# A CSSL-like Solution of ARGESIM Comparison “C7 - Constrained Pendulum” with ACSL

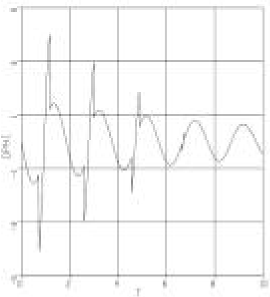
**M.Jach,K.Vogel, R. Hohmann;**
**Otto von Guericke-University Magdeburg;**
**Matthias.Jach@Student.Uni-Magdeburg.de**

**Simulator:** ACSL is a general purpose simulation language modelling systems by time-dependent differential equations and running on a wide range of computers (this solution run on a home PC, WIN 98.

**Model:** In the following model (for task a, b) we use the Schedules **`hit`** and **`hitlin`** to serve the discrete sections **`hit`** and **`hitlin`**, if the pendulum reaches or leaves the pin. The section with the suffix **`lin`** handles the linearized model (linear and nonlinear models in Task b) run simultaneously):

```
PROGRAM Constrained Pendulum Task A and B
LOGICAL cl , cllin
 CONSTANT pi = 3.141592654; pi6 = pi/6
 pi12=pi/12;mpi2=-pi/2;mpi6 =-pi6;
 mpi12=-pi12; TEND=9.99! Pi-frac, time 10 sec
 INITIAL ! Pendulum Parameters
  CONSTANT l=1,m=1.02,d=0.2,g=9.81,lp=0.7
  CONSTANT phi0=0.3,dphi0=0,php=0.2 !Default
  phi0lin=phi0;dphi0lin=dphi0
  ! Determine initial position of pendulum --
  ls=l-lp;cl=.false.;cllin=.false.
  IF(SIGN(1.0,php).NE.SIGN(1.0,phi0)) THEN
   la=l; lalin=l; ELSE
   IF(ABS(phi0).GT.ABS(php))THEN
    la=l; lalin=l; ELSE
    la=ls; lalin=ls ; cl=.true.;cllin=.true.
   ENDIF
  ENDIF
 END ! of INITIAL
 DYNAMIC
  DERIVATIVE ! Dynamics of pendulum
   ddphi = -(g/la)*SIN(phi) - (d/m)*dphi
   dphi = INTEG(ddphi, dphi0)
   phi = INTEG(dphi, phi0)
   SCHEDULE hit .XZ. (phi-php) ! P hits pin
   ddphilin=-(g/lalin)*philin - (d/m)*dphilin
   dphilin = INTEG(ddphilin, dphi0lin)
   philin = INTEG(dphilin, phi0lin)
   SCHEDULE hitlin.XZ.(philin-php)! hit(lin)
   deltaphi = philin-phi !error of lin eq.
  END ! of DERIVATIVE
  DISCRETE hit ! Change of Velocty and length
   cl = .NOT. cl ! switching in the following
   la = RSW(cl,ls,l);
   dphi= RSW( cl, dphi*l/ls, dphi*ls/l)
  END ! of DISCRETE hit
  DISCRETE hitlin ! as HIT linear
   cllin = .NOT. cllin
   lalin=RSW(cllin,ls,l)
   dphilin=RSW(cllin,dphilin*l/ls,
              dphilin*ls/l)
  END ! of DISCRETE hit
 CINTERVAL CINT = 0.01
 TERMT ( t .GT. tend,'Stop on time limit')
END ! of DYNAMIC
END !of Program
```

**Task a, Task b:** The following figures show the results of task a-1 (left) and task a-2 (right) – simultaneously results for task b arge obtained.



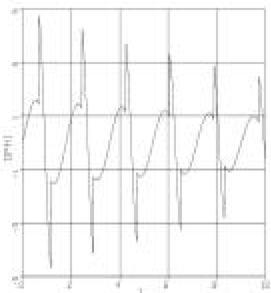


fig. 1: task a-1 (angle velocity dphi) fig. 2: task a-2

**Task c:** Our approach can be described as “brute-force” using only a minimum of information implemented in the model. Starting with the angle velocity $d\phi_0$ we are looking for, the pendulum moves to the right ($d\phi_0 > 0$) and swings then back or it goes to the left in the other case instantly.

If the absolute value of this initial angle velocity is sufficiently large, the pendulum reaches the angle of $-\pi/2$. On the other hand the pendulum does not reach this angle, if the initial angle velocity is too small.

We use three event-driven commands to catch the desired value by bisection: First we estimate the two limits of our interval a<b with sign(a)=sign(b) or with 0 as one limit. We start with the value absolutely higher.

- Event 1: The pendulum -coming from the right - reaches the pin, its length and angle velocity change. We suppose that this event always takes place.
- Event 2: The pendulum traverses the angle $-\pi/2$ which means that $d\phi_0$ was estimated (absolutely) too high. Therefore we stop the simulation, cut the initial angle velocity and try it again.
- Event 3: The pendulum leaves the pin coming from the left. That means that it was not stopped reaching the angle $-\pi/2$, so $d\phi_0$ was estimated (absolutely) too small. We stop the simulation, increase the value and try it again.

If we choose the start value sufficiently large and the other limit of the interval in the way that Event 2 does not occur, we have a classic bisection.
We get $d\phi_0$= -2.1847 after 53 iterations with a= -5 and b= 0 and $d\phi_0$= 2.29107 after 56 iterations with a= 0 and b= 5.

**C7 Classification: Model Segment Approach**

**Simulator: ACSL 11.8.4**