Dymola and Simulink in Co-Simulation: A Vehicle Electronic Stability Control case study

Theodor Ensbury[1] Nate Horn[1] Mike Dempsey[2]

[1]Claytex USA, Inc., United States, {theodor.ensbury,nate.horn}@claytex.com

[2]Claytex Services Ltd. Edmund House, Rugby Road, Leamington Spa, CV32 6EL, UK, mike.dempsey@claytex.com

Abstract

Deploying models as FMUs using the FMI standard interface is a common task; co-simulation enables models to be simulation environment agnostic. Users may wish to deploy models outside of the development environment for several reasons. One such example is control system development; Simulink is a popular choice for control development, but Dymola is the superior environment for multibody simulation. This paper presents a case study regarding deployment of a full multibody VeSyMA vehicle model developed in Dymola, deployed within Simulink, for the purpose of Electronic Stability Control (ESC) system development. Correlation of results in both environments indicates the validity of the vehicle experiment FMU generated in Dymola, when deployed in Simulink.

Keywords: FMI/FMU, Control system development, Electronic Stability Control

1 Introduction

Traditional, prototype centric vehicle development practices have been increasingly replaced by simulation in recent years. Offering multiple tangible benefits, from an overall reduction in development time and cost stemming from eliminating ancillary activities (travel, producing prototypical vehicles, systems etc.), to providing a consistent and repeatable virtual testing environment, simulation has become an integral element of Original Equipment Manufacturer (OEM) development processes. As far back as 2002, then DaimlerChrysler were deploying simulation with the express intention of expediting vehicle chassis control algorithm development (Quan-Zhong et al, 2002).

Beyond OEMs seeking to streamline engineering workflows, simulation is also becoming increasingly employed by regulatory bodies for homologation purposes. Presented by Adam Opel AG and IDIADA Automotive Technology SA in 2012, ECE Regulation 13-H (EU market) stipulates the requirement of passenger vehicles to be equipped with a brake actuated Electronic Stability Control (ESC) system. Whilst the primary or base model requires physical homologation, model variants can be homologated using simulation (Hahn et al, 2012).

1.1 Simulation tool selection challenges

Active vehicle dynamics problems, like active chassis control systems, present a two-domain problem. Electronic Stability Control (ESC) systems, where the controller computes the vehicle handling state, will apply a corrective yaw moment through a physical actuator, such as a/the brake(s). Selecting a single tool in order to approach the problem of developing an ESC control strategy is difficult, as the tool selected needs to be capable of not only accurately modeling the vehicle system dynamics, but also the controller.

Simulink, from MathWorks, is often chosen when developing control systems, owing to extensive signal processing libraries and code generation options. An acausal, multidomain tool with a strong multibody code, Dymola is ideally equipped to tackle the challenge of simulating the whole vehicle, as a combination of physical mechanical, fluid, thermal and electrical components.

1.2 Using FMI/FMU to deploy Dymola and Simulink together

Harnessing the FMI (Functional Mockup Interface) standard, the strengths of both Simulink and Dymola can be deployed together, without having to sacrifice model fidelity or simulation accuracy by only using one of the two tools.

Not limited to the interface of Dymola and Simulink, FMI provides an independent standard for the interface of simulation models (FMI, 2019). This enables a model developed in one tool to be packaged up as an FMU (Functional Mockup Unit) and deployed within a second tool. In this case, FMI enables a Dymola based vehicle model to be deployed within Simulink, to enable the controller to be developed using a detailed, physically representative vehicle model.

Deploying FMUs thus provides a robust solution to the problem of creating a detailed vehicle model for use within Simulink. Cheng et al (2010) studied multiple tools before deciding that utilizing both Dymola and Simulink together in Co-Simulation was the optimal approach to creating a powertrain modeling structure for Hybrid Electric Vehicles (HEVs). Control work was undertaken in Simulink (due to its comparative strength in this field versus Dymola) and the physical modelling undertaken in Dymola, citing that due to the complexity of HEVs, a more capable physical modeling tool compared to Simulink would be required. Whilst a HEV and standard internal combustion engine (ICE) vehicle equipped with ESC are not directly analogous, they share a similar trait in that they are both physical systems interfacing with control algorithms.

Therefore, using the example of developing an ESC model in Simulink, and a vehicle test experiment in Dymola, this paper will present the suitability of FMU deployment with Co-Simulation when utilizing Dymola models in a Simulink environment.

2 Modeling

The modeling requirement for the case study presented in this paper can be broadly broken down into 3 separate components. The modeling of the vehicle (in Dymola), the experimental setup (in Dymola) including top level interface to the ESC model, and the modeling of the electronic stability controller in Simulink itself.

All models and components modeled in Dymola are derived from models found in the VeSyMA suite of libraries from Claytex, presented previously (Hammond-Scott and Dempsey, 2018). Forming a complete vehicle level simulation solution, the suite is built upon the VeSyMA library itself, which comprises model templates and interfaces, in addition to supporting straight-line simulation. Subject specific libraries are built as extensions of the VeSyMA library, with a variety of areas covered such as VeSyMA – Suspensions, for detailed analysis of vehicle dynamics including 3D multibody suspension systems, high-fidelity tyre models, road and driver models; VeSyMA – Engines, a library devoted to high-fidelity internal combustion engine models; VeSyMA – Powertrains, a library of full 3D multibody driveline and gear train models; VeSyMA – Driver-in-the-Loop, which enables the deployment of vehicle models developed using the VeSyMA – Suspensions (or VeSyMA – Motorsports) libraries in Driver-in-the-loop (DiL) real time virtual environments, such as rFPro (Ensbury et al, 2019).

2.1 Vehicle Model – Dymola

Found in the VeSyMA – Suspensions library, the vehicle model used for this case study was the SmallFamilyMT example. As the name indicates, this vehicle model is a generic example of a small family (C – segment/compact car) vehicle. It is equipped with an idealized manual transmission, mated to a mapped petrol (gasoline) engine model. No front engine ancillary drives (FEADs), lubrication or fuel tank is modeled (a fuel mass is included separate to the chassis). A 1D rotational driveline model supplies drive torque to the front wheels only, through an open 1D differential model. Driveline rotational inertias were included.

Free to move in all 6 degrees of freedom (roll, pitch, yaw; lateral, longitudinal and heave), the vehicle

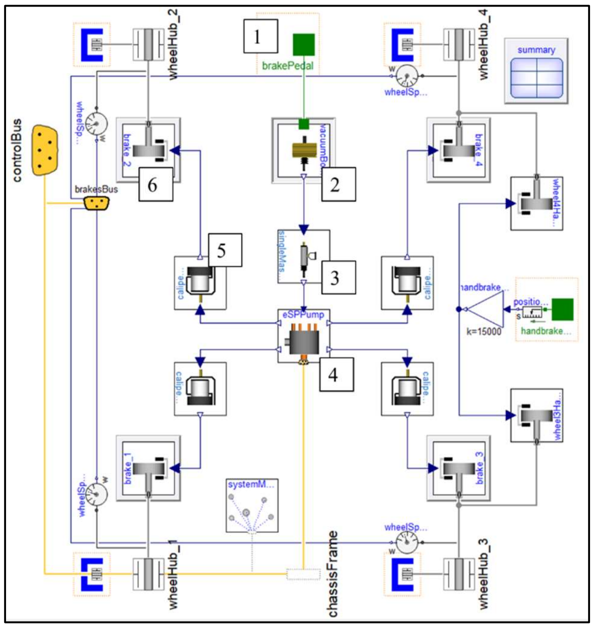

Figure 1 - Pseudo-hydraulic brake model compatible with ESC algorithm. A position input from the driverEnvironment (1) is fed into the vacuumBooster (2). This is output as a real signal to the master cylinder (3), becoming a real line pressure. The ESPpump (4) applies the additional pressure demand from the ESC algorithm along with the normal line pressure to the caliperPiston (5). A force is then applied from the caliperPiston (5) to the brake disc models (6).

platform comprised of full multibody double wishbone suspension at the front, with full multibody multilink rear suspension linkages. Both front and rear anti-roll bars (ARBs) were fully functional multibody models as well. Each specific suspension body had its own mass and inertia, including the ARBs. 1D translational force generation models were used for the springs and dampers, applied to the linkages through multibody mounts. No suspension compliances (bushings, lumped compliances etc.) were included. A non-linear, fully combined lateral and longitudinal slip tire model was also used, comprising of a Modelica implementation of the Pacejka Magic Formula (corresponding to MF6.2). Asymmetric tire behavior due conicity and ply-steer was included, with a kelvin spring damper vertical dynamics model (Pacejka, 2012). A pseudo-hydraulic braking model was utilized, illustrated in Figure 1. Brake pedal position, converted from a normalized driver model demand to a pedal position within the driverEnvironment model, actuates a table based vacuum booster (servo) model responsible for converting the pedal position to a plunger force, of type real. Brake line pressure is calculated in a single master cylinder model, from the cylinder piston diameter. Each wheel features a specific brake model, including

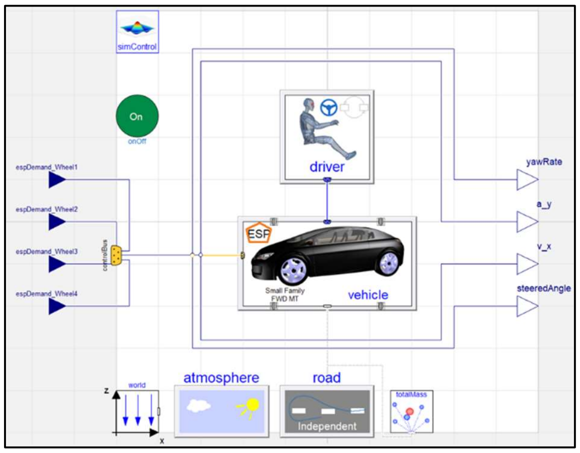

Figure 2 - The experimental setup prior to exporting as an FMU. Note the control bus inputs and outputs, these will become the FMU inputs and outputs when exported.

rotational inertia and mass effects. Pad/disc friction is accounted for by a controlled elasto-plastic friction model (Dankowicz, 1999). Normal force is used to control the pad friction, determined by the piston diameter in the caliperPiston model, which emulates the action of the piston within the brake caliper. No hydraulic effects were included, as the model was real-signal based. Four real control signals were present on the brakesControlBus (this is part of the Control Bus, therefore it is not shown in Figure 1, as only the Control bus connection is required to the ESPpump model), which indicates the normalized brake demand from the ESC algorithm regarding each individual brake model. Actuation of this demand was undertaken by adding brake pressure value to the system, before the caliper piston model itself. Gains within the ESPpump model convert the demand into a pseudo line pressure, which is then added to the brake line pressure generated as a result of the driver model actuating the brake pedal flange. Divide by zero protection is also present, preventing the ESC algorithm from applying a negative pressure to the brake disc friction model. When translated, 97 states are selected within the vehicle model.

2.2 Experimental setup – Dymola

To enable the assessment of an ESC algorithm, an experiment model was created in Dymola to push the vehicle close to the lateral adhesion limit. A slalom test was chosen for this task, with a closed loop driver model following the slalom path whilst holding the vehicle at the target speed of 14.5m/s. 11 slalom cones were placed 15m apart starting 50m from the vehicle initialization point, with the driving line a constant offset 1.25m away from the cones. No roughness or gradient were included in the road model. To accentuate difficult handing conditions, an 80kg boot/trunk payload was added to the vehicle.

In order to support interfacing an external ESC algorithm with the vehicle model, the ESC control demands for each wheel were represented by input classes at the top level of the experiment model, being fed onto the controlBus. A similar method was used to expose the vehicle system variables, written onto the controlBus inside the vehicle model, required by the ESC algorithm calculation, as per Figure 2.

The Cvode – variable order solver was chosen, as it does not require a further (Binary Model Export) license to export from Dymola (Dassault Systemes, 2019). Solver tolerance was 0.0001; results were logged at an interval of 0.01s. Experiment duration was 15s. Within the experiment model, 112 states were selected; 15 were selected in the driver model, the remaining 97 in the vehicle.

2.3 Electronic Stability Controller

For this example, a simple feedforward controller was created, where the vehicle yaw rate is compared to a reference yaw rate calculated by the controller. The controller presented here is a Simulink implementation of the brakeBased ESC controller model available in the

Suspensions library. Parameterisation was consistent between the Modelica and Simulink versions. Any error between the ideal yaw rate and the actual yaw rate of the vehicle will cause a control response from the controller. The ideal yaw rate is given by (Karogal and Ayalew, 2009), based upon the steered angle (δ), longitudinal velocity (V_x), understeer gradient (K_{us}) and wheelbase length (l):

$$r_{desired} = \delta * \frac{V_x}{{V_x}^2 * K_{us} + l} \tag{1}$$

Which is then subtracted from the measured vehicle yaw rate to determine the error. Once this error is calculated, it is then passed to a PD controller to determine the magnitude of the corrective braking action required by the controller. Brake command distribution to specific wheels is achieved using a logic table, dependent upon the lateral acceleration and understeer gradient signs. The understeer gradient (K_{us}) is calculated in the following way (Milliken and Milliken, 1995), from the steered angle (δ), longitudinal velocity (V_x), lateral acceleration (a_y) and wheelbase length (l):

$$K_{us} = \frac{\delta}{a_y} - \frac{l}{V_x} \tag{2}$$

A rudimentary protection based upon the steered angle and lateral acceleration of the vehicle was also included, to prevent ESC interference in straight-line driving. This was implemented in Simulink by making the ESC

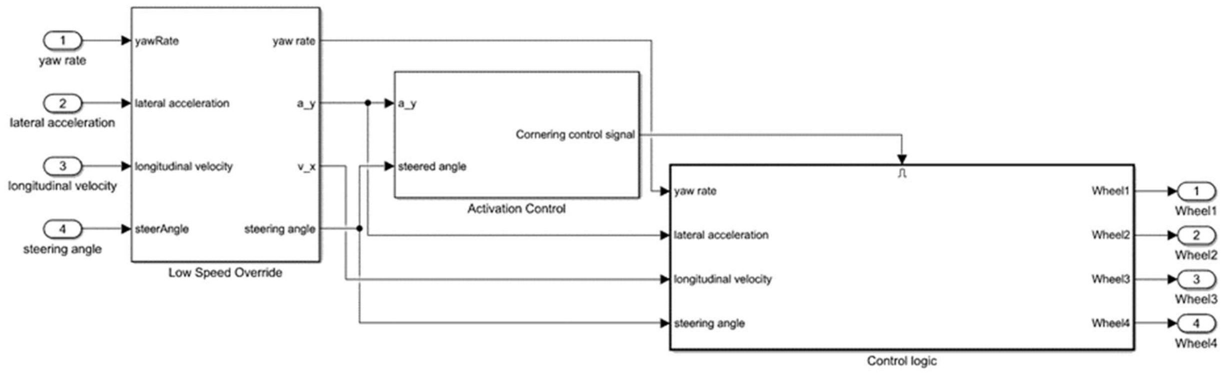

Figure 3 - The ESC algorithm model in Simulink. Note the Control logic subsystem features an enable input

control logic an enabled subsystem, see Figure 3, only enabled when the vehicle was deemed to not be travelling at low speed or in a straight line.

3 Simulation and results

3.1 Dymola – baseline

A baseline result was undertaken, using the Modelica implementation of the ESC algorithm, found within the VeSyMA – Suspensions library, simulated solely within Dymola. Experimental setup was exactly as listed in section 2.2. Cvode solver was used with a tolerance of 0.0001; results were logged at an interval of 0.01s. Two sets of results were taken; with and without an active ESC controller.

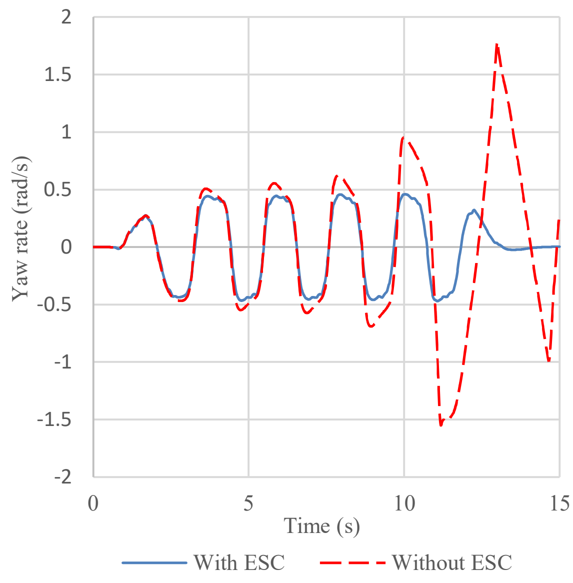

Figure 4 – Yaw rate comparison of Dymola baseline model with/without ESC

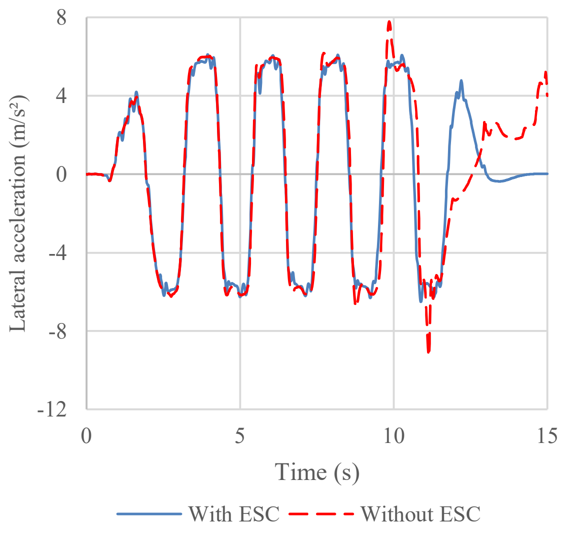

Figure 5 – Lateral acceleration comparison of Dymola baseline model with/without ESC

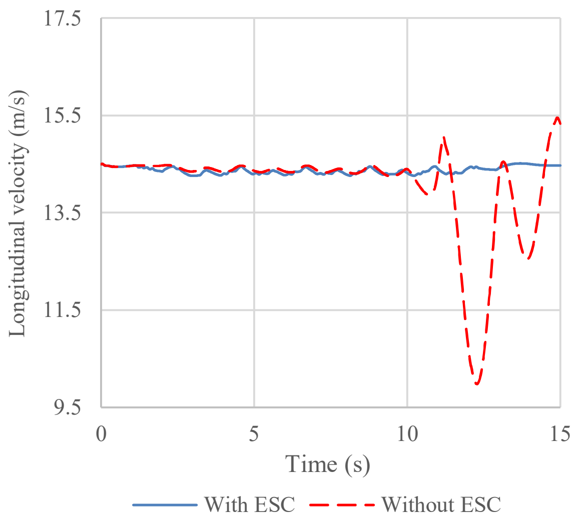

Figure 6 – Longitudinal velocity comparison of Dymola baseline model with/without ESC

Reviewing Figures 4, 5 and 6, the inherent instability of the vehicle model undertaking the test is apparent, as is the success of the ESC algorithm in constraining the vehicle yaw rate and preventing the vehicle from oscillating out of control at the end of the test.

3.2 Co-Simulation in Simulink

Testing of the co-simulation function within Simulink was conducted by exporting the experiment model, complete with vehicle model from Dymola. Version 2.0 Model exchange, and Co-simulation using Cvode were the options chosen, with resources copied to the FMU to enable the inclusion of the road file. String parameters were exported, with model description filters for protected variables and auxiliary variables also included. Only 64-bit binaries were compiled. Finally, the dialog was displayed during export to ensure the required resources were included.

Within Simulink, the FMI kit from Dassault Systemes was used to deploy the experiment FMU. 112 continuous states, 222 event indicators and 94727 variables were present in the FMU when loaded into Simulink. As indicated, the co-simulation option was selected, enabling the Cvode solver to handle the simulation of the vehicle and the experiment. Interestingly, no delay blocks were required to break the loop between the vehicle model and the controller within the Simulink environment. Simulink VariableStepAuto (tolerance 0.001) was used as the time integration option for the Simulink environment. Vehicle model results were logged to MatLab every 0.001s.

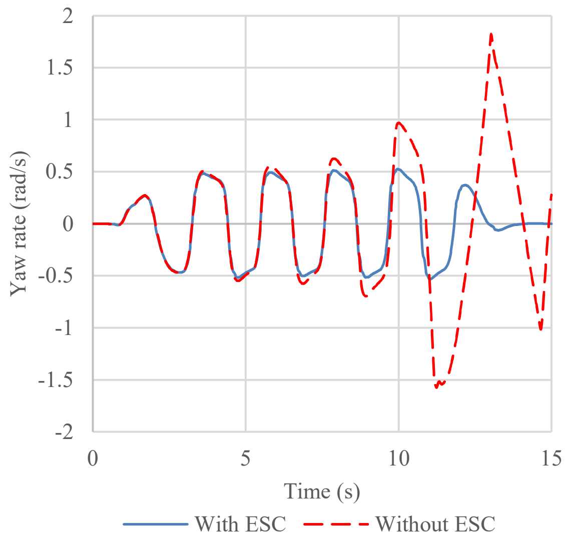

Figure 7 – Yaw rate comparison of FMU in Simulink with/without ESC

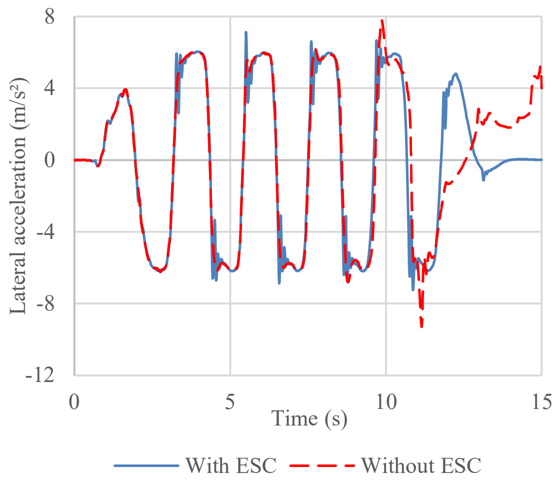

Figure 8 – Lateral acceleration comparison of FMU in Simulink with/without ESC

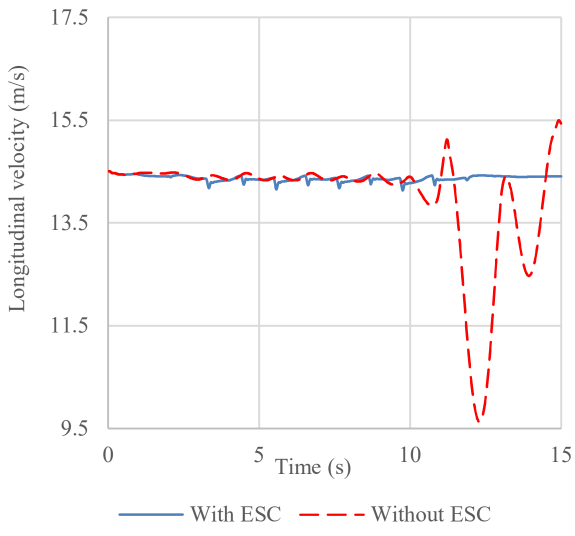

Figure 9 - Longitudinal velocity comparison of FMU in Simulink with/without ESC

Initial review of Figures 7, 8 and 9 (with and without an ESC model attached) indicate the simulation is performing as expected. Trends surrounding the vehicle performance consistent with the behavior displayed in Dymola.

3.3 Results comparison

To understand the difference in simulation performance arising from exporting the vehicle experiment as an FMU to Simulink and interfacing it with an external controller, several comparison plots were generated.

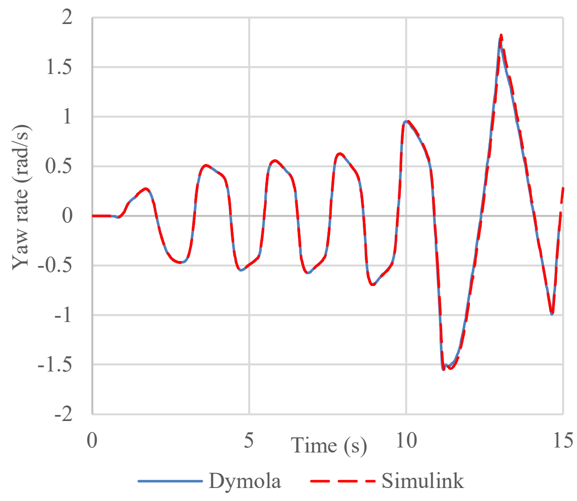

Figure 10 – Yaw rate comparison. Dymola baseline (no ESC) and Dymola plant FMU in Simulink (no ESC).

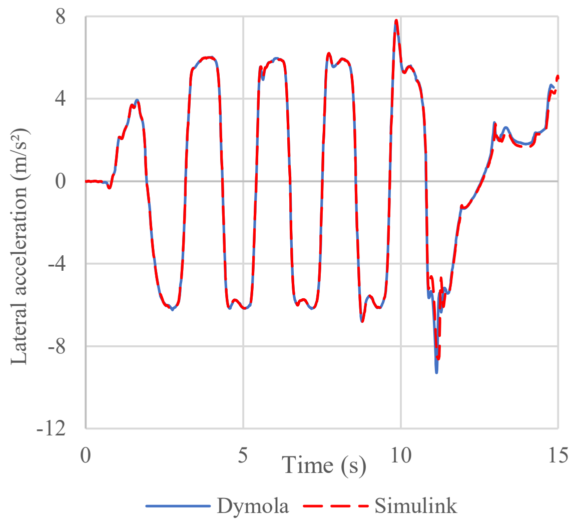

Figure 11 - Lateral acceleration comparison. Dymola baseline (no ESC) and Dymola plant FMU in Simulink (no ESC).

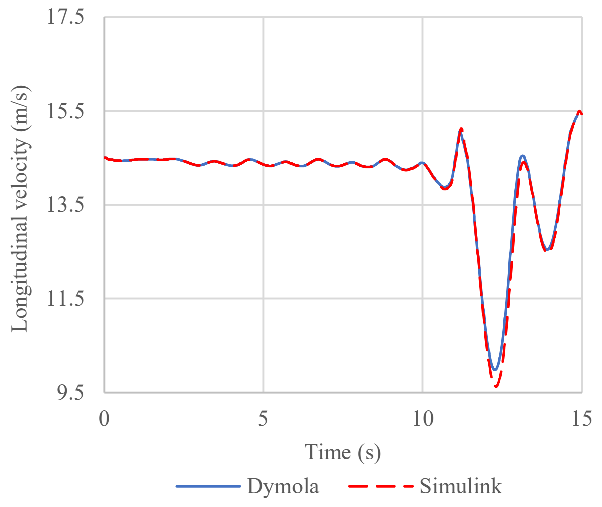

Figure 12 - Longitudinal velocity comparison. Dymola baseline (no ESC) and Dymola plant FMU in Simulink (no ESC).

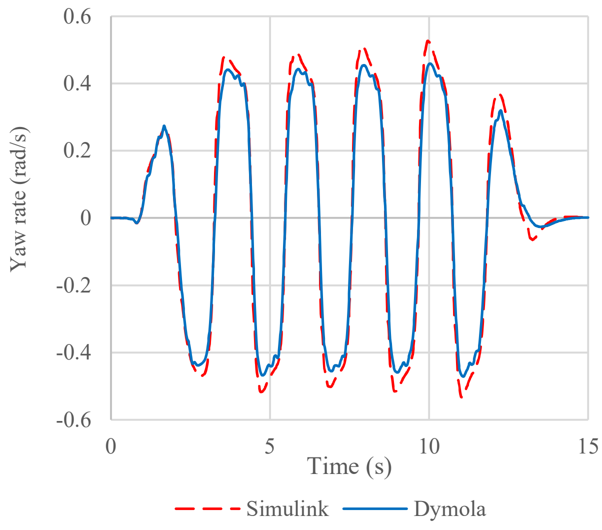

Figure 13 - Yaw rate comparison with ESC. Dymola baseline (with ESC) and Dymola plant FMU in Simulink (with Simulink ESC).

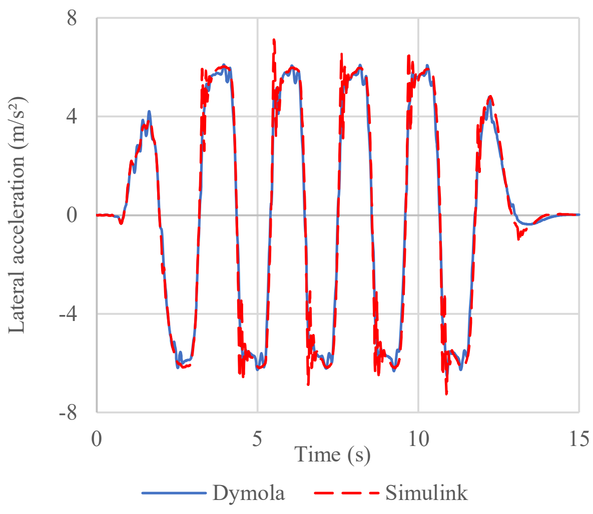

Figure 14 – Lateral acceleration comparison with ESC. Dymola baseline (with ESC) and Dymola plant FMU in Simulink (with Simulink ESC).

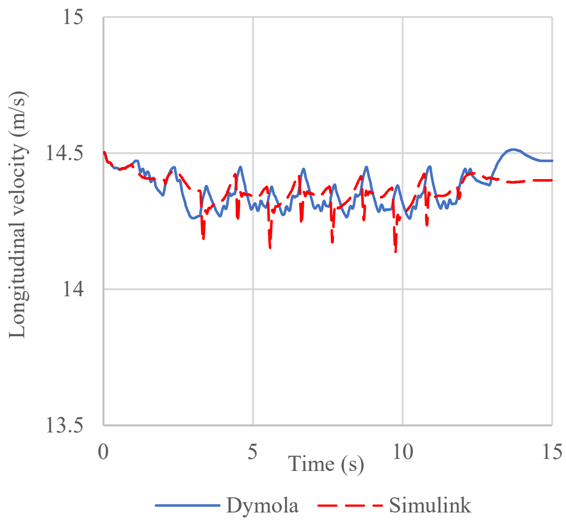

Figure 15 – Longitudinal velocity comparison with ESC. Dymola baseline (with ESC) and Dymola plant FMU in Simulink (with Simulink ESC).

3.4 Discussion of results

Review of the plots presented as Figures 10-15 immediately indicates there is a small difference in results, when comparing the vehicle experiment running in Dymola (without ESC, Figures 10, 11 and 12; with ESC, Figures 13, 14 and 15), and when running as a deployed FMU within Simulink (also). Present near the end of the test, where the vehicle is sliding/oscillating out of control, this is most markedly shown in the longitudinal velocity comparison. As there is no ESC controller attached to either of the tests in these results, effects from the coupling of the ESC model and the FMU can be discounted as the source of minor difference. Therefore, the logical explanation would be that the difference is due to the Simulink variable step solver interacting with the function of the Cvode solver exported from Dymola, within the FMU, likely with regards to event handing given the location of biggest disparity.

Moving onto the comparisons between the experiment with ESC (within Dymola) and the Simulink ESC implementation (connected to the FMU of the experiment), some small differences can again be observed. Generally, the vehicle model appears to be more unstable when controlled by the external controller in Simulink, versus the internal controller in Dymola. Higher, very localized peak lateral accelerations coupled with a higher yaw rate and larger drop in longitudinal velocity suggest this. Due to the differences in the test run without ESC above, some difference would be expected. Although the trends between the two tests correlate, indicating the ESC is interacting with the vehicle in the same manner, there is enough difference in the results to warrant comment. Some element of this error is likely to have been caused by the fact that the ESC controller is implemented within Simulink and thus is now being solved by the Simulink solver rather than the same solver as the vehicle model. Effectively, this means the controller would be running at different refresh rate between the two tests. Furthermore, as the two controller models were written in different languages (one in Simulink, one Dymola/Modelica), then there will also be a built-in difference in performance due to this. The next point to consider is the action of the ESC controller on the actions of the solver in each environment. When combined within the same model in Dymola, any action of the controller, such as zero crossing or state events, will trigger solver functions which affect the results and simulation of the whole model. Expanding upon this, it is worth remembering that the translated model from Dymola will differ depending on whether the ESC model is present or not. What this effectively means, is there is a possibility of the vehicle experiment model having slight differences in structure when comparing the FMU + Simulink ESC controller versus the native Dymola implementation. Instinctively, differences in model structure would likely lead to minor differences in results. Finally, it is worth remembering that the driver model within the experiments function as closed loop; differences in vehicle response will be amplified by a different driver model response.

3.5 Comments on CPU time

One of the biggest hurdles regarding using external FMUs when creating larger experiments is the effect on the CPU time, in part due to the overheads associated governing the interaction of the models.

Table 1 - Comparison of CPU time for various tests

Test type	*CPU time(s)*
Dymola (no ESC)	22.156
Dymola (with ESC)	51.518
FMU in Simulink (no ESC)	21.615
FMU in Simulink (with ESC)	128.1

Review of Table 1 indicates the expected result, of the vehicle experiment FMU running in Simulink (connected to a Simulink implementation of ESC) being the slowest, with the Dymola simulation with the Modelica ESC model being the next slowest. The margin of difference between the simulation time of the FMU examples in Simulink compared to that of the native Dymola implementation indicated the extra simulation overhead associated with syncing the two models together during time integration (Wang, et al 2017). Potential actions to minimize the simulation time penalty include evaluation of various Simulink solvers (a discrete fixed step rather than a variable step, due to the simple procedural nature of the ESC mathematics) and optimizing the sampling/exchange rates between the plant and the controller model. Both methods have been demonstrated to have significant impact on simulation time by Wang, et al (2017). As the simulation speed was not of primary importance to this case study, no actions were taken to optimize the execution speed, therefore one can assume the simulation time results presented can be improved upon with updated settings.

4 Conclusions

Overall, it can be concluded that the vehicle models, and experiment tests found within the VeSyMA – Suspensions library can be deployed effectively within Simulink as FMUs. This enables the end user to be able to take advantage of the strengths of each simulation environment, not being constrained to one or the other. Furthermore, the model developer in Dymola can be assured of the suitability of deploying vehicle models to other simulation tools which support FMI.

Differences in results can be seen to be minor, considering the complexity of the case study example and the level of interaction and integration between the ESC algorithm and the vehicle response. Whilst the vehicle in the Simulink deployment does appear to be closer to the edge of the control envelope, it is within a demonstrable state of control compared to running without an ESC algorithm. No experimentation with solver setting was made during this work, so it can be hypothesized that such differences in results could be tuned out by modifying tolerance or experimenting with some of the different solvers available within Simulink.

4.1 Further work

To understand more about the basic dynamics existing between the vehicle experiment FMU and the controller model, it would seem prudent to create a controller bench test, comparing directly the results of the controller models running in their native environments of Dymola and Simulink. This would enable any difference in the function of the controller with respect to the simulation environment to be isolated from the full vehicle results. In addition, experimentation using different solver settings would also be of interest to see whether some of the differences in results can be tuned out.

References

C. Cheng, A. McGordon, R. P. Jones and P. A. Jennings (2010). Comprehensive Forward Dynamic HEV Powertrain Modelling Using Dymola and MATLAB/Simulink. *6th IFAC Symposium Advances in Automotive Control, 2010.* DOI: 10.3182/20100712-3-DE-2013.00026

H. Dankowicz (1999). Modeling of dynamic friction phenomena. ZAMM 1999;79:399–409.

Dassault Systemes. 2019. *Dymola: Dynamic Modeling Laboratory User Manual 2, version 26.* Lund: Dassault Systèmes AB, pp 322.

T. Ensbury, D. Briant and M. Dempsey (2019). Virtual Proving Ground Testing: Deploying Dymola and Modelica to recreate Full Vehicle Proving Ground Testing Procedure. *Proceedings of the 13th Modelica Conference, 2019.* DOI: 10.3384/ecp19157

FMI. 2019. *FMI.* Online. Accessed 31st October 2019. Available at: https://fmi-standard.org/

K. Hahn, H. Holzmann, F. Weyer, M. Roemer, J. Webb and S. Bolthauser (2012). Simulation-based Certification of ESC Systems for Passenger Vehicles in Europe. SAE International. DOI:10.4271/2012-01-0235.

H. Hammond-Scott and M. Dempsey (2018). Vehicle Systems Modeling and Analysis (VeSyMA) Platform. *Proceedings of the 2nd Japanese Modelica Conference, 2018.* DOI: 10.3384/ecp18148

I. Karogal and B. Ayalew (2009). Independent Torque Distribution Stratergies for Vehicle Stability Control, SAE International. 2009-01-0456

W. F. Milliken and D. L. Milliken (1995). Race Car Vehicle Dynamics, 1st edition. Warrendale, PA: SAE International, pp 216.

Hans B. Pacejka (2012). *Tyre and Vehicle Dynamics*, 2nd edition. Oxford: Elsevier, pp 356-404.

Y. Quan-Zhong, J. M. Williams and J. Li (2002). Chassis Control Systems Development Using Simulation: Software in the Loop, Rapid Prototyping and Hardware in the Loop. SAE International. 2002-01-1565.

K. wang, C. Greiner, J. Batteh, and L. Li. (2017). Integration of complex Modelica-based physics models and discrete-time control systems: Approaches and observations of numerical performance. *Proceedings of the 12th International Modelica Conference, 2017. DOI: 10.3384/ecp17132527*