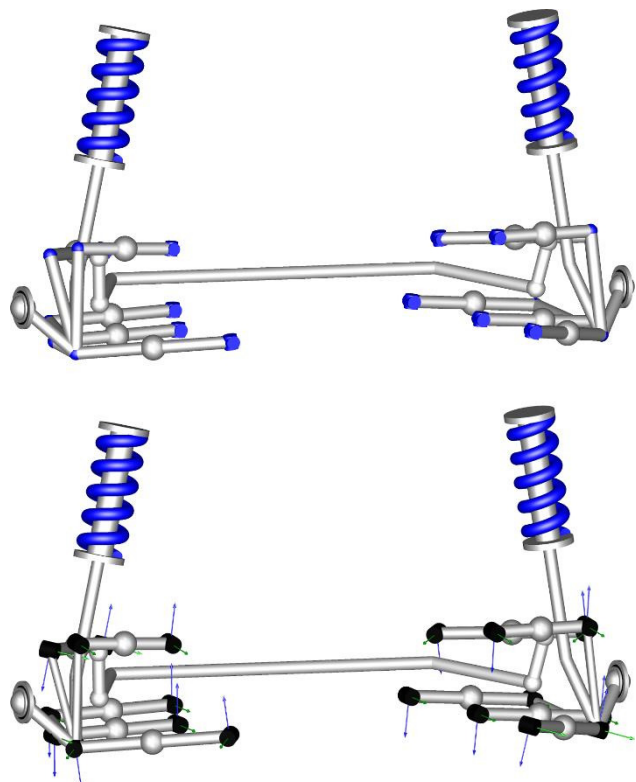
4 Road car suspension models

4.1 Overview

A new set of suspension models has been developed for the simulation of road car suspensions. The new family of models provides kinematic and elastic suspension models where the bushes can be simple linear models, nonlinear models or sophisticated elastomer models including frequency and amplitude dependent effects. Figure 6 shows the animation view of a rear multilink suspension in Dymola with two variants: the top image uses ideal joints and the lower one includes bushes.

The suspension models are defined using a template based approach with replaceable components allowing the links with ideal joints to be easily swapped for links with bushes at either ends. The bushes can then easily be redeclared to have the appropriate characteristics.
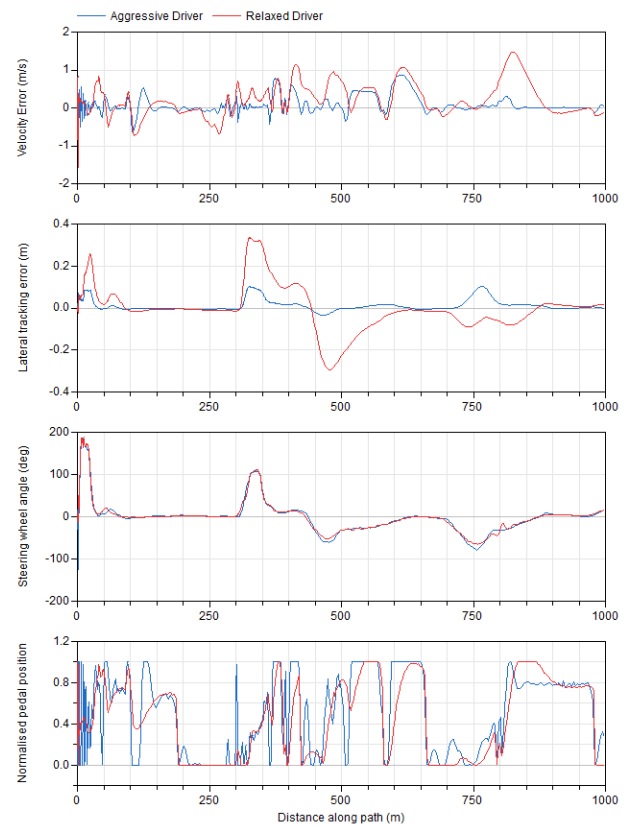

Figure 5: Comparison of aggressive (in blue) and relaxed (in red) driver model parametrizations. The plots show velocity tracking error, lateral tracking error, steering wheel angle and accelerator pedal position

Figure 6: Kinematic Multilink suspension (top) and elastic Multilink suspension model (bottom)

The templates provide support for the easy integration of flexible bodies based on reduced Finite Element models to define the structural compliance of links, control arms, and uprights. Figure 7 shows an example of the quarter car template for a front McPherson strut suspension using bushes and a flexible body for the lower control arm. The upright can also be easily replaced with a flexible body as it defines separate attachment points for the wheel centre, damper strut, lower ball joint, track rod and anti-roll mechanism.
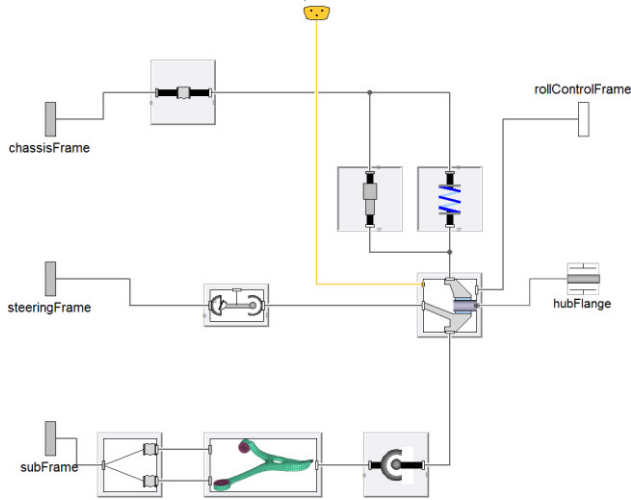
**Figure 7:** McPherson strut suspension with bushes and FE based lower control arm

4.2 Bush models

In road car applications the suspension bushes have a big effect on the ride and handling of the car, and have to be tuned to provide the right compromise between noise, vibration and harshness (NVH) and the desired handling characteristics. To support the tuning of these bushes at all stages of the design process a number of different characteristic models are provided including simple linear bushes and more sophisticated models with frequency and amplitude dependent characteristics.

In MultiBody simulation the modelling of elastomer mounts, such as suspension bushes, is usually done using a simple spring-damper element. This approach works satisfactorily provided the stiffness and damping terms are tuned to accurately capture the dynamic stiffness of the elastomer at the operating point being studied. In the case of superimposed oscillations with differing frequency and amplitudes, which an accurate road input would induce, then this approach is not adequate and more complex elastomer models have to be used (Persson, 2003) .

A sophisticated elastomer model has been implemented that allows the frequency and amplitude dependency to be captured whilst maintaining a relatively low computation cost (Pfeffer, 2002) and an automatic calibration method simplifies the parametisation of the model. Figure 8 shows how this

model is implemented in Modelica. It consists of a nonlinear force-displacement spring element that captures the static characteristic of the elastomer. In parallel with the spring element, is a linear damper and then a component array of frictional elements and spring-damper elements in series. The frictional elements are used to capture the amplitude dependent effects and the spring-damper in series, often referred to as a Maxwell model, are used to capture the frequency dependent effects.
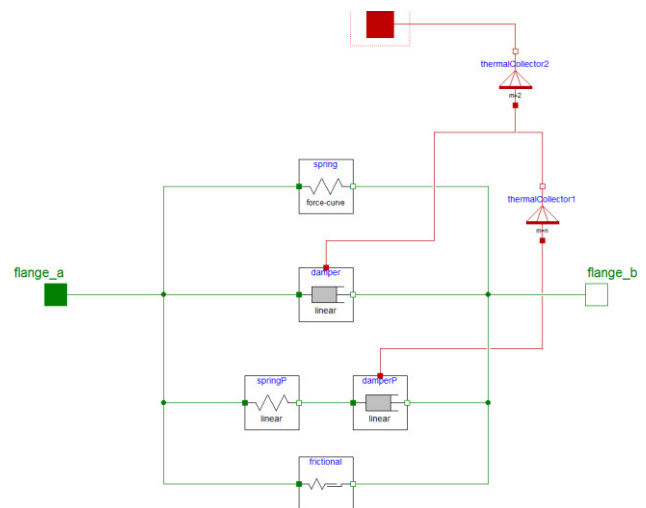
Figure 8: Elastomer with frequency and amplitude dependent characteristics

To parameterise the model the elastomer needs to be measured to get the static and dynamic stiffness and loss angle characteristics. Using this information, optimisation can be used to tune the model parameters, provided the user first decides on the sizes for the component arrays. The number of Maxwell models included determines the ability of the model to cover the frequency range of interest, with 1 Maxwell model it is possible to accurately represent the bush at one of the measurement points but at other frequencies the dynamic stiffness and loss angle will be incorrect, as shown in Figure 9. In the example shown, the parameters for the Maxwell model were calculated from the measurement data at 21Hz using the following method. The measurement results for dynamic stiffness and loss angle can be used to calculate the complex stiffness of the bush:

$$k_{dyn} = k_d(\cos\alpha + j\sin\alpha) \qquad (1)$$

Where k_d is the dynamic stiffness and α is the loss angle. The complex stiffness of the bush, without frictional elements, at a specific frequency can be calculated according to:

$$k_{dyn} = k_e + c.w.j + k_{maxwell} \qquad (2)$$

Where k_e is the elastic stiffness, c is the damping, w is the frequency in rad/s, and $k_{maxwell}$ is the complex stiffness of the Maxwell model. From equations 1 and 2 we can calculate the stiffness contribution that must

come from the Maxwell model. The complex stiffness of the Maxwell model is given by:

$$k_{maxwell} = k_m \left(\frac{w.j}{1 + t_r.w.j} \right) t_r \qquad (3)$$

Where k_m is the stiffness of the spring and t_r is the time response of the Maxwell model. k_m*t_r is the damping of the Maxwell model. Equations 1-3 can be solved as a nonlinear system consisting of 2 unknowns k_m and t_r to calculate the properties of a single Maxwell model tuned to work correctly at the selected frequency.
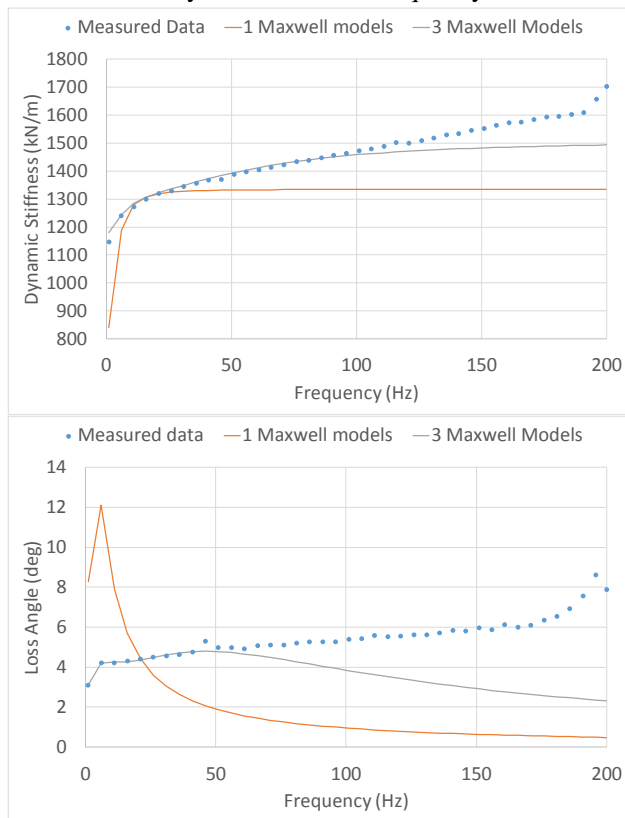
Figure 9: Dynamic stiffness and loss angle for a tuned model with varying numbers of Maxwell models

By increasing the number of Maxwell models in parallel to 3 and through the use of optimization to determine the parameter values it is possible to get the bush model to have a greater accuracy over a larger frequency range. Figure 9 shows the results of tuning the parameters to give a good fit between 1 and 46Hz. This would mean that the bush works well in the frequency ranges necessary to simulate primary and secondary ride effects. To cover a wider frequency range then additional Maxwell models would be needed but this increases the difficulty of the optimization problem, and as each Maxwell model adds 1 state to the model it also increases the computation time.

5 Real-time simulation

One of the key advantages of a Modelica based approach for vehicle dynamics analysis is the ability of Dymola to export simulation code that is capable of running in real-time. This has been used in Motorsport for running vehicle models as part of a Driver-in-the-loop simulator for several years (Dempsey, 2012). Typically though a race car does not include bushes in the suspension which makes the vehicle dynamics model simpler and easier to run in real-time even with structural compliance effects included. To run a suitably detailed road car vehicle dynamics model in real-time we need to take advantage of new capabilities in Dymola 2016 to parallelize a model (Elmqvist, 2014; Andreasson 2014).

Utilizing this approach we have been able to partition the model into a number of separate computation tasks: the body; front suspension; the left and right rear suspensions; and 4 tyre models. The model used is a saloon car with double wishbone front suspension and a multilink rear suspension. The front suspension uses ideal joints but includes compliance effects in the upright and the rear suspension includes bushes at the inboard and outboard ends of every link, see Figure 10. The powertrain model includes a mapped engine model, a 6 speed automatic gearbox with torque converter and it is front wheel drive. The model has been optimized to eliminate events and uses elastic friction models for the brakes and losses within the powertrain. The whole vehicle model consist of 243 states, of which 17 relate to the brakes and powertrain systems and the remainder are related to the suspension and tyres.
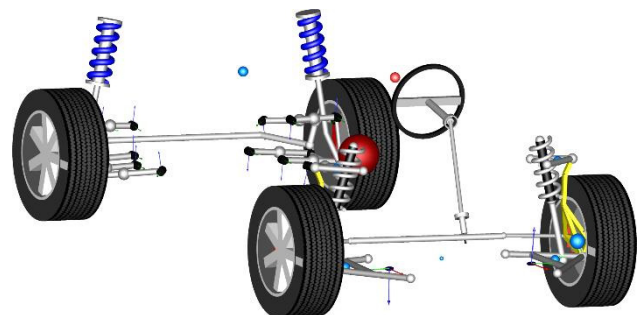

Figure 10: Animation view of the vehicle model

The model is partitioned into separate tasks using the decouple blocks available in Dymola as shown in Figure 11 for the rear suspension. The decouple blocks are used to break down the size of the implicit nonlinear system of equations related to the inline integration method. In the full vehicle model, if the decouple blocks are not used there is a large implicit nonlinear system of equations of size 139 but when these blocks are used this is broken up in to a set of 4 nonlinear systems of equations of sizes 21, 6, 38, and 38. The two systems of size 38 are related to the left and right rear suspensions. These smaller systems of equations are easier to calculate, and most importantly the jacobian used by the implicit inline integration method is much smaller and easier to compute.
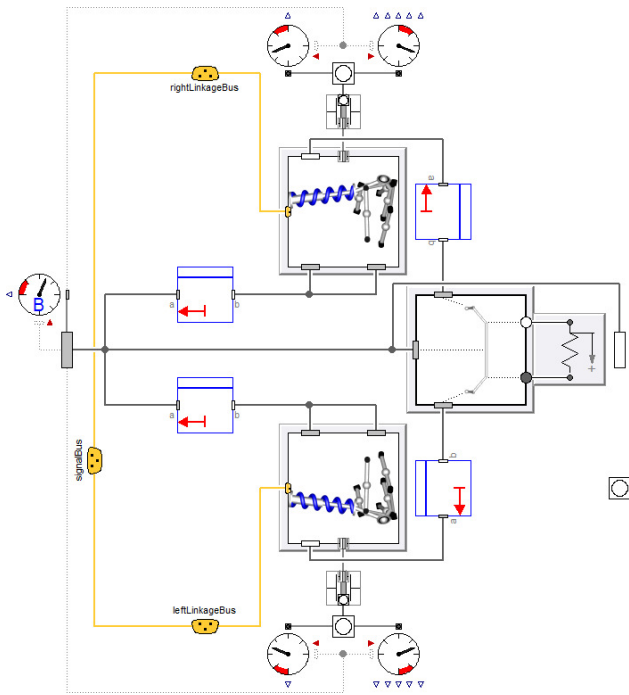
Figure 11: Rear suspension model including decouple blocks to partition the model into separate tasks for the left and right suspension

By splitting the model into these subtasks we have been able to run the model in real-time using a inline implicit Runge Kutta solver with a 1ms time step. Figure 12 shows the turnaround time for the model during a few laps of the Motorland Circuit of Aragon when running the vehicle with rFpro on a PC workstation. This track model uses high fidelity LiDAR data to define the surface which is provided to the model by rFpro TerrainServer as described in 2.1. The PC runs Windows 7 (64 bit) with an Intel Core i7 5960X processor overclocked to 4.2 GHz. There are very occassional overruns but these are small in magnitude, around 0.2ms, and infrequent which means they can be easily tolerated by the DiL system.
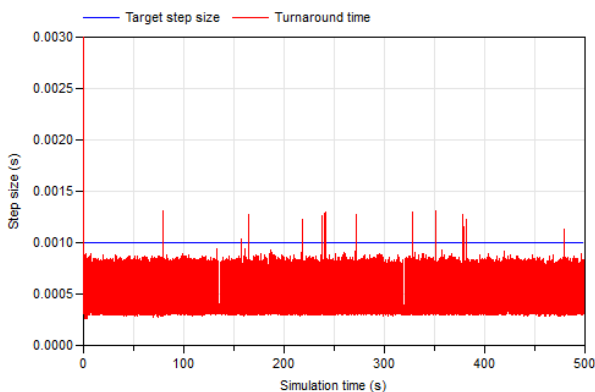
Figure 12: Turnaround time for the vehicle model running in rFpro on a track using LiDAR data

When configuring a model for use in real-time simulation there is always a trade-off to be made between performance and accuracy compared to running the model with a variable step solver. It is important to verify that the change in simulation results are minor when partioning the models and to find the best compromise between computation time and accuracy. Figure 13 shows a comparison of the simulation results obtained with this model driving at 55kph with a sinusoidal steering input. It shows that the variations found with the different real-time solver settings are small when comparing the results to those achieved with a variable step solver.
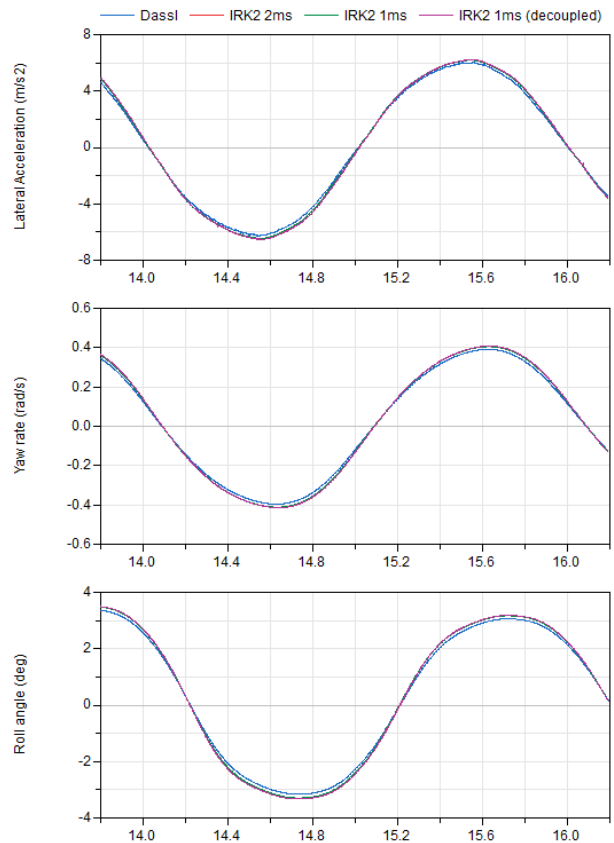
Figure 13: Comparison of lateral acceleration (top), yaw rate (middle) and roll angle (bottom) for the model running with Dassl compared to using Implicit Runge Kutta 2nd order fixed step solver at rates of 1ms, 2ms and with the decoupled model running at 1ms

6 Conclusions

The integration between the driver-in-the-loop system, rFpro and Modelica has been extended to enable the same track data to be used for offline and real-time simulation. This means the analysis work in Dymola can use the same high fidelity track data that is available to the driver-in-the-loop simulator. A new closed loop driver model has also been developed for use with these high fidelity tracks and it allows different driver behaviour to be assessed as well as comparing the handling effects of vehicle setup changes. These capabilities are available in a commercial Modelica library called TerrainServer.

New suspension models have been developed to support the transfer of these technologies from motorsport into road car applications. These new models enable higher fidelity suspension models to be created, and when coupled with the latest enhancements in Dymola, support parallelisation across multiple cores enabling a full MultiBody vehicle model with bushes in the suspension to be run in real-time at 1 kHz using standard PC hardware. These models will be available in future commercial Modelica libraries.

References

J.Andreasson, H. Elmqvist, J. Griffin, D. Henriksson, S.E. Mattson, H. Olßon, Real-Time Simulation of Detailed Vehicle Models using Multiple Cores, *12th International Symposium on Advanced Vehicle Control*, September 22-26, 2014, Tokyo, Japan

M. Dempsey, A.Picarelli, G.Fish, Using Modelica models for Driver-in-the-loop simulators, *Proceedings of the 9th International MODELICA Conference*, September 3-5, 2012, Munich, Germany, doi: 10.3384/ecp12076571.

H. Elmqvist, S.E. Matsson, H. Olßon, Parallel Model Execution on Many Cores, *Proceedings of the 10th International Modelica Conference*, March 10-12, 2014, Lund, Sweden, doi:10.3384/ecp14096363

C. Hoyle, How Mobile Lidar is Revolutionising Automotive Testing, *LiDAR News Magazine*, Vol. 4 No. 7, Spatial Media, 2014

H. Pacejka, Tire and Vehicle Dynamics (Third edition), *Butterworth-Heinemann*, 2012, doi:10.1016/B978-0-08-097016-5.01001-9

A. Persson and F. Karlsson, Modelling Non-Linear Dynamics of Rubber Busings – Parameter Identification and Validation, Division of Structural Mechanics, Lund University, 2003, http://lup.lub.lu.se/student-papers/record/3566697

P. Pfeffer and K. Hofer, Simple Non-Linear Model for Elastomer and Hydro-Mountings, *ATZ worldwide* 5/2002 Volume 104, 2002

rFactor Pro, *HD Terrain Server User Guide*, September 2014

Toso, A. and Moroni, A., Professional Driving Simulator to Design First-Time-Right Race Cars, *SAE Technical Paper* 2014-01-0099, 2014, doi:10.4271/2014-01-0099.