SIMULATION OF THE MEAN-VALUE INTERNAL COMBUSTION ENGINE IN MODELICA

M. Najafi[1], X. Shen[2], Z. Benjelloun-Dabaghi[2], P. Moulin[2]
[1]INRIA, Le Chesnay, France. [2]IFP, Rueil-Malmaison, France.

Corresponding author:
INRIA-Rocquencourt, Domaine de Voluceau, BP 105, 78153, Le Chesnay, France.
masoud.najafi@inria.fr
IFP, 1 Avenue Bois-Préau, 92852, Rueil-Malmaison, France.
{xudong.shen,zakia.benjelloun-dabaghi,philippe.moulin}@ifp.fr

Abstract. The Modeling and simulation play an important role in automotive engineering applications. Several well-known system based simulators (based on causal modeling) such as Matlab/Simulink and standard Scilab/Scicos allow Engineers to investigate and design control strategies for complex systems. There are also component based simulators (based on acausal modeling) such as AMESim give good insight for the design with the physical components. In this paper, we evaluate at IFP the potential of Modelica as an object-oriented programming language in modeling the vehicle engine. The model of the engine components is validated using real data logged from a spark-ignited two cylinders engine as well as models developed with other simulators. After modeling the engine, the Modelica model will be simulated in Dymola (V6.1) and Scicos simulators to perform a comparison between these two Modelica simulators.

KEY WORDS
Modelica language, modeling, simulation, mean-value combustion engine model (MVEM), Dymola, Scicos.

1 Introduction

Nowadays, it is out of question to design and implement an industrial system without performing a thorough analysis before building the real system. The analysis mainly consists of modeling and simulation of the system in normal and very often in severe critical conditions to predict the behavior or the performance of the system. Hence, a good modeling methodology as well as a suitable simulation environment is indispensable. There are two main approaches for modeling a physical system: system based modeling and component based modeling. In the first approach, the physical equations of the system are manually rewritten in explicit form, *i.e.,* in ODE (Ordinary Differential Equations) form. Then, the overall model is built by connecting basic causal modules, *i.e.,* blocks with explicit input and outputs. `Simulink` is an example of a tool based on this modeling approach. This modeling approach becomes inappropriate as the system size and its complexity grow. Furthermore, a physical system is very often naturally expressed in form of DAE (Differential-algebraic Equation) depicting a collection of relationship between variables and some of their derivative. These models cannot be expressed in explicit form, and even if possible, it can destroy sparsity and prevent the exploration of system structure. Physical equations are expressed with variables that usually have a physical significance. If the equations are changed into explicit form, less meaningful variables may produce. Furthermore, replacing a component or even changing a parameter value in a component can alter the relationship between variables and require different explicit models. Unlike the system based modeling, component based approach does not suffer from these shortcomings and offer several advantages. In the latter approach, the original model is built with components and physical equations are directly written in the model, so it is easier for the scientist or engineer to explore the model and see the effect of modeling changes and parameter variations. These advantages enable researchers to focus their attention on physical problems of interest. Also, there are numerical benefits for working directly with physical equations. Since equations can be expressed directly as DAE's, no more user manipulation is required and the equations can be solved by DAE solvers. There are several modeling languages based on this approach such as EL[1], EMSO[2], and Modelica[3] [5] languages. These languages provide the framework for developing model of physical components and then interconnect them to build a model with higher complexity.

Modelica is a relatively new object oriented language designed for modeling hybrid physical system with component based approach. In order to model a complex system in Modelica, first the model of the composing components are developed and then the components can be connected to construct the whole model. At IFP, engineers have been starting to evaluate the Modelica language in modeling and simulation of complex engine systems. In this paper the construction of a powertrain Modelica library for MVEM application containing all the basic components of internal combustion engine, *i.e.,* intake and exhaust manifolds, restriction pipes, valves, turbine, compressor, exchanger, cylinder, crank shaft, and various control volumes. Each component is singly tested and

[1] www.ecosimpro.com
[2] www.vrtech.com.br/rps/emso.html
[3] www.modelica.org

validated with a reference component model in the AMESim simulator which has been already validated. After building the engine components in Modelica, the turbo charged engine is constructed using the components in the library, and all the parameters are set to match with the real engine data. The whole model is then validated by comparing the simulation results with the engine test-bench measurements. The purpose of this work is not to achieve the best possible model of any specific engine, but to bring forward the concept of making a Modelica library which will enable the design of complex systems much faster.

An important characteristic of Modelica is the fact that the developed model can be simulated independent of the simulator. This allows the user to test and simulate the Modelica model with any Modelica simulator without being dependent on a particular simulator. IFP[4] has been involved in a few national and European projects with the objective of extending the usage of the Modelica language in modeling and simulation. One of these projects is the SIMPA[5] project launched by ANR[6]/RNTL to develop a free and open-source Modelica compiler for Scicos[7] and AMESim[8] simulators. The outcome of this project is a Modelica compiler[9]. At present, this compiler which is used by Scicos and AMESim supports most important features of the Modelica language. Several industrial applications have been developed with Scicos/Modelica at IFP [10, 8, 9], EDF[10] [11] and PSA[11]. Another important ITEA2 project is Eurosyslib that has the objective of developing Modelica libraries. In this work, in order to simulate the Modelica models, we use these two Modelica simulators *i.e.,* Scicos [2] and Dymola[12] and the advantages and flaws of each one will be numbered.

In this sequel, we first use the Modelica language to model a spark-ignited engine with a medium complexity. Then, after giving a brief description of the components of the engine, the overall engine model will be given. This model is then simulated with two Modelica simulators.

2 Modeling the engine library

In order to model the engine, the connector definitions together with the underlying models and functions for thermodynamic property calculations had to be developed from scratch. Once the fundamental base models are developed, a wide variety of systems could be built by reusing the base ones. The Modelica library of engine currently contains various engine components including control volumes, restriction pipes, pipe split, pipe converge, intake manifold, exhaust manifold, throttle, inter cooler, turbine, compressor, combined turbo charger, turbo charger with waste-gate, cylinder, crankshaft. Interested readers are refereed to [8, 15, 14, 6, 1, 3].

In order to model the gas and fuel thermodynamic behaviors we need their specific heat coefficients. For single species gas, NASA polynomials are used to reconstruct specific heats and enthalpy of a [4]. For fuels, Heywood polynomials are used to describe specific heats [6]. In engine applications (temperature less than 3000K), the NASA and Heywood method give good results.

The first step in building up the thermodynamic models for the powertrain library was to decide on the connector definition.For thermodynamic systems, the state of the working fluid is represented by pressure and temperature. In addition, the composition of the working fluid is represented by a vector containing the mass fractions of each gas species. In addition, each connection represents a path for energy and mass (of each gas species and the total value as well) to move through the system. Considering all these characteristics, the following codes give the definition of the thermodynamic connector of the system in Modelica.

```
connector ThermoPort
  parameter Integer n_gas=3  "No of gas species";
  Real p "[pa] pressure";
  Real T "[K] temperature";
  Real[n_gas] mfraction "mass fraction ";
  flow Real hdot "[J/s] heat flow";
  flow Real[n_gas] xmdot "[kg/s] mass flow";
end ThermoPort;
```

Taking into account one of the requirements of Modelica programming that the number of equations and the number of variables must balance. When two components are connected, the variables defined in the connector can not be calculated twice in each component. Thus all the thermodynamic components in this library can be classified into two main categories, capacity type and restriction type, in which capacity variables (pressure, temperature and mass fractions) and restriction variable (mass and heat flow) are calculated respectively.

[4] www.ifp.fr

[5] Simulation pour le Procédé et l'Automatique

[6] French National Research Agency

[7] Scicos is a free and open-source modeling and simulation software available at www.scicos.org

[8] www.lmsintl.com

[9] The compiler was developed at LMS Imagine.Lab

[10] Electricité de France

[11] Peugeot Citroën

[12] www.dynasim.com

2.1 Capacity base model

The capacity base model which represents a control volume is idealized to model mixing of in-flowing gases as constant giving uniform pressure and temperature throughout the volume which functions like a buffer of mass and energy. Any component containing a volume (intake and exhaust manifold, converge and split pipes and exchanger) is extended from this base model. The model is implemented with mass and energy dynamics. The states are total stored energy U and the vector of stored mass M. Pressure, temperature and mass fractions of each gas species are calculated by the following equations:

$$\begin{aligned} \dot{M} &= \dot{m}_1 + \dot{m}_2 \\ \dot{U} &= \dot{h}_1 + \dot{h}_2 \\ U &= M\,C_v\,T \\ \rho V &= M\,R\,T \end{aligned} \tag{1}$$

2.2 Restriction base model and pipe

The restriction base model represents a choked tube. The density difference at both ends of the restriction is ignored since the pressure difference is small enough to consider the working fluid is incompressible. A simple restriction pipe and a valve model are extended from this restriction base. In order to have a more elaborated model to make it closer to the real situation, a more detailed restriction pipe is modeled. In this model the Reynolds number is computed, friction factor in a thermal-multigas pipe as well as enthalpy and mass flow are considered.

2.3 Exchanger

The cooler extracts heat from the gas by setting it in thermal contact with a cooling medium. The exchanger model is extended from the volume base, and all the equations in the volume base are inherited, and thus represents a control volume of capacity type with heat transfer effect. The following equation represents the heat transfer rate based on the temperature difference between the component and the cooling medium.

$$\dot{U} = G\,A_{eq}\,(T_{coolant} - T_{gas})$$

Aeq is the equivalent area of the cooling medium. The heat transfer coefficient G is a measure of effectiveness of the thermal contact. The coefficient can be either parametrized or be determined by the flow through the cooler and the speed of the cooling medium via a two-dimensional map.

2.4 Turbine and compressor

The compressor is a restriction type in which the upstream and downstream temperatures, pressures and the turbo torque are considered as known. Mass flow and heat flow of the multigas mixture are then computed. In the model, due to high complexity of the system two-dimensional tables are used. In this 2-D tables the the compressor efficiency and mass flow-rate are computed as a function of turbine speed and upstream/downstream pressure ratio.

The turbine model is also considered as a restriction type in which the mass and heat flow are computed. Similar to the compressor, the efficiency and the mass flow rate is computed with 2-D tables. Neither the turbine nor the compressor includes any inertia, so they should be connected to some rotational mechanical component like a stiff inertial axle.

2.5 Combustion chamber

The cylinders are modeled as a MVEM (mean value engine model) which is a volume-less, steady flow pump heating the gas and extracting energy. This is far from real situation in the combustion chamber where the mass flow is highly turbulent as an effect of the intake and exhaust ports opening and closing. The fuel injection of this model does not take into account of the cycle effect and valve timing evolution as in real engine where the fuel is only injected once every cycle. The delay between input and torque output is not modeled either. Despite the above assumptions, the simulation result is satisfying for the MVEM application.

A more elaborated combustion model incorporating 2-zone wiebe combustion model is being developed at IFP. This model which includes valve timings, gas turbulence, mass and heat transfer and simple chemistry are used to achieve more precise simulation result as well as to study the emission control of the engine.

2.6 Other components

Additional components are constructed by extending the base models or connecting certain models presented above, which is the easiest way to create new components, including intake and exhaust manifold, branching pipes, turbo charger and turbo charger with waste gate.

3 Mean-value engine model

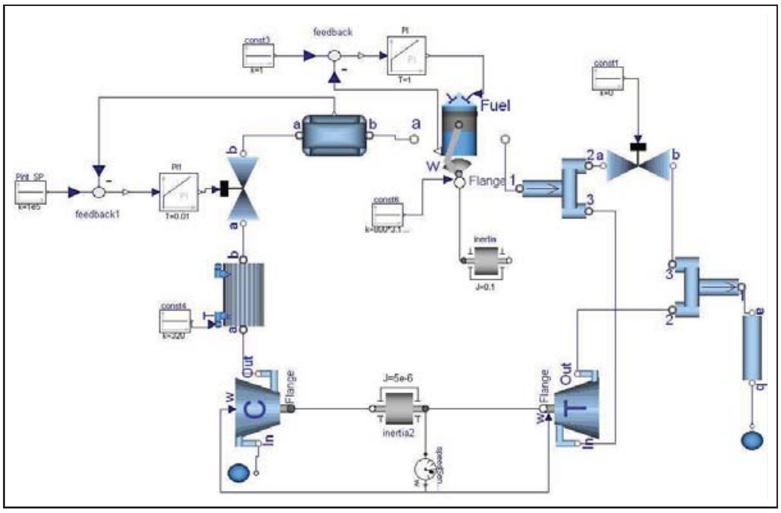


Figure 1: Engine model in Dymola

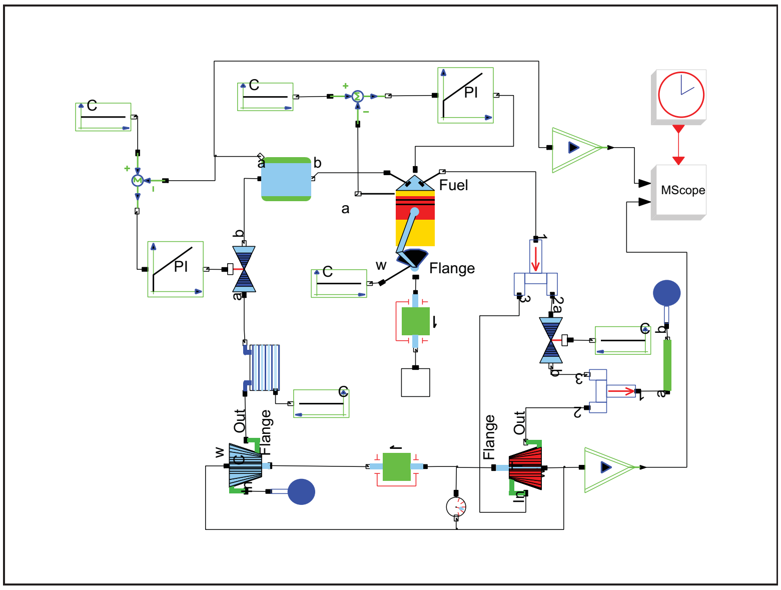


Figure 2: Engine model in Scicos

Each of the components in the library has been tested and compared with the equivalent component model in the AMESim simulator. Now it is the time to build the whole engine model. The test-bench engine is a direct injection gasoline engine and its model is composed of intake manifold, cylinder, exhaust manifold and pipes, waste-gate, turbine, compressor, exchanger and intake throttle. The complete engine model is thus built by connecting the components. Since in this work we intend to simulate the engine model with several Modelica simulators, the whole model is implemented in Scicos and in Dymola. The sketch of the model made in Dymola is given in Figure 1 and the same model in Scicos is given in Figure 2. All the components' parameters are set according to the real technical data of the engine, and the maps used in the model are calculated from the test-bench data. Fuel flow rate, intake throttle, turbo waste-gate coolant temperature are control inputs of the model.

As shown in Figures 1 and 2, a PI controller are used to tune the lambda value of the combustion chamber by adjusting the fuel injection rate around unity in all conditions. In addition, two other PI controllers is used to adjust the intake throttle and the waste-gate openings in order to obtain a desired intake pressure. In this model only the PI controller for the throttle is shown and the waste-gate is left fully closed. The real pressure in the intake manifold is feedback to the controllers to tune the throttle and waste-gate openings.

In order to validate the model the lambda value governed by PI controller set to unity and the engine speed set to 3000 rpm which is in the normal engine operating range. The test strategy is to open the intake throttle gradually while the waste-gate is fully open, and then close the waste-gate step by step when the intake throttle is fully open, as shown in Figure 3. Several model variables such as intake and exhaust temperatures and pressures, engine torque and turbine speed will be evaluated and compared with the experimental data from the test-bench.

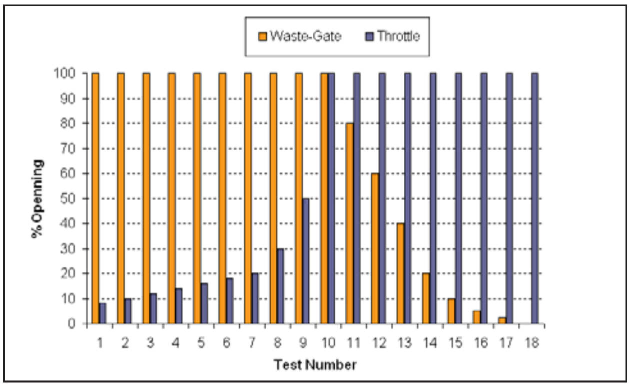


Figure 3: Throttle and waste-gate opening values used in the test

The output engine torque versus the intake manifold pressure at several operating points is plotted in Figure 4. This figure represents the engine torque resulting from the engine model as well as the real engine torque measured on the test-bench at several intake manifold pressures.

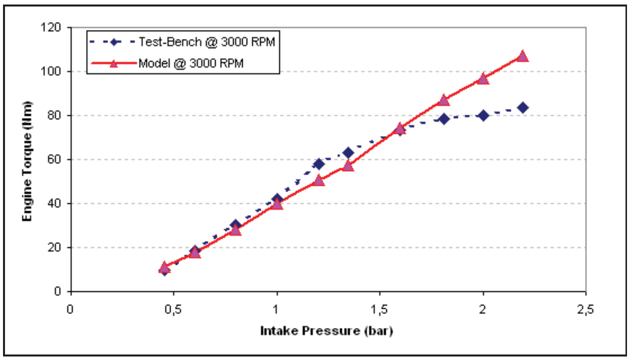


Figure 4: Comparing model and test-bench data for engine torque vs manifold pressure

There is a difference between the simulation result and the test data at higher intake pressures. The difference is in fact due to several simplifications made in the model. The most important simplification is to represent the two cylinders of the real engine by a MVEM cylinder, so only mean value is compared with the simulation results. Furthermore some frictions are ignored. For our purposes the model is however good enough.

In Figure 5, the transient response of some interesting variables of the model in Dymola are shown. The top subplot shows the output engine torque, the middle subplot shows the intake and outlet temperatures, and the bottom subplot shows lambda value of the test at 5000 rpm and intake pressure set point 1.2 bar.

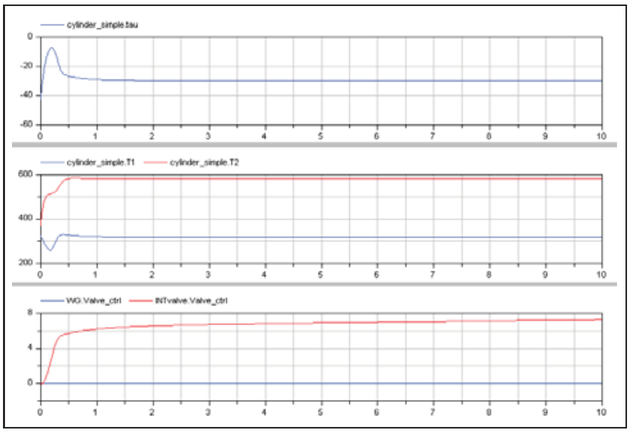
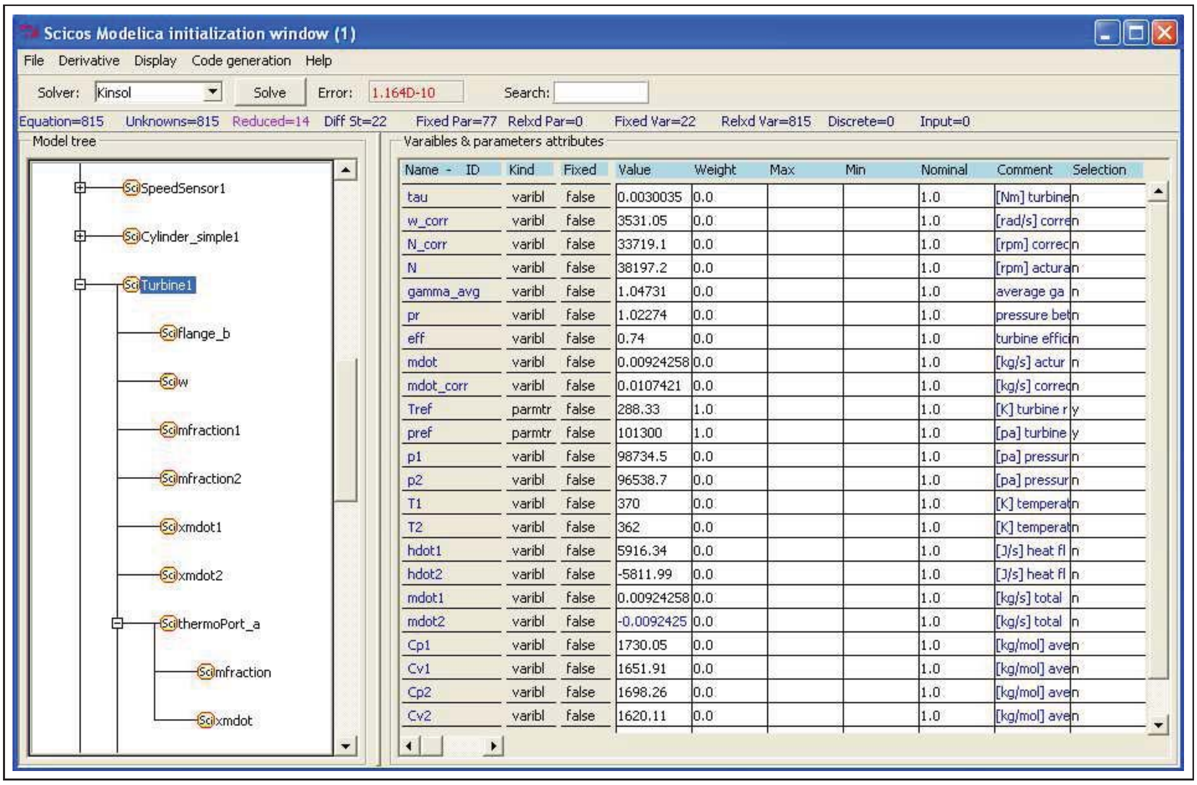


Figure 5: Simulation curves in Dymola

4 Comparison

With the clear equation based design on the component level, Modelica gives certain advantages over the Matlab/Simulink and standard AMESim simulators in terms of software cost, model complexity, and CPU running time. Several works have been done in IFP to perform a comparison between the causal modeling environment of



Solver: Kinsol Error: 1.164D-10 Search:

Equation=815 Unknowns=815 Reduced=14 Diff St=22 Fixed Par=77 Relxd Par=0 Fixed Var=22 Relxd Var=815 Discrete=0 Input=0

Varaibles & parameters attributes

Name - ID	Kind	Fixed	Value	Weight	Max	Min	Nominal	Comment	Selection
tau	varibl	false	0.0030035	0.0			1.0	[Nm] turbine	n
w_corr	varibl	false	3531.05	0.0			1.0	[rad/s] corre	n
N_corr	varibl	false	33719.1	0.0			1.0	[rpm] correc	n
N	varibl	false	38197.2	0.0			1.0	[rpm] actura	n
gamma_avg	varibl	false	1.04731	0.0			1.0	average ga	n
pr	varibl	false	1.02274	0.0			1.0	pressure bet	n
eff	varibl	false	0.74	0.0			1.0	turbine effic	n
mdot	varibl	false	0.00924258	0.0			1.0	[kg/s] actur	n
mdot_corr	varibl	false	0.0107421	0.0			1.0	[kg/s] corre	n
Tref	parmtr	false	288.33	1.0			1.0	[K] turbine r	y
pref	parmtr	false	101300	1.0			1.0	[pa] turbine	y
p1	varibl	false	98734.5	0.0			1.0	[pa] pressur	n
p2	varibl	false	96538.7	0.0			1.0	[pa] pressur	n
T1	varibl	false	370	0.0			1.0	[K] temperat	n
T2	varibl	false	362	0.0			1.0	[K] temperat	n
hdot1	varibl	false	5916.34	0.0			1.0	[J/s] heat fl	n
hdot2	varibl	false	-5811.99	0.0			1.0	[J/s] heat fl	n
mdot1	varibl	false	0.00924258	0.0			1.0	[kg/s] total	n
mdot2	varibl	false	-0.0092425	0.0			1.0	[kg/s] total	n
Cp1	varibl	false	1730.05	0.0			1.0	[kg/mol] ave	n
Cv1	varibl	false	1651.91	0.0			1.0	[kg/mol] ave	n
Cp2	varibl	false	1698.26	0.0			1.0	[kg/mol] ave	n
Cv2	varibl	false	1620.11	0.0			1.0	[kg/mol] ave	n

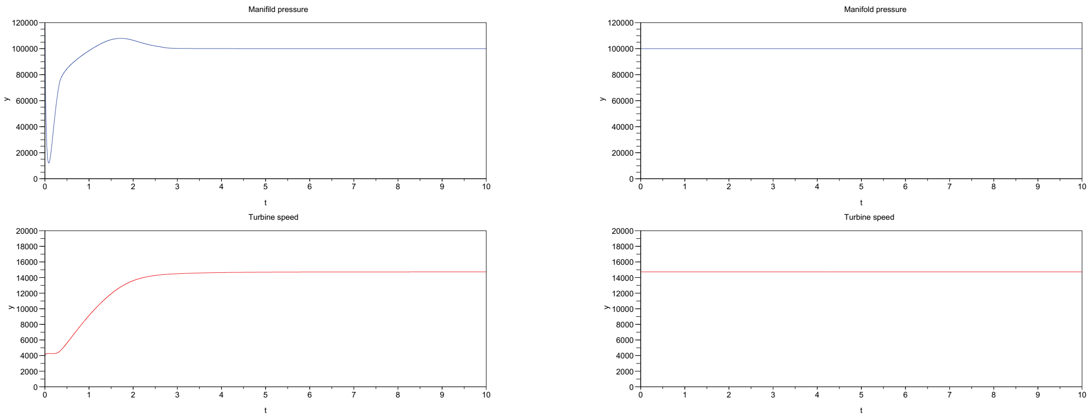
Figure 6: Screen shot of the initialization GUI in Scicos

Simulink[13] and Modelica, see for example [8, 9]. In this work we have concentrated on comparing two Modelica simulation environments, *i.e.,* Dymola and Scicos.

The developed engine model in Modelica contains more than 800 variables. The Dymola and Scicos compilers perform symbolic manipulation to reduce the size of the model's equation. The size of the engine model after the symbolic simplifications is reduce to about 50 equations. Although the simulation results in two simulators for the engine model are identical, each one has some advantages and some flaws in handling a Modelica model that are listed below.

- In many applications the user needs a scientific computation environment like Matlab. Dymola provides an interface with Matlab and Simulink. Scicos, being inherently developed in a Matlab-like environment, *i.e.,* the Scilab environment, does not need such an interface and a direct communication with Scilab is possible. This allows Scicos to develop mixed models, *i.e.,* develop models with causal blocks and Modelica blocks in the same environment [13] and can take the advantages of being in Scilab environment.
- Scicos does not support some features of the Modelica language. Most important unsupported features are index reduction and the algorithm section. Fortunately most thermo hydraulic systems (including the engine model) are index-one differential equations. Thus they can be simulated in both simulators. In the engine model, we used algorithm to develop maps and interpolation methods in Dymola. Since in Scicos this is not still supported we had to use some external C functions and use a DLL to import map data.
- Initialization of Modelica Models in Dymola or in general in Modelica is not easy and initialization method is not very clear. In Scicos a specialized graphical user interface is used to help the user to select the variables to be initialized and then the simulation can, for example, be started directly from the steady state. Finding the initial conditions for a simulation is not in general an easy task for complex models, that is why Scicos has incorporated several pure algebraic solvers for finding the consistent initial condition. In Figure 6, the graphical user interface of Scicos for the engine model has been shown. In Figure 7(a), the simulation result of a model starting from given initial values are shown and in Figure 7(b), the simulation result of a model starting directly from steady state is shown. In these Figures the top subplot is the intake manifold pressure and the other is the turbine speed [12, 7].
- Scicos can generate the analytical Jacobian which facilitates the task of the numerical solver and the numerical solver is more likely to integrate the model. This feature permitted Scicos to simulate the engine models faster than Dymola. This option should be turned off when the size of model is large.
- Dymola provides the simulation results for all of the variables in the model. Scicos, on the other hand, generates the outputs only for the variables selected by the user, *i.e., variables interfaced to the Scicos standard*

[13] Simulink has recently introduced a component based modeling in its environment



(a) start with transient response

(b) start directly from steady-state

Figure 7: Simulation result for different initial values in Scicos

environment. That means Scicos does not generate output results for all variables of the model. Although inconvenient, this features of Scicos permits to obtain a more compact C code, thus a faster simulation.

5 Future works

As aforesaid, a good point in Modelica is the fact that it is independent of the simulation software. Thus development of a model in Modelica does not induce a dependence on a particular simulation software. There are several Modelica compilers/simulators on the market. Some are commercial such as Dymola and LMS Imagine.Lab (AMESim) simulators, and some are free and open-source such as Scicos and OpenModelica[14]. In the next stage of this work, we intend to test present model as well as future models with other simulators, in particular, AMESIM and OpenModelica. The objective is in fact to understand the strong and weak points of each simulator.

6 Conclusion

With a full equation based design on the component level, the powertrain library is successfully constructed in the Modelica language and the engine model is made by applying the components in the library. Since Modelica models are independent of the simulator, in order to make a comparison we simulated our model with Dymola and Scicos and we made a brief comparison between these Modelica simulators. The developed engine model, being relatively simple, was not a good benchmark for testing the simulation time the compiler performances. In the future works, we will develop more elaborated models to do a thorough comparison between the compiler and the core simulators of these two Modelica simulators.

7 References

[1] C.F. Aquino. Transient a/f control characteristics of the 5 liter central fuel injection engine. 1981.

[2] S. L. Campbell, J-Ph. Chancelier, and R. Nikoukhah. *Modeling and simulation Scilab/Scicos*. Springer Verlag, 2005.

[3] Paul Degobert. Automobiles and pollution. *Society of Automotive Engineers, Inc. Translation of Automobile et pollution*, 1995.

[4] J. G. Ehlers, S. Gordon, S. Heimel, and B.J. Mc Bride. Thermodynamic properties to 6000k for 210 substances involving the first 18 elements. Technical Report NASA-SP-3001, NASA Center: Glenn Research Center, 1963.

[5] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, 2004.

[6] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill series in mechanical engineering, international edition, 1988.

[7] M. Najafi. Selection of variables in initialization of modelica models. *EOOLT 2nd International Workshop on Equation-Based Object-Oriented Languages, July 2008, Paphos, Cyprus*, 2008.

[8] M. Najafi and Z. Benjelloun-Dabaghi. Modeling complex systems with modelica in scicos: Application to mean value spark engine. *ESM conference, October 2007, St. Julians, Malta.*, 2007.

[14] www.ida.liu.se/projects/OpenModelica/

[9] M. Najafi and Z. Benjelloun-Dabaghi. Modeling and simulation of a drilling station in modelica. *Med'08, 16th Mediterranean Conference on Control and Automation, June 2008, Corsica, France.*, 2008.

[10] M. Najafi and Z. Benjelloun-Dabaghi. A new modelica modeling and scicos simulation for 0d/1d nonlinear complex system. *Oil & Gas science and technology journal.*, 2008.

[11] M. Najafi and D. Bouskela. Thermal-hydraulics applications developed in scicos. *ESM conference, October 2007, St. Julians, Malta.*, 2007.

[12] M. Najafi and R. Nikoukhah. Initialization of modelica models in scicos. *Conference Modelica 2008, Bielefeld, Germany.*, 2008.

[13] M. Najafi, R. Nikoukhah, S. Steer, and S. Furic. New features and new challenges in modeling and simulation in scicos. *IEEE conference on control application, Toronto, Canada*, 2005.

[14] Y. Nakayama and F. Boucher. Introduction to fluid mechanics. *Arnold, London, Great Britain.*, 1999.

[15] Daniel Silverlind. Mean value engine modeling with modelica. *M, Department of Electrical Engineering, Linkoping University.*, Master's thesis, 2001.