Comparison of Classical IF-Clause Modelling and State Chart Modelling for ARGESIM Benchmark C5 'Two State Model' in MOSILAB/Modelica

Gemma Ferdinand Kaunang, Günther Zauner, Vienna Univ. of Technology, Austria,

Simulator: Mosilab (**Mo**deling and **Si**mulation **Lab**oratory) is a simulation tool for complex technical systems. For the modeling process, MOSILAB uses the object- and equation-oriented model description language Modelica®, with a backwards-compatible extension to incorporate elements for describing model structure dynamics. An integrated development environment offers user support in every work step – from model building over simulation to post-processing. Besides the traditional component diagram, class and statechart diagrams are available to users for the model-based development process.

MOSILAB can be used for applications in automotive and energetic systems, mechatronics, as well as microsystems technology and others.

Model: This example tests the ability of the simulator to handle discontinuities of the type defined in ARGESIM benchmark 5. The problem is defined as follows

$$\frac{dy1}{dt} = c1 * (y2 + c2 - y1)$$

$$\frac{dy2}{dt} = c3 * (c4 - y2)$$

The model operates in two states:

1. the parameter $c2 = 0.4$ and $c4 = 5.5$ and it switches to state 2 if $y1 \geq 5.8$.
2. the parameters $c2$ and $c4$ change to $c2 = -0.3$ and $c4 = 2.73$. The model switches back to state 1 if $y1 \leq 2.5$.

The aim is to implement these equations and all the conditions above into Mosilab structure. Therefore two different ways are chosen to model these equations. The first solution is by using textual Modelica language. Designing the model is relatively easy by using the exact equations in the `equation` section and declaring all variables in the beginning section. The switching state is modelled in the `algorithm` section as follows:

```
algorithm
  when (y1>=5.8) then
     c2:=-0.3; c4:=2.73;
  end when;
  when (y1<=2.5) then
     c2:=0.4; c4:=5.5;
  end when;
```

Listing 1: Parameter event in a Modelica based `algorithm` section

The second solution is implemented by using the state chart approach, dividing the system into states, depending on the value of the variable y1. The method of modelling is quite similar to the first solution; namely, changing the switching state algorithm above to the statechart code as follows:

```
equation
  s2 = if y1 >= 5.8 then true else false;
  s1 = if y1 <=2.5 then true else false;
statechart
state C5MosilabStateSC extends State;
  State State1; State State2;
  State Initial (isInitial=true);
  transition Initial->State1
  end transition;
  transition State1->State2 event s2
    action
      c2:= -0.3; c4:= 2.73;
  end transition;
  transition State2->State1 event s1
    action
      c2:= 0.4; c4:= 5.5;
  end transition;
end C5MosilabStateSC;
```

Listing 2: State chart code

The program codes of all solutions are realized in the modelling section of Mosilab. Compiling the code, setting up the simulation parameters and simulation process are done in the simulation section. The results of the simulation run are shown in the post-processing section.

A-Task: The aim of task A is to simulate the system, by setting 1μsec as minimal stepsize, 80msec as maximal stepsize, relative tolerance of 1e-6 and `Dassl` as the integration method. Selecting the intial value 4.2 for variable y1 and 0.3 for variable y2, the result for value y1 is shown in Figure 1 for both model implementations. It took 0.1sec to simulate the task A for the classical Modelica solution and 0.3sec for the state chart solution.

Rel_Tol	1e-6	1e-10	1e-14
t_0	1.1088	1.1083	1.1090
t_1	2.1397	2.1394	2.1299
t_2	3.0588	3.0584	3.0592
t_3	4.0760	4.0757	4.0764
y1(5.0)	5.7988	5.7985	5.7997

Table 1: Time discontinuity and final value y1(5.0) with varying relative tolerance

solver	**Dassl**
t_0	1.1088
t_1	1.1220
...	...
t_{60}	4.9235
t_{61}	4.9370
y1(5.0)	5.7827

Table 2: Time series of state event and final value y1(5.0) with varying relative tolerance

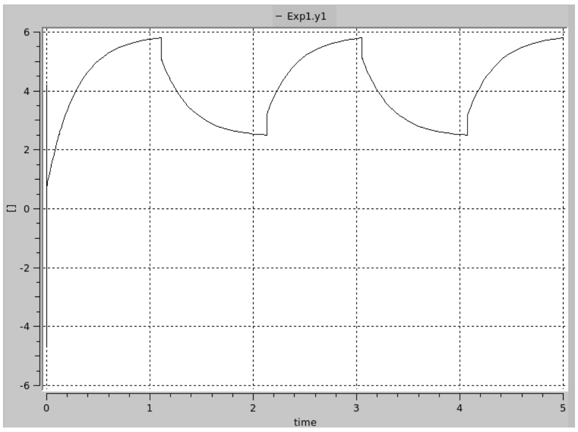

Figure 1. Time Curve y1

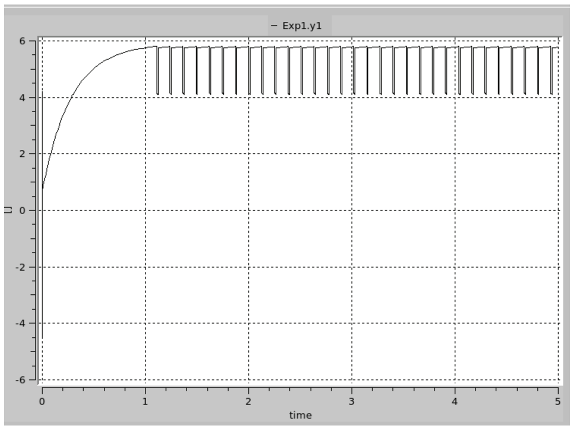

Figure 2: Time Curve of y1 for the settings of task D

B-Task: Based on Figure 1 the time discontinuity and the final value found can be seen at column two in Table 1.

C-Task: The parameter of relative tolerance is varied as follows: 1e-6, 1e-10 and 1e-14. Again, 1μsec is chosen as minimal stepsize. 80msec as maximal stepsize and `Dassl` as integration method. Table 1 shows the results of time discontinuity and final value y1(5.0) with varying relative tolerance.

D-Task: Changing the state 2 parameters c2 to -1.25, c4 to 4.33 and the switching condition to y1 $\leq$ 4.1 will result in a high frequent event of discontinuity for y1. Figure 2 shows y1 for both ways of implementation. The number of discontinuites found is 62.

The time discontinuities and the final values are shown in Table 2.

For all calculations Mosilab version 3.1 on Notebook Dell Latitude D630 Intel Centrino Duo was used.

Resumé: Mosilab is a simulation tool that offers several ways to model the system, because it is based on Modelica, an object- and equation-oriented model description standard. Furthermore the extension with state charts provides extra features for modelling structural dynamik systems.

The Dassl integration method implemented in Mosilab is an adequate state of the art solver for the given problem class.

Modelling the tasks in the textual mode gives faster simulation results than with state chart modelling. However the results remain the same.

Corresponding author: Günther Zauner
dieDrahtwarenhandlung Simulation Services
Neustiftgasse 57-59, 1070 Vienna, Austria
guenther.zauner@drahtwarenhandlung.at

Received: October 6, 2007
Revised: October 20, 2007
Accepted: November 3, 2007