A KNOWLEDGE-BASED APPROACH IN FUZZY CONTROLLER DESIGN

Cristian Gh. BOLDISOR, Vasile V. COMNAC, Simona V.P. COMAN

Department of Automation, Transilvania University of Braşov; cristiboldisor@yahoo.com

Abstract **– A fuzzy logic controller (FLC) provides a means of converting a linguistic control strategy based on expert knowledge into an automatic control strategy. A short survey of the FLC rule-base design strategies is presented, focusing on a control engineering knowledge based methodology. A fuzzy controller is designed for a temperature control application to test the proposed approach. System performances and method's viability are analysed.**

Keywords: *fuzzy logic controller, knowledge-based design; temperature fuzzy control system.*

1. INTRODUCTION

Although a large number of algorithms have been proposed so far, it is still hard to say there are some general, all accepted methods for designing fuzzy controllers and for finding their optimal parameters. Anyway, based on experience and the huge amount of literature available, a set of suggestions can set bounds to an initial, somehow standard approach [1] that will result in obtaining a controller by knowing few details about the process. Such methodology should be able to build at least a rough controller, which will be subsequently improved to satisfy higher performances (if required).
As often mentioned in literature, there are at least four main sources for finding and/or fine tuning control rules of either an initial controller or an improved one:
- Based on expert experience and control engineering knowledge [2],[3].
- Based on the operator's control actions and designer's knowledge [2],[3].
- Based on a fuzzy model of the plant (if available) or fuzzy identification [4].
- Based on learning, neural networks or genetic algorithms [3], [5].

There are a lot of practical strategies, from the simplest "try-and-see" to high complexity algorithms. However, when it comes to practical implementation, the main advantages of fuzzy control should be considered, along with performances, even during design stage. Hence, design must be guided by the often mentioned reasons for using fuzzy logic controllers:
- Fuzzy logic controllers are an efficient method to control ill-defined processes, since no explicit or accurate mathematical models of the process are needed to design the controller [2], [5].
- Fuzzy controllers are more robust than PID controllers because they can cover a much wider range of operating conditions, and it can operate with disturbances of different natures [1], [6].
- Control strategy consists of if-then rules, built from a usual vocabulary containing everyday words, which make it easy to read. Plant operators can embed their experience directly and controllers are easily customisable [6].
- Fuzzy logic enables non-specialists to design control systems, since it focus on linguistic knowledge rather than complex mathematical equations. It is easy to learn how fuzzy controllers work and how to design them to a concrete application [6].
- Developing fuzzy controllers is cheaper than developing model-based or other controllers to do the same thing.

2. A KNOWLEDGE-BASED FUZZY CONTROLLER DESIGN

When based on expert knowledge, fuzzy controller design problem has at least four steps: a) choose control system structure and controller type (P, PI, PD, PID), which determines the input and output variables; b) set universes and fuzzy sets for each fuzzy variable; c) set the control rules; d) set the scaling gains for measured crisp variables. We will shortly describe these steps.
The classical control system is having the controller located before the process and computes the control actions from the error values by applying an algorithm. Most fuzzy control applications consider as *inputs* of the fuzzy inference system both the *error* and the *derivative error* (also named *change-in-error*). The

reason is that not only the error value is important in taking a control action, but also the way that value changes. The *output* of the fuzzy controller is either the *control* action or its derivative value (*change-in-command*).

Control applications of fuzzy logic accommodates better with Sugeno-Tagaki inference system (than Mamdani). The main reason is that the defuzzyfication is easier and faster.

Several available commercial controllers use *standard universes* for input and/or output variables [1]. There are mainly two types of standard universes: a) the symmetrical range $[-L;+L]$ and b) the asymmetrical range $[0;+L]$. Using one or the other depends on the actual application, but for control systems the first type is more appropriate, since error variable can be negative. The values of this range limits depend first on implementation of the fuzzy controller.

For every input variable a family of *attributes* (*linguistic terms*) are defined. The choice of shape and width of all fuzzy sets is subjective; however there are a few rules [1], [2]:

- A term set should be sufficiently wide to allow for noise in the measurement;
- A certain overlap is desirable to avoid poorly defined states.
- A gap between two neighbouring sets must be avoided, because in the gap no control action is defined.

Usually, an odd number of attributes are chosen for input variables. The linguistic terms are simple fuzzy sets, with triangular or trapezoidal membership function, for an easy implementation on numerical devices.

In control applications, the most relevant linguistic term is the one of error variable that is representing zero. It strongly affects the steady-state performance. Simulations prove that for $e_{st,\max}$, maximum accepted steady-state error, the zero attribute must be defined within $[-e_{st,\max};+e_{st,\max}]$.

Linguistic terms of output variable are chosen uniformly distributed over the variable's universe and they are singletons. It is also important to have singletons at the limits of output variable's universe.

For setting a rule-base, let's first analyse a system known to have optimal transient response. Control theory presents the classical problem of a first order element with the K_p gain and the T_p time constant, for which optimal performance is obtained using an integrator as controller. The closed-loop system is a second-order system, with the step response similar to that shown in Fig. 1. The figure also shows the error signal.

By examining the error, we can express a relation between it and the control output that forces the system's output to follow the reference. Hence, we can say:

if *error* ***is*** *positive_big* ***then*** *command* ***is*** *positive_big*

or maybe more suggestive

… command ***should be*** *positive_big*

The same reasoning will produce more rules, e.g.:

if *error* ***is*** *positive_small* ***then*** *command* ***is*** *positive_small*

The rule base is subsequently refined by considering the derivative error, e.g.:

if *error* ***is*** *positive_small* ***and*** *derivative error* ***is*** *negative* ***then*** *command* ***is*** *positive_small*

Techniques to tune the *scaling gains*, offline or online (real time) have received the highest priority in literature due to the influence of the gains on the performance and stability of the system. First, the measured variables must be scaled to/from the chosen universe of discourse (for input and output variables). Second, there are many methods for further fine tuning of scaling gains.

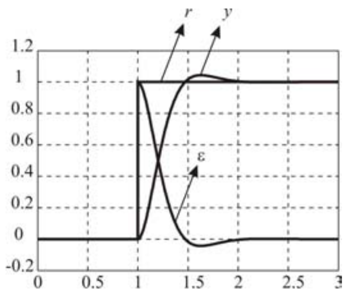

Figure 1: An optimal control system

Let's assume that for all variables we set a $[-L;+L]$ range. For error variable, the maximum possible value is measured at step time (see Fig. 1), and that value is

$$e_{\max} = r(t) - y(t)\big|_{t=step\ time} \quad (1)$$

That value corresponds to the positive limit of error's universe, $+L_e$. So we can set the scaling factor for error variable:

$$K_e = \frac{L_e}{e_{\max}} \quad (2)$$

and for derivative error:

$$K_{de} = \frac{L_{de}}{de_{\max}} = \frac{L_{de}}{e_{\max}}. \quad (3)$$

The maximum change-in-error is no larger than the maximum error: $de_{\max} \le e_{\max}$.

As considered above, $e_{\max}$ depends on reference signal. Changing the reference will cause an undesirable K_e value. A more general and still easy method would be:

$$K_e(t) = \frac{L_e}{r(t)}, \quad (2a)$$

$$K_{de}(t) = \frac{L_{de}}{r(t)} . \quad (3a)$$

which means that the gains are adjusted online by considering at any time the actual reference value as the maximum possible value of error. Hence, the scaling gains are not constants, but time-variant functions. The advantage is that changing reference's type or parameters would not lead to change the controller's input gains (and so avoid controller's redesign).

Another gains tuning approach consists in designing a classic PID controller, then to replace it with a fuzzy PID controller.

The scaling gain for controller's output is a gain factor "before" the process, so it acts like the proportional action of a PID controller.

Inspired by Ziegler-Nichols formula for the gain of a classical PID, we can calculate the scaling gain for output:

$$K_u = \frac{1}{K_p}\frac{T_p}{\tau} , \quad (4)$$

where: K_p is the plant's gain, T_p is the main time constant, τ is the dead-time.

It is worth mentioning here that the output gain will compensate plant's gain, time constant and dead-time, which is the reason for using this equation. Also, we have to mention that, usually T_p / τ is large enough to assure that for the limit of output variable's universe, L_u, the value of control action is the maximum technically possible value, $u_{\max}$.

The above presented strategy leads to the fuzzy control system depicted in Fig. 2.

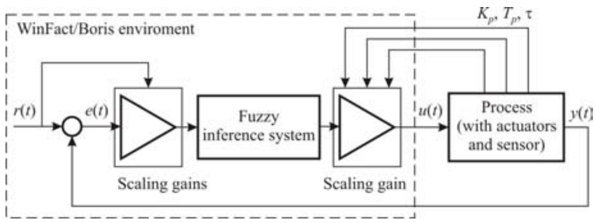

Figure 2: Proposed knowledge-based fuzzy control system

3. EXPERIMENT DESCRIPTION

The process subjected to fuzzy control in our experiment is a thermal process, shown in Fig. 3. The plant contains a 24[V] halogen lamp (1) that heats a small radiator (2) and a fan (3) feeding a uniform air jet that cools the radiator. A sensor (4) is used to measure the temperature and to set a corresponding voltage signal, with the ratio 1[V] = 10[ºC]: $K_{ratio} = 0.1$.

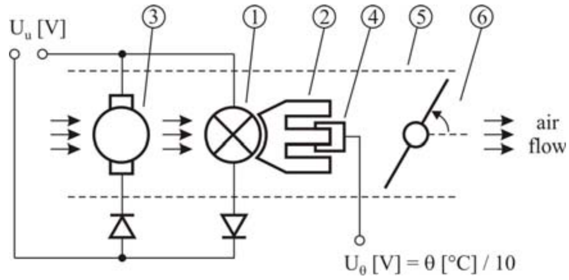

Figure 3. Process description

Air jet flows over the radiator through a notch (5) that can be blocked by an adjustable flap (6). Temperature is raised by applying a voltage on a halogen lamp and lowered by applying a voltage on the fan. The voltages are generated by two power amplifiers that work separately and exclusively, based on command signals from the controller.

The designed control system contains a data acquisition module (Profi-Cassy) and the Boris/WinFact software

The Profi-Cassy module has two inputs, to measure $[-10;+10]$ [V] signals, and two outputs, to generate $[-10;+10]$ [V] signals. Inputs and outputs work simultaneously and independent. Boris/WinFact is a simulation and real-time control environment that can be used for simulating control systems or for running as a digital controller, with Profi-Cassy as analogic-numerical interface.

The fuzzy inference system is Sugeno-Takagi type, having two inputs, error and derivative error, and one output, control action. The universe of each variable is the interval $[-1;+1]$:

$$L_e = L_{de} = L_u = 1$$

The variables' universes and linguistic terms are presented in table 1.

Table 1

Error: $[-1;+1]$	
NegBig	trapmf(-1, -1, -0.1, -0.01)
NegSmall	trimf(-0.1, -0.01, 0)
Zero	trimf(-0.01, 0, 0.01)
PosSmall	trimf(0, 0.01, 0.1)
PosBig	trapmf(0.01, 0.1, 1, 1)
Derivative error: $[-1;+1]$	
Neg	trimf(-1, -1, 0)
Zero	trimf(-1, 0, 1)
Big	trimf(0, 1, 1)
Control output: $[-1;+1]$	
NegBig	-1
NegSmall	-0.5
Zero	0
PosSmall	0.5
PosBig	1
trapmf = trapezoidal membership function *trimf* = triangular membership function	

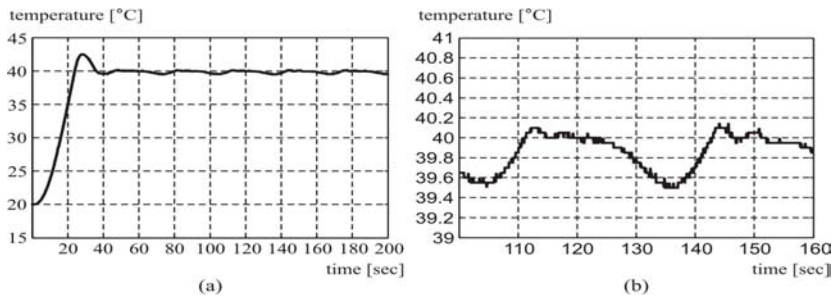

Figure 4: Recorded temperature: (a) transient response when changing reference from 20 ºC to 40 ºC; (b) detail of steady-state conditions

The rule-base is obtained by examining the second-order system known to have optimal transient response, and without considering the plants model (table 2).

Table 2

Error	Derivative error		
	Neg	Zero	Big
NegBig	-1	-1	-1
NegSmall	-1	-0.5	-0.5
Zero	-0.5	0	0.5
PosSmall	0.5	0.5	1
PosBig	1	1	1

4. RESULTS

In Boris control scheme, we set the reference temperature value to $r(t) = \theta^* = 40[°\text{C}]$.

Previously, we recorded step response of the open-loop system to in order measure process gain, dead-time, and time constant. The values are:

$$K_p = 1.2,\ T_p = 120\,\text{sec},\ \tau = 10\,\text{sec},$$

and the calculated scaling gains are:

$$K_e(t) = \frac{L_e}{r(t)K_{ratio}} = \frac{1}{4},\ K_{de} = \frac{L_{de}}{r(t)K_{ratio}} = \frac{1}{4},$$

$$K_u = \frac{1}{K_p}\frac{T_p}{\tau} = \frac{1}{1.2}\frac{120}{10} = 10.$$

With these scaling gains applied to the proposed fuzzy controller, the obtained performances for the control system are:

- in steady-state conditions, temperature slowly oscillates around 39.8 [ºC], which means that $e_{st} \approx 0.5\% (< L_e[\%] = 1\%)$;
- for transient response, overshoot is close to an optimal value: $\sigma \approx 6.25\%$;
- settling time is $t_s \approx 35[\text{sec}]$, satisfactory for a process with large time constant.

Figure 4 shows the measured output temperature of the process.

5. CONCLUSIONS

A control engineering knowledge based strategy for fuzzy controllers design is successfully used in a considerable number of control applications. The methodology follows some general guidelines mentioned in today available literature. Knowledge can reveal a set of if-then rules to describe the controller's correct actions for achieving performances. Furthermore, this approach follows the main reasons of fuzzy logic control, among which we mention: process modelling is not necessary and little experience is needed for a satisfactory control system.

The knowledge based controller was tested with success on a temperature control system. The temperature control is one of the most suitable applications of fuzzy control, since thermal processes have complex or ill-defined models, and are often affected by disturbances.

References

[1] Preitl, Ş., Precup, R.E. *Introducere în conducerea fuzzy a proceselor* (*Introduction to Fuzzy Control*), ed. Tehnică, Bucureşti, 1997.

[2] Jantzen, J. *Foundations of Fuzzy Control*, John Wiley & Sons, 2007.

[3] Zimmermann, H.J. (ed.), *Practical Applications of Fuzzy Technologies*, Springer, 1999.

[4] Lee, C.C. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I and Part II*, IEEE Transactions on Systems, Man and Cybernetics, vol. 20 (2), pp. 404-435, 1990.

[5] Nie, J., Linkens, D. Fuzzy-Neural Control: Principles, Algorithms and Applications, Prentice Hall, 1995.

[6] Reznik, L. *Fuzzy Controllers*, Newnes, 1997.

[7] Sala, A., Guerrab, T.M., Babuška, R. *Perspectives of fuzzy systems and control*, IEEE Fuzzy Sets and Systems vol. 156 (3), 2005, pp. 432-444.