

Model Reduction Techniques Applied to a Physical Vehicle Model for HiL Testing

R. Gillot* S. Gallagher** A. Picarelli* M. Dempsey*

*Claytex Services Ltd. Edmund House, Rugby Road, Leamington Spa, CV32 6EL
{romain.gillot, alessandro.picarelli, mike.dempsey}@claytex.com

**Ford Motor Company Ltd, Dunton Technical Centre, SS15 6EE
sgalla20@ford.com

Abstract

To build a full vehicle model entirely based on physical equations is a challenge (Dempsey M., 2006). To have this model to run fast enough so that it is suitable for Hardware-in-the-Loop testing is even more challenging. The level of detail in the physical representation of the vehicle can always be increased at the cost of simulation time. Even if the performance of the hardware is constantly improving, we still have to compromise.

As part of the MORSE (Model based Real-time Systems Engineering) project, model reduction techniques are developed and applied to a vehicle model. The results in terms of accuracy and simulation speed are then investigated.

Keywords: vehicle model, model reduction, real-time simulation, Hardware-in-the-Loop testing

1. Introduction

MORSE (Model based Real-time Systems Engineering) is a 2-year project in collaboration with Ford and AVL, co-funded through InnovateUK's Towards Zero Prototyping competition. The aim of the project is to develop predictive engine and vehicle models enabling virtual calibration of driveability control features and validation of On Board Diagnostics (OBD) fault paths. In order to satisfy these requirements, we need physical models with a high level of detail. We need, for example, a clutch with a detailed friction model, a gear set with torque reactions, a differential with force and torque reactions, compliant drive shafts, Pacejka tyre model, linear engine mounts, detailed suspensions, a crank angle resolved engine model. We use these models for Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) testing. Whilst simulation time is not a major concern for SiL, the models do have to run in real time and with no overruns to be used in the HiL environment. This is why we need model reduction techniques that will help us simplify our models to improve simulation speed while matching

the behaviour of the full model. The idea is to have two different models for two applications: the fully detailed

model for SiL testing and a reduced version, automatically generated and parameterized from the first one in order to match its results, for HiL testing. In this paper, we present the full vehicle model and its associated level of detail. Then we introduce the model reduction techniques and show how they are applied to each subsystem. The subsystems and their reduced equivalents are tested and the results compared. Finally the full vehicle model as well as the reduced vehicle model are run over a series of Tip-In/Tip-Out manoeuvres in the HiL environment and the trade-off between accuracy and simulation performance is investigated.

2. The Vehicle Model

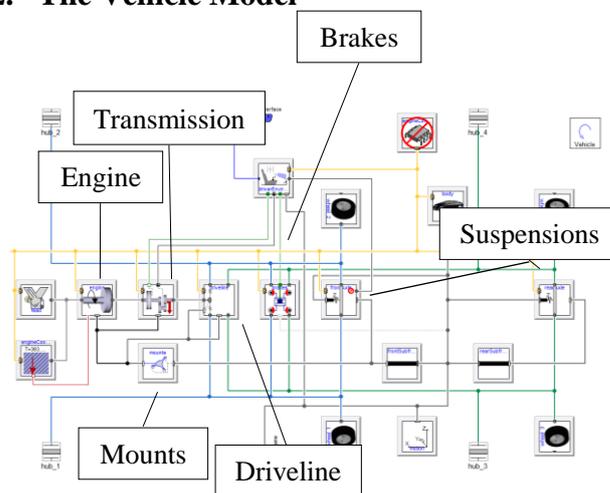


Figure 1. Detailed view of the vehicle model with all the subsystems.

In order to perform the driveability analysis, a certain level of detail is required in the vehicle model.

We require a mounts model using linear springs and dampers to constrain the motion in the three directions (x, y, z) as well as a transmission model with a clutch based on coulomb friction with a reliable handling of the stuck phase and a gear set that models the gears, gear

meshes and mesh losses and takes into account the torque reactions.

In the driveline, the drive shafts need to be compliant and to include backlash. The differential, in the same manner as the gear set, models the gear contact and considers the torque reactions.

The suspensions have a vertical degree of freedom (fore-aft motion can be included) and use a linear spring and damper (other spring and damper models are available if required).

Another critical component in driveability studies is the modelling of tyres; our models thus utilize the Pacejka slip model since it is the most commonly used model to investigate tyre dynamics.

The engine is not studied in this paper and sits outside of the vehicle model, a torque source coupled to a flywheel are used to transmit the torque from the engine to the transmission.

3. Model Reduction

a. Transmission

The physical gear set (Figure 2) is a multibody model (Dempsey M., 2009) that uses physical representations of gears, shafts, bearings and synchronizers. Gear engagement is achieved through translational mechanics flanges (in green in the Figure 2) providing a clamp load to the left or right flanges on the synchronizer dependant on the sign of the clamp force.

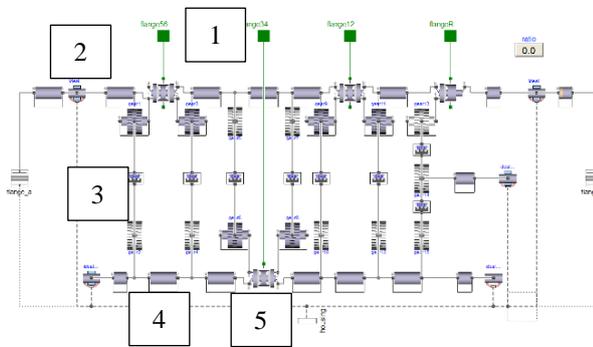


Figure 2. Physical gear set model (1: Translational flange, 2: Bearing, 3: Gear, 4: Shaft, 5: Synchronizer).

This is how the model reduction tool works internally: The physical gear set is run on a test rig (Figure 3) in 1st gear. The speed source ramps up from 0 to 6000 rpm. A load is attached to the gear set. The experiment is repeated several times, varying the load each time (from 30 to 360 N.m). The transmission is thus run over a range of speeds and loads.

This procedure is repeated for all the remaining gears. We now have a loss map for the transmission for all the operating points. This data is stored in a set of data records (one for each gear) through an automated procedure.

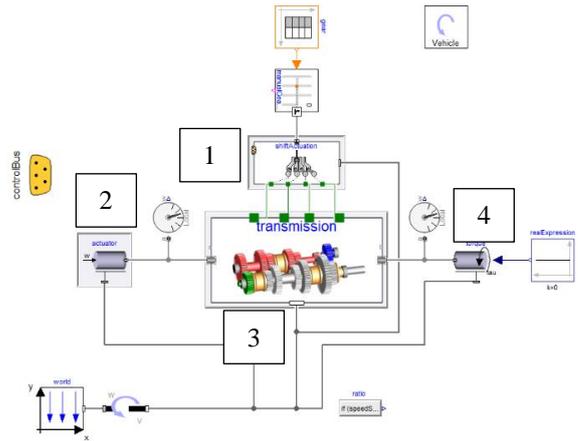


Figure 3. Gear set test rig (1: Shift mechanism, 2: Speed source, 3: Gear set, 4: Load).

The function then extends the reduced gear set model from the PTDynamics library (Figure 4) and populates the lumped losses component with the data records we just created. This lumped losses component interpolates the tables in the data records to give the losses depending on gear, speed and load.

The inertia of the whole physical gear set for each gear is also calculated. The reduced gear set has a lumped inertia component that will be populated with the data we just derived.

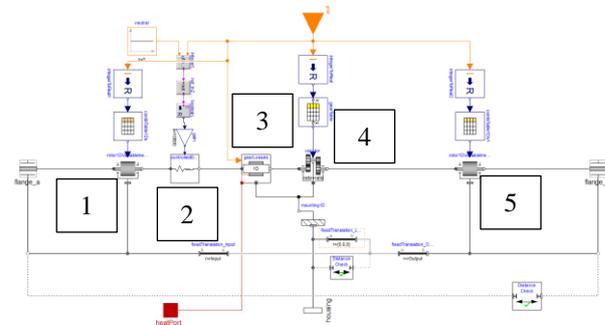


Figure 4. Reduced gear set model (1: Input shaft lumped inertia, 2: Clutch, 3: Lumped losses, 4: Ideal variator, 5: Output shaft lumped inertia).

The gear ratio is applied using an ideal variator which means there is a first order transfer function between the ratio input and the applied ratio. A clutch (item no. 2 in the Figure 4) is used in this model since the ideal variator does not give good results when in neutral.

$$\begin{aligned} \omega_a &= \omega_b * ratio \\ 0 &= \tau_a * ratio + \tau_b \end{aligned}$$

Where ω_a is the angular velocity at flange_a (input flange), τ_a is the torque at flange_a and ratio is the gear ratio. When in neutral gear, the equations become:

$$\begin{aligned} \omega_a &= 0 \\ 0 &= \tau_b \end{aligned}$$

The first equation forces the angular velocity at the input flange of the gear set to be zero and as a consequence the angular velocity of all the components rigidly connected to it, including the engine, to also be zero. The clutch in this gear set is always engaged except in neutral. The equation in the variator sets the angular velocity at the output flange of the clutch to zero but the input flange is free to rotate as the clutch is disengaged.

Let us run a fully detailed 6-speed gear set and its reduced version we derived using the model reduction function and compare the results. To do so, we run the models in all the gears, feeding in a torque of 80 N.m (see Figure 5).

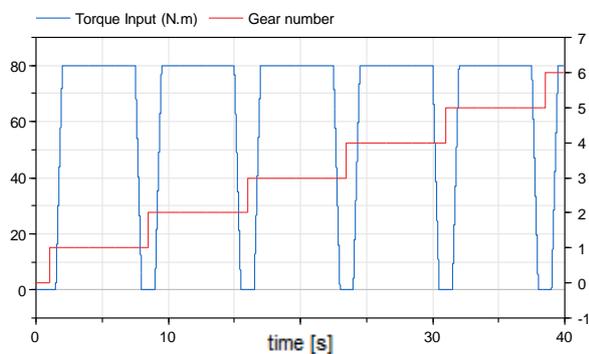


Figure 5. Test rig gear and torque inputs.

We can now have a look at the torque at the input and output of the two gear sets with different levels of detail:

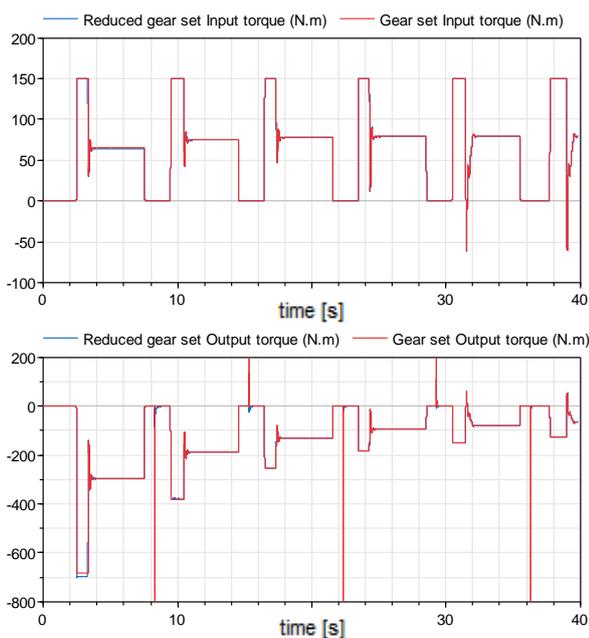


Figure 6. Input and output torque of a 6-speed gear set and its reduced equivalent.

The results of the reduced model match very well those of the full gear set, there is only a small discrepancy at around 3s. This is because in the table of lumped losses that we got using the function, the torque ranges only up to 350 N.m. The torque being outside of this range at the beginning of the test, Dymola has to extrapolate from the table of losses which leads to a small inaccuracy. The torque range will be extended in future work.

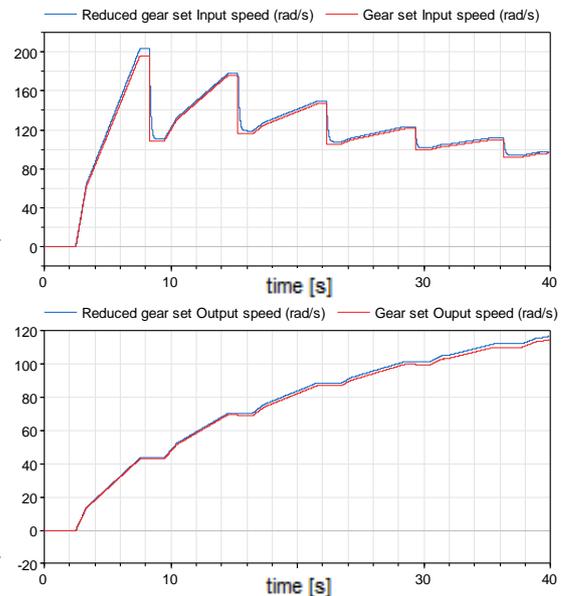


Figure 7. Input and output shaft speed of a 6-speed gear set and its reduced equivalent.

The speed curves match well too. The speed of the reduced gear set is slightly overestimated though. This comes from the inaccuracy in the torque curve at the beginning of the simulation (see Figure 6) which therefore calculates an acceleration that is too big. The relative error in angular velocity then gets carried until the end of the simulation but its magnitude does not increase.

The test lasts for 40s and the solver used is Radau II – order 5 stiff with a tolerance of 1e-5, this solver will be used to test all the subsystems in the following paragraphs. The improvements in terms of simulation performance are shown in the following table:

	Full gear set	Reduced gear set
Simulation time (s)	4.33	0.95
State events	204	36
Jacobian-evaluations	302	155

Most of the events happen during gearshift (see figure 8 below) so the savings in simulation time depend a lot on

the type of test we run (i.e. how frequently we change gear).

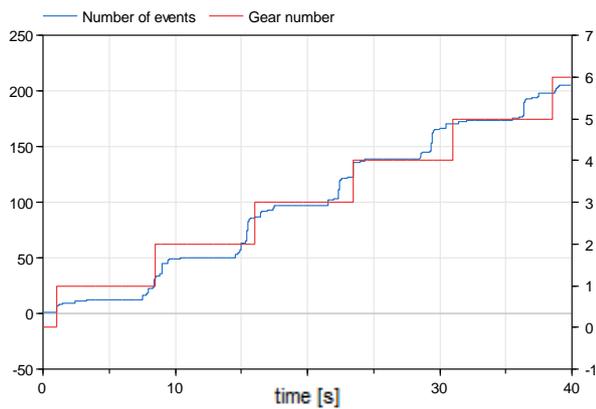


Figure 8. Correlation between number of events and gearshift.

We presented in this section a model reduction tool which is automatic, creates a reduced model that gives very similar results to the full model and runs faster. The new model is however a one-dimensional rotational model which is then not suited for studies where force or torque reactions are of prime importance.

b. Driveline

Here we take advantage of the fact that we are, in the scope of the MORSE project, only performing straight-line manoeuvres. The results we get on the left side of the car (wheel angular velocity, suspension’s spring force and position, driveshaft torque etc.) are thus very similar to the ones on the right-hand side, allowing some simplifications. We can use ideal force and torque sources to replace the physical actuators (translational and rotational springs and dampers) on one side of the vehicle. We arbitrarily chose to reduce the components on the right side. In the case of the driveline, we then reduce the right driveshaft, and keep the left one unchanged.

The differential required adaptation since it now only needs to transfer torque to one driveshaft. A standard open differential would transfer all the torque coming from the transmission to the right driveshaft as there is no load on it. This new model splits the torque independently of what is connected to its flanges. This approximation works because we only test the vehicle in a straight line and we assume that the road is ideal (i.e. uniform friction coefficient, no bumps or holes).

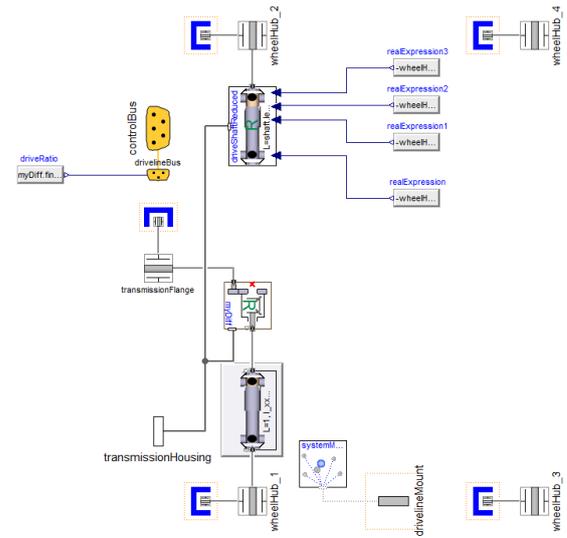


Figure 9. Driveline model with a complete left-hand side driveshaft (bottom) and a reduced right-hand side driveshaft (top).

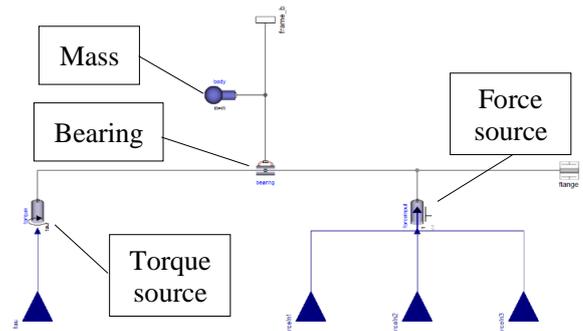


Figure 10. Reduced driveline using force and torque sources.

The reduced driveline uses ideal force and torque sources to replicate the behaviour of the other non-reduced driveshaft. The inputs to these force and torque sources are set to the sensed values in the non-reduced driveshaft.

We can switch between the full and reduced driveline by just double-clicking on the driveline subsystem at the vehicle level (see figure 1) and choosing between the two models. There is a Boolean parameter that is used to conditionally enable or disable components. The reduced model’s parameters are linked to the parameters from the full one so we do not need extra parameterization when switching between models.

We test the reduced driveline with a trapezoidal torque input and observe the torque and angular velocity at the wheel hubs:

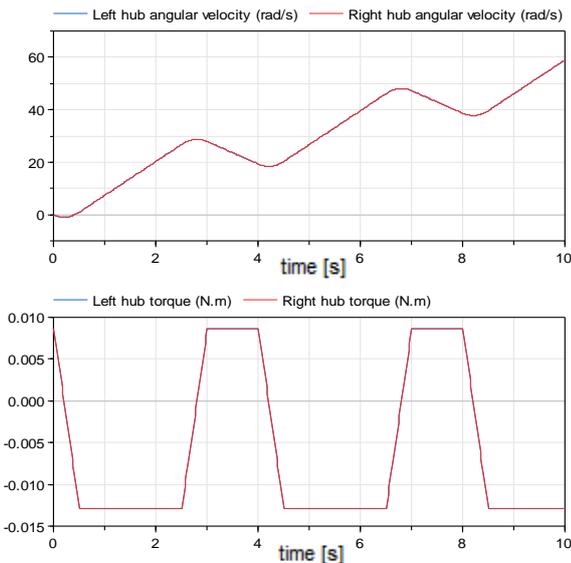


Figure 11. Angular velocity (top plot) and torque (bottom plot) at both front wheel hubs.

The results match perfectly. The benefits in terms of simulation speed and number of events are not shown in this section since the driveline itself is a subsystem that runs relatively quickly. The results will be investigated when testing the full vehicle model.

c. Suspensions

In this paper, we consider a one degree of freedom independent suspension with anti-roll bar. An optional steering connection can be used but we leave the model empty here since we only want to run the vehicle in a straight line. This empty steering model holds the steering frame in a fixed position. The linear anti-roll bar model uses the difference in z-heights to calculate a roll angle and apply a reaction torque.

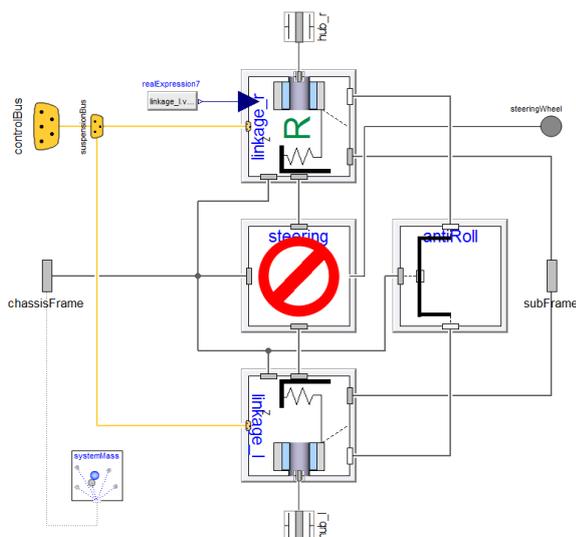


Figure 12. Front suspension model. The left linkage (bottom) is a physical suspension model while the right one (top) is reduced.

The left suspension is kept physical while the right one is reduced.

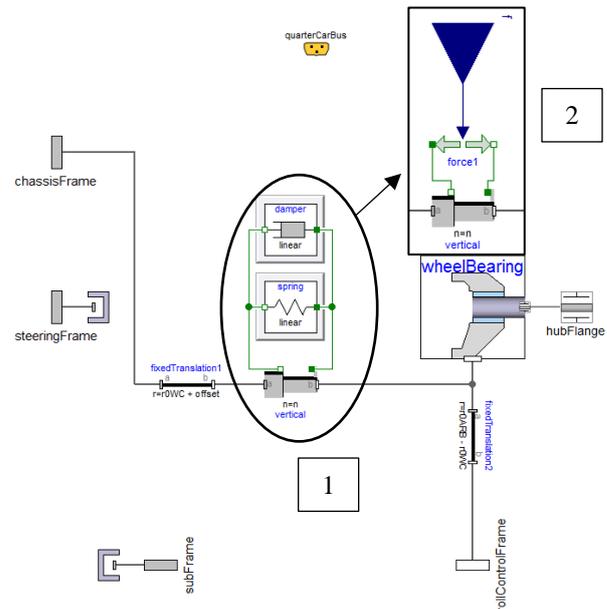


Figure 13. Physical suspension with spring and damper (1) and its reduced equivalent using a force source (2).

The suspension model only allows a vertical degree of freedom. It uses a linear spring and a linear damper. The fast oscillations that can happen when running this model are computationally very expensive.

In the reduced suspension model, the spring and damper are replaced by an ideal force source fed with the force value read at the full suspension model’s flange. The rest of the model, which is not computationally very intensive, is kept identical between the left and right side of the vehicle.

We test the suspensions on a test bed with a trapezoidal position input at both hubs.

In this ideal experiment, where the desired behaviour of the suspensions is exactly similar on both sides, the results of the reduced model match perfectly those of the full model. When tested in a vehicle, the forces and torques applied to the left and right suspension hubs will be slightly different, even during a straight-line manoeuvre (the effective rolling radius is never equal in all the wheels, the repartition of the vehicle mass is never perfect, etc.). The reduced model will ignore these differences and produce the exact same results on both sides. The inaccuracies being extremely small, they are completely acceptable for the applications targeted in the MORSE project.

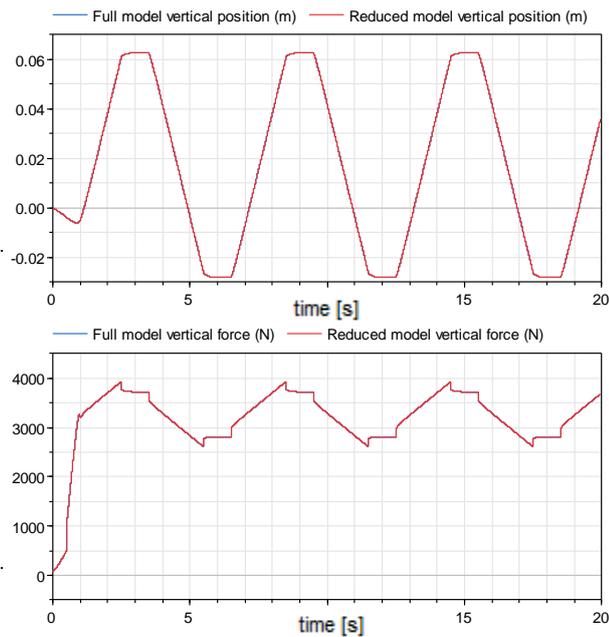


Figure 14. Suspension’s hub vertical position (top graph) and vertical force (bottom graph) for both the complete and reduced model.

d. Wheels

The wheels are reduced in the same way, a force and a torque actuator are used to account for the tyre dynamics in the front left and rear right wheels.

Once again we have to point out that in reality, each one of the tyres would behave slightly differently, even in a straight line. The reduced model ignores these differences and replicates exactly on the right side of the car what happens on the left side.

The reduced wheel model is not presented in detail in this paper since it is generated following the idea as the drive shafts and the suspensions.

4. Results

a. In Dymola

Hardware specifications: computer with Windows 10, processor is Intel® Core™ i7-4790K @ 4.00 GHz Quad-core.

In this section, we run a vehicle with several levels of model reduction on a series of Tip-in/Tip-out manoeuvres in 2nd gear. The levels of model reduction are as follows: Level 1: Full vehicle model. Level 2: Vehicle with reduced transmission only. Level 3: Vehicle with reduced transmission and reduced driveline. Level 4: Vehicle with reduced transmission, reduced driveline and reduced chassis (suspensions and wheels). Level 5: Vehicle with reduced transmission, reduced driveline and reduced chassis and only allowing longitudinal motion.

It is important to note that the interface of all the vehicle models is the same as they need to be able to dialog with the ECU without missing information. The simple vehicle model is thus capable of sending and receiving the same signals as the most detailed one.

The model is run first in Dymola. The simulation lasts for 56s. The solver settings are: Step size = 0.0005s, tolerance=1e-5, inline integration method = implicit Euler. This has indeed proven to be the quickest inline integration method for our application. The step size has been calculated to be the biggest time step that gives correct results when running the crank angle resolved engine model, which is the subsystem that requires the smallest sample rate.

The conditions of the tests are different from the conditions of the tests of the individual subsystems since we wanted to run the vehicle on a real manoeuvre like Tip-In, Tip-Out.

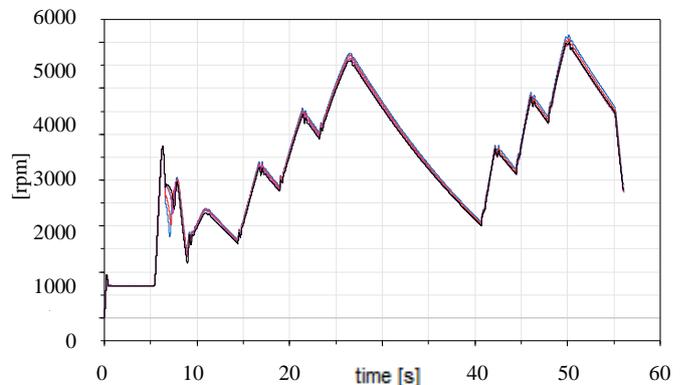


Figure 15. Engine speed. Blue: Level 1, Red: Level 2, Green: Level 3, Magenta: Level 4, Black: Level 5.

The maximum error in engine speed for each level of reduction is respectively: 1.24%, 2.69%, 2.68% and 3.05%.

The biggest error occurs at around 7s when we engage the clutch after engaging 1st gear (i.e. at pull-away when the vehicle starts moving). The magnitude of the error after that moment does not increase, it remains constant until the end of the simulation.

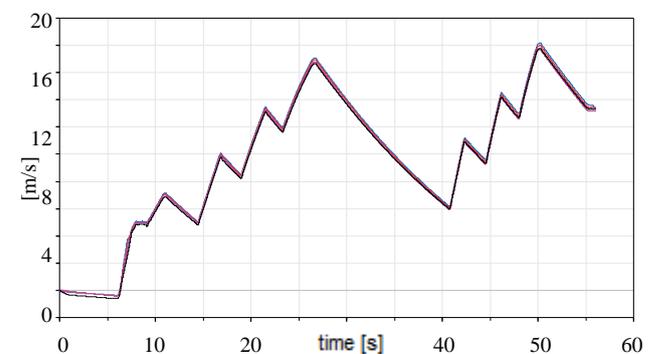


Figure 16. Vehicle speed. Blue: Level 1, Red: Level 2, Green: Level 3, Magenta: Level 4, Black: Level 5.

The maximum error in vehicle speed for each level of reduction is respectively: 1.01%, 2.41%, 2.40% and 2.60%.

We see a slightly negative vehicle speed at the beginning of the experiment, when the engine is idling and the vehicle is in neutral. This is attributable to the Pacejka tyre model which is inaccurate at very low vehicle speeds.

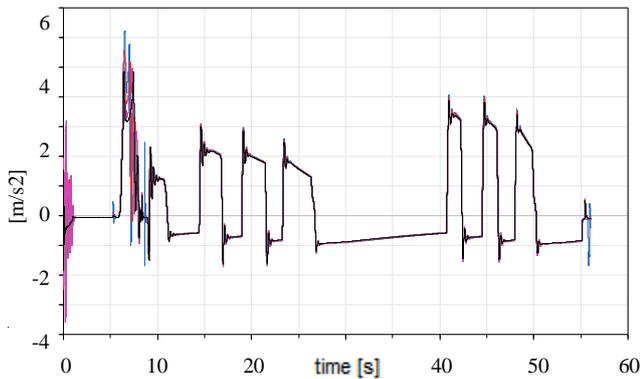


Figure 17. Vehicle acceleration. Blue: Level 1, Red: Level 2, Green: Level 3, Magenta: Level 4, Black: Level 5.

The acceleration plot shows good correlation between the models. There are oscillations at the beginning due to non-optimal initialisation and during clutch engagement at 7s.

In the table below, a time overrun happens when a time step in Dymola lasts longer than the corresponding amount of time in real life. For example, if we choose a step size of 0.5 ms, it should take less than 0.5 ms for the hardware to perform all the calculation before moving to the next step. Otherwise, all the equations do not have time to be solved before the next step and the results cannot be trusted anymore so this has to be avoided.

Simulation performance summary:

	Simulation time (s)	Number of events	Overruns
Level 1 (Full model)	62.2	46	>50
Level 2	42.5	26	>50
Level 3	36.2	28	>50
Level 4	33.8	27	>50
Level 5 (Fully reduced model)	16	16	12

We can see from the table above that each level of reduction improves the simulation time. The number of

overruns of the first four models is quite high and even though it seems to decrease when we reduce the model, it remains too high for the model to be tested in HiL.

b. In Hardware-in-the-Loop

Hardware specifications: dSPACE DS1005 PPC with 4 cores available.

The simplest and the most detailed vehicle models are run on the HiL rig with a step size of 0.0005 s. Due to time constraints, we only tested two vehicle models in HiL, the most detailed one and a reduced one.

The most detailed vehicle is the full vehicle model (i.e. the Level 1 vehicle in the last section).

The simplest vehicle is essentially the fully reduced vehicle (i.e. the Level 5 vehicle in the last section) with an elasto-plastic based friction model to reduce the number of events and thus the number of time overruns. We could see indeed that if the Level 5 vehicle was running very fast in Dymola it still generated a few overruns. The elasto-plastic clutch uses a single state and defines the friction in a continuous way without introducing events (Dupont P., 2002).

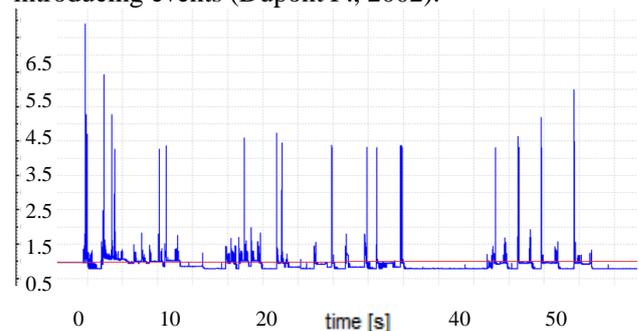


Figure 18. Most detailed vehicle's turnaround time (ms) (blue) and target step size (red).

Maximum turnaround time: 7.4 ms
 Minimum turnaround time: 0.28 ms
 Number of overruns: 55

This very detailed vehicle model is on average too slow on the current hardware and cannot achieve real-time performance. It also generates a high number of overruns.

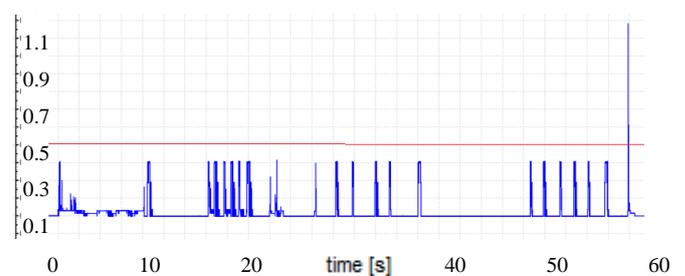


Figure 19. Simplest vehicle's turnaround time (ms) (blue) and target step size (red).

Maximum turnaround time: 1.18 ms
 Minimum turnaround time: 0.096 ms
 Number of overruns: 1

This simplified vehicle model is achieving real-time performance and generates only one overrun. The cause of the overrun is under investigation.

A comparison of the key variables is not very relevant here since, due to the high number of overruns, the results of the most complex model are rapidly drifting. However, despite these inaccuracies, the results are still matching well. The manoeuvre starts at time=0s, what happens before this time can be ignored.

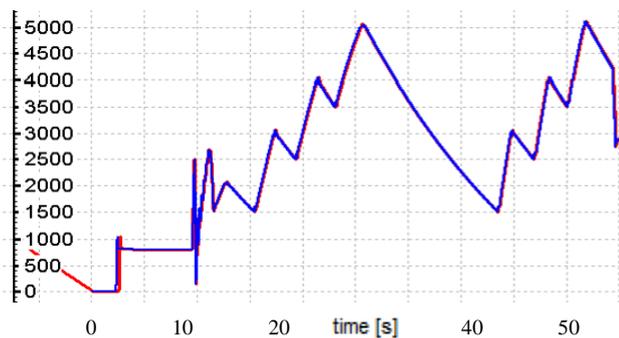


Figure 20. Reduced vehicle's engine speed (rpm) for single core (blue) and multicore (red) implementation.

The results are slightly different from what we got in Dymola because of a change in vehicle parameterisation (tyre and aerodynamic drag have been increased in the vehicle tested in Dymola, hence smaller vehicle speeds). Due to time constraints, a second test of the model in the HiL environment has unfortunately not been possible. The point here is not to compare the results between Dymola and HiL but rather to compare the results of the different vehicles.

The multicore capability will also be investigated in more detail in future work; it does not show a real benefit here since the controller we used is the software version and thus is not very CPU demanding. The crank angle resolved engine model has been, as part of this project, split into three sub models (Gallagher S., 2016): the mechanics part (included in the vehicle model), the combustion part (one for each cylinder) and the air path. We thus have 5 s-functions for the engine and vehicle to run and 4 cores available. Along with these models are the driver model and the CAN buses that also have to be run on these 4 cores. At the time of writing of this paper, it was still undecided how the repartition between the cores would be done.

5. Conclusion and Future work

Model reduction techniques for all the vehicle subsystems have been implemented and tested. The accuracy of the results is satisfying and the improvement in performance significant. The level of detail in the

chassis and driveline has been maintained the same as in the full models. While the reduced transmission has lost the 3D capability, it still outputs the correct speed and torque for all operating points. However, the fully detailed model is still very much needed. It is important to be able to model a vehicle with a physical representation and a high level of detail to accurately predict the vehicle behaviour to then be able to calibrate the reduced model.

A series of assumptions and simplifications have of course had to be made. The main one is that the results would be the same on the left and right hand sides of the vehicle since we test it in a straight line. This assumption is acceptable in the MORSE project as the small inaccuracies are acceptable. Moreover, we thought it was more interesting to compromise on the left/right discrepancy but to keep the same level of detail in the model rather than reducing the capability of the subsystems to maintain the models physical on both vehicle sides which is not of prime importance in a straight-line test.

More testing needs to be done in the Hardware-in-the-Loop environment: we need for example to test the vehicle over other manoeuvres than Tip-in/Tip-out, to include the detailed Dymola engine model and to explore further the multicore capability.

The reduced models (except the gear set) are multi-body and could be simplified further to one-dimensional subsystems if needed in order to still be able to achieve real-time performance once we will have integrated the Dymola engine in the vehicle.

References

- Gallagher S. *et al.* (2016) *Model-based Real-time Systems Engineering*, Loughborough, England, Powertrain Modelling and Control Conference.
- Dempsey M. *et al.* (2006) *Coordinated automotive libraries for vehicle system modelling*, Vienna, Austria, Proceedings of the 5th International Modelica Conference.
- Dempsey M. *et al.* (2009) *Investigating the Multibody Dynamics of the Complete Powertrain System*, Como, Italy, Proceedings of the 7th Modelica Conference.
- Dupont P. *et al.* (2002) *Single State Elasto-Plastic Friction Models*, IEEE Transactions of Automatic Control.