



Tiller M., Tobler, W.E., Kuang, M.:

Evaluating Engine Contributions to HEV Driveline Vibrations

2nd International Modelica Conference, Proceedings, pp. 19-24

Paper presented at the 2nd International Modelica Conference, March 18-19, 2002,
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Oberpfaffenhofen, Germany.

All papers of this workshop can be downloaded from
<http://www.Modelica.org/Conference2002/papers.shtml>

Program Committee:

- Martin Otter, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Institut für Robotik und Mechatronik, Oberpfaffenhofen, Germany (chairman of the program committee).
- Hilding Elmqvist, Dynasim AB, Lund, Sweden.
- Peter Fritzson, PELAB, Department of Computer and Information Science, Linköping University, Sweden.

Local organizers:

Martin Otter, Astrid Jaschinski, Christian Schweiger, Erika Woeller, Johann Bals,
Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Institut für Robotik und
Mechatronik, Oberpfaffenhofen, Germany

Evaluating Engine Contributions to HEV Driveline Vibrations

Michael Tiller, William E. Tobler and Ming Kuang

Ford Motor Company

Abstract

In order to comply with increasing consumer and regulatory demand for improved fuel economy and lower emissions, Ford Motor Company is developing a Hybrid Electric Vehicle (HEV) version of the Escape sport utility vehicle for production in 2003. Since HEVs typically have several different operating modes (e.g. electric launch, active neutral, regenerative braking), an important concern is the fact that each of these modes and the transitions between them lead to minimal driver perceived vibrations. In order to understand how the design and control of an HEV influences what is "felt" by the driver, we need to build models that accurately reproduce the dynamic response of the powertrain. In this way, the response for a given mechanical configuration and/or controller design can be evaluated.

A model targeted at prediction of driver perceived vibration was developed and validated against experimental data. However, one unexpected result of this work was to demonstrate that we could take the dynamic model used to reproduce the behavior described previously and, by using some advanced Modelica features, derive a second model that predicts the system efficiency of the transmission without having to create an entirely new model for that purpose. The system efficiency model was also validated against experimental data and showed very good agreement. The result is that rather than spending time creating and maintaining two different models (one for dynamic response and one for system efficiency) we were able to build one on the foundation of the other. Furthermore, we determined it was possible to generate a single model that could describe both types (i.e. dynamic and steady-state) of responses by merely changing the values of a few model parameters.

Introduction

The idea of using computer-aided methods to evaluate powertrain and vehicle NVH (i.e. noise, vibration and harshness) is not new [1,2]. Furthermore, the use of Modelica to model automotive systems is increasing [3,4,5,6,7]. However, the contribution that the internal combustion engine makes as a "forcing function" to a powertrain system is not typically examined in detail since the steady state operation of the engine is well understood and sufficient for most applications. For HEVs though, the engine starts and stops frequently, both with the vehicle in motion and at rest, and this

makes a significant contribution to vibration perceived by the driver.

In order to understand the effects of powertrain design and control on driver perceived vibrations, a detailed thermodynamic model of an Atkinson cycle internal combustion engine was developed and integrated with a detailed model of a hybrid electric transmission. The computational model was then validated against experimental data and showed very close agreement. With this validated dynamic model of the powertrain, we analyzed the effect that changes in the mechanical design of the powertrain had on the natural frequencies of the vehicle, examined the effect that engine control parameters (e.g. spark timing and valve timing) had on powertrain response and created realistic Simulink plant models which could be used to test different control strategies.

Physical Models

The focus of paper is on modeling the physical response of the powertrain. Issues about control system design or strategy are larger issues beyond the scope of this paper. Nevertheless, a good physical model of the powertrain can provide useful insights for both the hardware and control system designers.

In particular, we are interested in predicting the sensitivity of the powertrain response with respect to component design parameters and actuator commands. To preserve the effects of design parameters, it is generally necessary to provide design-oriented models built from first-principles based component models rather than models derived from empirical relationships or experimental data.

Our discussion of modeling efforts will start with some general modeling issues and then present details of the engine, transmission and vehicle models used in this work.

Control Signals

Both the engine and transmission subsystems contain components that require control signal inputs (e.g. spark timing and motor torque). One interesting problem that arises when actuator and sensor models are included is the need to communicate these control signals into and out of the physical model hierarchy. The difficulty is in managing the propagation of these signals especially in the context of replaceable components.

For example, we may develop an HEV model that includes an engine with certain control signals (*e.g.* spark timing and injector timing). At some later point, we may wish to create a variation of that model by simply extending the original model and replacing the engine with different engine model. The difficulty comes when the new engine happens to have a different set of control inputs (*e.g.* cam phasing). In order to propagate these new signals, this could require the model developer to add a whole new set of connectors up and down the model hierarchy.

To avoid this situation, we use something in our Modelica models that we call the `SignalBus` idiom. In this approach, all the signals associated with each subsystem are grouped onto a "master" bus (*e.g.* `eng_control_bus`) at the top-level of the model. The `SignalBus` idiom is useful because the component models only need to be aware of *the specific signals they require* and not all signals on the master bus.

```
connector SignalBus
  annotation (...);
end SignalBus;

model FuelInjector
  outer ControlBus eng_control_bus;
  // ...
protected
  connector ControlBus
    extends SignalBus;
    Ford.Types.Degree inj_start;
    Ford.Types.Degree inj_stop;
  end ControlBus;
end FuelInjector;

model FullVehicle
  inner EngineMaster eng_control_bus;
  Engine eng "has fuel injectors";
protected
  connector EngineMaster
    extends SignalBus;
    Ford.Types.Degree inj_start;
    Ford.Types.Degree inj_stop;
    Ford.Types.Degree spark_adv;
  end EngineMaster;
end FullVehicle;
```

Figure 1: Example of SignalBus Idiom

To implement the `SignalBus` idiom, we define an empty connector with a specific graphical annotation. Although not strictly necessary, it makes the bus connectors very easy to identify in diagrams. Next, inside each component requiring control signals (*e.g.* a fuel injector), we declare a specific bus type for that component (preferably in a `protected` section to clearly indicate that this definition is for internal use). The bus definition should include only the signals required by the component. This bus can then be

instantiated with the `outer` qualifier. The name of the instance should be that of the master bus where the signals ultimately reside. At the top-level, the master bus type must contain (at least) the union of all subsystem component buses and an inner instance must be declared. An example of the definitions and declarations required is shown in Figure 1.

The `SignalBus` idiom has the following advantages over signals. First, it avoids the necessity to place connectors at each level in the hierarchy. This is important because every change in control signals can potentially change the set of connectors and connections, a situation that becomes difficult to maintain. In addition, because the `outer` bus only has to be a subtype of the matching inner bus, the component models are only required to declare the signals they are interested in. This avoids dealing with complex combinatorial possibilities that result when all signals are included in a single connector definition. One disadvantage with `SignalBus` definitions is that responsibility for assigning the control signals is not clearly specified by the definition. Instead, this requires some discipline and understanding of the idiom.

Trying to decide on the best logical grouping for the signals could be an involved task. Fortunately, there are developing internal corporate standards or identifying and grouping control signals and the `SignalBus` idiom fits nicely into these emerging standards.

Reaction Torques

One limitation of the current rotational mechanics library in the Modelica Standard Library is the fact that it neglects reaction torques on rotational components. For example, consider the `IdealGear` model definition shown in Figure 2. The problem with the `IdealGear` model is that it contains the equation:

$$R\tau_a + \tau_b = 0$$

which, in general, results in the torques not summing to zero for this component. Since the torques represent the flow of angular momentum, angular momentum is not conserved.

```
within Modelica.Mechanics.Rotational;
model IdealGear "without inertia"
  parameter Real ratio "Gear ratio";
  Interfaces.Flange_a flange_a;
  Interfaces.Flange_b flange_b;
equation
  flange_a.phi=ratio*flange_b.phi;
  0=ratio*flange_a.tau+flange_b.tau;
end IdealGear;
```

Figure 2: Standard IdealGear Model

While for many applications the models in the Modelica Standard Library are sufficient, it is necessary to include an additional flange in applications where the entire

geartrain assembly has the potential to rotate. For our application, we are interested in the motion of the transmission housing and engine block and as a result, we must include a special flange on many of our component models (e.g. electric motors, spur gears, crank-slider mechanisms) to account for the reaction torque which ultimately causes vibrations in the powertrain casing. In most cases, it is necessary to formulate the reaction torques by considering conservation of momentum and conservation of energy.

Engine Modeling

One of the key features of the models developed for this application is the ability to predict the torque generated by the engine during startup and shutdown. In order to predict this torque, it is necessary to model some of the detailed thermodynamic processes of the engine (e.g. breathing, compression, combustion). Fortunately, we had already developed, prior to this application, a library of thermodynamic components for the purpose of studying engine behavior [6,8,9].

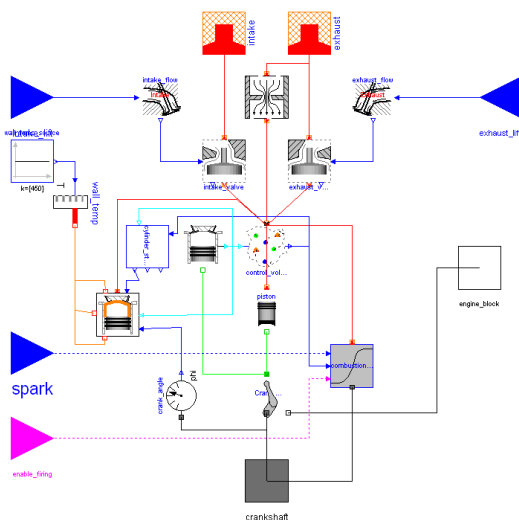


Figure 3: Combustion Chamber Processes

The engine model uses the same geometry and valve timing as the intended production engine to ensure that the predicted torque fluctuations have the same characteristics as the actual engine. To accomplish this, the engine model was developed such that it could reproduce effects due to throttle position, spark timing, cam phasing, valve lift profiles, engine geometry and injection timing during both startup and shutdown. This involves modeling the behavior of manifold filling and emptying, variable valve timing mechanisms, combustion and the application of cranking torque. Several of these behaviors are represented in the combustion chamber schematic shown in Figure 3.

Transmission Modeling

Using rotational components that account for the necessary reaction torques, construction of transmission models is straightforward. Unlike our previous transmission models which included hydraulic subsystems [4], the only complex behavior in the hybrid transmission is related to frictional elements and these can all be captured using the components in the Modelica Standard Library. In other words, no complex models had to be developed in order to build a reasonable model of the transmission.

To accurately predict the behavior of the hybrid transmission, several effects must be considered. First, a non-linear spring is connected to the input shaft of the transmission to isolate the transmission from the high-frequency torque fluctuations produced by the engine. In addition, the differential on the output side of the transmission includes a single backlash used to represent the backlash distributed throughout the transmission.

Vehicle Modeling

The vehicle response model is quite simple and neglects effects due to tire and suspension compliance. Currently, we treat the vehicle as a single mass connected by a kinematic tire model. The only real detail of the vehicle model is in the modeling of the front halfshafts (our current model handles only the front wheel drive configuration of the powertrain) which are modeled as non-symmetric compliances connecting the transmission to the wheels. In the future, we plan on refining our model to include suspension, tire and driveline details so that we are able to predict driver seat accelerations due to powertrain vibrations. It has been shown previously [4] that such large and complex models can be expressed in Modelica and simulated using Dymola.

Analyses

Dynamic Response

The dynamic response of the powertrain is due to the various inertias (e.g. gears, shafts, flywheel) and compliances (e.g. halfshafts and engine mounts) distributed throughout the system. The compliances are all modeled as linear with the exception of the isolation element on the input shaft of the transmission which is modeled as a piecewise linear spring. In addition to the inertias and the compliances there are several non-linear elements. While the transmission includes several frictional elements, they are not involved during start stop operation. Finally, as previously mentioned, all backlash in the transmission is lumped at the differential on the output shaft.

The dynamic response of the transmission can be modeled in Dymola and as we shall see later in the 'Validation' section, the results show close agreement

with experimental results. Because of the simplicity of our current vehicle model, the results we focus on are the halfshaft torque trajectories. However, interpreting the time domain results by inspection of the trajectories is not a very good way of establishing the "quality" of the startup or shutdown operation. Instead, we use a signal processing algorithm which reduces the time domain trajectories down to a scalar value. Using this information, we can then generate plots of the startup and shutdown quality as a function of spark timing, cam phasing, *etc.* While we are not in a position to discuss the results of such analyses, they have proved quite useful in identifying what factors contribute to powertrain vibrations.

Steady-State Response

One of the unexpected results of this work was to demonstrate that additional types of analyses could be performed using the model initially developed for studying dynamic powertrain response. Once we had established our ability to predict dynamic response of the powertrain, we were asked whether we could apply our model to understanding some experimental efficiency data taken on a powertrain dynamometer. The experimental results had shown what appeared to be anomalous data points during the testing and the question was whether the model could explain these anomalies.

To study the problem, we went back to our dynamic response model and made all the geartrains in the transmission replaceable. We then created a new transmission model for studying steady-state efficiency issues by extending our dynamic model and redeclaring all the geartrains so that steady-state efficiency data for each geartrain could be provided to the model. In other words, we took our original dynamic model and redeclared all the gears to include more detailed gear models necessary for studying steady-state efficiency.

In addition to redeclaring some of the components, several additional modifications were required. However, none of these modifications required changes to the original model but could instead be accomplished via the modification semantics in Modelica and by the addition of some new components. The first modification was to add some slight parasitic losses for some of the frictional elements. These losses could be introduced through modifications to the parameters in the original model. The other big change for the steady-state response was to eliminate the compliances since they only play a role in the dynamic response of the powertrain.

Thankfully, eliminating the compliances did not require crude methods like making the stiffness of the elements extremely large. This would not have eliminated the dynamics but would have just shifted the natural frequencies until they were extremely high. Instead, we employed a technique which is quite easy in Modelica.

We created the `RigidBypass` model shown in Figure 4. Placing an instance of this model in parallel with all our compliances allowed us, just by changing the value of the `rigid` parameter, to eliminate completely all compliance in the model.

Another important difference between the dynamic and steady-state response models is how they were used. The dynamic response model was used in the context of a vehicle simulation where the vehicle moves in response to the output torque of the transmission. On the other hand, the steady-state response model was used to reproduce the results of experiments conducted on a powertrain dynamometer where the speeds of various elements were fixed. To analyze our model we had to place the transmission on a virtual powertrain dynamometer. Once again, the experimental and model results showed good agreement.

```

model RigidBypass
  import Modelica.Mechanics.Rotational;
  parameter Boolean rigid;
  Rotational.Interfaces.Flange_a a;
  Rotational.Interfaces.Flange_b b;
equation
  if rigid then
    a.tau + b.tau = 0;
    a.phi = b.phi;
  else
    a.tau = 0;
    b.tau = 0;
  end if;
end RigidBypass;

```

Figure 4: RigidBypass Model

Frequency response

Another type of analysis that we could do quite easily with these models was to study the frequency response of the powertrain. With this capability, we could then study the effect that different design and control changes had on the poles of the system.

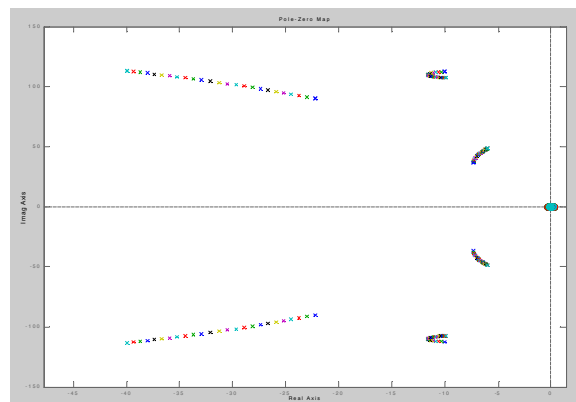


Figure 5: Design Dependence of Poles

To perform this analysis we used the "Linearize" and scripting functionality in Dymola [10] to generate a

linear time invariant system of equations for several different sets of design parameters. To do this properly, we needed to find a state where the backlash was taken up and the engine isolation spring was in the appropriate behavioral regime. We could then use the "Itviewer" functionality in MATLAB [11] to visualize the poles and zeros and to study how the poles moved in response to changes in hardware or controller design. Figure 5 shows one example of how the poles are visualized.

Validation

Dynamic Response

To validate the dynamic response of the powertrain, we used experimental data collected from vehicle tests conducted on our test track. The experiments involved starting the engine and looking at the resulting engine speed and halfshaft torque trajectories. The tests themselves were conducted with a closed loop controller. For our validation, we extracted the actuator signals used in the test and applied them in an open loop fashion to our model. The vehicle testing consisted of 27 different experiments involving 9 different controller strategies.

Figure 6 shows a comparison between the engine speed measured during the testing (dotted green line) and what the model predicts (blue line) based on the same actuator commands. The effects of the first few compression strokes can be seen as distinct bumps in the engine speed profiles.

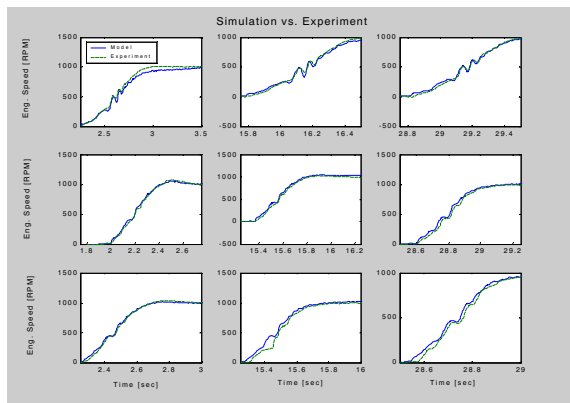


Figure 6: Validation of Dynamic Engine Response

Similarly, Figure 7 shows a comparison (during the same experiments) of the halfshaft torque predicted by the model compared to the halfshaft torque measured in the experiment. The halfshaft torque results are sensitive to the initial crankshaft position and the initial gap in the backlash.

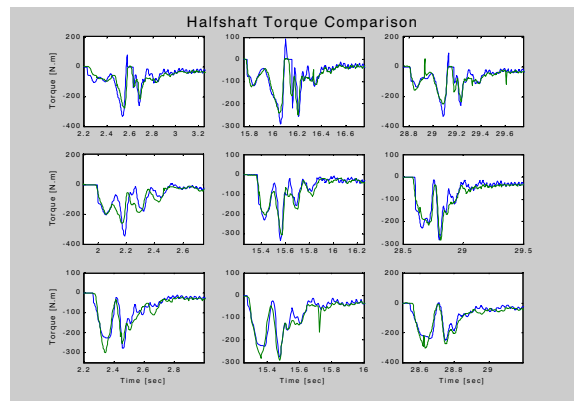


Figure 7: Validation of Dynamic Driveline Response

Steady-State Response

The steady-state response of the transmission was also validated by comparison to experimental data. However, the steady-state response is based on powertrain dynamometer data. Again, we saw good agreement between our model and the experimental data taken over a range of different operating conditions. Figure 8 shows a comparison between the experimental data (black bars) compared with the model predicted efficiency (red stars). The important thing to note in this data is how well the model predicts the conspicuously low efficiency present in some of the tests.

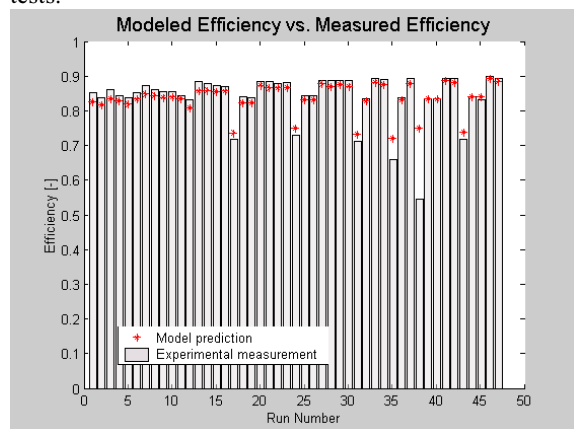


Figure 8: Validation of Steady-State Efficiency

Conclusions

There are several important points to be made about this modeling project. The models described in this paper were constructed from data about the individual components that appear in the model. Whenever such data was available, we used it. The only exception was a slight modification to the crankshaft inertia to demonstrate better agreement in engine speed trajectories. For component data that is not easily obtained or measured (e.g. damping ratios), we started by using "rule of thumb" numbers (which showed reasonable agreement) and then we made some small

adjustments, within reasonable limits, to calibrate those parameters so that we could achieve the agreement shown in the validation figures.

Another important point to make is the flexibility and reusability that is inherent in Modelica models. This is evidenced by our ability to do component level and powertrain level validation studies, the flexibility of using the models in different contexts (i.e. with different causalities) and the ability to reuse the dynamic models to reproduce steady-state response characteristics. This flexibility combined with the efficient code generation and solution methods in Dymola ensured that the model development and analysis process was able to provide accurate answers in a timely manner.

References

1. M. C. Tsangarides, W. E. Tobler and C. R. Heermann, "Interactive Computer Simulation of Drivetrain Dynamics", SAE 850978, *Surface Vehicle Noise and Vibration Conference Proceedings*.
2. M. C. Tsangarides and W. E. Tobler, "Dynamic Behavior of a Torque Converter with Centrifugal Bypass Clutch", SAE 850461, *Surface Vehicle Noise and Vibration Conference Proceedings*.
3. M. Otter, M. Dempsey and C. Schlegel, "Package PowerTrain. A Modelica library for modeling and simulation of vehicle power trains", *Modelica Workshop 2000 Proceedings*, pp. 23-32.
4. M. Tiller, P. Bowles, H. Elmqvist, D. Brück, S. E. Mattson, A. Möller, H. Olsson and M. Otter, "Detailed Vehicle Powertrain Modeling in Modelica", *Modelica Workshop 2000 Proceedings*, pp. 169-178.
5. M. Tiller, "Introduction to Physical Modeling with Modelica", Kluwer Academic Publishers, ISBN 0-7923-7367-7
6. C. Newman, J. Batteh and M. Tiller, "Spark-Ignited Engine Cycle Simulation in Modelica", *Second International Modelica Conference Proceedings*.
7. C. Puchalsky, T. Megli, M. Tiller, E. Curtis, N. Trask, Y. Wang, "Modelica Applications for Camless Valvetrain Development", *Second International Modelica Conference Proceedings*.
8. H. Tummescheit, M. Tiller, "Modellierung von Ottomotoren", *Objektorientierte Modellierung Physikalischer, Teil 17*.
9. M. Tiller, H. Tummescheit, C. Davis, N. Trigui, "Powertrain Modeling with Modelica", *ASME 2000 Congress Proceedings*.
10. H. Elmqvist, D. Brück, S. E. Mattsson, H. Olsson and M. Otter, "Dymola User's Guide, version 4.2a", *Dynasim AB, Sweden*
11. "MATLAB Release 12" (documentation), The MathWorks, Inc., Natick, MA.