



S. Soejima:

**Examples of usage and spread of Dymola
within Toyota.**

Modelica Workshop 2000 Proceedings, pp. 55-60.

Paper presented at the Modelica Workshop 2000, Oct. 23.-24., 2000, Lund, Sweden.

All papers of this workshop can be downloaded from
<http://www.Modelica.org/modelica2000/proceedings.html>

Workshop Program Committee:

- Peter Fritzson, PELAB, Department of Computer and Information Science, Linköping University, Sweden (chairman of the program committee).
- Martin Otter, German Aerospace Center, Institute of Robotics and Mechatronics, Oberpfaffenhofen, Germany.
- Hilding Elmqvist, Dynasim AB, Lund, Sweden.
- Hubertus Tummescheit, Department of Automatic Control, Lund University, Sweden.

Workshop Organizing Committee:

- Hubertus Tummescheit, Department of Automatic Control, Lund University, Sweden.
- Vadim Engelson, Department of Computer and Information Science, Linköping University, Sweden.

Examples of usage and the spread of Dymola within Toyota

Shinichi Soejima

Toyota Motor Corporation

ABSTRACT

This paper describes some usage examples of Dymola within Toyota. Dymola has been used in many divisions, including Engine, Drivetrain, and Chassis. In the Drivetrain division, acceleration performance and noise vibration are being evaluated. In the Engine division, we have constructed an Engine library for Diesel Engine Control. Using the model, we have analyzed behavior of mass flow, pressure, and temperature in all parts of the engine. In addition evaluation of a designed control system was done.

Finally, some improvements that we would like to see in Dymola are stated.

1. Introduction

In the automotive industry, as performance is improved, the system within the car becomes more complicated. Because of this, new development process is required because it takes too much effort using the conventional development process. To improve system performance and reduce development time, simulation, derived from a physically based model, is a very important technique. Therefore, various simulation models and tools have been used in Toyota.

In controls field, model based controller design has been used recently. When we design a control system, a model which expresses the physical phenomenon correctly is required. A useful simulation tool which includes these models is indispensable. Dymola has many standard libraries, and good model reusability. It's convenient for modeling. Dymola is one of the various simulation tools that have been used to reduce development time at Toyota.

2. Position of Dymola in Development Process

Dymola has been used in many phases in the development process. When we designed vehicle drivetrain specifications, simulation analysis results were used. To construct the drivetrain models, we have been mainly used Dymola. For a simulation package, it is important that it is easy to construct models. It is also important that a model once built is able to be easily re-used. Therefore, Dymola is convenient because it has a drivetrain library, and if the library doesn't contain a model which we need, we can make the necessary model easily.

In the design process of a control system, model based controller design has attracted much attention, and is being applied in many controller designs. Figure 1 shows the typical development process when we design

a model based controller. Dymola has been used in some of the steps in this process. At the 1st step (Modeling) we made a model for controller design from physical equations, and experimental data. It is required that a model developed to evaluate the performance of a controller be simple and accurate to a certain extent. In this step, making a model easily and quickly is required, so Dymola is used here.

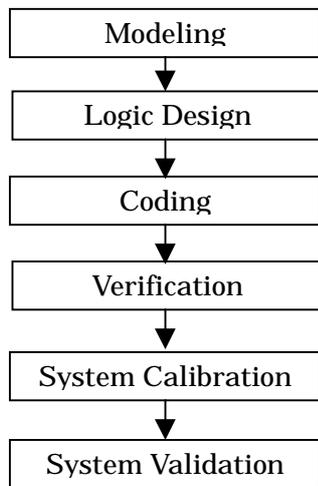


Figure 1: Model based process

In the 2nd step (Logic Design) we design a controller in simulink, and evaluate the performance of the controller on a PC. The control logic is coded in simulink, because MATLAB/SIMULINK has powerful functions useful in the design of a controller. In this step, we evaluate the control logic in a closed loop with the model which is made in the 1st step. In the 3rd step (Coding) the control logic is encoded in C to put in an ECU. In the 4th step (Verification) we evaluate the C code to make sure it free of bugs. In the 5th step (System Calibration) we adjust parameters and gains which are

included in the logic. In the final step (System Validation) we evaluate the entire ECU system which includes the gain adjusted control logic downloaded to it. In that evaluation, we use a Hardware In the Loop Simulation or HILS. HILS requires a plant model which can be calculated in real time. That model was being constructed in Simulink until now, but recently increasingly models constructed in Dymola and transferred to an S-function are being used. We think that this is a natural change, because models which are coded in Dymola are increasing.

In this way, the use of Dymola has been becoming increasing recently.

3. Usage of Dymola

Dymola is used mainly in the drivetrain and engine divisions.

3.1 Drivetrain Division

In the drivetrain field, Dymola has been used for analyzing the vehicle acceleration, shock, and noise vibration performance, and fuel economy.

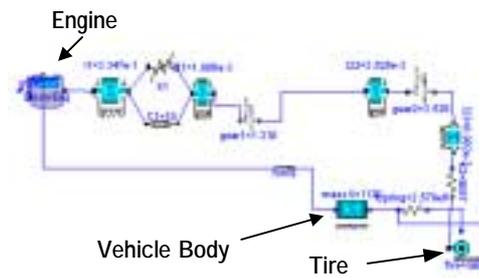


Figure 2: Drivetrain model

Figure 2 shows one of the drivetrain models we have used. This model includes an engine, clutch, manual transmission, crank shaft, driveshaft, tire, and vehicle body. The engine, clutch, and tire models were constructed by Toyota. The engine model calculates the cyclic torque. To evaluate

noise vibration, an accurate engine torque is required. Therefore, the torque is calculated from the cylinder pressure which was measured in an actual engine. The tire model calculates a longitudinal force using the slip ratio which is calculated from wheel speed and vehicle speed. The acceleration, shock and noise vibration performance is evaluated using this Dymola simulation. The information obtained from this evaluation is feedback into the design of vehicle specification.

Dymola has also been used in a closed loop fashion. The amount of fuel injected into the engine is obtained from a simulink model of the engine control logic. The amount of fuel is then used to calculate the output torque of the engine and this is input to the drivetrain model in Dymola. (See Figure 3.) In this case, at first we made the drivetrain model using Dymola, and then translated it to an S-function to use in Simulink. Then, we combined that drivetrain model with the engine control logic which was coded in simulink. We made the plant model in Dymola because it was easier to use than simulink.

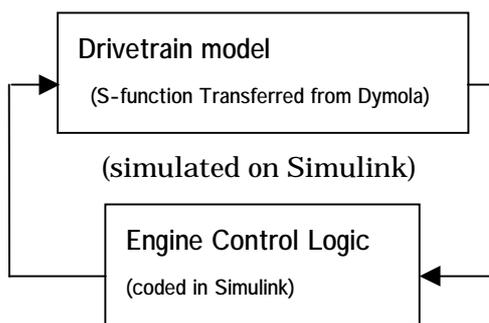


Figure 3: Engine controller-drivetrain model

3.2 Engine

Dymola has a many standard libraries, but regrettably, there is no engine library. To improve the accuracy of engine control, and to reduce development time, model based

controller design is extensively used. We have constructed an engine model for the purpose of analyzing certain engine phenomena, and for deciding the appropriate model to use within the engine control logic. Figure 4 shows a diesel engine model.

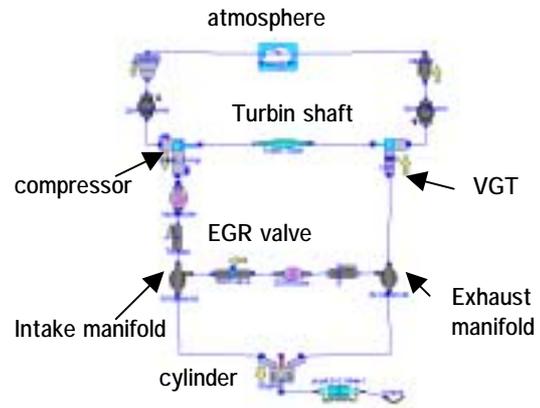


Figure 4: Diesel Engine Model

The engine model was built by bringing together smaller models representing each part of the engine (Intake manifold, Compressor, Cylinder etc). Each engine part model is registered in a library. These models were built using Mean Value Engine Model^{[1][2]}, therefore oscillation of torque or cylinder pressure is not calculated. The conservation of energy, mass, and degrees of freedom are used to derive the model. Figure 4 shows an engine with a Variable Geometry Turbo or VGT, and an Exhaust Gas Recirculation system or EGR.

The EGR system reduces nitrogen oxide (NOx) by recirculating exhaust gases back to the intake manifold. The VGT system increases the exhaust pressure by restricting the flow of burned gas using vanes installed at the entrance. To achieve low emission levels, it is important that we control the VGT and EGR correctly. But the intake manifold pressure and fresh air/EGR gas flow into the engine is influenced by the VGT and EGR simultaneously. It is required that a controller calculates each

effect while the controller controls these actuators. This is a very complex phenomenon, and many papers^[3] have published using a model based approach to the controller design. We have been using model to design our control logic.

We want to use this engine library for other engine types (other than diesel with VGT and EGR). There are many engine type which we would like to simulate. With Dymola, it is very easy to change a model configuration, we could make an engine model without the VGT or without the EGR system easily, by just removing the VGT or the EGR, and joining the lines.

3.3 HILS

We are increasingly using S-function models which have been constructed in Dymola. One particular area of use is HILS. In this case, the model is calculated in fixed calculation step, and the calculation speed is very important. Up to now, we have built the model in simulink considering the calculation speed. We want to be able to put a model which was made using Dymola into a HILS system.

In HILS, using a fixed calculation step creates some problems. When we use the equation(1) to solve an air mass flow by in a simulation, when the up stream pressure(p_1) and down stream(p_2) are nearly equal, the ratio p_2 / p_1 goes to 1, and the slope of dm / dp becomes infinite. Because of this, the mass flow and the up and down stream pressure calculated by the equation oscillate (see figure5.). This is a typical problem in a fixed step solver.

$$\dot{m} = CA\sqrt{2p_1\rho_1}\Phi \quad (1)$$

$$\Phi = \begin{cases} \sqrt{\frac{\kappa}{\kappa-1} \left[\left(\frac{p_2}{p_1} \right)^{\frac{2}{\kappa}} - \left(\frac{p_2}{p_1} \right)^{\frac{\kappa+1}{\kappa}} \right]} \cdot \dots \text{if } \left(\frac{p_2}{p_1} \right) > \left(\frac{2}{\kappa+1} \right)^{\frac{\kappa}{\kappa-1}} \\ \left(\frac{2}{\kappa+1} \right)^{\frac{1}{\kappa-1}} \sqrt{\frac{\kappa}{\kappa+1}} \cdot \dots \dots \dots \text{if } \left(\frac{p_2}{p_1} \right) < \left(\frac{2}{\kappa+1} \right)^{\frac{\kappa}{\kappa-1}} \end{cases}$$

To solve problem, we can set the sampling time short, but in a real time simulator, we don't have this freedom to shorten the sampling time. Therefore, we have coped with this problem by changing the model. So, we end up with two models: one for HILS use and one for evaluation of the control logic.

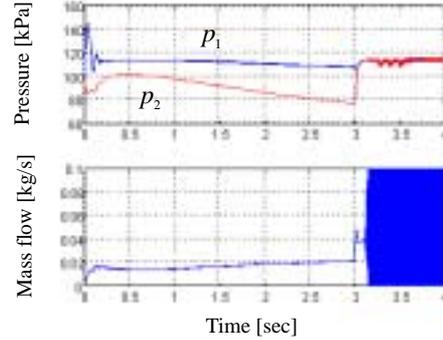


Figure 5: Fixed calculation step problem

4. Improvement requests of Toyota

As this paper has described, the usage of Dymola is increasing within Toyota. We have asked the Dymola users within Toyota, "What would you like to see as an improvement in Dymola?". From this, we think that the most important request concerns the calculation time in simulink when we use a model which was coded in Dymola and then transferred to an S-function. This problem is as follows. When we perform a simulation with a model in Dymola, the calculation speed is very fast. But, when we transfer the model to an S-function, and perform the same simulation in simulink, it takes much longer. We have heard that this is caused by the solver performance of simulink, and it is difficult to shorten the calculation time. In simulink, when the simulation calculation time is long or the simulation gives up, it is fairly easy to change the way the simulation is written to get around this problem. Unfortunately, when a

model is written using the standard Dymola libraries, it is not so easy to change how the model is written. Therefore, in Dymola when the simulation calculation time is long, we have no recourse action. Because of this, we feel it is very important to ensure that the simulation time is as short as possible in Dymola.

Other improvements we would like to see are in the user interface. We would like to make below things more user friendly.

- (1) Plot window (Graph).
- (2) Edit Window (Editor).

5. Modelica within Toyota

In Toyota, almost all Dymola users use the Dymola language, and they admit that Dymola is an excellent simulation tool. Regrettably, we don't understand the advantage of the Modelica language at present. We want to watch what happens with Modelica, and then ascertain when to change to Modelica language.

6. Conclusion

In this paper, examples of usage of Dymola were presented. Within Toyota in the drivetrain and engine division, Dymola has been used in many phases: design of vehicle specification, design of control logic, HILS

in the step of system validation. In these divisions, Dymola is thought in the development phase of a necessary and indispensable tool.

Finally, some desired improvements were stated. We are hoping that Dymola will be come a more convenient and useful simulation tool to fit our user needs, and that development time can be further reduced by using Dymola.

References

- [1] Hendricks, E., and Sorenson, S. C., "Mean Value Modelling of Spark Ignition Engines", SAE Technical Paper No.960616
- [2] Hendricks, E., Chevalier, A., Jensen, M., and Sorenson, S. C., "Modelling of the Intake Manifold Filling Dynamics", SAE Technical Paper No.960037
- [3] M. J. van Nieuwstadt, I.V. Kolmanovsky and P. E. Moraal, "Coordinated EGT-VGT Control for Diesel Engines: an Experimental Comparison", SAE Technical Paper 2000-01-0266