

C7 Constrained Pendulum – DYMOLA

Law-oriented Modelling

Simulator: DYMOLA – Dynamic Modeling Laboratory – provides a powerful object-oriented modeling and simulation environment with MODELICA support. The usual need for manual conversion of equations to a block diagram is removed by use of automatic formula manipulation, so that physical laws can be used for modelling. Before experiments are performed the code is compiled which speeds up simulation runs.

MODELICA, a quasi-standardized description for law-oriented modelling supports hierarchical structuring, reuse and evolution of large and complex models independent from the application domain and acausal modelling based on differential and algebraic equations. DYMOLA is able to understand MODELICA model notations.

Model: In this approach DYMOLA's equation layer, which is part of the model editor, has been used. To build the linear model even the graphical model editor could have been used, but for the non-linear model there exists no basic class so the equation layer is the easiest way to describe it. Because DYMOLA chooses state-variables on his own and is able to handle state-events automatically it is rather easy to describe the model. The code to declare parameters and variables has the following form:

```
parameter Real m=1.02,g=...
Real phi(start=pi/6),...
Real v,l,...
```

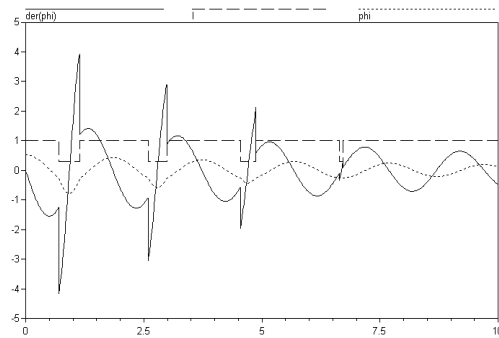
The code to describe the non-linear model:

```
l = if phi > phi_pin then l1 else l2;
v = l*der(phi);
m*der(v) + m*g*sin(phi) + d*l*der(phi) = 0;
```

Task a: Simulation of the System: After setting the initial values and specifying the simulation interval the simulation is started out of DYMOLA's main window. Because of DYMOLA's automatic state-event handling it is just necessary to specify initial conditions for the angle and the angular velocity (from where the initial pendulum length is determined by means of the the derived model equations).

Parameters are changed in the main window too, but for more complex parameter variations the MODELICA language with its experiment environment could be used.

The following plot shows the solution for $\varphi_0 = \pi/6$ and $d = 0.2$.

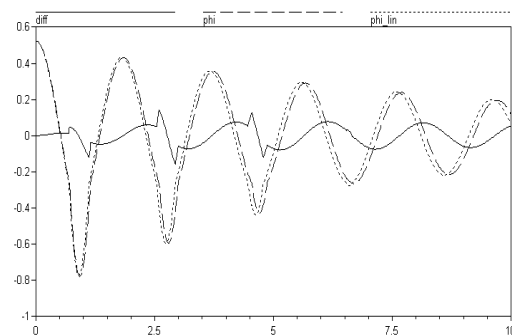


Task b: Linearization: The linearized model is implemented in the same way as the non-linear model, just substituting $\sin \varphi$ with φ . To compare the different results both models have to be simulated in parallel. DYMO-SIM, DYMOLA's simulation environment, offers no tools to compare results from different simulation runs. The code to be added is:

```
l_lin = if phi_lin > phi_pin then l1 else l2;
v_lin = l_lin*der(phi_lin);
m*der(v_lin) + m*g*phi_lin +
d*l_lin*der(phi_lin) = 0;
```

```
diff=phi-phi_lin ;
```

The following plot shows the angle of the linear pendulum and the non-linear pendulum and the difference as a function of the simulation time.



Task c : Boundary Value Problem: This problem can be transformed to an initial value problem by integrating the equations backwards in time. The simulation run has to be stopped when the given start-value is reached, which may be easily implemented in DYMOLA because of its automatic state event handling.

The solution is the angular velocity at $t = 0$, which is approximately 2.1847.

Nikolaus Viertl
ARGESIM, Vienna Univ. of Technology
viertl@mail.com

