

WHITE PAPER

Building Custom Vehicle Models with Virtual Vehicle Composer App

Simulations are widely used in vehicle development to identify issues early and reduce prototype testing. However, creating a vehicle model is a complex and time-consuming process. All the subsystems not only require the right model fidelity but also need to be tediously connected. To maximize its use and adoption, the model should be straightforward to navigate and should provide flexibility to make changes if needed. Key challenges like this limit the use of modeling and simulation in vehicle development.

The Virtual Vehicle Composer app overcomes several of these challenges. Through its background automation, it connects all the subcomponents allowing the user to create a vehicle model in a few minutes. The model then can be viewed in Simulink®, and if needed, the user can alter or substitute sub-models to their more specific simulation objectives. This paper illustrates the Virtual Vehicle Composer app and an example use case in which various powertrain architectures models were created, simulated, and analyzed.

Why a Virtual Vehicle Composer?

Content and complexity of a vehicle model depends on its end use. For example, to simulate EV range or fuel economy, models of engine, e-motor, battery, transmission, and other subsystems need to be connected, whereas for analyzing ride and handling, models of steering, suspension, brakes, tires, and other subsystems are necessary. For complete closed-loop simulations, along with the driver model, production-representative control models and algorithms need to be included to replicate vehicle behavior.

Creating such a large-scale model is time-consuming and often results in slow simulation speeds. If the fidelity of the subsystem models needs to be changed, it can require a significant tear-up of the model. Even less severe changes to the model would require verification testing and debugging. So, additional overhead is necessary to maintain the model, and active management is required to prevent proliferation of independent model versions. As a result, such a model would not be suitable for mass adoption without excessive maintenance overhead.

The Virtual Vehicle Composer app overcomes many of these challenges. The user can quickly build a vehicle level model for their architecture of interest in a few minutes. The model can be opened and viewed in Simulink and, if necessary, can be modified. With its default settings, this generated model can simulate any drive cycle from a comprehensive list of cycles (such as FTP, highway, and WLTC) and maneuvers (such as wide open throttle launch, double lane change, and constant radius test). The user can change the default parameters to match their representative vehicle through a UI. They can develop new control algorithms on top of existing controls or bring an entire new controller into Simulink through model-in-the-loop (MIL), software-in-the-loop (SIL) and even hardware-in-the-loop (HIL) implementations. They also have the flexibility of modifying the plant model, adding fidelity using Simscape™, for example, or importing custom plant models from third-party specialty software into Simulink. Such models enable evaluation of interactions between subsystems, overall system performance and behaviors, and trade-offs while reducing the need for physical testing.

What is Virtual Vehicle Composer?

Virtual Vehicle Composer, available with Powertrain Blockset™ and Vehicle Dynamics Blockset™, is a MATLAB-based app (Figure 1), that allows the user to build vehicle models by automatically connecting the subsystem plant and control models. A drop-down menu in app's UI (Figure 2) allows the selection of the desired powertrain architecture from conventional to EV and various hybrid alternatives. The choice to include lateral vehicle dynamics is also available, depending on user's simulation application. Once satisfied with the parametrization, the user can then click the "Virtual Vehicle" icon (Figure 2) to build a model. The model created opens in Simulink, as shown in Figure 3. To start the simulation, the user can click the "Run" button. Once the simulation runs are complete, the results can be viewed using the "Simulation Data Inspector" button within the app.

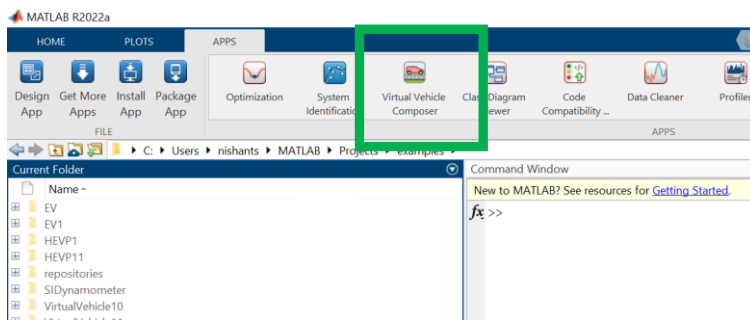


Figure 1: Virtual Vehicle Composer app access from the MATLAB Toolstrip.

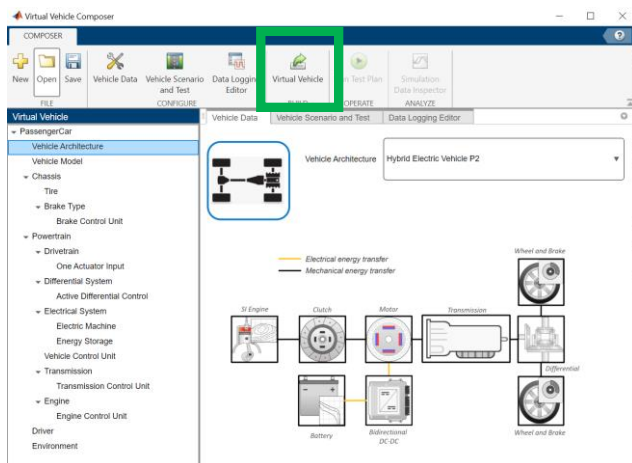


Figure 2: Virtual Vehicle Composer app user interface.

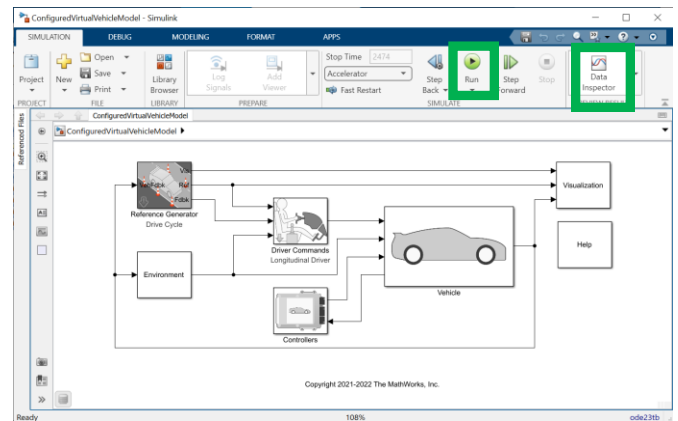


Figure 3: P2 hybrid model created using the Virtual Vehicle Composer app.

Use Case: Evaluating Multiple Powertrain Architectures

Using the methodology shown in the previous section, multiple vehicle models were created using the Virtual Vehicle Composer app. Figure 4 shows the position of the motor for each of the HEV models. In addition, a conventional and an EV model were also created. The models were then simulated for the WLTC Class 3 cycle and results were compared to illustrate the difference in engine and motor operation for these architectures. In case of hybrid applications, the supervisory control strategy, which resides in the Controllers block (Figure 3), decides if the engine or motor or both should be used, by analytically optimizing for least equivalent energy consumption.

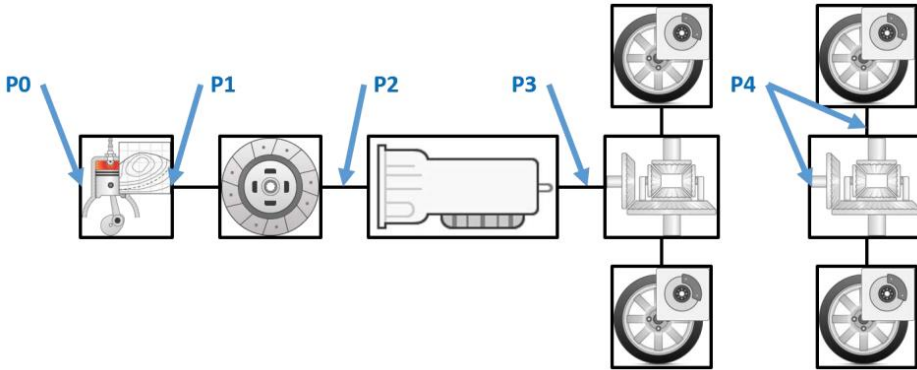


Figure 4: e-motor positioning alternatives in HEVs.

For reference, Figure 5 shows all the motor torque curves used for simulations including the EV motor torque curve. Identical P1, P2, P3, and P4 motors were assumed and have higher torque/power than P0 in this example. The engine assumed was a 1.5L turbo and remains unchanged between these models. The same 8 speed automatic stepped gear transmission was assumed for conventional and hybrid simulations.

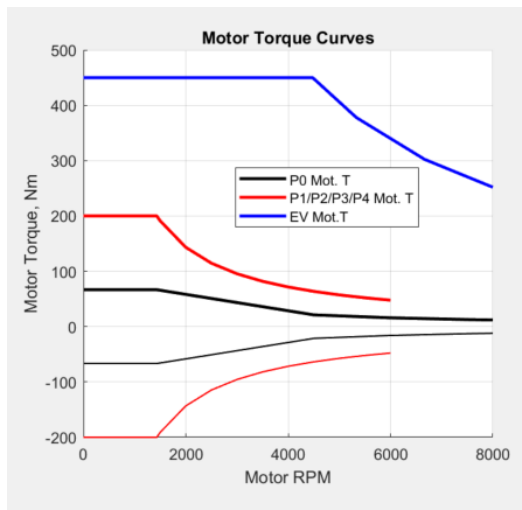


Figure 5: e-Motor torque curves used in simulations.

Figure 6 show the results of shift in engine operation due to P0 hybridization. The optimizer eliminates engine operation at the inefficient low-speed region (highlighted yellow area), leading to an improved fuel economy.

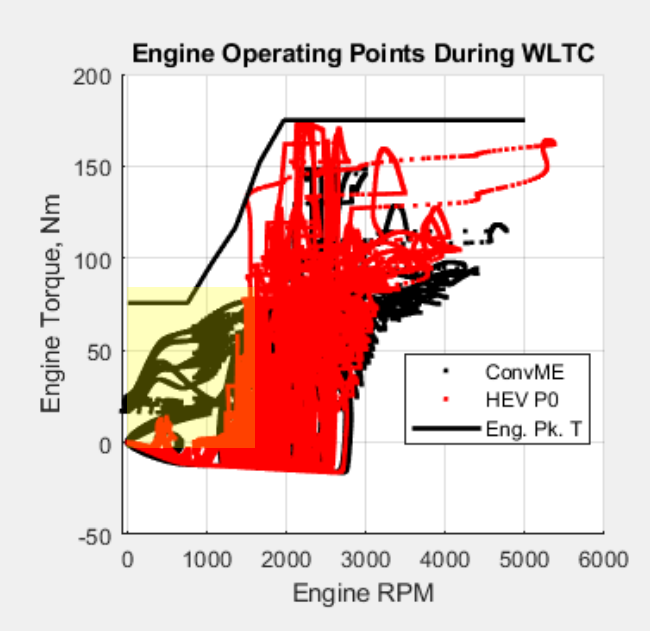


Figure 6: Engine operation change with hybridization.

Between P0 and P1, higher speed engine operation is reduced due to earlier gear upshifts with the higher torque P1 motor eliminating further engine inefficiencies (highlighted yellow area in Figure 7).

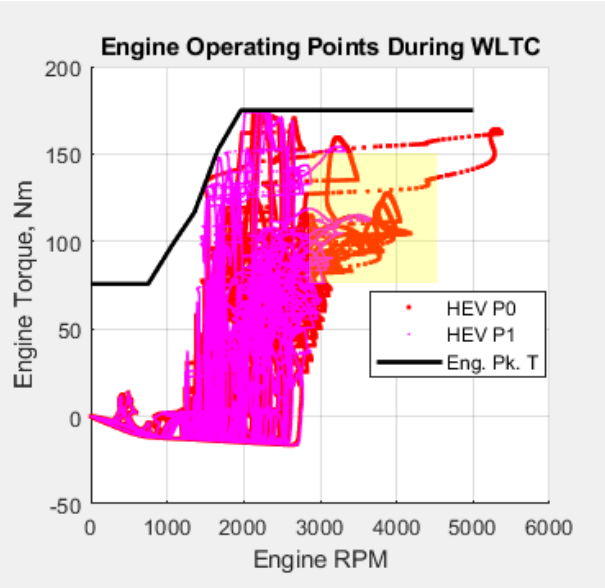


Figure 7: Engine operation change between P0 and P1.

© 2022 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

As expected, there is not a significant difference between P1 and P2 engine operation (Figure 8). However, P2 can provide higher regeneration by disconnecting the engine during deceleration, thereby diverting the energy towards charging the battery instead of losing to engine drag.

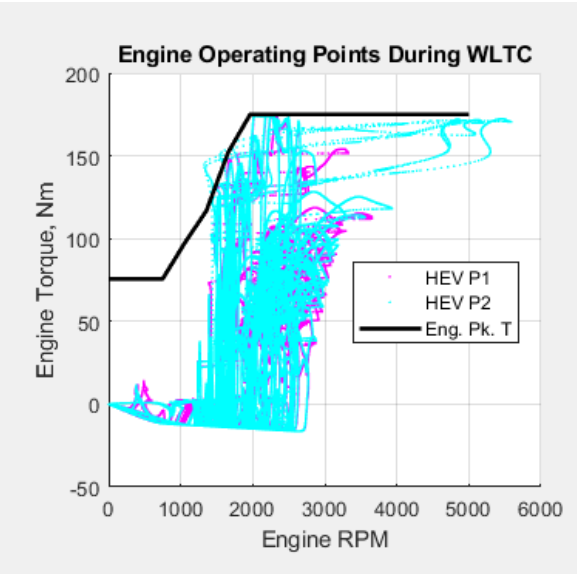


Figure 8: Engine operation change between P1 and P2.

The e-motor operating points can be seen in Figure 9. P0 motor operates at higher speeds due to the engine’s FEAD (front-end accessory drive) pulley ratio unlike other hybrids. P1 and P2 do not require peak low-end motor torque for WLTC due to downstream trans gear multiplication, which is not the case for P3/P4. And for EV, there is surplus motor torque still available during the WLTC operation.

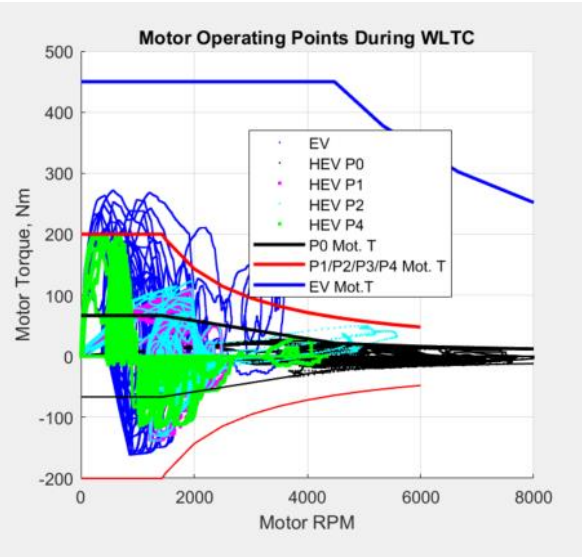


Figure 9: e- motor operating points over WLTC cycle.

Using this type of analysis in combination with real-world drivability scenarios, various motor torques can be investigated to select the right motor for an application. Similarly, the trade-offs between powertrain architectures and their impact on fuel economy, range, and other factors can be investigated along with many other possibilities of virtual assessment.

Conclusions

The Virtual Vehicle Composer app enables users to create complex vehicle models by automating the process of combining subsystem models and provides the flexibility to make changes if needed. As a result, users can devote more time to analyzing the results and conducting evaluations.

Learn More

[Virtual Vehicle Composer Documentation](#)

[Building a Virtual Vehicle for Large-Scale Simulation Studies – Video](#)