Technical University of Denmark

DTU

Tuning-Rules for Fuzzy Controllers

Jantzen, Jan

DTU Library
Technical Information Center of Denmark

TUNING-RULES FOR FUZZY CONTROLLERS

Jan Jantzen

Technical University of Denmark, Electric Power
Engineering Department, DK-2800 Lyngby, DENMARK.
e-mail: STARDUST at NEUVM1

ABSTRACT

A theoretical study has shown that a fuzzy controller for a single-input-single-output process can be tuned in four intuitive steps, much like classical PID (proportional-integral-derivative) controllers. A fuzzy lookup table controller becomes a PID controller if the lookup table is replaced by a summation. The relationships between the fuzzy gain factors and the PID tuning parameters are simple mathematical ratios and products. The results were applied to fuzzy controllers in simulation and on a laboratory water tank rig. Experiments show significant time-savings during tuning.

INTRODUCTION

In a local graduate course on fuzzy control, the students used to spend an hour or more tuning a fuzzy controller. By contrast, it takes 10 - 15 minutes to tune a classical PID (proportional-integral-derivative) controller. The students mostly used trial and error for the fuzzy controller, because good, reliable tuning-rules are hard to find. The controlled object is a water tank rig in the laboratory. The objective is to adjust the water level after a step in the setpoint as fast as possible, with the smallest overshoot, and avoiding steady-state offset.

A study was then carried out in order to find an easy and fast way to tune a fuzzy controller for a single-input-single-output process.

The result is a set of tuning-rules (see the section 'Tuning-rule transfer'). Computer simulations of the water tank rig demonstrate, that the tuning progresses much like in the PID case (see Fig 3). Mathematical relationships link the tuning parameters of the fuzzy and PID controllers (see Table I and appendix).

The fuzzy controllers in question are table-based [4]. Referring to Fig.1, they use setpoint error, E, and change in error, CE, for input. The controller output is usually an incremental 'change in output', CO, which is then integrated. Alternatively, the controller output can be the control signal, O, directly. These two controller configurations will be referred to as the 'CO-controller' and the 'O-controller'.

The gain factors -- often named GE, GCE, GCO (GO for the O-controller) -- must be tuned to get the best transient response, and they are thus central to the study.
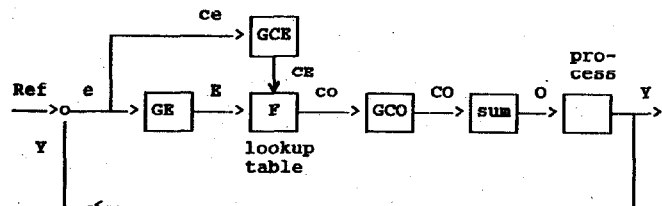

Fig. 1. Fuzzy CO-controller

The inputs, E and CE, are clipped and quantized to fit onto discrete universes on the two axes of the table. The choice of resolution is the designer's choice between sensitivity and computing time; 13 integer points centered around zero has been used as a standard universe by several researchers, [4], [8]. The numbers in the table result from a fuzzy inference, [9] involving several linguistic statements like: 'IF E is Largepositive and CE is Largepositive THEN CO is Largenegative', [2]; the Largepositive and Largenegative are fuzzy sets defined on the standard 13 point universe.

Originally, the error axis of the table had a double zero, [4], due to a user-friendly definition of E and CE. Yamazaki, [8] got rid of the double zero, by redefining E to be a function of Ref - Y, and CE a function of e(n) - e(n-1), like common practice. By this definition, the lookup table acts as a discrete phase plane.

Tuning a fuzzy controller

The literature on tuning-rules is rather sparse; the following is a summary.

It is common to view the gain factors as scaling factors, [3], [4], [8]. Thus the role of GE and GCE is to map the variation of e and ce onto the standard universe to make the best use of the lookup table. The role of GCO (GO) is to keep the control signal within its operating limits. This results in the tuning-rules:

- keep abs (GE * Elarge) equal to 6

- keep abs (GCE * CElarge) equal to 6

- keep abs (GCO * COlarge) equal to the operating limits

- keep abs (GO * Olarge) equal to the operating limit

where 'abs' means absolute value.

A fuzzy controller typically allows the process to wander a bit around the steady state (limit cycles). This is due to the discrete nature of the controller. It can be avoided to an extent by making the controller more sensitive to deviations from the setpoint, i.e. increasing GE. If the admissible steady state error is Epsilon, then

- keep abs (GE * Epsilon) less than or equal to 0.5

Thus E will jump to a neighbouring cell, if the error is larger than Epsilon, [3]. On the other hand, GE can get too large and make the system unstable. Pedrycz has found that

- for increasing GE, the response is faster with larger overshoots and significant oscillations;

- changing GCE hardly influences the rise-time and the peak-time, [5].

Some authors have found that the ratio GCE/GE is important.

It is generally believed that the CO controller is equivalent to the PI-controller, [4, p 1187], [6], or at least 'superficially' similar, [8, p 190].

However, tuning still remains an open problem: it is not clear how GE, GCE, GCO (GO), and the sampling time affect overshoot, rise time and steady state error -- in contrast with the PID controller.

Tuning-rule transfer

The central finding of the study is the following: if we omit the clipping and quantization of E and CE, and replace the lookup table in Fig. 1 by a simple summation point, then we get a pure PI controller. This can be shown by block-diagram manipulations, or mathematically (see Appendix for details).

In case of an O-controller, the same trick would turn that into a pure PD controller. To get a fuzzy PID controller, the designer can include a third term, integral error IE, in the O-configuration (see Appendix).

A discrete PID controller can be written

$$O(n) = Kp*[e(n) + (1/Ti)*\sum_{0}^{n} ej*Ts + Td*(e(n)-e(n-1))/Ts]$$

where O is the controller output, index n is the time instant, Kp is the proportional gain, and e is the error. The error e equals Ref - Y (cf. Fig. 1). The Ti, Td and Ts are integral, derivative and sampling time respectively. Table I below holds the resulting relationships between the gains of the PID and fuzzy controllers. The sampling time Ts appears along with GCE in the formulas. It would disappear if e(n)-e(n-1) were divided by Ts, in accordance with usual practice. Notice that there is a degree of freedom in the fuzzy controllers: there is always one more gain factor than in the PID-case.

There is an empirical PID tuning approach, widely used, which is suitable for a transfer to fuzzy controllers, [7], [Foxboro in 1]:

1. Remove all reset (1/Ti=0) and derivative action (Td=0), and tune the proportional mode (Kp) to give the desired response characteristics, ignoring any steady state offset.

2. Increase the proportional gain (Kp), and attempt to restore the response characteristics by adjusting the derivative time (Td). Repeat until the proportional gain (Kp) is as large as possible.

3. Adjust the reset time (Ti) to remove the offset.

Using the formulas in Table I, it is possible to make similar tuning-rules for the fuzzy controllers:

1. Remove all integral action, i.e keep GE very small and GCE large (CO-controller) or GIE = 0 (fuzzy PID). Remove all derivative action, i.e. put GCE = 0 (O-controller and fuzzy PID). Tune the proportional gain, cf. the Kp column in the table, to give the desired response characteristics (overshoot), ignoring steady state offset.

2. Increase the proportional gain (cf. the Kp column), and attempt to restore the response characteristics by adjusting the derivative time (cf. the Td column). Repeat until the proportional gain is as large as possible.

3. Adjust the reset time (cf. the Ti column) to remove the offset.

4. Use the extra degree of freedom to place the gains according to the section 'Tuning a fuzzy controller'.

TABLE I. Relationships between fuzzy and PID gains

Fuzzy	Kp	Td	1/Ti
CO-controller	GCE*GCO	0	GE/(GCE*Ts)
O-controller	GE*GO	(GCE*Ts/GE)	0
Fuzzy PID	GE*GO	GCE/GE	GIE/GE

Computer simulations will serve to evaluate and illustrate the usefulness of the tuning-rules next.

SIMULATIONS

The simulated object is a laboratory water tank rig. The objective is to adjust the water level after a step in the setpoint. A pump feeds the water tank and a load pump takes water out again, see Fig. 2. The block-diagram was turned into the discrete state-space form using the basic MATLAB computer package. The PID and fuzzy controllers have been coded in MATLAB m-files, and the simulation progresses iteratively.

The first series of experiments (Fig. 3, top row and bottom left) uses a CO-controller with a linear lookup table. That is, each element in the table is the sum of the numbers on the axes; thus the largest table element is 12. The step responses in Fig. 3 (top left), are indeed similar to proportional control with increasing proportional gain. The gain GCO is useful in adjusting the proportional gain, because it does not affect the integral gain, cf. Table I. The steady-state error decreases with increasing GCO, because the proportional gain increases (cf. Table I). The system oscillates perhaps a little more than genuine P control. In Fig. 3 top right, the integral gain increases with increasing GE (cf. Table I). The steady-state error disappears and the overshoot increases, as in a PI controller. The size of GE should preferably be larger than 6. Otherwise, the universe, and the lookup table, will not be used to the full extent when there is a unit step in the setpoint.

Fig. 3, bottom left, compares the tuned CO-controller with the equivalent (in the sense of Table I) genuine PID. The fuzzy controller has less overshoot and longer rise time. This 'sluggishness' is characteristic of the fuzzy controller, and often referred to as robustness. It is interesting to note, though, that the explanation lies with the clipping and quantization in the fuzzy controller, being the only difference between the two controllers in the figure.

The O-controller and the fuzzy PID controller are easier to tune than the CO-controller, and similar graphs can be produced for them.
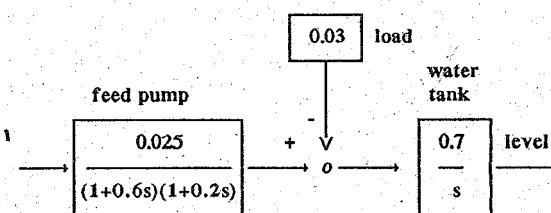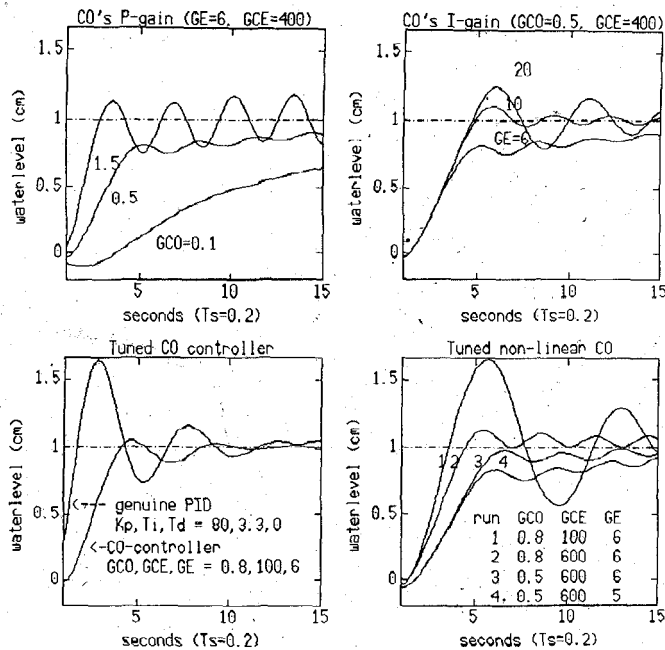
Fig. 2. Laboratory water model.

Fig. 3. Successive tuning. Top left: CO-controller with linear look-up table, tuning the proportional mode. Top right: CO-controller with linear lookup table, tuning the integral mode. Bottom left: CO-controller with linear table and the equivalent PID controller. Bottom right: CO-controller with non-linear lookup table.

TABLE II.
Non-linear lookup table used in Fig. 3, bottom right.

		CHANGE IN ERROR											
	-6	-5	-4	-3	-2	1	0	1	2	3	4	5	6
-6	-7	-7	-7	-7	-7	-7	-7	-7	-4	-4	-4	-6	0
-5	-6	-6	-6	-6	-6	-6	-6	-5	-5	-4	-4	-6	0
-4	-6	-6	-6	-6	-6	-6	-6	-4	-4	-4	-4	-6	0
-3	-5	-5	-4	-4	-4	-4	-4	-4	-4	-4	-4	-5	0
E -2	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	4	4
R -1	-4	-4	-4	-4	-4	-4	-4	2	2	3	4	4	4
R 0	-4	-4	-4	-3	-1	-1	0	1	1	3	4	4	4
O 1	-4	-4	-4	-3	-1	-1	4	4	4	4	4	4	4
R 2	-4	-4	4	4	4	4	4	4	4	4	4	4	4
3	0	6	4	4	4	4	4	4	4	4	4	6	6
4	0	6	4	4	4	4	6	6	6	6	6	6	6
5	0	6	4	4	6	6	7	7	7	7	7	7	7
6	0	4	4	4	7	7	7	7	7	7	7	7	7

The second series of experiments (Fig. 3, bottom right) uses a non-linear lookup table for the CO-controller, Table II. The table is Mamdani's table, [4], but modified according to the modified definitions of E and CE. Even this controller shows proportional and integral behaviour. The performance seems worse, but that is not surprising, since the table was designed for another process.

CONCLUSION

The new tuning-rules have in fact cut the time the students need to tune the laboratory water tank rig, from one hour to 10 - 15 minutes. The CO-controller is a little harder to tune, than the O-controller and the fuzzy PID controller proposed in Appendix, because its integral action cannot be removed completely. In that respect, it is easier to use the fuzzy PID configuration; here the integral and derivative gains can be put to zero independently and at will.

A future study will try and transfer other PID tuning-rules, for example the Ziegler-Nichols recommendations, using the established link between fuzzy and PID controllers.

LITERATURE

[1] Astrom, K.J.: "Reglerteknik: en elementaer introduktion", (Automatic Control: an Elementary Introduction). Lund's Technical University: Institutionen for Reglerteknik, 1982, 5.1-5.46 (chap. 5: "PID reglering").

[2] Holmblad, L.P & J.-J. Ostergard: "Control of a Cement Kiln by Fuzzy Logic." Fuzzy Inf. Decision Processes, 1982, 389-399.

[3] Jespersen, T.: "Self-organizing Fuzzy Logic Control". Technical University of Denmark: Electric Power Engineering Department, report no 8107, 1981.

[4] Mamdani, E.H.: "Application of Fuzzy Logic to Approximate Reasoning". IEEE Trans Computers, 1977, c-26(12), 1182 - 1191.

[5] Pedrycz, W.: "Fuzzy Control and Fuzzy Systems". New York and Toronto: Wiley, 1989.

[6] Tang, K.L. & R.J. Mulholland: "Comparing Fuzzy Logic with Classical Controller Designs". IEEE Trans on Systems, Man, and Cybernetics, 1987, SMC-17(6), 1085-1087.

[7] Smith, L.C.: "Fundamentals of Control Theory". Chemical Engineering, Deskbook Issue, 1979, 86 (22), 11-39.

[8] Yamazaki, T.: "An Improved Algorithm for a Self-Organising Controller, and its Experimental Analysis." Queen Mary College, London: Dept. of Electrical and Electronic Engineering, 1982 (Ph.D. thesis).

[9] Zadeh, L.: "Outline of a new Approach to the Analysis of Complex Systems and Decision Processes." IEEE Trans Smc, 1973, 3, 28-44.

APPENDIX: Analogies between PID and fuzzy controllers

This appendix establishes the mathematical relationships between fuzzy and PID (proportional-integral-derivative) controllers.

A discrete version of a PID controller can be written

$$O(n)=Kp*[e(n)+(1/Ti)*\sum_{0}^{n}ej*Ts+Td*(e(n)-e(n-1))/Ts] \qquad (1)$$

85

where O is the controller output, index n is the time instant, Kp is the proportional gain, and e is the setpoint error. The Ti, Td and Ts are integral, derivative and sampling time respectively.

The fuzzy CO-controller produces an incremental change in output, CO, which is a function F of the inputs to the controller (Fig. 1)

$$CO(n)=GCO*F(E(n),CE(n)) \qquad (2)$$

The function F includes clipping, quantization and a lookup in a table. Replacing F by a summation, one can turn eq. (2) into a form similar to eq. (1)

$$
\begin{aligned}
CO(n) &= GCO*[E(n) + CE(n)] \\
&= GCO*[GE*e(n) + GCE*ce(n)] \\
&= GCO*[GE*e(n) + GCE*(e(n) - e(n-1)] \\
&= GCO*GCE*[GE*e(n)*Ts/GCE*Ts+e(n)-e(n-1)] \quad (3)
\end{aligned}
$$

Integrating the left hand side and the right hand side provides

$$O(n)=GCO*GCE*[(GE/(GCE*Ts))*\textstyle\sum_{1}^{n}e(j)*Ts+e(n)] \qquad (4)$$

Comparing terms in eqs. (1) and (4), it is clear that GCO*GCE is equivalent to Kp, and GE/(GCE*Ts) is equivalent to 1/Ti. Consequently, the fuzzy CO-controller is similar to a PI controller.

The fuzzy O-controller produces a control signal, also as a result of a function F

$$O(n) =GO*F[E(n),CE(n)] \qquad (5)$$

Replacing F by summation again, yields

$$
\begin{aligned}
O(n) &= GO*[E(n) + CE(n)] \\
&= GO*[GE*e(n)+GCE*ce(n)] \\
&= GE*GO*[e(n)+(GCE/GE)*ce(n)] \\
&= GE*GO*[e(n)+(GCE*Ts/GE)*(e(n)-e(n-1))/Ts] \quad (6)
\end{aligned}
$$

Comparing terms with eq. (1), one sees that GE*GO is equivalent to Kp, and GCE*Ts/GE is equivalent to Td, i.e. the fuzzy O-controller is similar to a PD controller.

To get a fuzzy controller similar to a PID controller, one could add a third input, integral error IE, to the O-controller configuration. If we define

$$IE(n) = GIE*ie(n) = GIE*\textstyle\sum_{0}^{n}e*Ts,$$

$$ce(n) = (e(n) - e(n-1))/Ts$$

the fuzzy PID controller is,

$$O(n) = GO*F[E(n), CE(n), IE(n)] \qquad (7)$$

The function F includes a lookup in a three-dimensional table.

Replacing F by summation yields

$$O(n) =$$

$$GO*[E(n) + CE(n) + IE(n)] =$$

$$GO*[GE*e(n) + GCE*ce(n) + GIE*ie(n)] =$$

$$GO*[GE*e(n)+GCE*(e(n)-e(n-1))/Ts+GIE*\textstyle\sum_{0}^{n}e*Ts] =$$

$$GO*GE*[e(n)+GIE/GE*\textstyle\sum_{0}^{n}e*Ts+GCE/GE*(e(n)-e(n-1))/Ts]$$

$$(8)$$

Comparing terms with eq. (1), one sees that GO*GE is equivalent to Kp, GIE/GE is equivalent to 1/Ti, and GCE/GE is equivalent to Td, i.e. a PID controller.