A Java-supported Approach to ARGESIM Benchmark C9 'Fuzzy Control of a Two-Tank System' with Anylogic

Thomas Mair, Robert Leidenfrost, Andreas Platschek, Shabnam Tauböck
Vienna University of Technology, *Shabnam.Tauboeck@.tuwien.ac.at*

Simulator: Anylogic is an object oriented simulation software, supporting Agent Based, Discrete Events and Systems Dynamics based simulations. This comparison primarily uses the Systems Dynamics - based approach, because it enables a very smooth way to model the waterflow in the system. For fuzzy control design, Java was used in the Anylogic environment.

Modelling. In our approach, the System is split up into two basic blocks, the model of the two tank system (the plant) and the Fuzzy controller. The plant is implemented by System Dynamics approach, as shown in Figure 1, using stock variables. Every tank in the system is mapped as a stock variable in the model. Therefore we have a stock variable x1, representing the water level of the first tank, x2 for the second tank, in represents the water pipe, and out the sink. The flow in the system is described by two differential equations; here they both have been split up in two parts, the part defining the water flowing into the tank and the part flowing out of the tank. These parts of the differential equation are distributed into the three flow variables u, v1 and v2 in the plant model:

$$u = 0.067 * u\_\text{var}$$

$$v1 = 0.06624 * 0.4 * \sqrt{|x1 - x2|} * sign(x1 - x2)$$

$$v2 = 0.0605 * 0.3 * r * |x2|^{0.48}$$

So the the resulting formulas for the water levels in the tanks are pretty simple:

$$x1 = u - v1, \quad x2 = v1 - v2$$

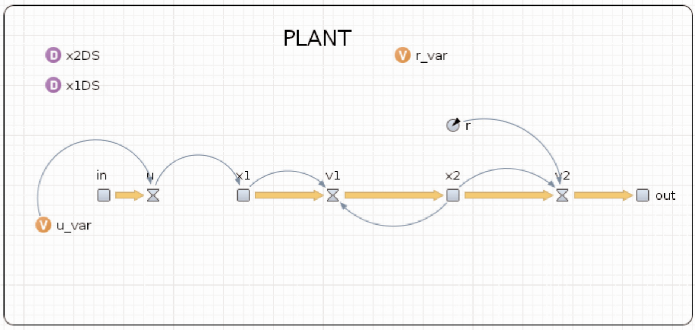

Figure 1: C9 - System Dynamics model in Anylogic.

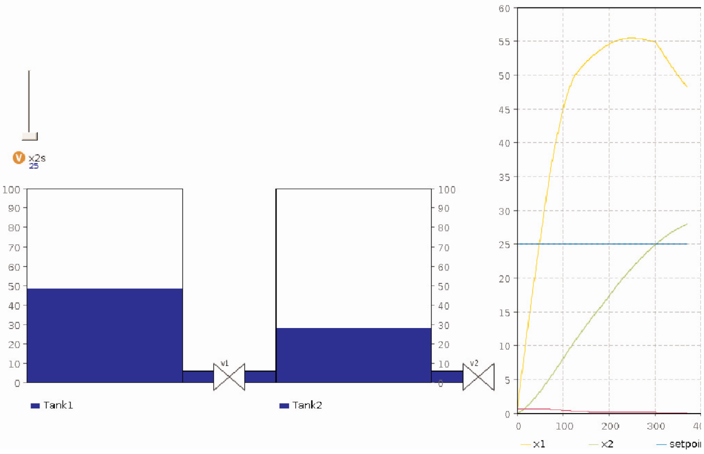

Figure 2: Direct visualization of tank dynamics in AnyLogic.

Controller Model. The second big part of the model is the fuzzy controller. We implemented the fuzzy rules into a Java function, which then is called periodically. The fact that Anylogic is based on Java enables the users to lookup mathematical functions (e.g. max, min, sqrt, ...) in the Java documentation. This makes live really easy. Anylogic also allows the user to import Java classes, making it possible to call any functions im-plemented in Java. In the following a code snippet for the control algorithm FC1 (the code has a total of 140 lines):

```
if (ex2 < 0)
{ ex2n1 =1;}
else if (ex2 >= 0 && ex2 <= 10)
{ ex2n1 =( ex2 /(0 -10) -10/(0 -10) );}
else
{ ex2n1 =0;}
if ( ex2 >= 0 && ex2 <= 20)
{ ex2nl =( ex2 /(0 -20) -20/(0 -20) );}
else
{ ex2nl =0;}
if ( ex2 >= 0 && ex2 <= 10)
{ ex2p1 =( ex2 /(10) );}
else if ( ex2 >= 10 && ex2 <= 30)
{ ex2p1 =( ex2 /(10 -30) -30/(10 -30) );}/
  ..............
* *** putting it all together *** */
p1 = min(ex2nl , x1p3 );
p2 = max( min(ex2nl , x1p2 ), min(ex2p1 , x1p3 ));
p3a = max(min(ex2nl , x1p1 ), min (ex2p1 , x1p2 ));
.......
p7 = max(p7a , p7b);
p8 = min(ex2p3 , x1nl );
nla = max(min(ex2n1 , x1nl ), min (ex2n1 , x1p1 ));
nlb = max(min(ex2n1 , x1p2 ), min (ex2n1 , x1p3 ));
.......
nlh = max(nlf , nle);
u_var = ((1/3) *nl *0.5 + 1.25* p1 + 2.5* p2 + 3.75* p3
        + 5* p4 + 6.25* p5 + 7.5* p6 + 8.75* p7 +
        (10 - 0.5) *p8 *1/3) /( nl /2 + p1 + p2 + p3 + p4 +
.......       p5 + p6 + p7 + p8 /2);
```

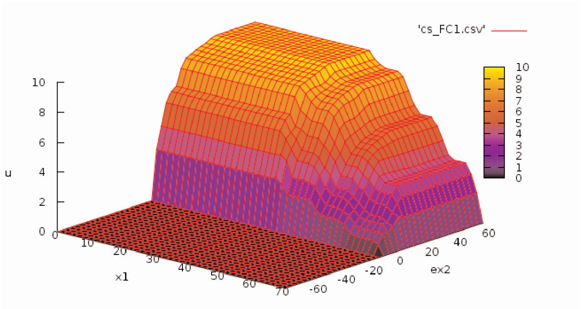

Figure 3: Control surface of FC1.

Anylogic provides a big set of different 2D diagram types, to visualize the results. For this comparison, we abused a stack chart diagram, to make a simplified visualization of the two tank system. A simulation, showing this visualization is shown in Figure 2.

A - Task: Computation of Controller Surfaces. The control surface of the controller represents the output of the controller at all possible input combinations. To get these, two counting variables (because of two inputs - x1 and ex2) are used to iterate of the ranges of the two inputs, as specified. These variables i and j range from 0 to 40 and from -20 to 20, resp. To get the same amount of points within the different ranges of x1 and ex2, j is multiplied with an appropriate factor, so we get the ranges specified: x1 - [0 ... 70] and ex2 - [-70 ... 70], 41x41 points = 1681 data points. Unfortunately, Anylogic does not provide the 3D surface plots, so the results of the simulation were exported and the surface plot was done by gnuplot. The result for Task A2, the control surface plot of FC1, is shown in Figure 3.

Specifying the simulation time is non-trivial. First it takes time to build the model: 5 seconds - first build after starting Anylogic, and 0.4 seconds - rebuilding. The simulation time itself very much depends on the scale of the model time to the real time. We decided to use the 512x speed for our simulations. This lead us to a Simulation time ta_{FC1} = 3.8 s and a ta_{FC2} = 3.7 s, ratio is ta_{FC1} / ta_{FC2} = 1.027 - almost no difference.

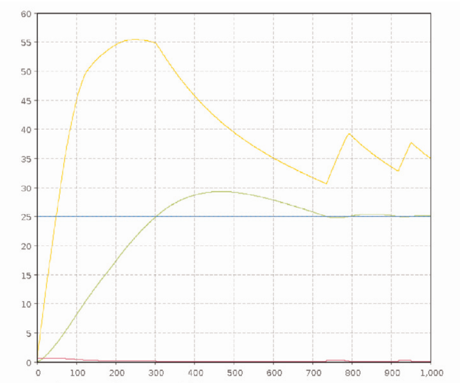

Figure 4: States and control, FC1.

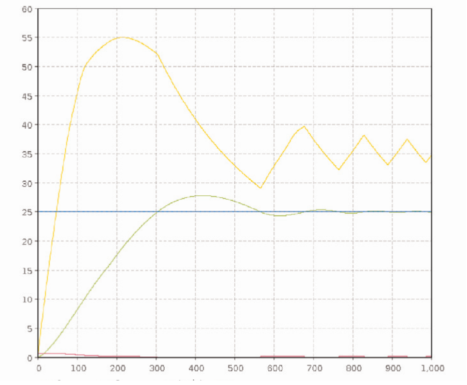

Figure 5: States and control, FC3.

B - Task: Simulation of the System. Simulation was performed in the ‘standard’ AnyLogic environment. Figure 4 shows the output of the system using fuzzy controller FC1. As can be seen the waterlevel x2 of the secont tank first overshoots the setpoint of 25 and then swings into the setpoint after 700 s. As mentioned before, measuring simulation time works only on a relative relative basis in AnyLogic. With same speed factor as in Task A1, simulation time was about 1.3 s, with no difference for FC1 and FC2.

C - Task: Weighted fuzzy control. The last task included a new Fuzzy controller (FC3), using FAM interference. The rule base is the same as with FC2, but some of the functions are weighted with a factor - simple changes in the Java - programmed control algorithm. Calculation time for controller surface and simulation time does not really change, compared with Task A and Task B, resp. Results of simulation with FC3 (Figure 5) show that the system swings into the setpoint much faster.

Résumé: The SD modelling approach in Anylogic allows for easy modelling of the plant in C9, but AnyLogic does not offer any control model library, neither classic contro nor fuzzy control. So all algorithms must be programmed in Java. On the other side, AnyLogic can be seen as Java programming and development environment, so that programming tasks are easy - for Java people.

Corresponding Author: Shabnam Tauböck,
Thomas Mair, Robert Leidenfrost, Andreas Platschek
Institute for Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria
Shabnam.Tauboeck@.tuwien.ac.at

Received: June 27, 2006
Revised: October 5, 2006
Accepted: October 12, 2006