Modeling of Conventional Vehicle in Modelica

Wei Chen, Gang Qin, Lingyang Li, Yunqing Zhang, Liping Chen

CAD Center, Huazhong University of Science and Technology, China

chenw@hustcad.com

Abstract

Modelica is a modern language used to model physical systems. The language is object-oriented, non-causal1 and the models are mathematically described by differential algebraic equations. The characteristic of modelica language make it very suited to define model libraries with reusable components, model complex applications involving parts from several application domains, and many more useful facilities.

InteDrive library was created for simulating automotive driving performance, fuel consumption and emissions. The library is yet under developed by Huazhong University of Science and Technology in modelica language. The aim of the library is to provide the user with an easy to use and highly replaceable set of vehicle component models, and predict the vehicle performance, especially fuel consumption for a given cycle.

The main components of this library and their applications are introduced in this paper. The simulation was carried out by MWorks, which is a general modeling and simulation platform developed by Huazhong University of Science and Technology. The simulation results were compared with ADVISOR. The easy and fast modeling process shows that modelica is very useful for the modeling and simulation of vehicles.

Keywords: InteDrive library; Simulation; Modelica; MWorks

1 Introduction

Automotive manufacturers have been striving for decades to produce vehicles which satisfy customers' requirements at minimum cost. Many of their concerns are on fuel economy, road performance and driveability. Improving fuel economy is both a political concern of alleviating dependency on foreign fuel and a customer preference of reducing vehicle operating cost. Consumers also expect vehicles to provide satisfactory performance with desirable driving comfort. So it is very necessary to predict the vehicle performance when the vehicle is design.

There have been a lot of program tools to predict vehicle performance, for example, cruise by AVL, ADVISOR, PSAT, etc. However, these tools are block-oriented and demand a huge amount of manual rewriting to get the equations into explicit form. Hence, these tools are less extensible, and hard to reuse. It brings too much inconvenience to the user.

Modelica is an object-oriented language for modeling of large and heterogeneous physical systems. The language is object-oriented, non-causal1 and the models are mathematically described by differential algebraic equations. These characteristics make it fast in modeling and easy to reuse modeling knowledge[1]. Modelica has been used to model various kinds of systems and proved to have superiority over traditional tools in modeling efficiency, especially over Matlab/Simulink.

InteDrive library was developed in modelica language to predict automotive driving performance, especially for fuel consumption. The aim of the library is to provide the user with an easy to use and highly replaceable set of vehicle component models.

MWorks is under developed by Huazhong University of Science and Technology. It is a general modeling and simulation platform for complex engineering systems which supports visual modeling, automatically translating and solving, as well as convenient postprocessing. The current version is based on Modelica 2.1 and implements almost all the syntax and semantics of Modelica.

A vehicle model was built with InteDrive library, and the simulation was carried out with MWorks. The simulation results are compared with ADVISOR, and show the correctness of the model.

2 Components

The InteDrive library contains some components of a conventional vehicle, such as engine, clutch, gearbox, etc. The modeling process have referred the paper[2-4]. The present structure of InteDrive can be viewed in Fig.1.

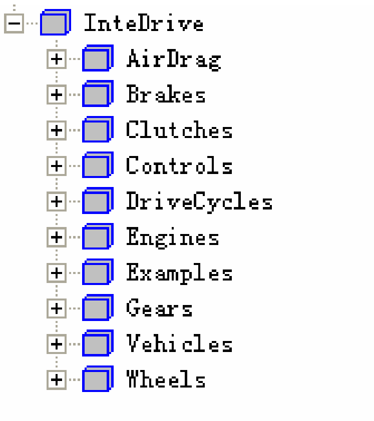

Fig.1 The structure of the InteDrive

The components of the InteDrive library are described as below.

2.1 Vehicles

The vehicle is one of the main objects in a model. This component contains general data of the vehicle, such as nominal dimensions and weights. The library presents only dynamic models for the longitudinal motion of the vehicle. So a sliding mass may represent as vehicle body. The model will be developed in the future for considering the load transfer to the rear or front axle when the vehicle is accelerating or braking.

2.2 DriveCycles

The drive cycle is the vehicle speed trace versus time. It is very useful for the evaluation of fuel economy and emissions. When the simulation was carried out, the vehicle speed must follow the speed profile to calculate the fuel consumption and emissions. This package includes tables for several driving cycles. At present, it contains the NEDC, UDDS, NYCC, HWFET and some standard driving cycles. Any other cycle can be added easily if desired.

Interpolation to the cycle table was used to get the vehicle speed when the simulation was carried out.

2.3 Clutches

The clutch contains the model of a friction clutch as used in cars with manual gear boxes. It is controlled by the driver via the clutch pedal position. In this paper, we adopt the clutch model in modelica standard library, and made some modification for simplification. The maximum normal force was changed to maximum transferable torque. So the parameters of the clutch may be acquired more easily.

2.4 Brakes

This is described by braking data and dimensions. By the implementation of a specific braking factor it is possible to model disc brakes as well as different forms of drum brakes. In this paper, a brake model in modelica standard library was adopted.

2.5 Gears

This package contains the gears in a vehicle, such as gear box, differential, final drive, etc.

The engine torque is turned into a power take-off torque by considering the transmission, the mass moments of inertia, the moment of loss. The modeling of gear box can be referred to the paper[3-5].

2.6 Engines

The component engine contains a model for a combustion engine. The engine was modeled by a structure of characteristic curves and maps. As the characteristic curves for the full load, the fuel consumption and others can be freely defined by the user. It is possible to define a gasoline engine as well as a diesel engine. Interpolation to the fuel map was used to get the fuel consumption. The emission can be calculated also if the emission map was defined.

2.7 Wheels

The wheels and tires link the vehicle to the road. In this paper, a block called IdealGearR2T in modelica standard library was used to model the wheel. It converts the rotational motion to translational motion. A force acted on the wheel to model the rolling resistance.

2.8 Controls

The vehicle is controlled to make the vehicle speed follow the driving cycle profiles. The controls include throttle control, brake control, gearbox control, clutch control and so on.

The throttle and the brake are controlled by PI controllers. The input to the PI controller was the error of vehicle speed acquired by simulation and the speed requested by the driving cycle. If the vehicle speed exceeds the reference speed, the driver controls the brake to let the vehicle slow down. If the vehicle speed is lower than the reference speed, the driver controls the throttle to let the vehicle accelerate. The PI parameters were tuned to control the vehicle properly.

The gearbox controller shifts the gears according to the vehicle speed. It is necessary to define the up- and downshifting velocities always only for one gear less than are available in the gear box (i.e. for a five step gear box, only for four gears the up- and downshifting velocities have to be defined). As can be seen in figure 2, the upshifting velocity of the 2nd gear means that at this velocity the gear box control is upshifting from the 2nd into the 3rd gear. The downshifting velocity for the 2nd gear means that at this velocity the gear box control is downshifting from the 3rd into the 2nd gear.

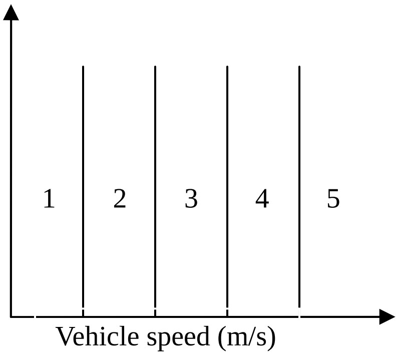

Fig. 2 Shift strategy by vehicle speed

The clutch control determines whether the clutch should be fully engaged, fully disengaged in this paper. The state of the clutch depends on the requirements of the drivetrain.

If the gear is changing, for an upshift or downshift, the clutch is disengaged. If no (positive) torque is required of the engine and or the speed required of the engine is less than its idle speed then the clutch is disengaged. Otherwise, the clutch is engaged.

2.9 AirDrags

This component models the air resistance force act on the vehicle. Usually, the aero dynamic resistance force is approximated by simplifying the vehicle to be a prismatic body with a frontal area A_f. The force caused by stagnation pressure is multiplied by aerodynamic drag coefficient c_d that models the actual flow conditions.

$$F_a = \frac{1}{2}\rho_a A_f c_d v^2$$

Here, v is the vehicle speed and ρ_a is the density of the air. The parameter c_d must be estimated using CFD programs or experiment in wind tunnels.

3 MWorks

MWorks is a general modeling and simulation platform for complex engineering systems which supports visual modeling, automatically translating and solving, as well as convenient postprocessing. It is under constant developing by Huazhong University of Science and Technology. The current version is based on Modelica 2.1 and implements almost all the syntax and semantics of Modelica.

MWorks has features as follows:

a. With modern integrated development environment styles, it provides friendly user interfaces such as syntax high-lighting, code assist etc.;

b. Based on object-oriented compiler framework, it perfectly supports almost all the syntax and semantics of Modelica;

Using self-adapting solving strategies, it can agilely solve differential equations, algebraic equations and discrete equations.

MWorks Studio is a visual modeling environment which supports drag-drop modeling based on Modelica Standard Library. It is also an integrated development environment integrating with translator, optimizer, solver and postprocessor.

As a developing tool, this studio provides many modern IDE styles to promote the users' conveniences just as Eclipse or Microsoft Visual Studio does, such as real-time syntax highlighting, content assist, code formatting, outlining etc.

The snapshot of MWorks Studio is shown as Figure 3.

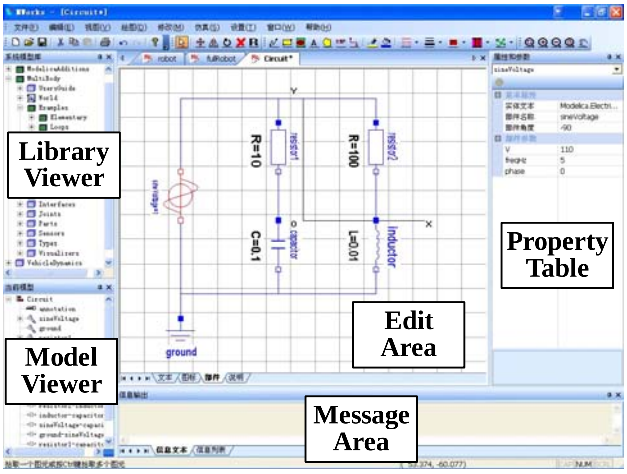

Fig. 3 Snapshot of MWorks Studio

The library viewer illustrates all predefined system libraries, all loaded user libraries and other top models in the memory. The model viewer shows all components of the current model. The edit area is visual modeling, text modeling, icon-editing or information area, and the status is chosen by tag. The property table displays all properties of selected element in the model, and the properties can be edited here. The message area displays all messages in the checking, translating, or simulating, including status and error messages. The error can automatically be located by double clicking error message.

The auxiliary functions of real-time syntax highlighting, content assist, code formatting and outlining are provided in the text modeling status.

The solver of MWorks includes two primary modules: collection of algorithms and console of solving strategies. Solver provides different basic algorithm alternatives for users to select appropriate one. Now, a series of algorithms for different kinds of equations have been collected in the solver, such as SUNDIALS.

More details about MWorks can be referred to paper [6].

4 Simulation

A complete conventional vehicle was built by drag components from InteDrive library. The vehicle model can be seen in figure 4. The vehicle was modeled by a forward-facing approach include the driver model, which controls the throttle, brake, clutch, gearbox to make the vehicle speed follow a given driving cycle.

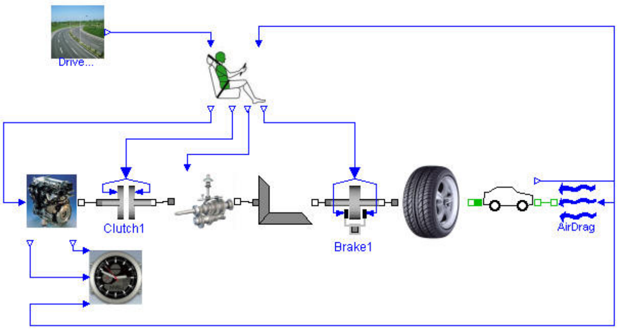

Fig. 4 Vehicle model

The key parameters for the vehicle simulation are listed in Table 1.

Table 1 Key parameters of the vehicle

Components	**Key Parameters**
Engine	Maximum power: 41kW@5700rpm Maximum torque: 81Nm@3477r/min
Final Drive	3.77
Transmission	3.5676/2.008/1.3289/1.0/0.7525
Vehicle Mass	1000Kg
Wheel Radius	0.282m
Rolling Resistance	0.09
Wheelbase	2.6m
Frontal Area	$2m^2$
Coefficient of Air Drag	0.335

The other parameters can be referred to ADVISOR2002, with the vehicle config file called CONVENTIONAL_default_in. The simulation works was carried out with MWorks.

The vehicle is driven with the UDDS driving cycle and the actual vehicle speed is given in figure 5. The total distance of the UDDS was 11.991 km. The difference between desired speed from the UDDS driving cycle and the actual speed is shown in figure 6. It seems the vehicle has been controlled as desired. The signal from the gear box can be seen in figure 7. The fuel information was shown in figure 8. The fuel use was 0.69 liter at the end of the cycle. So if the vehicle drives a distance with 100 kilometers, the fuel consumption was 100/11.991*0.69=5.75 liter. The result of ADVISOR is 5.9 liter. It shows the correctness of our models.

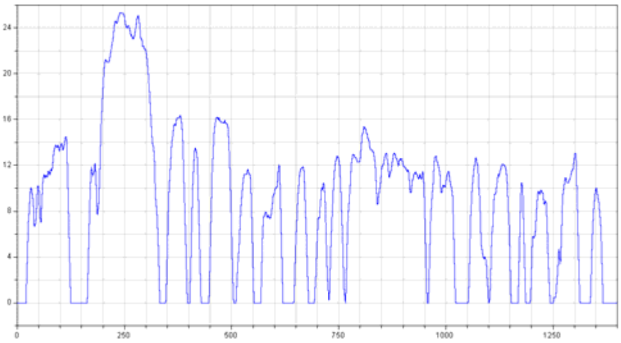

Fig. 5 Actual speed for the reference car during the UDDS driving cycle. X-axis shows time [s], y-axis vehicle velocity [m/s].

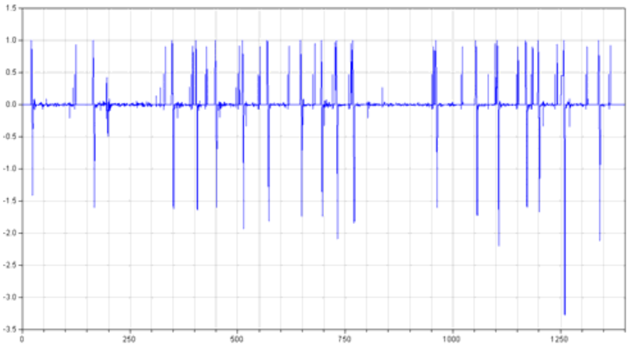

Fig.6 Difference between UDDS speed and actual vehicle speed during the simulation. X-axis shows time [s] and y-axis shows velocity [m/s].

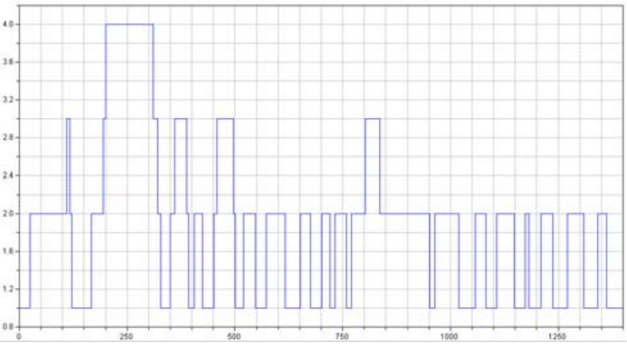

Fig.7 Gear number during the simulation. X-axis shows time [s] and y-axis shows gear number.

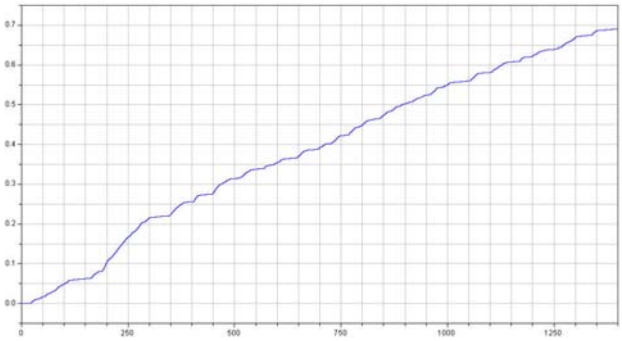

Fig.8 Fuel consumption during the simulation. X-axis shows time [s] and y-axis shows fuel [L].

5 Conclusions

In this paper, a library for modeling of simulation automotive fuel consumption is introduced. It uses the interfaces from the Modelica and ModelicaAddtions packages to be compatible with other libraries. The easy and fast modeling process shows the superiority of modelica language in modeling.

InteDrive provides various vehicle component models to simplify for the user to build the vehicle model according to their needs. The modular structure of the model design allows to take advantage of the Modelica language.

6 Future work

The library is under constant development. The models in the library will be modified with more detailed description. Much more models such as viscous-clutch, torque converter, CVT, and so on will be developed. Much more components with electrical modules will be developed, and the library is aimed to model conventional vehicles and electric vehicle.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No. 60574053), the National High-Tech Development 863 Program of China (Grant No. 2006AA110105).

References

[1] Modelica Association. http://www.modelica.org.

[2] Fritzson, P., et al. The Open Source Modelica Project. Proceedings of the 2nd International Modelica Conference, 18-19 March, 2002. Munich, Germany.

[3] P. Nobrant. Driveline Modelling using MathModelica. Master's thesis. Linkopings universitet, Linkoping, Sweden, 2001.

[4] Johanna Wallen. Modelling of Components for Conventional Car and Hybrid Electric Vehicle in Modelica. Master's thesis. Linkopings university, Linkoping, Sweden, 2004.

[5] Otto Montell. Advanced concepts in Modelica and their implementation in VehProLib. Master's thesis. Linkopings university, Linkoping, Sweden, 2004.

[6] FAN-LI Zhou, LI-PING Chen, etc. MWorks: a Modern IDE for Modeling and Simulation of Multidomain Physical Systems Based on Modelica. Proceedings of the 5th International Modelica Conference, 2006.