Directly Programmed Fuzzy Control in ARGESIM Benchmark C9 'Fuzzy Control of a Two-Tank System' using Dymola

Anton Sodja, University of Ljubljana, Slovenia; *asodja@gmail.com*

Simulator: *Dymola* 5.3d, Dynamic Modeling Laboratory , is an object oriented simulation environment for acausal modeling (textual and graphical), simulation and visualisation of continuous, hybrid and discrete models. Dymola is able to understand Modelica, a unified textual and graphical modelling language offering many libraries for applications. Simulation can be carried out either in Dymola's own simulation environment or included in Simulink (only under Windows). For this solutions, Dymola version 5.3d for Linux was used (WWW.DYNASIM.COM).

Modelling. The model is composed of the Fuzzy Controller Model, the Plant model, and standard blocks from the Modelica Block Library (sum block, source block) - Figure 1. The plant model is described textually by the system governing ODEs using Dymola notation, put into a graphical block.

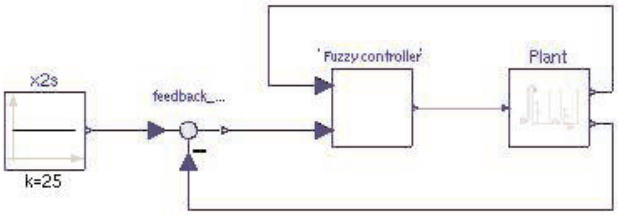

Figure 1: Top-level model: fuzzy controller and plant in closed loop.

There is no fuzzy control module or block as a part of Modelica Standard Library available inside the Dymola environment. The model was implemented on top of components from `Modelica.Blocks.Interfaces` library. The fuzzy controller is comprised of hierarchical connections of simple blocks. Input fuzzification blocks are made up of blocks for every fuzzy set. Those blocks are derived from Modelica's SISO block and linearly interpolate input between points in table that are given as a parameter to the every instance of a block:

```
parameter Real values[2];
parameter FuzzySetMembership membership[2];
algorithm
 if u <= values[1] then y := membership[1];
 elseif u <= values[2] then
   y := interpolate(values[1:2],
                    membership[1:2],u);
 elseif u <= values[3] then
   y := interpolate(values[2:3],
                    membership[2:3],u);
 else y := membership[3];
 end if;
```

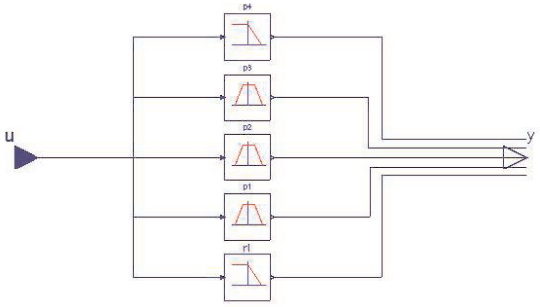

Figure 2: Input fuzzification structure - connected input fuzzy set blocks.

For different membership set's shapes different blocks have to be written. Structure of input fuzzification block can be seen on Figure 2. The same approach is used for the interface engine, it is looped through a 2D rules table and a *max-prod* operation for every rule is performed:

```
parameter Integer IFTable[rows,cols];
algorithm
 y := zeros(ny);
 for i in 1:rows loop
  for j in 1:cols loop
   y[IFTable[i,j]] :=
   max(y[IFTable[i,j]],(set_v[i]*set_h[j]))
  end for;
 end for;
```

For defuzzification every output set membership block provides two outputs: deviation from center and surface under degree of membership or degree of membership itself, resp. Those outputs are then simply routed to a block which calculates an output with selected defuzzification formula - center of gravity:

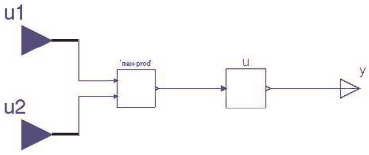

Figure 3: Fuzzy controller - two input fuzzification blocks, interface engine, ouput defuzzification block.

```
Interfaces.RealInput vars[nin];
  Interfaces.RealInput weights[nin];
  Interfaces.RealOutput y;
equation
  y = sum(vars*weights) / sum(weights);
```

A - Task: Computation of Controller Surfaces. The controller model is used as stand-alone Dymola model, with a discrete ramp signal on both inputs, with x_1 starting at 0 and going up to 70 and e_{x2} running from -70 to 70 for every x_1 change. For surface plotting MATLAB was used due to lack of 2D-plotting abilities in Dymola.

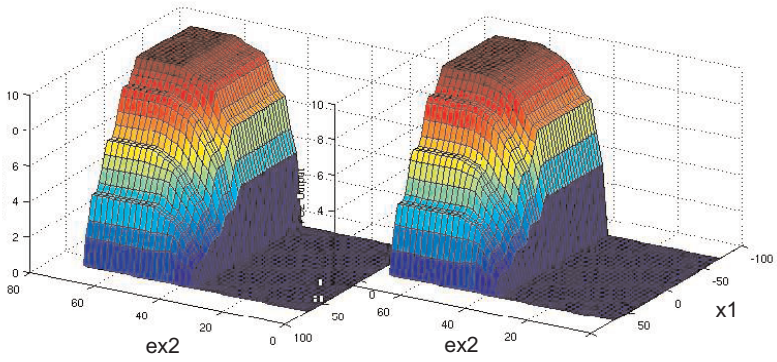

Figure 4: Control surface of FC1 (left) and FC2 (right).

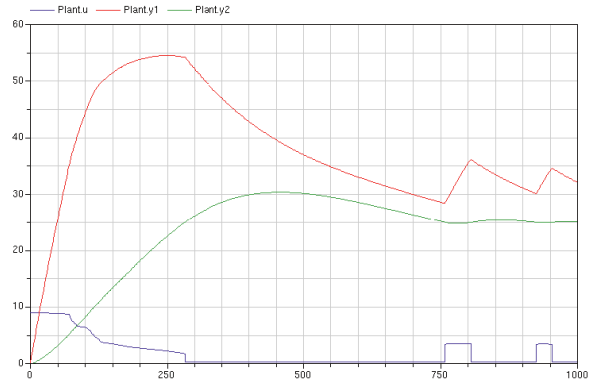

Figure 5: States and control for step reference - FC1 (triangluar membership functions).

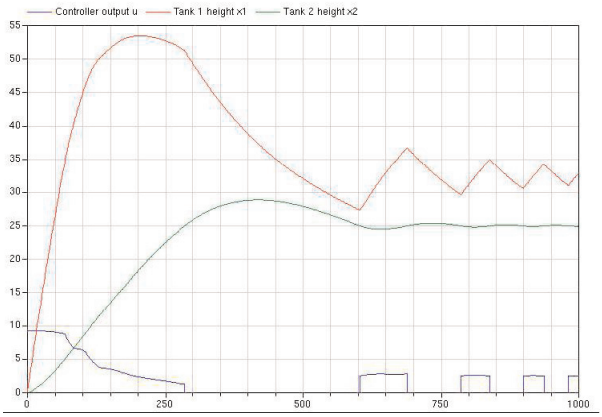

Figure 6: States and control for step reference - FC2 (singleton membership functions).

Data were extracted using `dymload.m` and `dymget.m` (MATLAB scripts provided together with Dymola). The result vectors need to be re-shaped into 2-D representation. Both surfaces (Figure 4, at left FC1 with triangular functions, at right FC2 with singletons) look very similar, differences can be only seen for high values of x1, at ex2 ~ 45, and at top of the surfaces.

Duration of calculation was measured by standard Unix utility `time` command, since model is compiled. No significant difference between calculation of FC1 and FC2 exist, with $ta_{FC1} = 0.872$ s, $ta_{FC2} = 0.884$ s, and with ratio $ta_{FC1} / ta_{FC2} = 0.9864$ s.

B - Task: Simulation of the System. Simulation of the whole system (Athlon XP-M 2500+) was performed in the Dymola environment and took 108 ms for a timespan of 1000 s in case of FC1, and 105 ms in case of FC2 (ratio is $tb_{FC1} / tb_{FC2} = 1.0286$).

Also not much difference can be seen in performance of both controllers (results for transient behaviour in Figure 5 and Figure 6). As expected, fuzzy control with singletons switches more often.

C - Task: Weighted Fuzzy Control. Weighting is implemented by extending the `InterfaceEngine` block with an additional table of weights. The calculated output set membership values are then multiplied by those weights.

```
parameter Real Weights[rows,cols];
  ...
y[IFTable[i,j]] := max(y[IFTable[i,j]],
Weights[i,j]*(set_v[i] * set_h[j]));
```

Calculation time for FC3 surface plot is $tc_{FC3} = 0.932$ s, agin very close to calculation times for FC1 and FC2.

Résumé: In this Dymola / Modelica solution, for plant model and for the discrete control structure standard Modelica blocks are used. Fuzzy control has to be programmed directly in textual Modelica code, making use of table function features and of very similar structure for all controller types, with and without weighting (singletons also programmed directly). The Dymola fuzzy control model can be used standalone, computing control values for arbitrary input values directly. For this solutions, Dymola version 5.3d for Linux was used.

Corresponding Author: Anton Sodja,
Laboratory of Modelling, Simulation and Control (LMSC),
Faculty of Electrical Engineering
Univ. of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia
asodja@gmail.com

Received: October 20, 2006
Revised: November 18, 2006; December 10, 2006
Accepted: December 15, 2006