# Fuzzy Controller Update 1/29/03

Following is description and outputs of FCL controller which now can process fuzzy output sets.

Slides 2&3 show the FCL input file. As before there are two inputs and two outputs. The two outputs now have the same rule base outputs, in the previous example they were inverses of each other, now the two outputs should show similar results, any differences will be from use of different fuzzy output sets.

Slide 4 shows the input/output fuzzy sets.

Slide 5 shows the input variable values in the first two plots, and the valve 2 outputs in the bottom two plots for singleton and rectangle fuzzy output sets.

Slide 6 shows the vavle2 outputs for the singleton, ramp & trapezoid, and ramp & triangle & trapezoid output fuzzy sets.

I would like to next work on integration into the hse as a fuzzy controller, and update the hse documentation and a testcase.

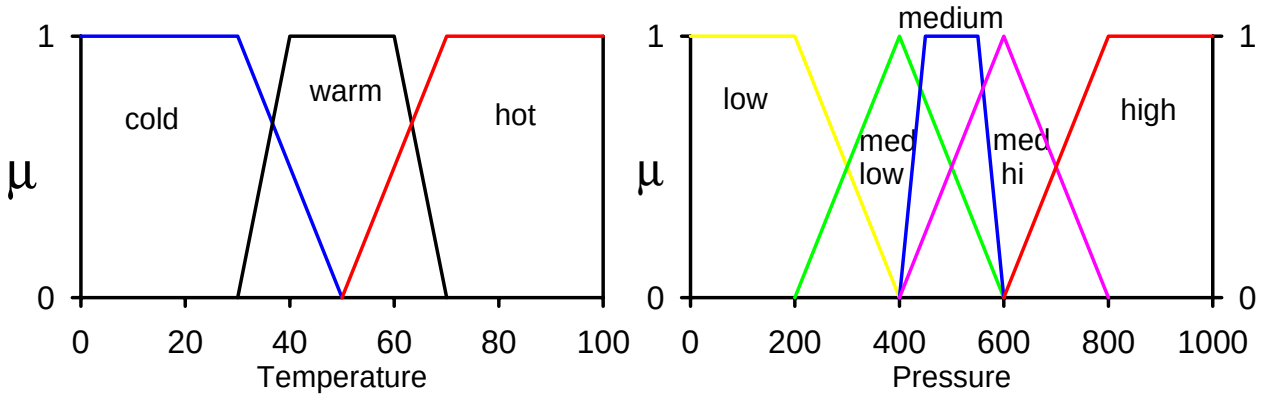
Following that, it seems that application testing of the fuzzy controller (as a decision tool) should be accomplished with some ‘realistic’ decision variables.

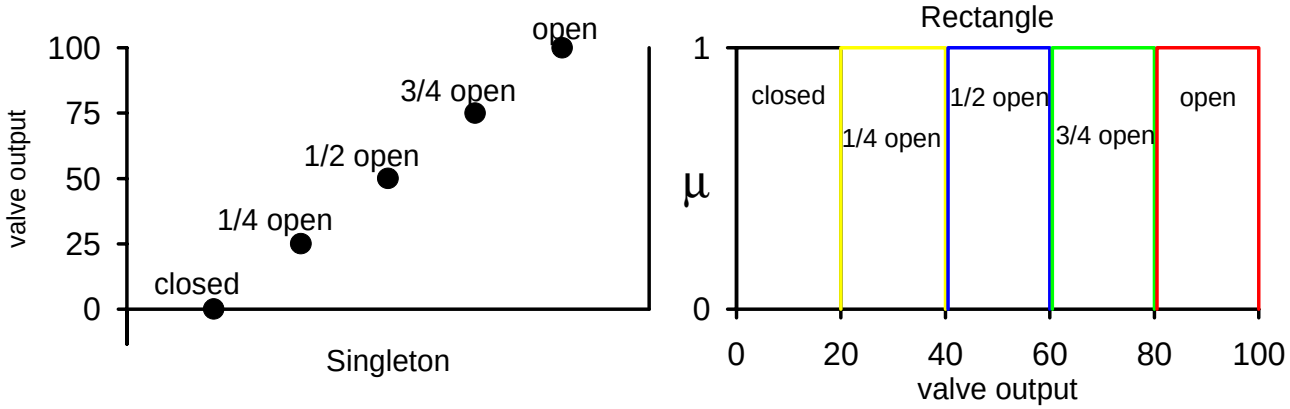
The next conceptual hurdle seems to be investigation of how to apply the fuzzy controller as a non-fuzzy rule-based decision tool. I can think of two approaches. One is to judicously design the FCL so that the system is in effect not fuzzy, i.e use singletons and definitions of fuzzy sets to limit the fuzzy inferencing, or second, to augment the FCL definitions with additional functions that allow non-fuzzy I/O processing (singleton inputs and outputs without membership value inferencing), and implementation of rule-functions (lookup tables or curves as I/O filters.) This probably needs some discussion, viz. do we even care about non-fuzzy rule processing, or is a fuzzy rule-based approach going to be the ‘standard’? Once this is achieved, I hope that the result can be applied as the ‘default’ Control Supervisory Algorithm in the MSE.

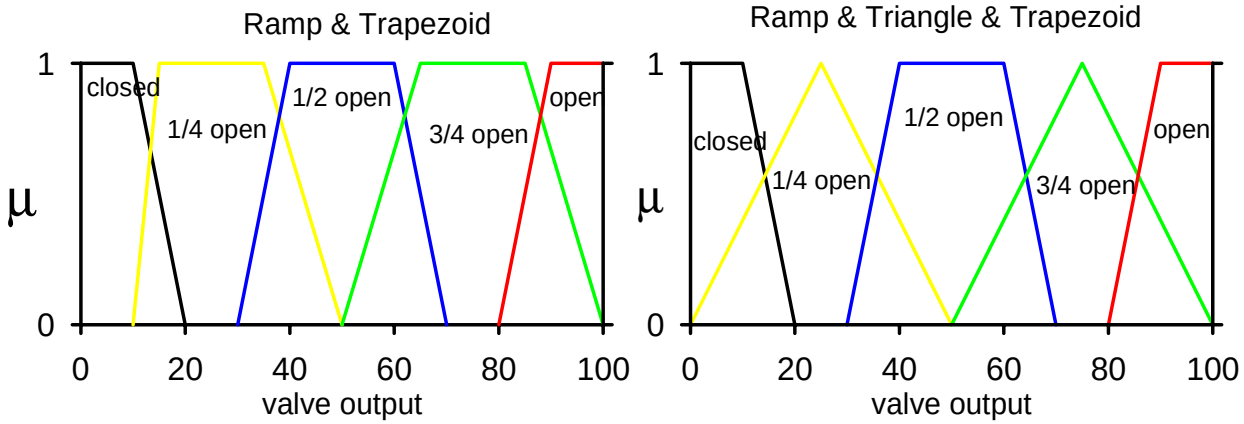
In conjunction with the latter objectives, I would like to take some time to write a couple of papers, probably in regards to the sigmoid controller, and perhaps a look at the nonlinear dynamics of a hydrologic network response.

```
FUNCTION_BLOCK Fuzzy_FB
VAR_INPUT
        // temperature and pressure input variables
        temp    : REAL;
        pressure : REAL;
END_VAR
VAR_OUTPUT
        // dual valve output
        valve1 : REAL;
        valve2 : REAL;
END_VAR
FUZZIFY temp
        // cold ramp, warm trapezoid, hot ramp
        TERM cold := (30, 1) (50, 0);
        TERM warm := (30, 0) (40, 1) (60, 1) (70, 0);
        TERM hot  := (50, 0) (70, 1);
END_FUZZIFY
FUZZIFY pressure
        // ramp, triangle, trapezoid, triangle, ramp
        TERM low     := (200, 1) (400, 0);
        TERM med_low  := (200, 0) (400, 1) (600, 0);
        TERM medium   := (400, 0) (450, 1) (550, 1) (600, 0);
        TERM med_high := (400, 0) (600, 1) (800, 0);
        TERM high     := (600, 0) (800, 1);
END_FUZZIFY
DEFUZZIFY valve1
         // All outputs are singletons
         TERM closed          := 0;
         TERM quarter_open     := 25;
         TERM half_open        := 50;
         TERM threeQuarter_open := 75;
         TERM open                    := 100;
         ACCU: MAX;
         METHOD: COG;
         DEFAULT:= 0;
         RANGE:= (0, 100);
END_DEFUZZIFY
DEFUZZIFY valve2
         // singleton output terms
         TERM closed          := 0;
         TERM quarter_open     := 25;
         TERM half_open        := 50;
         TERM threeQuarter_open := 75;
         TERM open                    := 100;
         // rectangular output terms
//       TERM closed          := (0,  0) (0,  1) (20, 1) (20,  0);
//       TERM quarter_open     := (20, 0) (20, 1) (40, 1) (40,  0);
//       TERM half_open        := (40, 0) (40, 1) (60, 1) (60,  0);
//       TERM threeQuarter_open := (60, 0) (60, 1) (80, 1) (80,  0);
//       TERM open                                          := (80, 0) (80, 1) (100,1) (100, 0);
         // Ramp and Trapezoidal output terms
//       TERM closed          := (10, 1)  (20, 0);
//       TERM quarter_open     := (10, 0)  (15, 1)  (35, 1) (50, 0);
//       TERM half_open        := (30, 0)  (40, 1)  (60, 1) (70, 0);
//       TERM threeQuarter_open := (50, 0)  (65, 1)  (85, 1) (100, 0);
//       TERM open                                          := (70, 0)  (80, 1);
         // Ramp, Triangle and Trapezoidal output terms
//       TERM closed          := (10, 1)  (20, 0);
//       TERM quarter_open     := (0,  0)  (25, 1)  (50, 0);
//       TERM half_open        := (30, 0)  (40, 1)  (60, 1) (70, 0);
//       TERM threeQuarter_open := (50, 0)  (75, 1)  (100, 0);
//       TERM open                                          := (80, 0)  (90, 1);
         ACCU: MAX;
         METHOD: COG;
         DEFAULT:= 0;
         RANGE:= (0, 100);
END_DEFUZZIFY
```

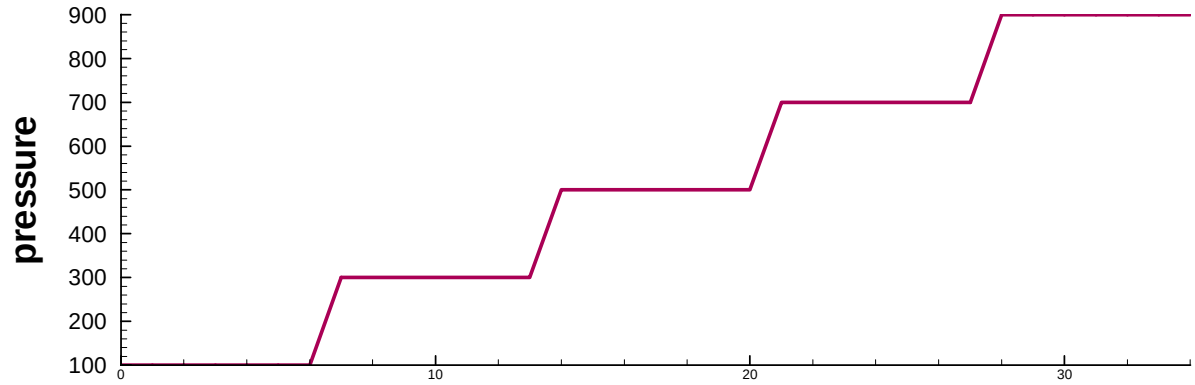
```
RULEBLOCK No1
        AND : MIN;
        OR  : MAX;
        ACT : MIN;
        //
        RULE 1: IF temp IS cold AND pressure IS low
                    THEN valve1 IS closed, valve2 IS closed;
        //
        RULE 2: IF temp IS cold AND pressure IS med_low
                    THEN valve1 IS quarter_open, valve2 IS quarter_open;
        //
        RULE 3: IF temp IS cold AND pressure IS medium
                    THEN valve1 IS half_open, valve2 IS half_open;
        //
        RULE 4: IF temp IS cold AND pressure IS med_high
                    THEN valve1 IS threeQuarter_open, valve2 IS threeQuarter_open;
        //
        RULE 5: IF temp IS cold AND pressure IS high
                    THEN valve1 IS open, valve2 IS open;
       //
        RULE 6: IF temp IS warm AND pressure IS low
                    THEN valve1 IS quarter_open, valve2 IS quarter_open;
        //
        RULE 7: IF temp IS warm AND pressure IS med_low
                    THEN valve1 IS half_open, valve2 IS half_open;
        //
        RULE 8: IF temp IS warm AND pressure IS medium
                    THEN valve1 IS half_open, valve2 IS half_open;
        //
        RULE 9: IF temp IS warm AND pressure IS med_high
                    THEN valve1 IS half_open, valve2 IS half_open;
        //
        RULE 10: IF temp IS warm AND pressure IS high
                     THEN valve1 IS threeQuarter_open, valve2 IS threeQuarter_open;
        //
        RULE 11: IF temp IS hot AND pressure IS low
                     THEN valve1 IS open, valve2 IS open;
        //
        RULE 12: IF temp IS hot AND pressure IS med_low
                     THEN valve1 IS threeQuarter_open, valve2 IS threeQuarter_open;
        //
        RULE 13: IF temp IS hot AND pressure IS medium
                     THEN valve1 IS half_open, valve2 IS half_open;
        //
        RULE 14: IF temp IS hot AND pressure IS med_high
                     THEN valve1 IS quarter_open, valve2 IS quarter_open;
        //
        RULE 15: IF temp IS hot AND pressure IS high
                     THEN valve1 IS closed, valve2 IS closed;
END_RULEBLOCK
END_FUNCTION_BLOCK
```


1
0
μ
cold
warm
hot
0
20
40
60
80
100
Temperature
medium
low
med
low
med
hi
high
0
200
400
600
800
1000
Pressure

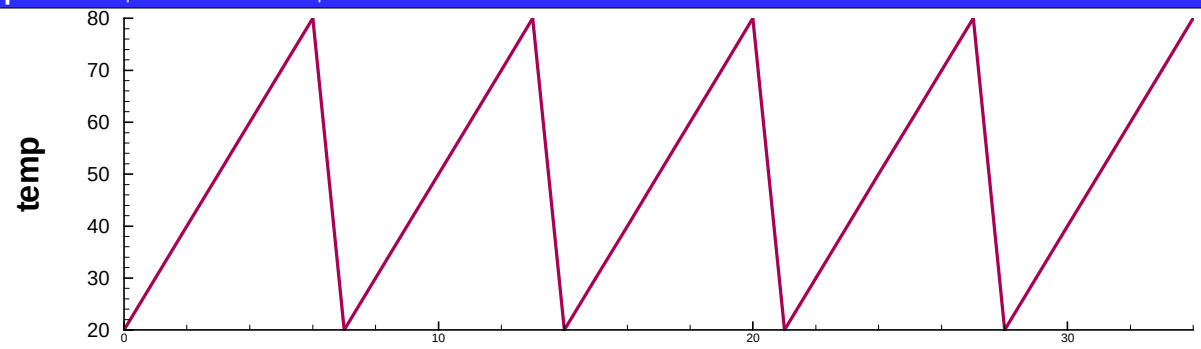

valve output
100
75
50
25
0
open
3/4 open
1/2 open
1/4 open
closed
Singleton
Rectangle
closed
1/4 open
1/2 open
3/4 open
open
valve output


Ramp & Trapezoid
closed
1/4 open
1/2 open
3/4 open
open
valve output
Ramp & Triangle & Trapezoid
closed
1/4 open
1/2 open
3/4 open
open
valve output

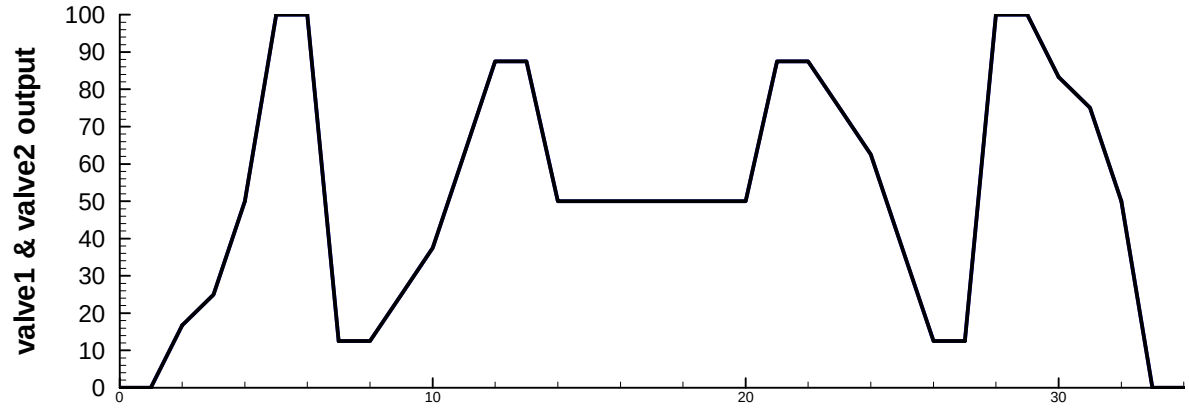
## Pressure | 29 Jan 2003



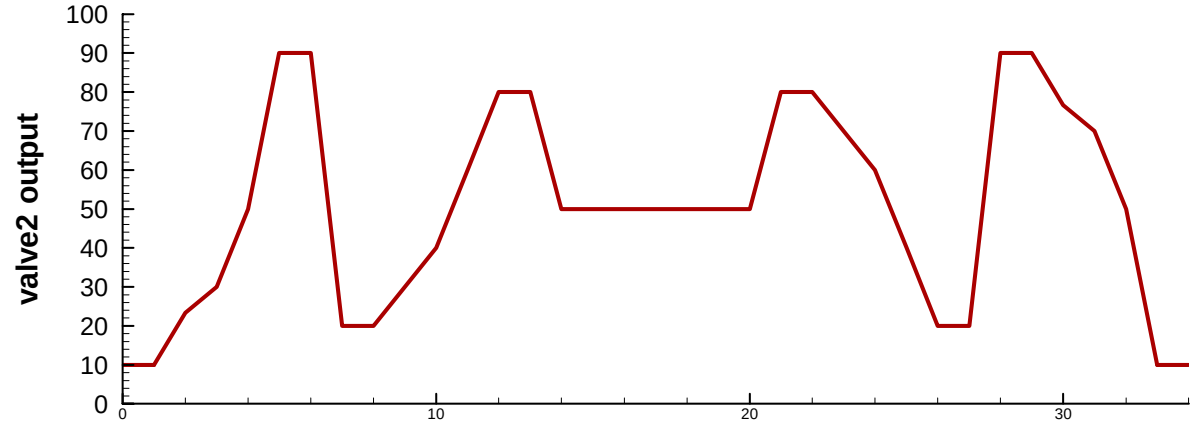

## Temperature | 29 Jan 2003



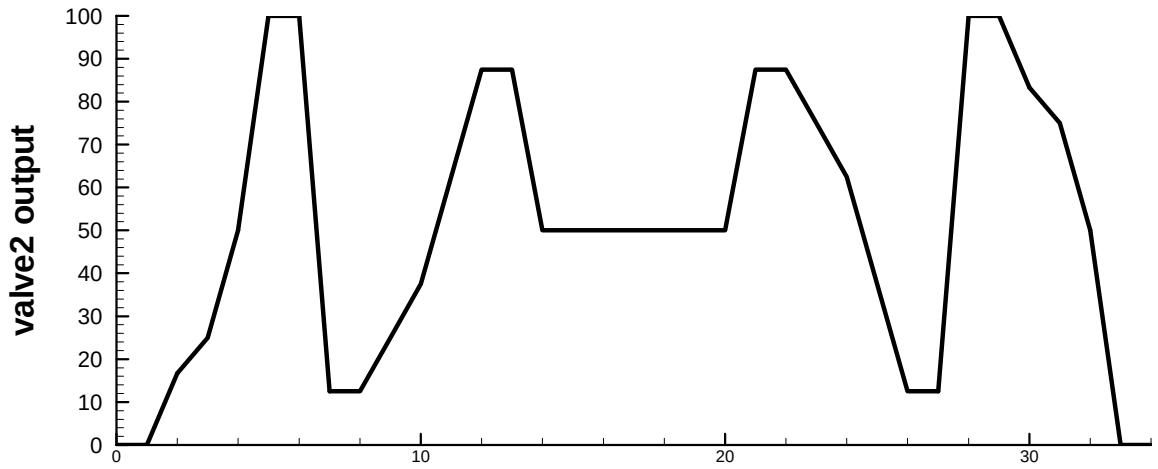

## Singleton Outputs | 29 Jan 2003



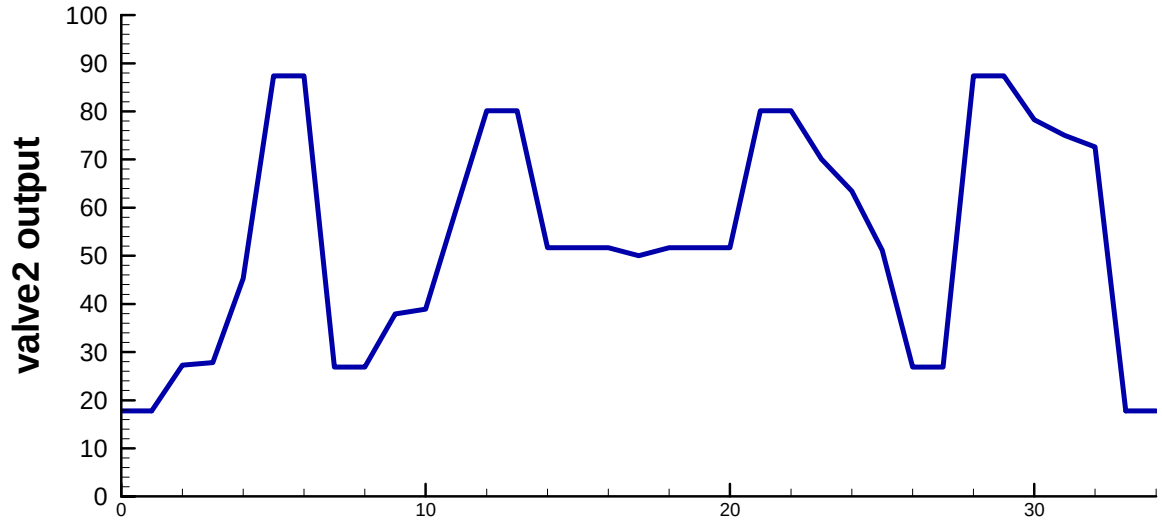

## Rectangle Outputs | 29 Jan 2003



**Singleton Output | 29 Jan 2003**




**Ramp & Trapezoid Output | 29 Jan 2003**




**Ramp & Triangle & Trapezoid | 29 Jan 2003**