Soejima S., Matsuba T.:

Application of mixed mode integration and new implicit inline integration at Toyota

2nd International Modelica Conference, Proceedings, pp. 65-1 – 65-6

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

Application of mixed mode integration and new implicit inline integration at Toyota

Shinichi Soejima
Toyota Motor Corporation

Takashi Matsuba
Toyota Techno Service Corporation

The HILS (Hardware In the Loop Simulation) is a popular technique to debug control logic of vehicles. Previously, only simplified models could be used to achieve real time performance in the simulations. On the other hand, quite detailed models of engine, drivetrain, hydraulics and brake system were developed with Dymola in recent years. Therefore we would like to use these models also in HILS. However, real time is difficult to obtain for stiff model components, such as the hydraulics, because integrators with fixed step size must be used. With explicit methods very small step sizes are needed to ensure stability. With implicit methods large nonlinear systems of equations have to be solved. Both approaches seem to be not feasible. To improve this situation, the new inline and mixed mode integration technique introduced by Dymola is evaluated for an engine model and results are reported.

1. Introduction

Concerns over fuel consumption and environmental problems have brought about a demand for higher performance in automobiles. To achieve this, the development of highly advanced systems using control technologies that incorporate the use of numerous actuators and sensors has been progressing. The composition and control of such systems is becoming increasingly more complicated. However, at the same time, reducing the length of their development period is also necessary. Applying simulation is essential for achieving this task. Particularly, HILS (Hardware In the Loop Simulation) is widely utilized in the debugging of ECUs (Electronic Control Units). In HILS, the ECU carries out its operation in real time, and as a result, the model is also required to carry out its operation in real time. Simple models with experimental data tables and transfer functions have been used for HILS so far. However, the demand is rising for Dymola models, that have been developed during the design of control logic, to be used in HILS without changes.

Physical models have typically a large span of time constants making them stiff for real time calculation. When using the explicit Euler method, the step size must be less than the fastest time constant in order to maintain numerical stability. However, in real-time simulation using HILS, the step size cannot be set shorter than the length of time necessary for calculating the new values of the model variables. The implicit Euler method allows a larger step size to be used. However, it implies that a nonlinear system of equations needs to be solved at each step. Reducing the size of the non-linear problem is advantageous. Dymola [1] exploits the method of inline integration [3,4] to support this. The discretization formulas of the integration methods are combined with the model equations. To reduce the size of the resulting non-linear problem, Dymola analyses the structure of the problem and manipulates it symbolically. The symbolic manipulation has recently been improved [4]. The improvements include also inline integration of higher order methods to obtain better accuracy for larger steps.

Another method, "mixed-mode integration", of reducing the size of the system of non-linear equations is to use explicit discretization on slow states and implicit on fast states. The problem is then to find which states that are slow and which that are fast. A method based on linearization and eigenvalue analysis was presented in [6]. Unfortunately, it is not straightforward to use this method.

This paper reports results from applying inline integration and mixed mode integration to two real applications.

2. Application: Engine model

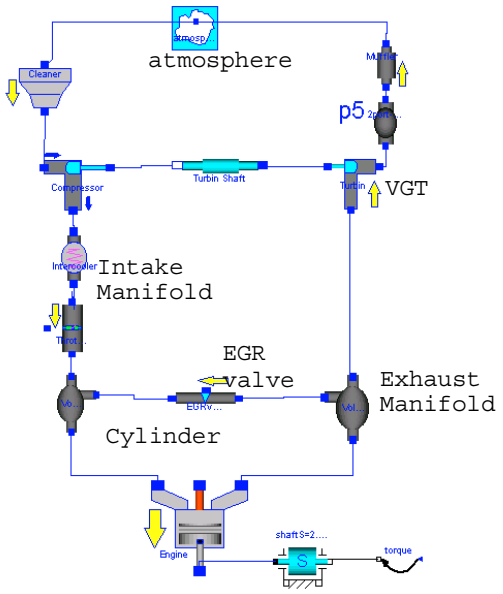

Figure 1: Engine model

Figure 1 shows the structure of the engine model which was used for evaluation. The engine has a Variable Geometry Turbo (VGT) and an Exhaust Gas Recirculation (EGR) system. 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 to control the VGT and EGR correctly. A complication is that both the VGT and the EGR influence the intake manifold pressure and fresh air/EGR gas flow into the engine.

The model is a mean value engine model. The variation of torque or cylinder pressure during a cycle is not calculated. The model builds on conservation of energy and mass and Newton's equations of motion. The amount of air mass flow and EGR gas flow, and the pressure and temperature of every part of the engine are calculated. The model has 796 unknowns. After Dymola's elimination of constant and alias variables at translation, 183 nontrivial and time-varying variables remain. The model has 26 continuous time states. The mass flows of the air or EGR gas are calculated by the Equation (1).

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

$$\Phi = \begin{cases} \sqrt{\frac{\kappa}{\kappa-1}\left\{\left(\frac{p_2}{p_1}\right)^{\frac{2}{\kappa}} - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa+1}{\kappa}}\right\}} & \cdots if\left(\frac{p_2}{p_1}\right) > \left(\frac{2}{\kappa+1}\right)^{\frac{1}{\kappa-1}} \\ \left(\frac{2}{\kappa+1}\right)^{\frac{1}{\kappa-1}}\sqrt{\frac{\kappa}{\kappa+1}} & \cdots\cdots\cdots if\left(\frac{p_2}{p_1}\right) < \left(\frac{2}{\kappa+1}\right)^{\frac{1}{\kappa-1}} \end{cases}$$

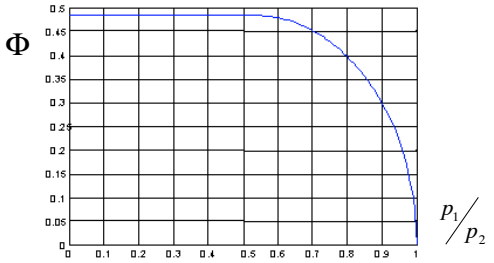

Figure 2: The flow characteristics

The equation shows that the flow changes are large when there is a small difference between the upper pressure and the lower pressure. It implies that the time constants of the dynamics change. As a result, the use of explicit Euler method with fixed step size, requires a small step size for the simulation.

2.1 Evaluation of inlined explicit Euler

We compared the calculation time of two 10 seconds simulations, one in which the explicit Euler inline integration method was applied and one in which a non-inlined explicit Euler method was used. Table 1 shows the results. We can see a 39% performance improvement when using the inlined explicit Euler in Dymola.

The results are labelled "Simulink" when simulating the S-function generated by Dymola in Simulink. The `sim´, `tic´ and `toc´ commands were used for timing. It should be noted that using the simulate button in Simulink 3.0 (Matlab 5.3) made the simulation of these models about 50% slower. This situation has been improved in Simulink 4.0 (Matlab 6.0).

It should be noted that there was not any significant improvement in speed when simulating in Simulink. The reason has not yet been determined.

Table 1: Performance for explicit Euler when simulating the engine model for 10 seconds using an Intel Pentium II 350 MHz processor.

	Integration Method	Step [ms]	CPU Time [s]
Dymola	Explicit Euler	0.1	41
	Inlined Expl Euler	0.1	25
Simulink	Explicit Euler	0.1	46
	Inlined Expl Euler	0.1	44

2.2 Evaluation of mixed mode integration

Next we evaluated mixed mode integration. The partition with 4 fast state variables discussed in [6] was used. Table 2 shows the results of a comparison of calculation time when mixed mode integration was used with the Engine model and when a non-inlined explicit Euler method was used. The mixed mode integration method showed very high performance. The improvement of the performance is about 85%. And without mixed mode integration, the model simulation needed 0.1 ms of step size to ensure calculation stability. On the other hand, with mixed mode integration the calculation was still stable with a step size of 1.0 ms.

The explict Euler method cannot be used for HIL simulation, because a step size of 0.1 ms is needed, which is too short for HILS calculation. The mixed mode integration method allows HIL simulation as the results from running on dSPACE 1005 (PowerPC 750, 480 MHz) indicate. However, there are problems of using the mixed mode approach in general

1. It is difficult to find the suitable partitioning. The operation is difficult to generalize because the partitioning depends on the characteristic of each model, and thus requires trial and error analysis on each occasion.
2. Once partitioning has been made, stability is not guaranteed when the input is changed.

Table 2: Performance for mixed mode integration when simulating the engine model for 10 seconds using an Intel Pentium II 350 MHz processor.

	Integration Method	Step [ms]	CPU Time [s]
Dymola	Explicit Euler	0.1	41
	Mixed Mode	1	6.1
Simulink	Explicit Euler	0.1	46
	Mixed Mode	1	7.1
dSPACE	Mixed Mode	1	Not Realtime
	Mixed Mode	1.3	Realtime

2.3 Evaluation of inlined implicit Euler

In order to provide more easy to use method than the mixed mode integration method, the implicit line method reported in [6] has been considerably improved [4]. The performance results of using this method are shown in Table 3.

Table 3: Performance for inlined implicit Euler integration when simulating the engine model for 10 seconds using an Intel Pentium II 350 MHz processor.

	Integration Method	Step [ms]	CPU Time [s]
Dymola	Explicit Euler	0.1	41
	Inlined Impl Euler	0.1	14.3
Simulink	Explicit Euler	0.1	46
	Inlined Impl Euler	0.1	15

It can be noted that the simulation time was about 15 seconds on the Pentium II processor, i.e. 1.5 ms/step. Since dSPACE DS1005 according to Table 2 needs more CPU time per step it would need at least 2 ms to calculate one step. However, using offline simulation in Dymola, it was determined that convergence was not obtained when the step size 2 ms was used.

Faster HILS environments based on Pentium processors are available, for example xPC from MathWorks. Dynasim made test using an Intel Pentium 4, 1.6 GHz processor. Table 4 shows the performance results for simulating in Dymola on xPC. The model could then be run in real-time.

Table 4: Performance for inlined implicit Euler integration when simulating the engine model for 10 seconds using an Intel Pentium 4, 1.6 GHz processor.

	Integration Method	Step [ms]	Comp Time [s]
Dymola	Explicit Euler	0.1	11.4
	Inlined Impl Euler	1	4.6
Simulink	Explicit Euler	0.1	13.6
	Inlined Impl Euler	1	4.6
xPC	Inlined Impl Euler	1	Realtime/4.5s

3. Application: Hydraulic system

The hydraulic system is one of the most important systems in an automobile. It plays a crucial role in the power train and the drive train. As the hydraulic system exhibits very complicated characteristics, it is very useful to simulate it. Additionally, the ability to simulate such systems in real time is very important. So far real-time simulation of hydraulic systems has not been possible. Hydraulic systems are typical examples of very stiff systems. It is necessary to use implicit solvers.

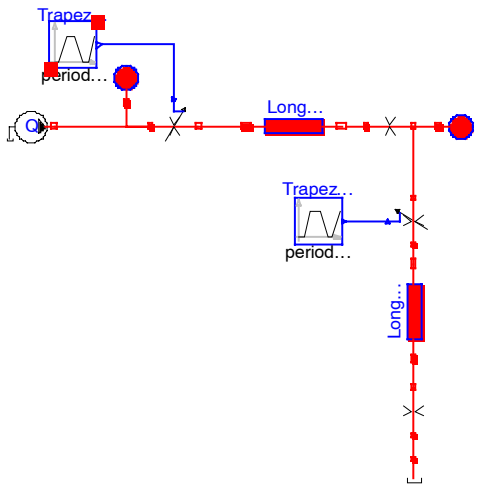

Figure 3: Hydraulic actuator system.

As the next application, consider simulation of an actuator which is usually used to control the pressure in automobiles. Figure 3 shows the structure of the model developed by using components from the HyLib hydraulics library [2]. There are a number of valves and pipes between a high-pressure source and low-pressure parts such as the brake at a wheel. The dynamics within the pipes cannot be neglected, because they are several meters long. The valves adjust the pressure of the lower part by very fast actuation. This introduces very fast pressure changes and the pressure wave propagate with the speed of sound which is approximately 1300 m/s. The result may be oscillations in the range 400-500 Hz generating vibration and noise within the automobile system. See Figure 4. It is important to analyze this phenomenon in order to develop ways of reducing the vibration and noise.

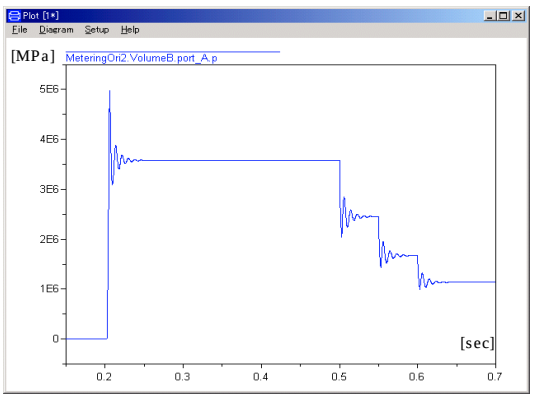

Figure 4: Pulsation in the long hydraulic line

The two pipes of the model are each 3 meters long. Each pipe is modeled by 15 segments, where each segment is 0.2 m long and has 5 continuous time states. The model has in total 164 states.

It is not possible to use explicit Euler for real-time simulation, because the model requires the step size to be less than 0.001 ms to ensure the stability of the simulation.

A the third order implicit inline Runge-Kutta integration method, RK3, with a step size of 0.5 ms was used to obtain both stability and desired accuracy. The symbolic manipulation of Dymola reduces the size of the nonlinear system to be solved by a nonlinear solver from 164 to 10.

Table 5 shows the performance results for the Pentium II processor when simulating 1 second.

Table 5: Performance for inlined implicit RK3 when simulating the hydraulic actuator model for 1 s using an Intel Pentium II 350 MHz processor.

	Integration Method	Step [ms]	CPU Time [s]
Dymola	Explicit Euler	0.001	283
	Inlined Impl RK3	0.5	3.1
Simulink	Explicit Euler	0.001	253
	Inlined Impl RK3	0.5	3

It should be noted that the inlined implicit RK3 gave a speed up of a factor of 91. However, the simulation is 3.1 times too slow to run in real-time. Therefore Dynasim made corresponding test shown in Table 6 for Pentium 4, 1.6 GHz. Real-time performance was then achieved.

When attempting to test on xPC, the diagnostic `Failed to download´ was received. More investigations have to be made to determine the cause.

Table 6: Performance for inlined implicit RK3 when simulating the hydraulic actuator model for 1 s using an Intel Pentium 4, 1.6 GHz processor.

	Integration Method	Step [ms]	CPU Time [s]
Dymola	Explicit Euler	0.001	59.1
	Inlined Impl RK3	0.5	0.71
Simulink	Explicit Euler	0.001	50.9
	Inlined Impl RK3	0.5	0.6

These results show that the improved implicit inline integration method is very effective in this case. Conventionally it was difficult to handle hydraulic models using fixed step explicit Euler and process the model within a practical calculation time. Using the improved implicit inline integration method it is possible to handle hydraulic models and process them at high speeds even in fixed step situations.

This characteristic is desirable for real time simulation. Implicit inline integration produced a remarkable improvement for this simple hydraulic model. Furthermore, it has also shown considerable effectiveness even for models that are more complicated. The representative characteristics of this method are as follows.

- This method enables the generation of a code which is suitable for real-time simulation, even in the case of models that have the property of oscillation, such as the hydraulics system.

- This method makes the handling of models with larger step size possible.

4. Conclusions

We evaluated inline integration and mixed mode integration which were developed to improve calculation performance. Improved implicit inline integration, which has recently been developed, was also evaluated. It was confirmed that these methods effectively increased the ability to handle models in real time. In particular the improved implicit inline 3rd order integration method could handle the model very efficiently even if it was a hydraulic model.

On the other hand, it was generally the case that setting step size for fixed step size integrators remained a problem. Even inline integration needs trial and error testing to find a suitable step size. In addition, an increasing number of modeling beginners are using these kinds of tools. These users are not always experts in modeling or control. It would therefore be desirable if they could more easily obtain improved performance using the techniques we evaluated. We hope that such method will be established and be applied to the physical models, which were made for controller designing in order to enable HILS.

The improved implicit higher order integration method has given a break-through in simulation speed for stiff models and for discreticized partial differential equations originating for example in long hydraulic pipes.

References

[1] Dymola. *Dynamic Modeling Laboratory*, Dynasim AB, Lund, Sweden, http://www.Dynasim.se.

[2] P. Beater, "*Modeling and digital simulation of hydraulic systems in design and engineering education using Modelica and HyLib*", Modelica Workshop 2000 proceedings pp 33-40

[3] H. Elmqvist, F. Cellier, M. Otter: *Inline Integration: A new mixed symbolic/numeric approach for solving differential-algebraic equation systems.* Proceedings European Simulation Multiconference, June 1995, Prague, pp: XXIII-XXXIV.

[4] H. Elmqvist, S.E. Mattsson, H. Olsson: *New Methods for Hardware-in-the-loop Simulation of Stiff Models,* Proceedings of Modelica Conference 2002. Modelica homepage: http://www.Modelica.org.

[5] Modelica. Modelica homepage: http://www.Modelica.org.

[6] A. Schiela, H. Olsson, "Mixed mode integration for Real-time Simulation", Modelica Workshop 2000 proceedings, pp 69-75

[7] S. Soejima, "Examples of usage and the spread of Dymola within Toyota", Modelica Workshop 2000 proceedings, pp 55-59