Modelling Environment for Holistic Vehicle Simulation

Nikolaos Kalantzis[1], Tom Fletcher[1], Antonios Pezouvanis[1], Kambiz Ebrahimi[1], Mark Cary[2], Bob Lygoe[2]

[1] Department of Aeronautical & Automotive Engineering, Loughborough University, Leicestershire UK,
[2]Ford Motor Company Limited, Dunton Engineering Centre, UK

Abstract: As the complexity of road vehicles increases with time, the importance of CAE tools to the product development cycle increases as well. A holistic vehicle simulation capability is necessary for front-loading component, subsystem, and controller design, for the early detection of component and subsystem design flaws, as well as for the model-based calibration of powertrain control modules. The current document explores the concept of holistic vehicle simulation by means of developing and testing a Simulink-based multi-disciplinary modelling environment (MME), modular in nature and capable of connecting to subsystem models developed in different environments, thus supporting holistic vehicle simulation on a company-wide scale. The developed environment is tested via the integration of subsystem models built in different commercial software packages within the environment. The simulation results generated from equivalent vehicle models developed in three competing platforms are compared and the advantages and limitations of the different methods of model integration to the master holistic vehicle simulation are discussed.

Keywords- co-simulation, heterogenous, simulation environment, model integration, automotive

1-Introduction

Computer Aided Engineering (CAE) involves the replacement of physical prototypes with mathematical representations of an engineering system with the intent of testing these mathematical models in the virtual world and extracting results that are relevant and applicable to the real world [1]. This enables the various research & development departments within an engineering entity to reduce the number of physical prototypes, and generate optimal component or subsystem designs and test them well before neighbouring components or subsystems are available (front loading) [2], [3], [4]. In addition, the increasing complexity of modern conventional, electric, and hybrid powertrain [5], [6], combined with the tendency of automotive companies to increase the offerings of a vehicle model increases the calibration data volume requirements tremendously [1]. Therefore, the implementation of CAE tools has the potential to reduce the cost and time of vehicle development while allowing for better designs [3].

Addressing the dependability of subsystem performance on their interaction with the whole system is very important, and for this reason, designing the subsystems to work in synergy is the current design approach for an integrated design approach and an early detection of design flaws [4], [7], [8], [9]. It is more common than not for each department to use CAE tools that are best suited for the application [1], [2], and while this makes sense considering the need for discipline-specific component libraries and numerical solvers, a heterogenous collection of subsystem models built in non-compatible environments is in most cases not useful for carrying out a holistic vehicle simulation. Thus, the usefulness of multi-domain simulation is recognized within the automotive industry [5].

Multi-domain simulations may not just involve different modelling tools but also different time scales [7] and required numerical solvers [8]. The need for a holistic vehicle simulation is currently being addressed by engineering software developers through the integration within their software of interfaces that allow for the connection of a model with models built in other engineering software either via model import or model export.

1.1.Platform Connectivity

One of the most common methods of connection to other platforms is through the export or the import of an FMI compliant model [1], [2]. There are currently two generations of FMI, and each generation is subdivided to model exchange (ME) which runs on the solver of the host, and co-simulation (CS) which is compiled with its dedicated solver.

Another very popular method of connection between two platforms is via tool coupling co-simulation [3], [6], [10]. Under this method, each model is simulated within its native platform with one of the involved platforms assuming the role of the master model setting the simulation pace and managing the flow of signals, and one or more connected platforms assuming the role of the slave models. Most automotive simulation platforms support co-simulation with Simulink in the role of the master but also the slave model.

1.2.The need for a generic holistic vehicle simulation environment

While one may argue that the connection between two simulation platforms via a combination of FMU import and co-simulation interfaces may be the fastest route towards a holistic vehicle simulation, the fact that each department tends to use a modelling and simulation platform specific to its area of expertise means that under such arrangement, it is more likely than not for each department to setup their own holistic vehicle simulation with the master simulation running on their chosen software. Such practice may seem to save time in the short term, but it can be the cause of numerical inconsistencies and complicate communication between departments, and in the long run, it may increase the model development and maintenance costs on a company-wide scale. In addition to the above, the design exploration/optimization capabilities of each department are constrained by the capabilities of the design exploration module integrated within the selected master simulation platform.

From the above, one may acknowledge the need for a universal modelling and simulation environment to serve on a company-wide scale as the master model of a holistic vehicle simulation. The modelling environment of a holistic vehicle simulation must possess the following characteristics:

- It must support the connection to a wide array of modelling and simulation platforms
- It must be generic in nature and applicable on a company-wide scale across the vehicle design lifecycle without the need for any changes in the top-level architecture. Therefore, the data buses used for the communication between the main subsystems must support all signals that may possibly be needed by any given department within an automotive company.
- It must be modular and easily adaptable with the user being able to transition from a high-fidelity configuration utilizing the maximum number of signals to a low-fidelity configuration utilizing the minimum necessary number of signals and vice versa under the same general high-level layout and model setup procedure.
- It must be capable of carrying out the master vehicle simulation while running each subsystem model under different numerical solver configurations to cater for the special requirements of different energy domains, model fidelities, and numerical solvers
- It must support complicated simulation control and data post-processing
- Featuring a straightforward method of developing and integrating user code to carry out bespoke simulation control schemes, data post processing tasks, and data visualization would be highly advantageous

1.3.The case for a MATLAB/Simulink based modelling environment

Simulink is a general modelling and simulation tool considered as standard across a very wide array of industries and disciplines. Its wide scope and set of component libraries are wide enough to make for a universal modelling and simulation platform and this may occasionally discourage potential adopters aiming to equip graduate level engineers who may have limited modelling and simulation experience with a holistic vehicle simulation tool. While there may be a few valid points to support these concerns, Simulink has features that can turn the perceived weaknesses into strengths when considered in the role of a holistic vehicle simulation environment:

- Due to the proliferation of the platform, the collective proficiency with the software across the company should be expected to be the highest compared to other modelling simulation platforms
- It features a vast array of built-in and third-party component libraries including signal flow and physical model components (Simscape), and is highly suitable for the development of control strategies
- Simulink simulation can be controlled via MATLAB and this means that the user may take advantage of the vast library of MATLAB functions to implement highly sophisticated simulation control and data post-processing including optimization, surrogate model generation, as well as data visualization
- Most modelling and simulation software tools support the connection of the natively generated models to Simulink master models either through co-simulation (featuring interface blocks for both sides of the co-simulation to enable data transfer) or through the compilation of the native model into a MATLAB S-function which is directly

importable to Simulink. In addition to the above, third party add-ons enable the importation of FMU models to Simulink

- It is possible for the user to build a master model which references the subsystem models running on separate Simulink model files. Such a configuration enables the simulation of a master Simulink model running under a variable step solver while each subsystem runs as a slave model under different numerical solver and time step settings. In addition, it is possible to allocate one slave model per CPU core, thus allowing for multithreaded simulation which has the potential to accelerate the simulation speed dramatically
- The capability of MATLAB to build a Graphical User Interface enables the model builders to build UIs that are tailor made for a specific application, thus simplifying the work flow for the end user considerably and cancelling out the complexity associated with the generic nature of the software

The fact that the combination of MATLAB/Simulink features presented in the list above is not encountered in competing software packages comprises a very strong argument for the use of the platform in the role of a holistic/multidisciplinary vehicle simulation environment that integrates the subsystem models built in specialized platforms within a holistic vehicle simulation. The aim of the current document is the development of a Simulink-based multi-disciplinary modelling environment (MME) to serve as a universal holistic vehicle simulation platform on a company-wide scale. The development of the MME is described in section 2. Section 3 presents the application of the MME using subsystem models built in a variety of commercial engine modelling and powertrain modelling software and discusses and compares the simulation results. Finally, section 4 draws conclusions on the observations made in the sections 2 and 3.

2. Simulink-Based Multi-Disciplinary Modelling Environment (MME)

The main purpose of the developed MME is to integrate the subsystem models built in several different heterogenous platforms, thus enabling for a holistic vehicle simulation. To ensure the applicability of the environment on a company-wide scale, the top-level Simulink block diagram and the involved data busses must be able to accommodate all possible uses of the MME.

The top-level Simulink block diagram of the developed MME is shown in Figure 1. As can be observed, the block diagram consists of three main blocks. The “WAVE & PCM” block, the “Driver” block, and the “Vehicle Harness” block. The blocks communicate with each other using data busses. To allow for the reference to external Simulink slave models, the data bus objects are already saved and are loaded when the master model is loaded. the input and output of ports within slave Simulink models destined to communicate with the master Simulink model via data busses must have their data types within their signal attributes set to the respective data bus. The “WAVE & PCM” is a Simulink Model block within the MME block diagram referring to a separate Simulink model named “WAVE_and_PCM”.

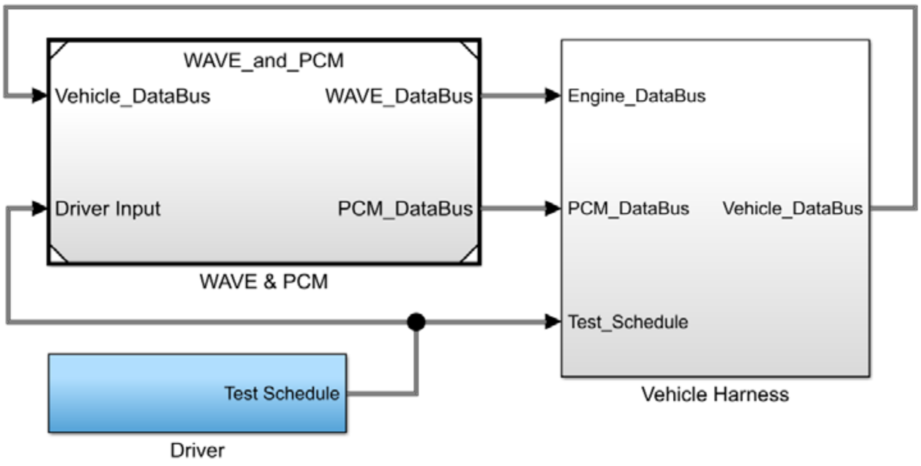

Figure 1 Top-level Simulink block diagram of the MME

“WAVE_and_PCM” model whose block diagram is shown in Figure 2 contains a virtual representation of the Powertrain Control Module (SiL PCM) named “PCM”, and the “Wave Harness” block. The “Wave Harness” block contains a WAVE-RT Simulink interface block which calls a crank angle resolved, real-time capable Gasoline Turbo Direct Injection (GTDI) engine built and parameterized in Ricardo WAVE and exported to WAVE-RT. In addition to the WAVE-RT interface, the “Wave Harness” block includes blocks and connections that convert the signals of the input data buses (PCM_DataBus and VehicleDataBus) to forms that are readily useable by Ricado WAVE engine models. In addition, it contains block diagram sections that calculate additional outputs such as HC and CO emissions, exhaust gas O_2 concentration, BMEP, IMEP, and the breathing efficiency of the engine at the given operating point.

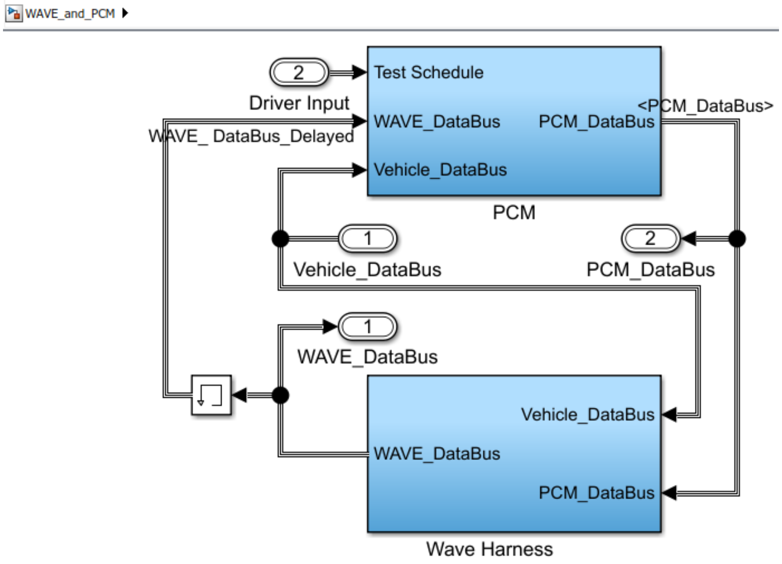

Figure 2 Block diagram of the "WAVE_and_PCM" model

The “Vehicle Harness” block whose content is shown in Figure 3 selects all signals that may be necessary for the operation of a vehicle model from the data buses “Engine_DataBus”, “TST_DataBus”, and “PCM_DataBus” and supplies them to the “Vehicle Model” block. The “PCM_DataBus” input data bus has been included as a place holder for the case in which the engine model is integral to the vehicle model. The “Vehicle Model” block contains the interface block to the vehicle model. An interface may be in the form of a third-party co-simulation interface or a reference block (Simulink Model) to a slave Simulink model containing and running the vehicle model under a different numerical solver configuration than that of the master holistic vehicle model. Alternatively, a Simulink or Simscape vehicle model may be added into the “Vehicle Model” block and ran as part of the master vehicle model. In terms of the vehicle model outputs, the “Vehicle Harness” block combines all output signals of the “Vehicle Model” block into the “Vehicle_DataBus” bus.

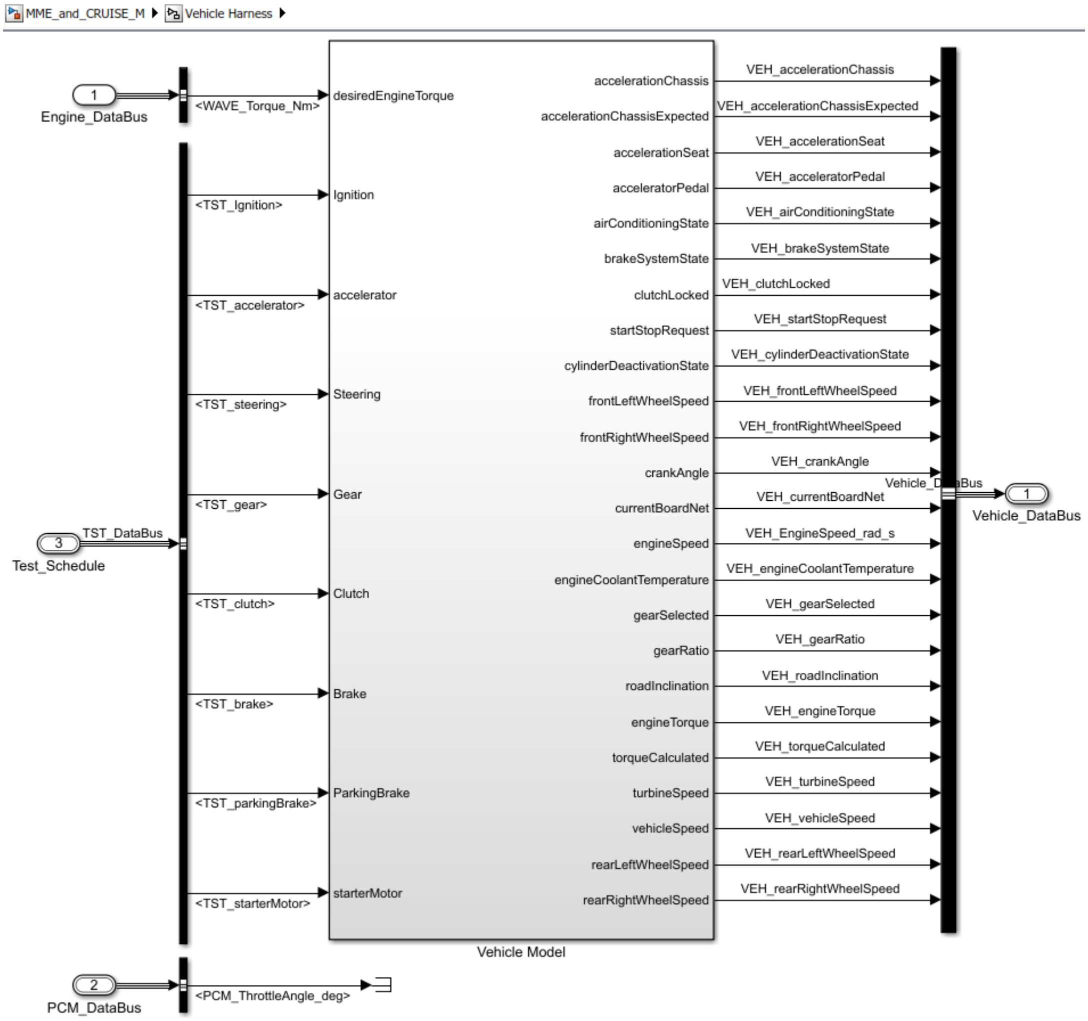

Figure 3 Content of the "Vehicle Harness" block

To test the modularity of the developed MME, three vehicle models have been built in AVL CRUISE M, Ricardo IGNITE, and LMS AMESIM to serve as slave vehicle models to the developed MME. The three models have been built to resemble each other as closely as possible in both structure and parameterization to exhibit as a close behaviour as possible.

The content of the “Vehicle Model” shown in Figure 4 connects the main MME to an external slave Simulink model (CruiseSF) which in turn contains a MATLAB S-function vehicle model exported by AVL CRUISE M. The reason for running the CRUISE M generated S-function as a referenced slave model is the fact that the MME master simulation must run under a variable step solver while the S-function vehicle model must be simulated under a fixed step solver of a 10^{-5} sec time step. One thing to notice when looking at Figure 4 is the unused I/O ports which serve as placeholders for when they are needed. The non-connection of some of the output ports does not prevent the simulation of the holistic vehicle model.

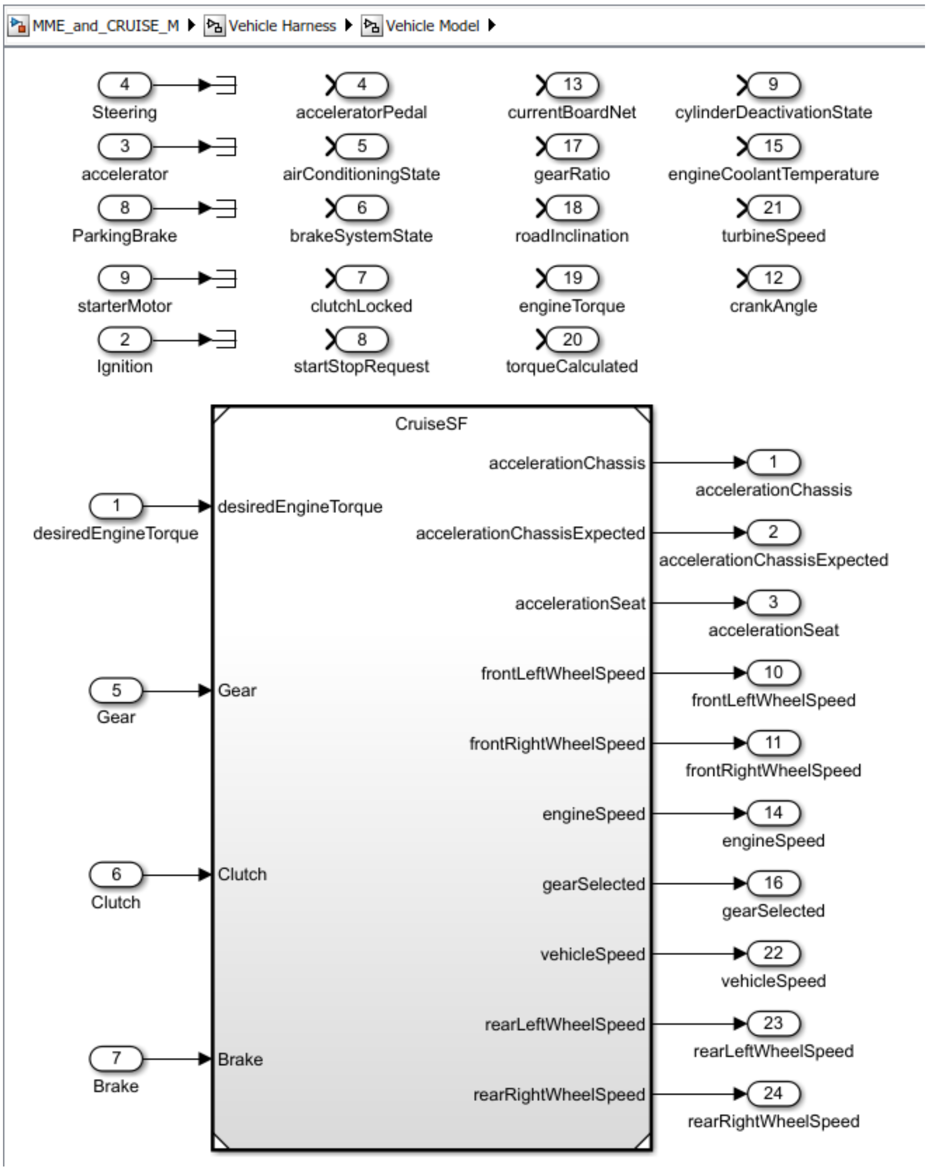

Figure 4 Content of the "Vehicle Model" block - Connection with slave Simulink model (AVL CRUISE M)

The second configuration of the tested MME vehicle model shown in Figure 5 involves the connection of the MME to a vehicle model built in Ricardo IGNITE. The connection of the master MME model to the slave IGNITE model is achieved through the addition of two IGNITE co-simulation interface blocks within the "Vehicle Model" block with one interface providing the time step of the co-simulation, and the other interface exchanging data with the IGNITE vehicle model. In this case, the IGNITE vehicle model is simulated in IGNITE using the solver and simulation time step that have automatically been selected by IGNITE. It can be observed, that this configuration makes use of only two vehicle model outputs compared to the ten outputs used in the above configuration. Again, the unconnected I/O ports of the "Vehicle Model" block serve as placeholders and do not affect the operation of the MME.

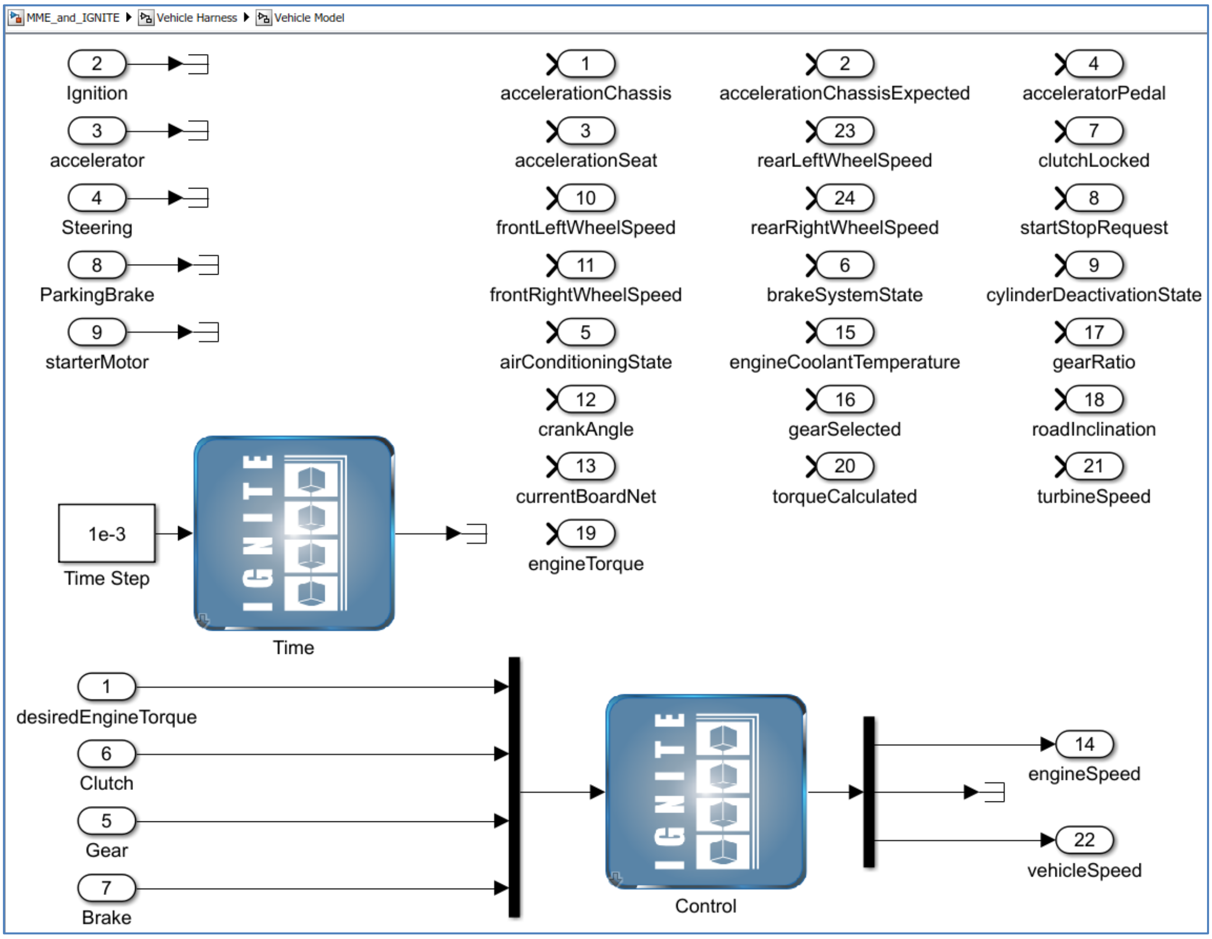

Figure 5 Content of the "Vehicle Model" block – Co-simulation with Ricardo IGNITE slave vehicle model

The third configuration of the tested MME involves the co-simulation of the MME with an LMS AMESIM vehicle model. As shown in Figure 6, the connection of the MME with the AMESIM vehicle model is achieved through the addition of an AMESIM co-simulation interface Simulink block within the "Vehicle Model" block. The slave vehicle model is simulated within LMS AMESIM (under a solver configuration automatically selected by AMESIM) and exchanges data with the MME via the AMESIM co-simulation interface

Simulink block. In this case, eight of the output ports of the "Vehicle Model" block are connected to the vehicle model while the rest of the blocks are left unconnected, thus showcasing once again the modularity and flexibility of the developed MME.

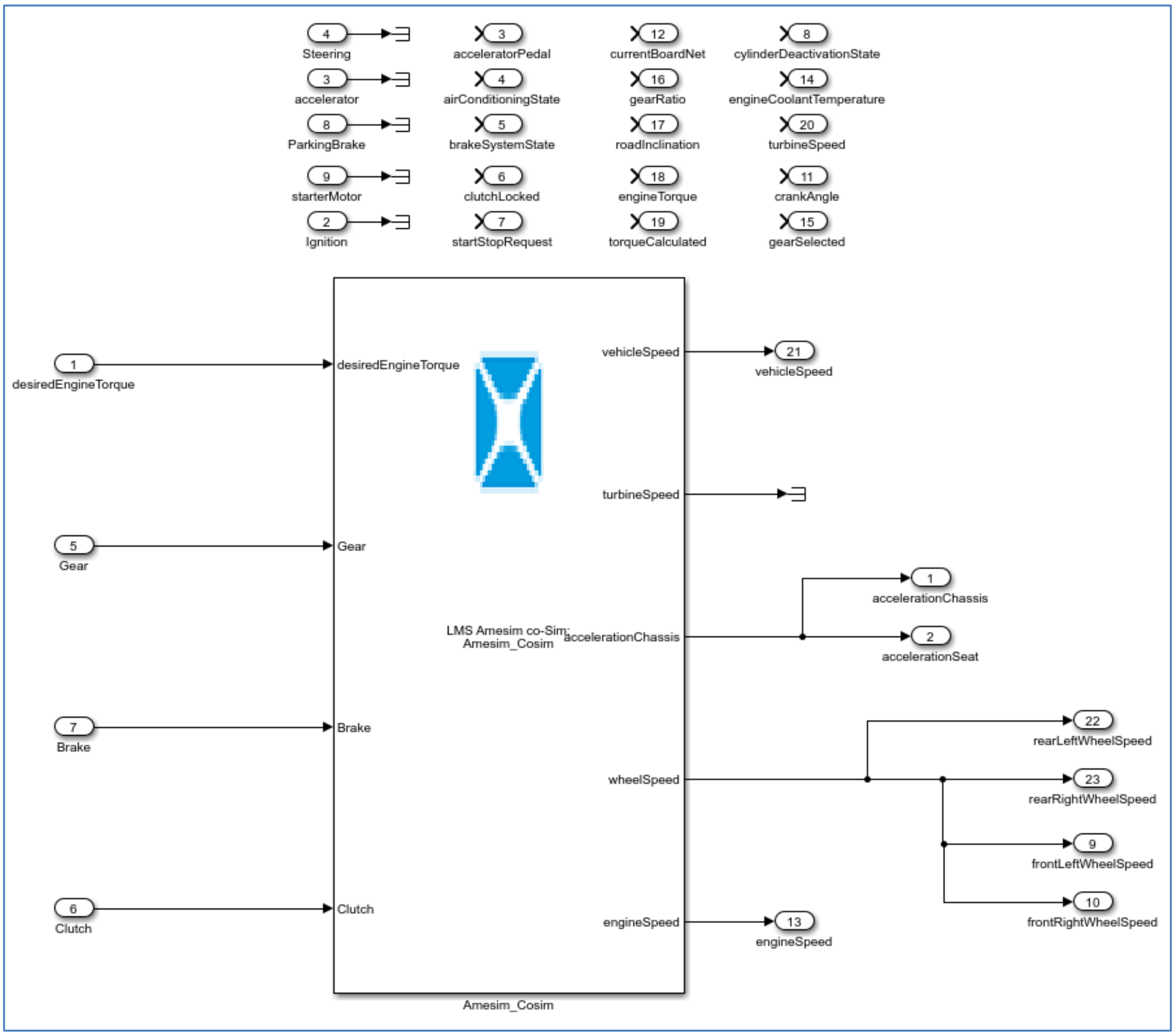

Figure 6 Content of the "Vehicle Model" block – Co-simulation with LMS AMESIM slave vehicle model

3. Simulation and comparison of results

Following the description of the developed modelling environment and the three MME configurations, this section validates the developed MME concept by means of simulating the three configurations of the MME described above and comparing the generated simulation results. For this purpose, a constant desired engine load input will be applied to the holistic vehicle model.

Constant blocks within the "Driver" block provide the "WAVE_and_PCM" slave model with the desired engine load, and the vehicle slave model with desired gear, brake signal, and clutch signal through the "TST_DataBus".

The simulation of the holistic vehicle model starts with an initial vehicle speed of 20km/h and stops at 30 seconds. The clutch is kept engaged, and the brakes are kept disengaged throughout the simulation. The desired engine load is constant throughout the simulation.

The engine speed traces of the MME simulated with CRUISE M generated S-function, IGNITE, and AMESIM vehicle models are plotted in Figure 7. It can be observed that all third-party vehicle models have been successfully integrated within the MME and communicate with the WAVE-RT engine model, as well as the Simulink based PCM. The engine speed traces of all tested configurations exhibit shapes that behave in a similar manner and are closely located with one another. All traces exhibit an oscillatory behaviour of a higher amplitude within the 14” to 18”, and 24” to 27” intervals.

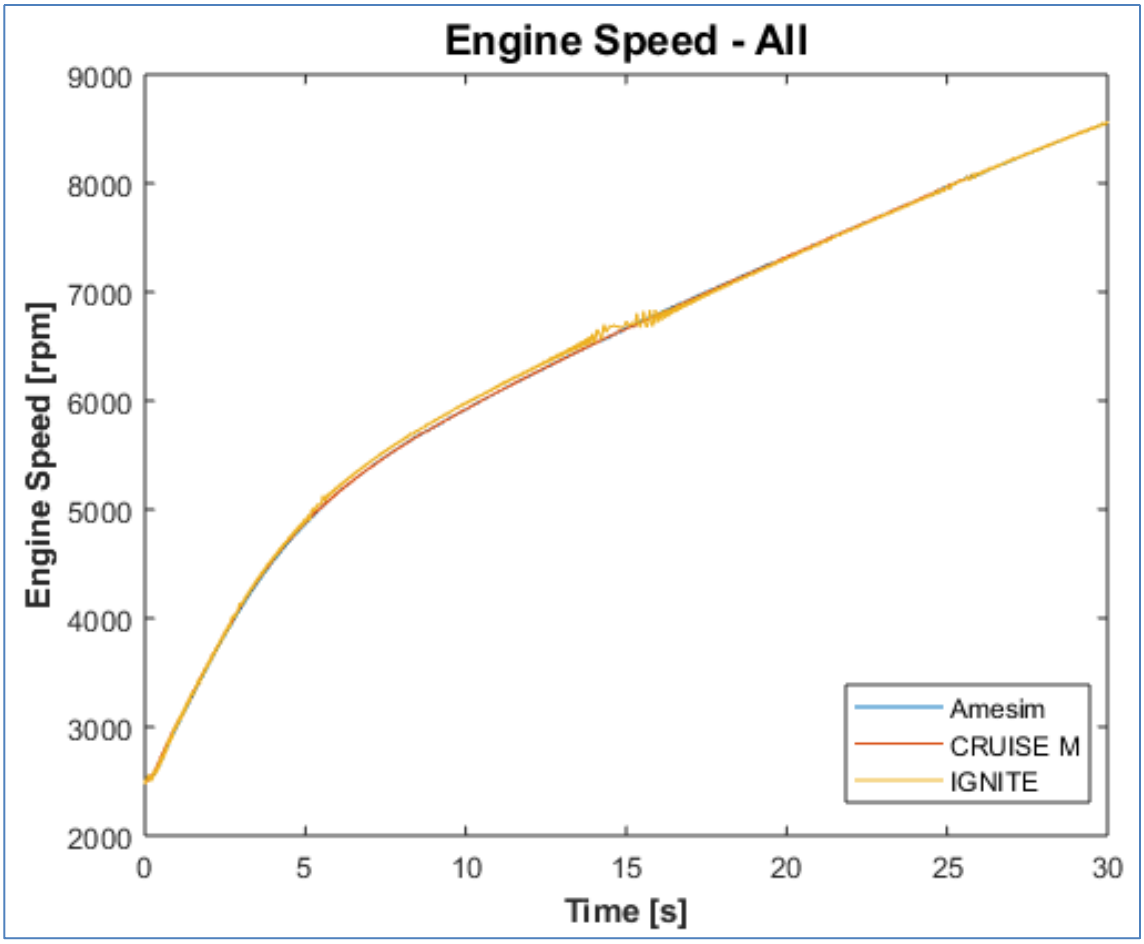

Figure 7 Engine speed vs. Time plot of the simulated MME-Vehicle combinations

As expected, the vehicle speed traces plotted in Figure 8 follow the trends exhibited by the respective engine speed traces with all vehicle speed traces being located at a very close proximity with one another. The oscillatory behaviour exhibited by the engine speed traces is also present in the respective vehicle speed traces of Figure 8.

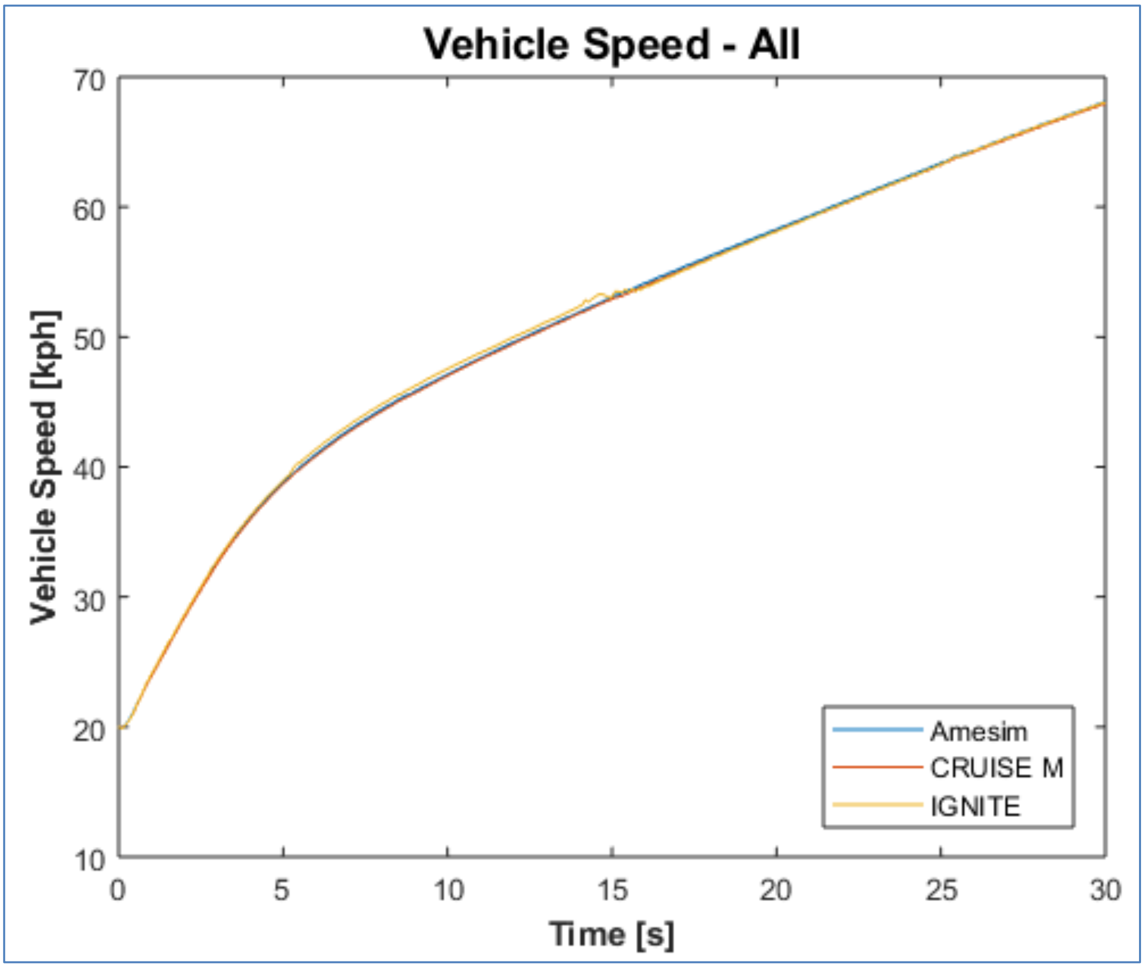

Figure 8 Vehicle speed vs. Time plot of the simulated MME-Vehicle combinations

The closer inspection of the traces in the plots of Figure 9 and Figure 10 reveals the effects the co-simulation and the interaction between separate models have on the results generated by the holistic vehicle model. The small differences in the behaviour of the overall vehicle models can be attributed to differences in the co-simulation communication interfaces used by the tested platforms. Because of the use of the engine speed (calculated within the vehicle model) as a feedback signal to the engine and PCM models, small differences in the communication of signals between the different configurations cause small differences in calculated acceleration. In addition to the above, small differences in the architecture of the models as well as in the solvers employed by the individual platforms introduce additional error in the calculated values.

The cause of this behaviour is the oscillatory nature of the torque generated by the crank angle resolved WAVE-RT engine model that excites the driveline and drift in and out of phase with the driveline natural frequencies. Another observation on the oscillations of vehicle speed traces is that while they depict the nature of the system correctly, their high magnitude highlights the need for the selected stiffness values for the driveline shaft components used for this study to be replaced by the calibrated values. Despite this, the vehicle models still served well the purpose of demonstrating the advantages offered by the developed MME by being capable of integrating distinct solvers dedicated to different physical domains into a holistic vehicle simulation.

While small differences in calculated engine speed and vehicle speed values have been observed between the three tested MME configurations, these can be characterized as small and well within an acceptable tolerance.

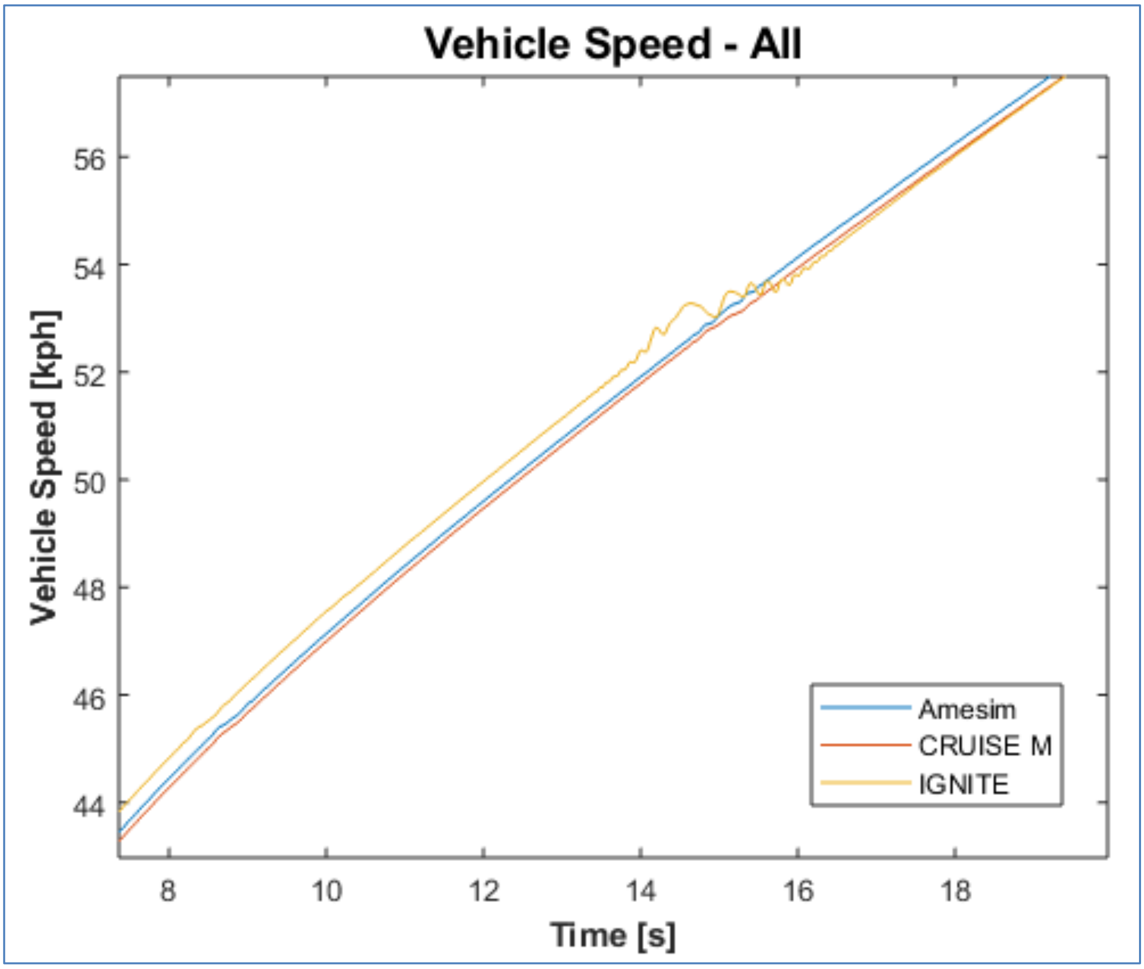

Figure 9 Detailed Vehicle Speed Comparison

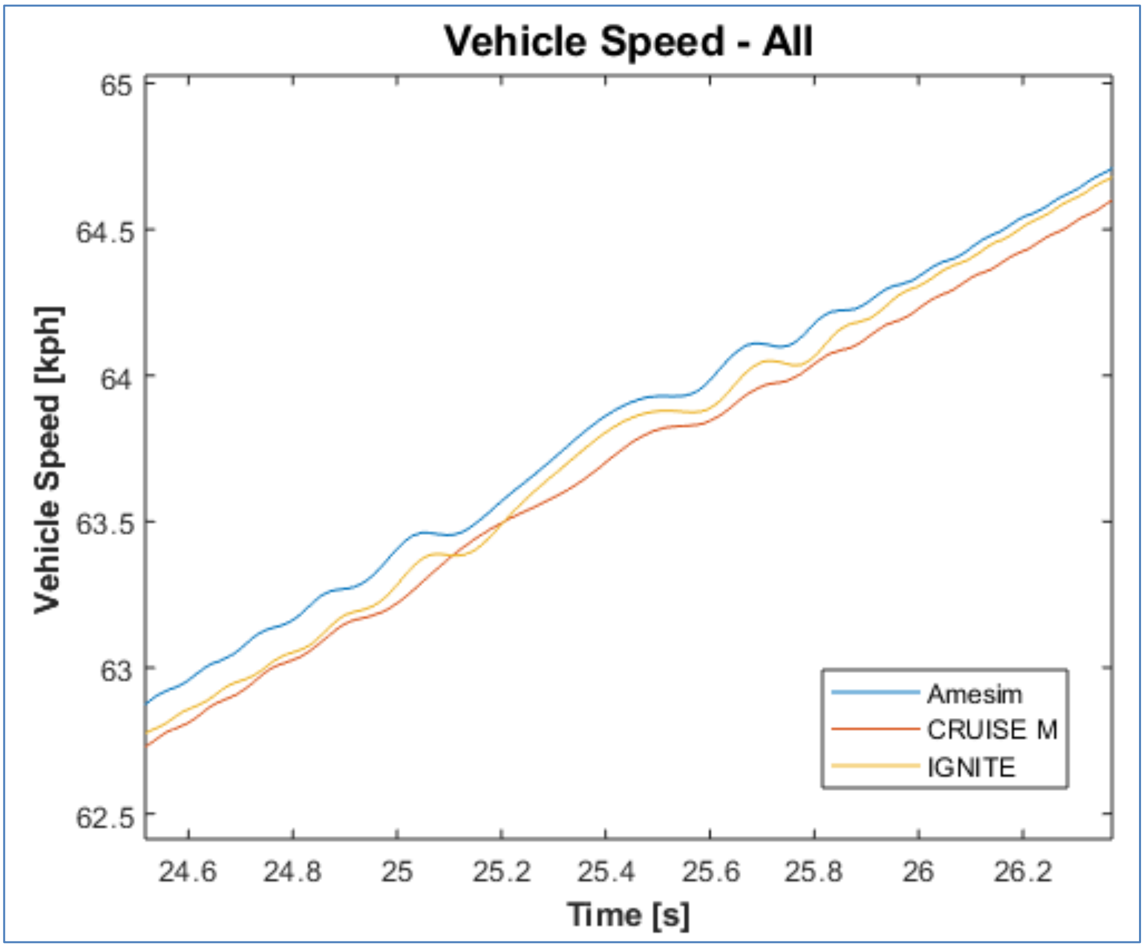

Figure 10 Detailed Vehicle Speed Comparison (2)

The capability of the developed MME of integrating subsystem models of varying capabilities built on many different platforms in one holistic vehicle simulation allows for building numerous holistic vehicle model configurations. The model fidelity and chosen modelling environment of the constituent subsystem of these configurations depends on the subject of the study, as well as the position of the study within the product development cycle.

A team working on a very early stage of the vehicle design lifecycle may make use of the MME to integrate a simple mean value or response surface engine model and a rigid driveline vehicle model to formulate the set of requirements/specifications for the design.

In later stages of the vehicle design lifecycle, engineering teams may use the MME to integrate high-fidelity subsystem models to carry out detailed design of the subsystem under investigation. The predictive nature of the crank angle resolved engine model enables the engineers to carry out numerous studies for which, a mean value engine model would be of limited usefulness. One of the most important applications of the crank angle resolved model are the studies of the effect of changes in engine calibration on vehicle fuel consumption for a given drive cycle. The user can experiment with injection, ignition, valve timing parameters, as well as boost control to optimize fuel consumption. In terms of the vehicle model, a relatively low fidelity vehicle model with rigid driveline components and a large timestep in the order of 10^{-2}-10^{-1} sec may supply the engine model with the necessary signals at an accuracy adequate for the purpose while allowing for a very high simulation speed. Another application of the crank angle resolved engine model is carrying out noise, vibration & harshness (NVH) studies, in which case, the torque profile of the engine model provides a realistic excitation signal to a high-fidelity vehicle model employing a fully elastic driveline as in the case of the current work. Such a vehicle model requires a very small simulation time step in the order of 10^{-5} seconds and for this reason, it is expected to slow down the global simulation to a considerable extent.

4. Conclusion

The aim of the current document was the development of a Multi-Disciplinary Modelling Environment (MME) to serve as the basis of a generic holistic vehicle simulation platform useful throughout all phases of the vehicle design lifecycle. For the selection of the platform to play the role of the master model, several criteria were considered, among which, platform proliferation, connectivity with other software, multiple solver and multi-thread capability, wealth of component libraries, design exploration, optimization, analysis, and surrogate model generation capabilities. Since MATLAB/Simulink satisfies all the above criteria to the highest extent, it has been selected to become the basis of the development of the MME. The developed MME is characterized by increased modularity and connectivity to third-party modelling platforms and can accommodate multi-solver simulation on a single or multiple CPU cores. It allows the users to setup countless combinations of subsystem model fidelities with models built in different environments. Thus, it enables the creation of holistic vehicle models useful on a company-wide scale across the vehicle design lifecycle while retaining its main environment architecture.

The capabilities of the MME have been tested by generating three different holistic vehicle model combinations, all of which make use of a WAVE-RT crank angle resolved engine model while integrating vehicle models built in three different platforms to the same parameters and architecture. Following the simulation of all combinations, the MME has been found successful in integrating the tested MME combinations as the simulation results are in a close proximity with one another throughout the simulation interval, and any small differences are within typical validation tolerances. The minor differences in calculated results are attributed to small differences in model architecture, communication between the master and the slave vehicle models, used numerical solvers.

Following the above, it can be concluded that the developed Multidisciplinary Modelling environment fully meets the requirements for model integration of subsystem models running under different solver configurations into a holistic vehicle model and it has achieved the aim of being a suitable generic vehicle simulation tool for studies across the vehicle design lifecycle.

References:

[1] L. Mikelsons and R. Samlaus, "Towards Virtual Validation of ECU Software using FMI," in *Proceedings of the 12th International Modelica Conference May 15-17, 2017,* 2017, pp. 307–311.

[2] N. Pedersen *et al.*, "Co-simulation of distributed engine control system and network model using FMI & SCNSL," *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 261–266, 2015.

[3] T. Wu, K. Han, L. Pei, and C. Zhao, "Co-Simulation Study of Coordinated Engine Control Focusing on Tracked Vehicle Shift Quality," *J. Autom. Control Eng.*, vol. 2, no. 2, pp. 160–165, 2014.

[4] J. Fitzgerald *et al.*, "Collaborative Modelling and Co-simulation in the Development of Dependable Embedded Systems," in *Integrated Formal Methods - IFM*, 2010, pp. 12–26.

[5] Julia Veintimilla Porlán, "Co-simulation Methodologies for Hybrid and Electric Vehicle Dynamics," UNIVERSIDAD POLITÉCNICA DE MADRID, 2016.

[6] M. Maharun, M. Bin Baharom, and M. S. Mohd, "Modelling and control of 4WD parallel split hybrid electric vehicle converted from a conventional vehicle," *World J. Model. Simul.*, vol. 9, no. 1, pp. 47–58, 2013.

[7] L. T. Kyllingstad, M. Rindarøy, and D. Eirik, "Distributed Co-Simulation of Maritime Systems and Operations," *CoRR*, vol. 1701.00997, 2017.

[8] P. Casoli, A. Gambarotta, N. Pompini, and L. Riccò, "Development and application of co- simulation and ' control - oriented ' modeling in the improvement of performance and energy saving of mobile machinery," *Energy Procedia*, vol. 45, pp. 849–858, 2014.

[9] F. R. L.M. Reyneri, E. Bellei, E. Bussolino, L. Mari, "Codesign and Cosimulation of Automotive Systems Based on Matlab / Simulink," in *Seminario Anual de Automática, Electrónica Industrial e Instrumentación, SAAEI*, 2002, no. May.

[10] J. J. Eckert, F. M. Santiciolli, S. Costa, F. C. Corrêa, H. J. Dionísio, and F. G. Dedini, "VEHICLE GEAR SHIFTING CO - SIMULATION TO OPTIMIZE PERFORMANCE AND FUEL CONSUMPTION IN THE BRAZILIAN STANDARD URBAN DRIVING CYCLE," in *XXII Simpósio Internacional de Engenharia Automotiva*, 2014, vol. 1.