Fuzzy Logic for PID Tuning: Simple Design with Inference Rules

Nelson Campos [1] and Antonio Marcus Nogueira Lima[1] and Saulo Oliveira Dornellas Luiz [1]

***Abstract*— This paper describes inference rules for tuning PID controllers for process control. Fuzzy logic is being applied to determine the controller parameters. The Mamdani method with 6 rules is being used here for tuning the PID controllers.**

I. INTRODUCTION

The Proportional-Integral-Derivative (PID) Controllers are so far the most widely used in the industry because of its simplicity and large range of operations. There are many techniques for PID design, but in this present work we will constraint our approach for the Ziegler & Nichols method for tuning the coefficients of the controller. [1]

The transfer function of a PID controller has the following form:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

where Kc, Ki, and Kd are the proportional, integral, and derivative gains, respectively. The equation (1) could be written as:

$$G_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{2}$$

Tunning rules would be applied to determine the parameters in the equations (1) and (2). According to [2], the following assertions give information about the PID parameters:

1) Increasing K_c decreases period and vice versa
2) Increasing K_c increases overshoot and vice versa
3) Increasing K_c decreases rise time and vice versa
4) Increasing K_c increases damping and vice versa
5) Decreasing T_i increases overshoot ratio and vice versa
6) Decreasing T_i increases damping and vice versa
7) Decreasing T_i decreases stability and vice versa
8) Increasing T_i decreases overshoot and vice versa
9) Increasing T_d increases stabilty and vice versa
10) Increasing T_d decreases rise time and vice versa.

All the features are self-explanatory except for the damping that is defined as follows:

$$damping = \frac{SecondPeak - FirstValley}{FirstPeak - FirstValley} \tag{3}$$

II. FUZZY LOGIC FOR PID TUNNING

In order to design a PID Controller based on inference rules, the process of the equation (4) was used as a case of study.

$$H(s) = \frac{e^{-s}}{(10s+1)(10s+1)} \tag{4}$$

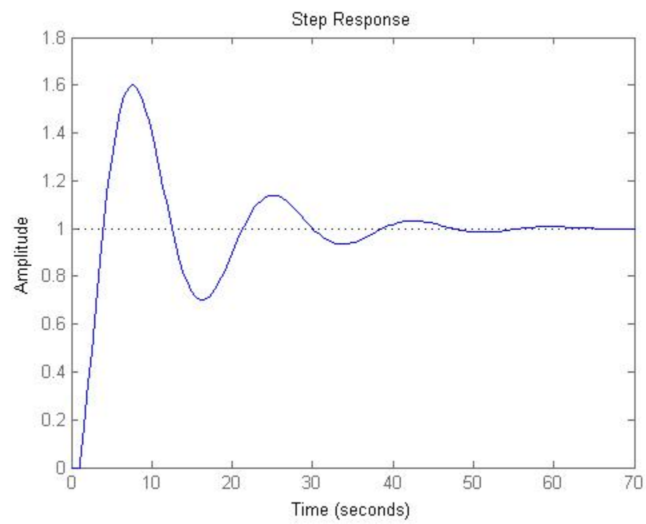

Fig. 1. Closed-loop output of the (4)

TABLE I
ZIEGLER & NICHOLS METHOD

Controller	Kp	Ti	Td
P	0.5Ku	Infinity	0
PI	0.45Ku	Pu/1.2	0
PID	0.6Ku	Pu/2	Pu/8

This is a second order process with a lag of one second. The closed-loop output of (4) is showed in Figure 1.

A traditional PID controller was used in order to compare the step response in the closed loop of the system with the proportional gain Kc = 15, Ti = 6.75 and Td = 1.69.

Table I shows the Ziegler & Nichols tuning method for the parameters based on the critic gain and the period where the system starts its oscillation.

Figure 2 shows the proposed Fuzzy Gain Scheduler. The approach here takes advantage of the rules 1 to 10 and some reasoning skills are used to calculate the controller parameters.

Fuzzification is the process of changing a real scalar value into a fuzzy value. This is achieved with the different types of fuzzifiers (membership functions).

Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. It is typically needed in fuzzy control systems.

The most commonly method for fuzzy inference is the Mamdani methodology. Mamdani's method was among the

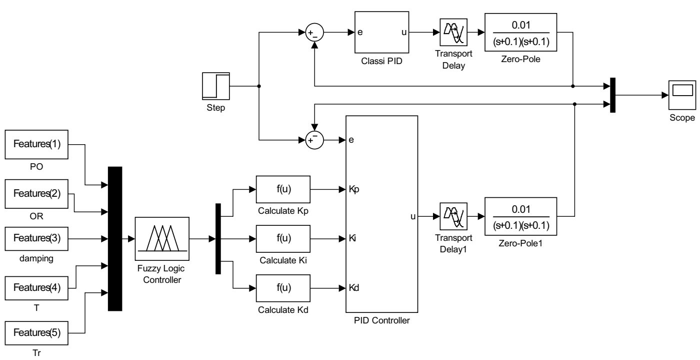

Fig. 2. The proposed Fuzzy Gain Scheduler

first control systems built using fuzzy set theory.

The Mamdani inference expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

The controller gain scheduler proposed here is Mamdani-type and has 5 inputs: Percentual of Overshoot (PO), Overshoot Ratio (OR), damping, period (T) and rise time (Tr). These features are extracted from the matlab function calculateFeatures in which expects a transfer function and its step response and returns the desired features. These features are defined in the fuzzy toolbox as gaussian member funcitons with two ranges: Small and Big. As described before, in the Mamdani-type method the outputs are also member functions. The ouputs are the gain parameters of the PID are their member functions are also of gaussian type. The inputs and outputs of the proposed scheduler can be seen in Figure 3. Figure 4 shows the PO member functions and Figure 5 shows output Ti member functions.

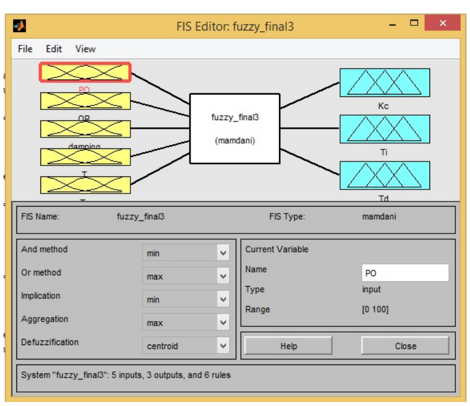

Fig. 3. The Fuzzy Inputs and Outputs

The rules 1 to 10 were defined in the Fuzzy Rule Editor.

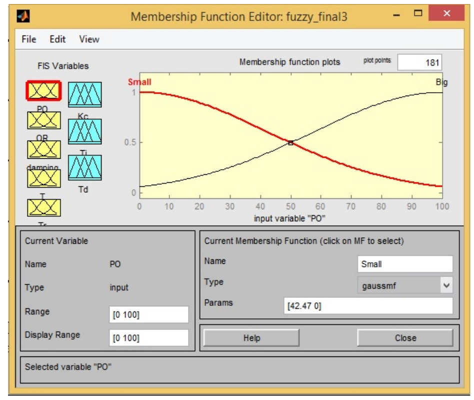

Fig. 4. PO Member Funtions

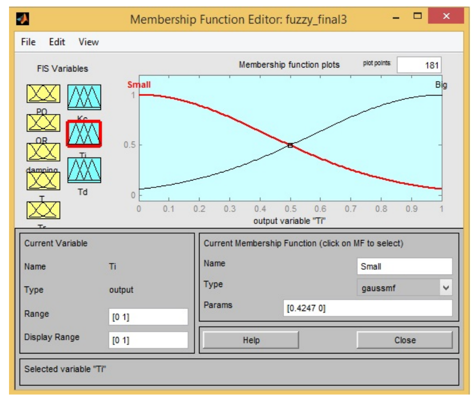

Fig. 5. Ti Member Functions

These rules infer the behavior of the output variables in order to the laws dictated by them. Figure 6 shows the Rule Editor of the Fuzzy toolbox.

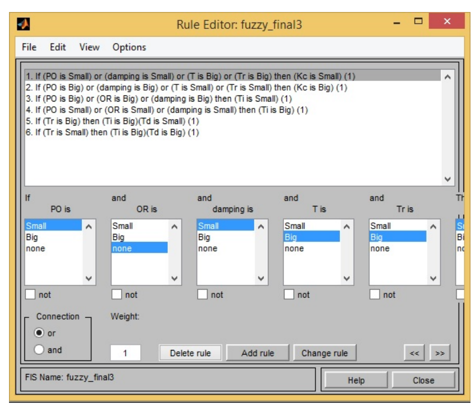

Fig. 6. The Fuzzy Rule Editor

The rules can also be verified graphically as can be seen in Figure 7. Figures 8 and 9 show the generated surface of the gain Kc in function of OR and PO, and the gain Ti in function of the damping and T, respectively.

The result of the two outputs: the classic PID (in yellow) and the Fuzzy Gain Scheduler (in purple) are showed in Figure 10;

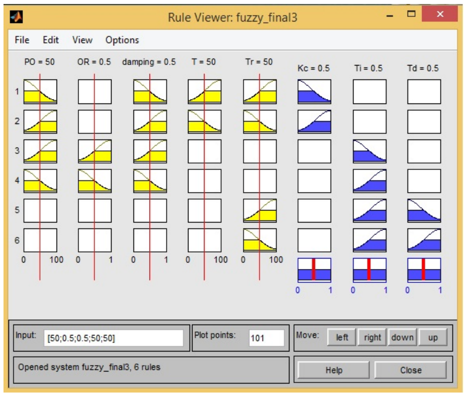

Fig. 7. The Fuzzy Rule Viewer

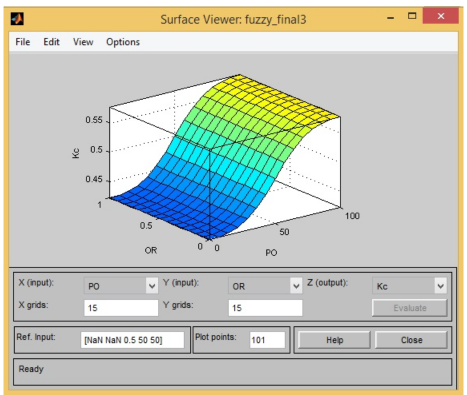

Fig. 8. The Surface Viewer for KcxPOxOR

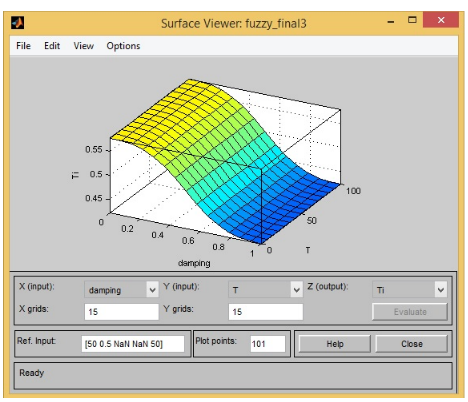

Fig. 9. The Surface Viewer for TixTxdamping

The range of the outputs Kc, Ti and Td are defined to be between 0 and 1 in the Fuzzy Toolbox. However it may not cover all the desired scenarios in the PID controller. An adjustment inspired in [3] is done as follows:

$$Kc = (Kc_{max} - Kc_{min})Kc^{fuzzy} + Kc_{min} \tag{5}$$

$$Kd = (Kd_{max} - Kd_{min})Kd^{fuzzy} + Kd_{min} \tag{6}$$

Different of [3], we also define the third output as the integrative gain as follows:

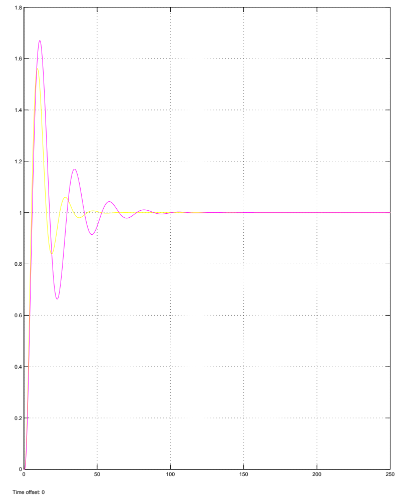

Fig. 10. The two outputs of the system

$$Ki = (Ki_{max} - Ki_{min})Ki^{fuzzy} + Ki_{min} \tag{7}$$

The parameters Kc^{fuzzy}, Ki^{fuzzy} and Kd^{fuzzy} are in the range [0,1] and are normalized with equations (5) to (7). The minimum and maximum values of each parameters are constrained with the Table I. For example, according to the Ziegler & Nichols method, $Ki = 2 * Ku/Tu$. Thus, $Ki_{min} = 1.8 * Ku/Tu$ and $Ki_{max} = 2 * Ku/Tu$. This reasoning is analog to ther others parameters.

III. CONCLUSIONS

In this work was presented a simple Fuzzy PID gain scheduler. Although there are controversies about the Fuzzy logic applications [4], when there is no interest in build analytic and mathematical models to a given process, common sense and some rules plus reasoning skills may be applied to design a feasible application and Fuzzy logic sound good in this situation.

IV. APPENDIX

```
function Features = calculateFeatures(F, y, t)
N = size(t,1);
dy = zeros(N,1);
info = stepinfo(F);
zero_cross = 0;
t_cross = 0;
index = 0;


for i=2:N
  dy(i) = y(i)-y(i-1);
end


for i=1:N
    if abs(y(i)-1) < abs(dy(i))/2
        t_cross(i) = t(i);
        index(i) = i;
        zero_cross = zero_cross+1;
    end
end

%Remove all the elements which values are 0
t_cross = t_cross(t_cross ~=0);
index = index(index ~=0);

P1 = max(y);
PO = 100*(P1-1);
riseTime = info.RiseTime;

if size(index,2) == 2 || size(index,2) == 3
    T = NaN;
    P2 = 0;
    OR = 0;
    damping = 0;
    disp('System without oscillation')

else if size(index,2) > 3 % If the system has oscillation
  T = t_cross(3)-t_cross(1);
  t0 = index(1);
  t1 = index(3);
  size(index)
  t2 = index(4);

  V1 = min(y(t0:t1));
  P2 = max(y(t1:t2));
  OR = (P2-1)/(P1-1);
  damping = (P2-V1)/(P1-V1);

  if OR < 1e-6
      disp('System Unstable')
  else disp('System with oscillation')
  end

else
    T = NaN;
    PO = NaN;
    OR = NaN;
    damping = NaN;
    riseTime = NaN;
    disp('Unstable System')
end
end
Features = [PO; OR; damping; T; riseTime];
```

REFERENCES

[1] Ziegler, John G., and Nathaniel B. Nichols. "Optimum settings for automatic controllers." trans. ASME 64.11 (1942).

[2] Litt, Jonathan. "An expert system to perform on-line controller tuning." Control Systems, IEEE 11.3 (1991): 18-23.

[3] Zhao, Zhen-Yu, Masayoshi Tomizuka, and Satoru Isaka. "Fuzzy gain scheduling of PID controllers." Control Applications, 1992., First IEEE Conference on. IEEE, 1992.

[4] Zadeh, Lotfi A. "Is there a need for fuzzy logic?." Information sciences 178.13 (2008): 2751-2779.