

Integration of complex Modelica-based physics models and discrete-time control systems: Approaches and observations of numerical performance

Kai Wang¹ Christopher Greiner¹ John Batteh² Lixiang Li²

¹ Ford Motor Company, USA, kwang37@ford.com, cgreiner@ford.com

² Modelon, Inc., USA, john.batteh@modelon.com, lixiang.li@modelon.com

Abstract

A Modelica-based air conditioning (A/C) system model has been integrated, closed-loop, with related S-function-based controls in the Simulink environment. The integration was performed with two different approaches, with a DymolaBlock-based S-function for the A/C model, and as a co-simulation FMU. The simulation performance of the integrated model needs to be sufficiently fast for the purpose of vehicle-level simulations and optimizations. This paper will discuss the integrated modeling of A/C system and associated control systems over a dynamic drive cycle, and the associated numerical performance issues discovered, as well as some approaches taken to increase said performance.

Keywords: Modelica, discrete, variable, integration

1 Introduction

As CAE simulations become more complex, the need for computational efficiency increases in order to provide timely solutions and analyses. One facet of this complexity is the integration of multiple software modeling tools and environments in order to utilize the most capable computational technologies for the different features of these complex system models. Physical plant models may be developed in Modelica and require variable step solvers to capture both fast and slow continuum dynamics while discrete time-based control systems may be developed in C-code or Simulink and require fixed time step solvers. Integrating these plant and control models into a single environment can result in computational inefficiencies due to conflicting solver time step requirements. This paper will discuss the integrated modeling of an automotive vapor compression air conditioning system and associated control systems over a dynamic drive cycle, and the associated numerical performance issues discovered, as well as some approaches taken to increase said performance.

2 Overview of the physical plant model - A/C refrigerant components, refrigerant system, and cabin

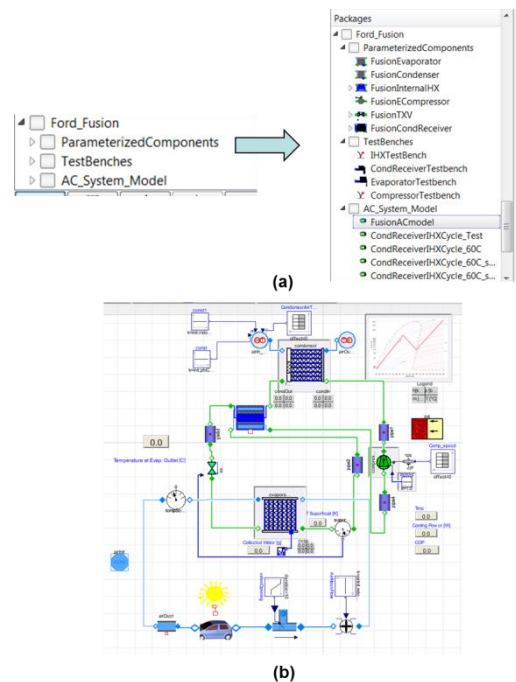


Figure 1 (a) Hierarchy structure of the A/C model (b) Modelon A/C model layout

For this study, only models of the complete refrigerant system and the vehicle interior (cabin) were required to simulate the physics of interest. The plant models are Modelica-based and developed in Dymola 2015FD01, utilizing component and refrigerant models from Modelon-supplied libraries. The structure and layout of the A/C model package are shown in Figure 1. The Dymola package browser in the upper-left of Figure 1(a) displays the package hierarchy and is shown with the A/C model selected. There are three sub-packages under the A/C model, namely, parameterized components, test benches, and A/C system model package. The parametrized components are specific, populated component models which are

used in the refrigerant circuit models. They include an evaporator, compressor, condenser, internal heat exchanger, and Thermostatic Expansion Valve (TXV), as well as piping and hoses. The test bench sub-package is a workspace for users to customize or calibrate individual component instances, and includes test benches for condenser, evaporator, and compressor models. The A/C simulation model sub-package consists of two main sub-models, the refrigerant circuit and the cabin model (including the air duct, blower model, etc.) as shown in Figure 1(b). Included in the refrigerant circuit model are the evaporator, compressor, condenser, internal heat exchanger, piping, and hoses. The cabin model consists of cabin interior, an air duct model, temperature blend door model, and the blower model.

The SC03 drive cycle is part of the EPA regulatory automotive 5-cycle fuel economy method. SC03 is a full vehicle chassis dynamometer test performed with the vehicle A/C unit operating with an ambient temperature of 95°F (35°C) and a solar loading on the vehicle of 850 W/m². The cycle represents a 3.6 mile (5.8 km) route with an average speed of 21.6 mph (34.8 kph), maximum speed 54.8 mph (88.2 kph), and duration of 596 seconds. The vehicle speed profile for the SC03 test is shown in Figure 2. The SC03 drive cycle was chosen for this study as it is the only cycle of the EPA 5-cycle method that requires the air conditioning system to be operating.

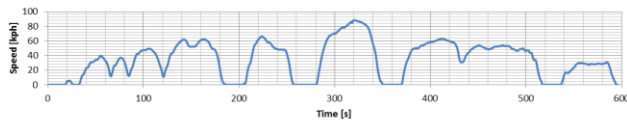


Figure 2 Vehicle speed race for SC03 cycle

3 Overview of the control systems models

The control systems required for the electrified automotive air conditioning system operation typically consist of the climate control head, compressor control, condenser fan control, active grille shutters, and electric water pump, as shown in Figure 3. The climate control head is the interface between the occupant and the climate control system. It controls the overall operation of the climate system, including cooling or heat request, blower airflow setting, airflow mode setting (location of discharge air), recirculation versus fresh air, cabin and evaporator temperature settings, and automatic or manual operational mode. The control head also controls the states of ancillary systems such as auxiliary heaters, glass fogging detection, heated/cooled seats, heated backlight/windshield, and heated steering wheel. Explicit modeling of a closed-loop control head is not included in this study, as the climate system settings are manually set at the start of the cycle and remain static throughout the drive cycle,

and the evaporator target temperature time trace comes from actual test data.

The compressor control modulates the compressor speed based on temperature request and refrigerant discharge pressure. The climate control head determines an appropriate target evaporator outlet temperature profile and the compressor control sets the compressor speed to achieve the target temperature using a PI control algorithm. If the compressor discharge pressure exceeds a specified limit, compressor speed is reduced and modulated using a PI controller to maintain a maximum allowed discharge pressure.

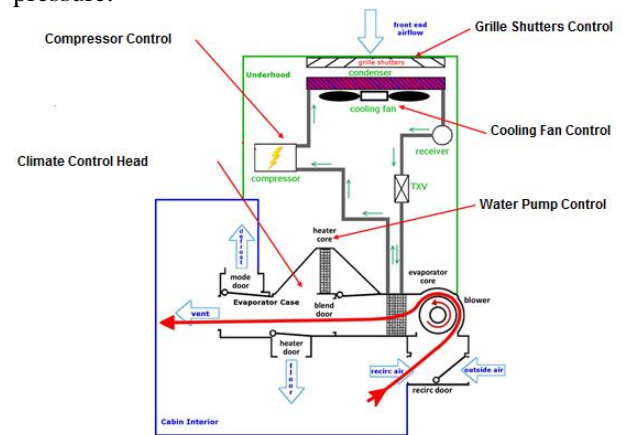


Figure 3 Automotive air conditioning control systems

The cooling fan control regulates the underhood front-end airflow fan speed in order to maintain the thermal management of systems requiring airflow through their associated heat exchangers. These systems typically include the A/C system, engine coolant, engine oil, and transmission oil systems. When the A/C system is operating the fan control calculates a desired fan speed/duty cycle based on compressor discharge pressure and ambient temperature. Fan speeds/duty cycles are also calculated for the other thermal systems and an arbitrator function determines the maximum required fan speed/duty cycle then commands the fan. Above certain vehicle speeds the fan speed/duty cycle is reduced to take advantage of ram air effect on front-end airflow.

Active grille shutters are used to balance aerodynamic drag and front end airflow/thermal management system requirements, closing down at higher vehicle speeds to reduce drag and opening more at lower speeds to enhance front end airflow. Similar to the cooling fan control, each thermal system has a desired shutter opening calculated and an arbitrator determines the maximum opening required and commands the shutters. For the A/C system, the desired grille shutter opening is calculated based on compressor discharge pressure and ambient temperature and then combined with a vehicle speed multiplier to account for aerodynamic effects.

Each of the control systems described above are implemented in the Matlab/Simulink (Version R2014a) environment as pre-compiled S-functions. The source of the S-functions is based on the C-language code implemented on the actual vehicle. While specific control strategies are hard-coded into the S-functions themselves, all control calibration parameters are user accessible at run time. Also, as each of the controls are discrete time-based systems operating with timing loops of 10 and 100 milliseconds, execution of the control models in Simulink require fixed step solvers.

4 Integrated Dymola/Simulink Model - Methods of model integration

In Dymola, the A/C plant model operates in an open loop fashion with the time varying inputs for a simulation prescribed in advance. While this is useful for a number of scenarios, such as plant model development and verification, substantially more value can be realized when the plant models are integrated with control systems in closed loop, thereby allowing more complex system performance and optimization studies to be conducted. Simulink is used here as the integration environment, and there are multiple methods for incorporating a Dymola model into Simulink.

The first integration approach is using the DymolaBlock S-function interface. This Dymola option allows models developed in Dymola to be compiled as S-functions incorporated directly into Simulink, enabling the powerful physical modeling capabilities of the Modelica language to be combined with the controls orientated approach of Simulink. Figure 4 depicts the A/C model integrated into Simulink as a DymolaBlock S-function.

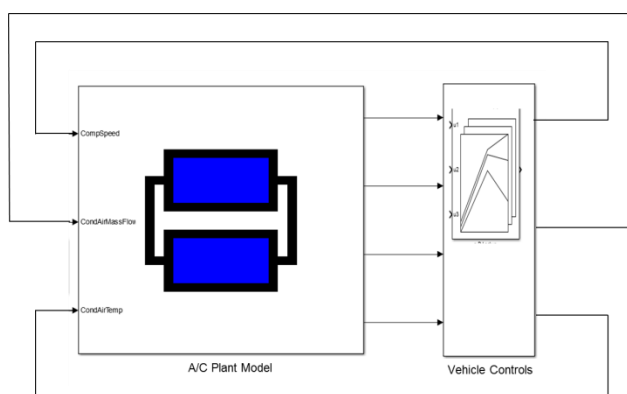


Figure 4 Integrated A/C model as a DymolaBlock S-function in Simulink

The second integration method utilizes the Functional Mock-up Interface (FMI) for model exchange or co-simulation. FMI defines a standardized modeling interface to be implemented by an executable module called a Functional Mock-up Unit (FMU). The FMI functions are used (called) by a simulation environment to create one or more instances of the

FMU and to simulate them, typically together with other model elements. An FMU may either have its own embedded numerical solvers (FMI for Co-Simulation) or utilize the simulation environment's own solvers (FMI for Model Exchange). [1].

In this application, we only consider the FMI for Co-Simulation option for two reasons. First, FMI for Model Exchange is very similar to utilizing an S-function function approach, like the DymolaBlock, as they both use the Simulink solver, and the DymolaBlock process is already incorporated into our modeling process. Second, due to the nature of the continuum behavior of the A/C system physics, and the need for a variable time step solver for the plant model, FMI for Co-Simulation allows the use of Dymola's solver for the physics in conjunction with Simulink's solver for the controls. Figure 5 shows an A/C model FMU in Simulink, with open loop inputs.

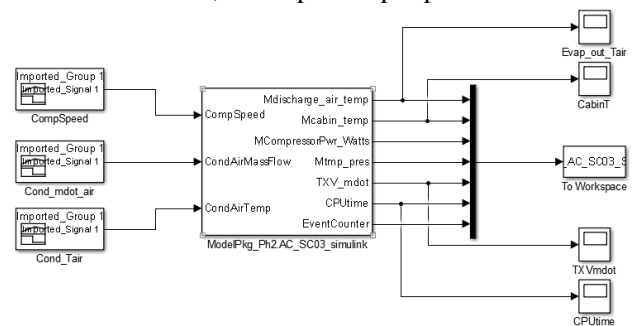


Figure 5 Integrated A/C model FMU in Simulink

Closed-loop control modeling requires the A/C system plant model to respond, minimally, at near real-time and produce physical and realistic outputs for the control systems to properly act upon [3]. In this study, there are three primary physical outputs from the A/C system plant model, specifically, the evaporator air out temperature, the vehicle cabin interior air temperature, and the compressor refrigerant discharge pressure. The evaporator air out temperature is the primary feedback signal used by the compressor controller to modulate the compressor speed, while the other two signals are used to a lesser degree. The input signals to the A/C system plant model from the control systems and Simulink-based physics models are the compressor speed, condenser airflow and air inlet temperature. The A/C system airflow through the evaporator and cabin interior is determined by the cabin blower fan and controlled by the Climate Control Head. In the SC03 drive cycle the blower is set to its maximum speed and is modeled as a constant input into the A/C model. Additional physics required by the integrated model are defined in Simulink, including the condenser airflow and air temperature. These physics models are coupled to both the control systems as well as the A/C model, as appropriate.

5 Model performance and solver strategies

The A/C system model, including the associated control systems, is one subsystem in a much larger and more complex total vehicle model. Because of this, it is desirable to optimize the computational performance of each subsystem in order to minimize the impact on the total vehicle model simulation time. The aforementioned A/C model integration approaches were selected to give the broadest range of solver and simulation settings to minimize the subsystem computational time.

The baseline performance for the A/C system model is defined as the physics-only Modelica model running the SC03 cycle open-loop in the Dymola environment, utilizing the DASSL variable time step with a solver tolerance of $1e-05$. Baseline simulation run time was 185 seconds, about one third of the real cycle time of 600 seconds. Next, the S-functions version of the Dymola model was run open-loop in the Simulink environment. Due to the numerical stiffness of the mathematics, only the Simulink ode15s variable time solver was capable of reliable solutions, and we used a solver tolerance of $1e-05$ to provide sufficient solution accuracy. The S-function simulation time averaged 183 seconds, essentially identical to the native-mode Dymola model. These results are shown as the first two bars in Figure 6. The actual SC03 cycle time comparator of 600 seconds is the right-most bar in the figure.

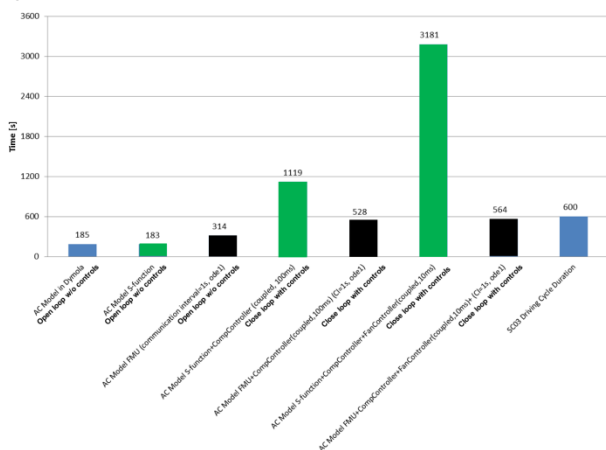


Figure 6. Comparison of the A/C system model computational performance

Next, we consider the closed-loop integration of the Dymola S-function with the 100ms fixed sampling rate compressor control and the 10ms sampling rate fan control. In Matlab R2014a, if any part of a Simulink model requires a variable step solver, the variable step solver must be the master solver for the simulation. However, Simulink does allow the user to define fixed sampling rates for parts of the model. For the integrated Dymola S-function model, we utilized the ode15s variable time step solver as the master

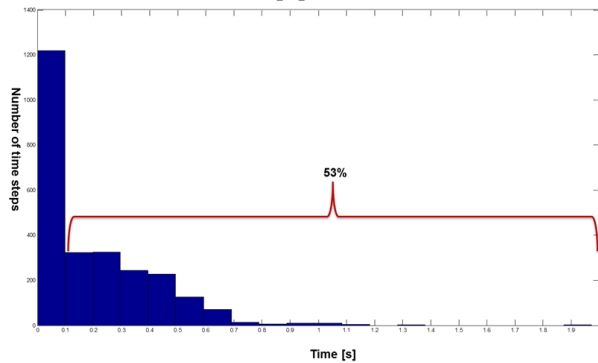
solver and specified a fixed sampling rate of 100ms for the compressor control, in order for the controls system to operate realistically. Likewise, when the cooling fan control was added to the integrated model, the fan control was set to run at a 10ms sampling rate, consistent with its actual operation.

The A/C model with the compressor control model completed the SC03 cycle in a time of 1119 seconds, as shown in the fourth bar of Figure 6, almost twice as slow as real time. When the fan control was added to the A/C model and compressor control the simulation time for the SC03 cycle increased dramatically to 3181 seconds, more than five times the actual cycle time.

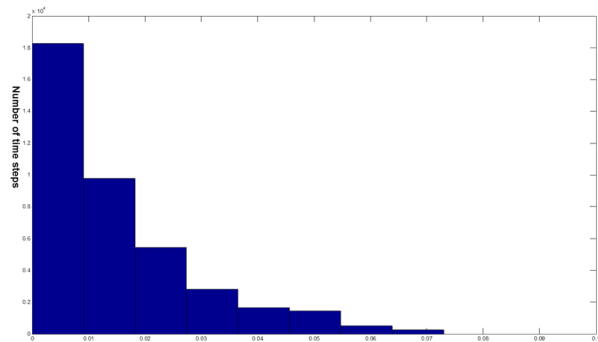
To rationalize the performance degradation we need to understand the interactions between the continuous, variable step solver used for the refrigerant system physics, and the discrete time-step solvers used by the control systems. With the integrated compressor control, the control module has to communicate 100ms. Additionally, when the fan control is also connected, it must communicate every 10ms. These sampling/exchange rates force the variable time step solver operating on the physical plant to synchronize the inputs and outputs of the Dymola S-function at the communication interval defined by the control system sampling rate. To illustrate this effect, the Simulink model was instrumented to record the timing rate of the variable time step solver. Figure 7a shows a histogram of the time step sizes used by the ode15s solver for the stand-alone, open-loop Dymola S-function model for the SC03 cycle. 53 percent of the time steps were larger than 100ms. Figure 7b shows the results of the A/C model with the integrated 100ms compressor control. The largest variable time step for this example was less than 80ms, and a large percentage 10ms or less. Finally, the combined A/C plant with compressor and cooling fan controls variable solver time steps are shown in Figure 7c, where the maximum step size is even less than the 10ms sampling rate of the fan controller. Combining variable and fixed step rates in Simulink models is referred to as a hybrid system in the Simulink documentation, as is detailed as follows:

“A hybrid system is a system that has both discrete and continuous states. Strictly speaking, any model that has both continuous and discrete sample times is treated as a hybrid model, presuming that the model has both continuous and discrete states. Solving such a model entails choosing a step size that satisfies both the precision constraint on the continuous state integration and the sample time hit constraint on the discrete states. The Simulink software meets this requirement by passing the next sample time hit, as determined by the discrete solver, as an additional

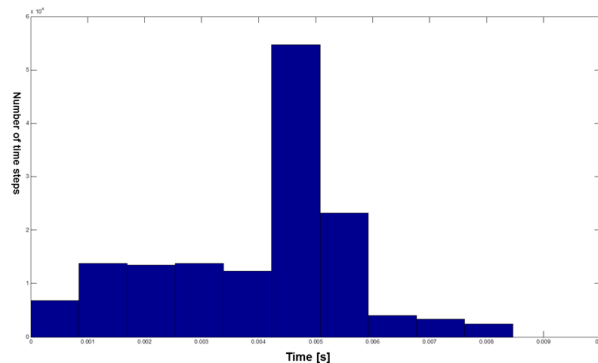
constraint on the continuous solver. The continuous solver must choose a step size that advances the simulation up to but not beyond the time of the next sample time hit. The continuous solver can take a time step short of the next sample time hit to meet its accuracy constraint but it cannot take a step beyond the next sample time hit even if its accuracy constraint allows it to.” [2]



(a). Standalone A/C Model S-function (183 seconds)



(b). A/C Model S-function+Compressor Control S-function (Coupled) (1119 seconds)



(c). A/C Model S-function+Compressor Control+Cooling Fan Control (Coupled) (3181 seconds)

Figure 7 Comparison of variable solver time step of the A/C model without/with compressor and fan control

As we attempted to optimize the computational performance of the combined A/C system and associated control systems, this approach of utilizing a Dymolablock-based S-function combined with the discrete time-based controls in Simulink was unable to deliver a sufficient level of performance, given the versions of the tools utilized, Dymola 2015FD01 and Matlab R2014a.

Next, the computational performance of the A/C model, as a co-simulation FMU, coupled with the control systems in Simulink, was evaluated. The primary benefit of FMI for co-simulation is that the FMU utilizes a native-mode solver from its parent tool, and the integrating environment uses its own appropriate solver, and the communication interval between the FMU and the integrating environment can be independently specified. After a series of tests, it was determined that a communication interval of one second, between the FMU and the Simulink-based controls, was sufficient to capture the required accuracy and dynamics of the of the A/C plant model to support vehicle-level cycle simulations. The following results discussed here are only for a communication interval of one second. The A/C model FMU utilized the Dymola DASSL solver, and the Simulink solver used was ODE1 with a fixed time step of 10ms, consistent with vehicle-level simulations.

Running the A/C model FMU coupled to open-loop inputs in Simulink resulted in a run time of 314 seconds, 70% slower than the stand-alone S-function, seen as the third bar in Figure 6. While this degradation was not expected, it was most likely due to the generation of time events in the FMU associated with the open-loop inputs, and not explored in depth here. Combining the A/C model FMU with the compressor controls resulted in a run time of 528 seconds, indicated by the fifth bar in Figure 6. Then, after adding the cooling fan controls, the simulation time only increased to 564 seconds, about a 7% increase in simulation time. Even with both controls systems integrated with the FMU, the model was able to execute faster than real time, as opposed to the large simulation times recorded utilizing the S-function-based plant model approach.

The benefit of the FMU co-simulation modeling approach is due to the variable time step solver for the physical plant model not having to synchronize lock-step with the discrete-time systems in the model, thus allowing the variable solver to run, on average, larger time steps permitted by the physics.

There have been recent performance enhancements and tool developments in the co-simulation software space, as well as enhancements to hybrid-solver simulations in more recent versions of Matlab, but those studies are not yet complete and could not be included in this paper.

6 Conclusions

A Modelica-based A/C system model has been integrated, closed-loop, with related S-function-based controls in the Simulink environment. The integration was performed with two different approaches, with a DymolaBlock-based S-function for the A/C model, and as a co-simulation FMU. The simulation performance

of the integrated model needs to be sufficiently fast for the purpose of vehicle-level simulations and optimizations. A study has been performed to evaluate and improve the simulation performance of the integrated model. The execution time of A/C model coupled with controls using Modelica FMI/FMU in closed-loop is faster than the real time of the SC03 cycle, and it is 5 times faster than the same A/C model using Dymola-Simulink interface S-function. Combining models that require both variable and fixed time step solvers can lead to serious numerical performance issues and care must be taken in evaluating potential solutions of these hybrid models. These performance issues are expected to grow as the complexity of physics and control models continue to increase, and the demand for faster model turnaround does, likewise.

References

- [1] T. Blochwitz:, M. Otter, M. Arnold, C. Bausch and H. Elmqvist, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models," in *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy, Dresden; Germany, 2011.*
- [2] MathWorks. [Online]. Available: <http://www.mathworks.com/help/simulink/ug/modeling-dynamic-systems.html?refresh=true>.