Chapter 2 Fuzzy Logic Controllers – Preliminaries

Introduction

Regarding automatic control principles, which are in continuous development, some important trends are apparent. The first one is in strongly application-oriented studies, dedicated to controllers, mainly in the PID type. For instance, the Ziegler-Nichols criterion for their adjustment is a famous and well-known result in the field of automatic control. Since then, more adcanced controllers have appeared, and more refinement of the theoretical tools has been needed for their succesful design and fruitful utilisation.

As usual in any engineering field, one may observe a variety of novel and deeply theoretical approaches that have made it possible to formulate and solve more mature and abstract problems in a general setting. Such milestones are, for instance, the state-space approach, stochastic methods, optimal control, filtering, and/or state estimation.

Some proposals looked academic, but rapid developments in technology made their realisation easy. Therefore, some problems concerned with the numerical complexity of control/identification algorithms no longer exist. However, there are still some limitations. These concern the amount and character of the information that has to be gathered to ensure efficient performance of the controller. In its absence, one can expect a poor performance of the sytem under control. The realm of model identification, for the design of control algorithms, has at its disposal tools based on advanced methods of statistics, experimental design, and multivariable nonlinear function optimisation. What is more usual these days is that the models constructed are claimed to be more precise, covering a specifity of the system controlled, which leads to more complex model equations, with many parameters to be estimated. Thus, we are near to the so-called "sin of overmodelling" []. Moreover, there is a trend towards the "mathematisation" of control problems, as mentioned in the Zadeh's papers [].

In practice, however, the human being can take reasonable control action, even in time-varying conditions of the process, their nonlinearities, and existing disturbances.

All these reasons make this approach an attractive one. A general scheme in which this computerised control protocol, called fuzzy controller, works is depicted in Figure 2.1. Here, its inputs are constituted by measures of satisfaction of performance indices that are optimised by the human operator. They are not given explicitly, and the structure presents only the conceptual links.

Nevertheless, several problems in the implementation of this control algorithm have to be solved. First, one needs a control protocol reflecting the main properties of the control strategy. Secondly, for efficiency, one needs a tool that is flexible enough to cope with a linguistic categiries in the control protocol and precise enough to allow its computer implementation. Fuzzy sets offer a convenient formal apparatus to meet booth requirements.

The original activites to develop a control system based on fuzzy logic theory, (fuzzy logic controller, FLC) centered around Qeen Mary College in London. Key to that development was the work of E. Mamdani and his students in the Department of Electrical and Electronic Engineering. Richard Tong, of nearby Cambridge, was another key figure in the development of fuzzy logic control theory. The first application of fuzzy set theory to the control of systems was by Mamdani and Assilian [], who reported on the control of a laboratory model steam engine. Recent applications of fuzzy control [] have pointed a way for an effective utilization of fuzzy control in the context of

complex ill-defined processes that can be controlled by a skilled human operator without the knowledge of their underlying dinamics.

2. 1. Basic Consruction of the Fuzzy Logic Controller

In control engineering, fuzzy controllers are extensively studied. Case studies present their application in various process-control systems previously controlled manually. One has to understand clearly the notation of "fuzziness" in technical systems (electrical, mechanical, chemeical, etc.) and its role in information processing. At first glance, it may seem artificial to apply fuzzy sets in such systems. But such a judgment is not valid. In control strategy applied by the operator, we can recognise some concepts that are evidently fuzzy in their nature, for instance the goals of the control (e.g. 'keep *close* to the desired trajectory with a *quite low* control effort' contains the vague expressions *close*, *quite low*).

The FLC is a tool for processing a fuzzy form of information in a nonfuzzy or fuzzy scheme of reasoning. Here is one application of fuzzy controllers in the so-called 'soft' sciences (i.e. where the human being plays a central role): a model of organisational behaviour using linguistic rules describing relationships at a mining company [].

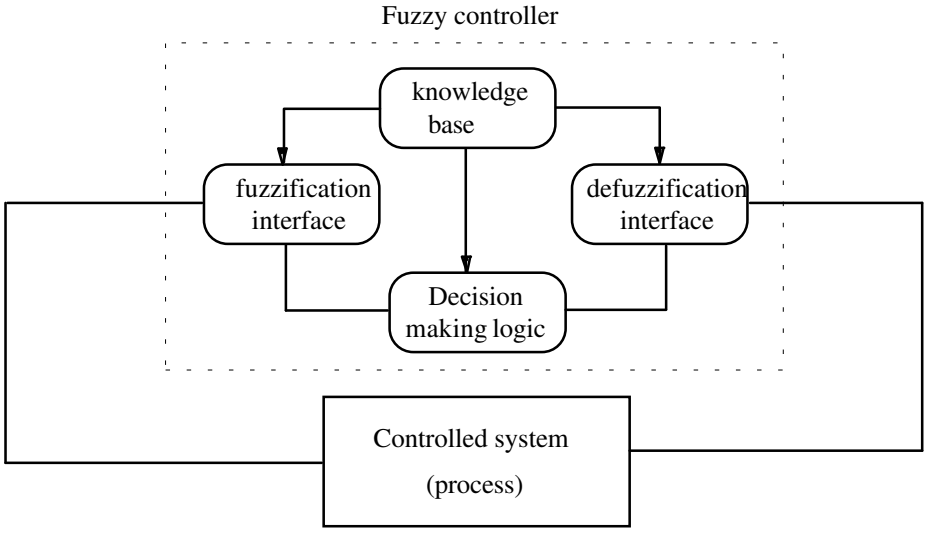

Figure 2-1. Basic configuration of fuzzy logic controller

A main, if not unique source of knowledge to cunstruct the control algorithm comes from the control protocol of the human operator. The protocol consists of a set of conditional 'if-then' statements, where the first part of each contains a so-called condition (antecedent) while the second (consequent) part deals with an action (control) that has to be taken. Therefore, it conveys the human strategy, expressing which control is to be realised when a certain state of the process controlled is observed.

2.1.1 A model of organisational behaviour using linguistic rules describing relationships at a mining company

An example from a textbook for cement-kiln operators [] deals with the control rules listed below:

Case	Condition Part	Action to be Realised	Reason
11	BZ OK OX low BE OK	Decrease fuel rate slightly	To raise percentage of oxygene
12	BZ OK OX low BE high	(a) Reduce fuel (b) Reduce ID fan speed	To increase percentage of oxygene to activate (b) to lower back-end temperature and maintain burning-zone temperature
17	BZ OK OX low	Reduce ID speed slyghtly	To lower percentage of oxygene

The rules indicate how the operator has to adjust the fuel rate, the kiln speed, and the temperature in the burning zone (BZ) of the kiln, the oxigen (OX) percentage in the exhaust gases, and the temperature at the back-end (BE) of the kiln. The remarkable fact is that the rules contain the condition as well as the action part (for instance, OX *high*, OX *low*, reduce ID fan speed *slightly*) of the linguistic terms that reflect the operator's knowledge of the process. Also, they give a clear impression of the level of precision at which one has to perform computations in order to mimic a human's behaviour. The linguistic terms suggest work with sets rather than with single numerical quantities. Hence, it is not suprising, that fuzzy sets came into this area of investigation.

While building the linguistic protocols for process, two main types of question are relewant in the construction of the fuzzy controller []:

– questions abaut the operator's own behaviour: e.g. what would you do in such-and-such situation?

– questions abaut the behaviour of the process: why does such-and-such a situation occur?

In general, as stated in [], a verbal protocol analysis is a fruitful technique for modelling human action in process control. The main aspects to be taken into account are:

– chracteristic of human behaviour,

– development of process-controll skills,

– individual differences between process operators,

– task-factors affecting performance,

– organisation of the operator's control behaviour.

We now examine the basic construction of the fuzzy controller: a formal construction dedicated to coding the linguistic protocol and equipped with inference capabilities.

We focus on single-output: single-output process in which control is determined in the light of the satisfaction of two criteria relating to two variables, namely error and change of error, where error is the difference between a specified set point and the actual output value.

For error, change of error, and control, linguistic values are encoded into fuzzy sets, the so-called reference fuzzy sets. These values are used by the operator to describe the control procedure. More precisely, we denote them by fuzzy sets E_1, E_2, ..., E_n, E_1, DE_1, DE_2, ..., DE_m, and U_1, U_2, ..., U_n, respectively. They are defined in the universe of discourse *E*, *DE*, and *U*, respectively. Links existing betweeen the linguistic values are, as mentioned above, given in the form of conditional statements (rules):

$$\textbf{If } (E_i \textit{ and } DE_i) \textbf{ then } U_i$$

i = 1, 2, ..., N, where N denotes the number of the rules. E_i, DE_i, and U_i are fuzzy sets taken from the collection of the reference fuzzy sets alredy introduced.

A set of conditional statements to control a single-input: single-input process is shown i the Table below [].

Error	Change of Error						
	NB	NM	NS	Z	PS	PM	PB
PB	-	-	NM	NB	NB	NB	NB
PM	-	-	NM	NB	NB	NB	NB
PS	PS	PS	NO	NB	NB	NB	NB
PO	PM	PM	PS	NO	NS	MN	MN
NO	PM	PM	PS	NO	NS	NM	NM
NS	PB	PB	PM	PM	NO	NS	NS
NM	PB	PB	PB	PB	PM	-	-
NB	PB	PB	PB	PB	PM	-	-

where all the entries of the matrix are fuzzy sets of error, change of error, and control. The abbreviations used above mean:

PB:	Positive Big
PM:	Positive Medium
PS:	Positive small
PO:	Positive Zero
NO:	Negative Zero
NS:	Negative Small
NM:	Negative Medium
NB:	Negative Big

From the table, we can deduce the control strategy:

- if error and change of error are zero (in the sense of the linguistic labels, i.e. PO, NO), the control applied is Negative Zero (also in the sense of the respective linguistic label of control)
- if error is Positive Big or Medium and the same as the change of error, we have to use a Negative Big Control to reach the state in which error and change of error are reduced to zero
- if error is Negative Big (or Medium) and the same holds for the change of error, the control is set to the greatest positive value (i.e. Positive Big) to ensure that they are reduced significantly.

For the above example, all three spaces, *E*, *DE*, and *U* are finite, thus

$E = \{e_1, e_2, ..., e_{n1}\}$, $DE = \{de_1, de_2, ..., de_{n2}\}$, $U = \{u_1, u_2, ..., u_{n3}\}$.

2.1.2 Steps of constructig fuzzy controller

Early studies [] showed the construction of the fuzzy controller to be a seqence of the next steps:

1. Development of fuzzfication interface which involves the following functions:
   (i) Measures the values of input variables,
   (ii) performes a scale mapping that transfers the range of values of input variables into corresponding universe of discourse,
   (iii) performs the function of fuzzification, that is, converts input data into suitable linguistic values which may be viewed as labels of fuzzy sets.
2. Construction of the knowledge base, which comprises a knowledge of the application domain and the attendant control goals. It consists of a "data base" and a "linguistic (fuzzy) control rule base":
   (i) data base provides necessary definitions, which are used to define linguistic control rules and fuzzy data manipulation on FLC,
   (ii) the rule base characterizes the control goals and control policy of the domain experts by means of a set linguistic control rules.
3. Development of decision making logic, which is the kernel of the FLC; it has the capability of simulating human decision naking based on fuzzy concepts and inferring fuzzy control actions employing fuzzy implication and the rules of inference in fuzzy logic.
4. Constructing the defuzzification interface performing the following functions:
   (i) a scale mapping which converts the range of values of outut variables into corresponding universe of discourse.

(ii) defuzzification, which yields a nonfuzzy control action from an inferred fuzzy control action.

For the above mentioned example, which is a SISO (single-input-single-ouput) system in general, the representation of the knowledge contained in the rules can be written as follows:

if E_1 *and* DE_1 **then** U_1

or else

if E_2 *and* DE_2 **then** U_2

or else

.
.
.

or else

if E_N *and* DE_N **then** U_N

that is, in the form of the fuzzy relation that combines all the rules:

$$R = R_1 \cup R_2 \cup ... \cup R_N = \bigcup_{i=1}^{N} (E_i \times DE_i \times U_i), \qquad (2.1)$$

Note that the union of R_is reflects the modelling of 'or else', the conjunction standing in the control protocol. Also, the Cartesian product is mainly considered to be represented by the min operator, however, one can expect any t-norm. For a discussion on a suitable choice of the operator, we refer to a presentation below. Thus, in virtue of (2.1), the fuzzy relation R has the membership function:

$$R(e_1, de_j, u_k) = \max_{1 \le i \le N} [E_1(e_1) \wedge DE_i(de_j) \wedge U_i(u_k)]. \qquad (2.2)$$

The fuzzy control obtained via a compositional rule of inerence:

$$U = (E \times DE) \circ R \qquad (2.3)$$

Inserting the values of the membership function of E and DE as well as R, one has:

$$U(u_k) = \bigvee_{e \in E,\ de \in DE} [E(e_1) \wedge DE(de_j) \wedge R(e_1, de_j, u_k)]$$

The output of the above mentioned controller, too, is a fuzzy set of control. The are several ways of tackling the problem of defuzzification, reviewed below.

2.1.3. Some Defuzzification Methods

2.1.3.1. Max procedure

The first and simplest method relies on choosing the control value for which the membership function attains a maximum, namely:

$$\max_{u \in U} U(u) = U(u_0)$$

A difficulty arises when more than one element of U possesses this maximal value and thuse u_0 is not uniquelly determined. For instance, one can take randomly one of the elements with the highest grade of membership, ot treat their avarage value as the representative one (mean-maxima approach), thus:

$$u_0 = \sum_{i=1}^{r} \frac{u^i}{r}$$

where

$$u^i \in \{u \in U | \max_{u \in U} U(u) = U(u^i)\}$$

and this set consists of 'r' elements. The above mentioned algorithm does not consider the shape of the membership function of control. This information is completely lost, which in many situations may simplify things. As a consequence, we may expect u_0, as a function $u_0 = g(x)$ of the nonfuzzy input of x of the controller, to be discontinuous. As eill be shown later, $g(x)$ has a relay-like function. The shapes of the membership fuunction of the fuzzy inputs of the controller are not significant; only their points of intersection have an inference on u_0.

2.1.3.2. Centre-of gravity procedure

The centre of gravity of the fuzzy control U is:

$$u_0 = \frac{\int_U u \cdot U(u)}{\int_U U(u)}$$

(we assume that both the integrals exist). Here, u_0 is determinated bearing in mind a surface of the fuzzy set U. In comparison to the max-procedure, we observe, that $u_0 = g(x)$ has no switching point, but a smooth transition between the control values for different x is visible.

2.1.3.2. Indexed maximum or centre-of gravity method

We modify both methods so that u_0 is computed according to:

$$u_0 = \arg \max_{u \in U(\alpha)} U(u)$$

$$u_0 = \frac{\int_{U(\alpha)} u \cdot U(u)}{\int_{U(\alpha)} U(u)}$$

with $u(\alpha) = u \in U | U(u) \geq \alpha\}$

where α forms a parameter varying as [0, 1]. The idea is that the threshold level α allows us to eliminate in the computation of u_0 those elements of the membership function that have nonsignificant grades f membership (lower then α), which may cause biased results. This threshold has to be determined with respect to disturbances in the input data and the properties of the process controlled. For instance, if no bang-bang (relay-type) control is accepted, α should be low.

To illustrate, let us consider a simple computation example of a fuzzy set of control, with triangular membership function (Figure 2-2). Note that it is asymmetrical abaut its highest value, so that we can observe its influence on the value of u_0. This phenomenon can also be seen in figure 2-2.

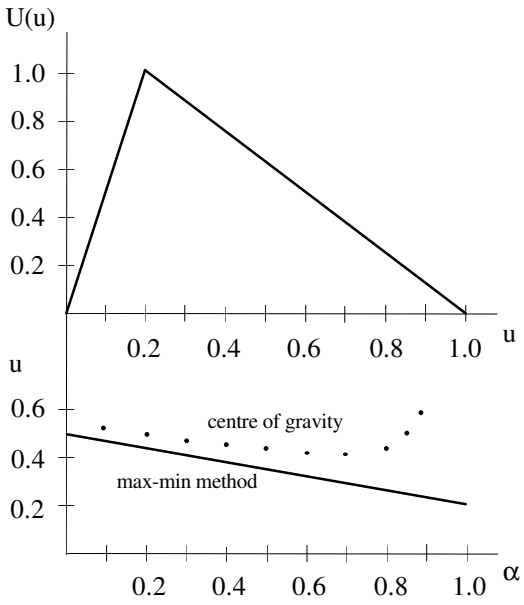

Figure 2.2.
Membersip function of a fuzzy set of control (above) and
its nonfuzzy representation {(2.9), (2.1)} for different values of α (below).

Returning to the general presentation of the fuzzy controller, we take a more detailed look in figure 2-3. The detailed structure of the fuzzy controller contains also scaling factors, *CE, CDE*, and *CU*. So the input and output of the controller are changed proportionally. Hence, the error e' and change of error de' entering the controller are modified accordingly:

$$e = CEe', \quad de = CDEde' \quad \text{and } u = CUu'.$$

Their role is to tune the fuzzy controller to obtain the desired dynamic properties of the process controll loop.

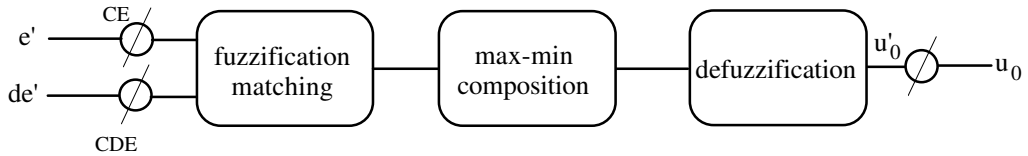

Figure 2.3. The functional blocks of fuzzy controller

From the structure of the controller, one can easily see thet the fuzzy relation requires a lot of memory space to be saved. Recalling n_1, n_2 and n_3, the numbers of the elements of the universes of discourse, R needs storage $n_1*n_2*n_3$ elements. Hence, more event a moderate size of *E*, *DE* and *U*, this might be a problem. For $n_1 = n_2 = n_3 =10$, the fuzzy relation consists of 1,000 elements. Thus, for implementation, it is important to look for another way of calculating the fuzzy control U that avoids forming the fuzzy relation of the controller.

Fortunately, for the fuzzy relation and the composition operator, this problem can be solved easily. The fuzzy relation R does not have to be stored explicitly. We show the result collected in terms of:

Proposition 2-1.

For a fuzzy controller with a fuzzy relation given by (2.2) and a rule of computation specified by (2.3), the fuzzy control U is a fuzzy set in *U* with the membership function:

$$U = \bigcup_{i=1}^{N} (\Lambda_i \cap U_i) \tag{2.4}$$

where Λ_i is a fuzzy set in *U* with a constant membership function equal to:

$$\Lambda_i(u) = \max_{e\in E,\ de\in DE} [E_i(e) \wedge DE_i(de) \wedge E(e) \wedge DE(de)].$$

In other words, (2.4) becomes:

$$U(u) = \bigvee_{1\le i\le N} [\Pi(E|\ E_i) \wedge \Pi(DE|\ DE_i) \wedge U_i(u)].$$

The proof results from straghtforward computations:

$$\begin{aligned} U(u) &= \bigvee_{1\le i\le N} \{ \max_{e\in E,\ de\in DE} [E(e) \wedge DE(de) \wedge E_i(e) \wedge DE_i(de)] \wedge U_i(u)\} = \\ &= \bigvee_{1\le i\le N} \{ \max_{e\in E,\ de\in DE} [E(e) \wedge E_i(e) \wedge DE(de) \wedge DE_i(de)] \wedge U_i(u)\} = \\ &= \bigvee_{1\le i\le N} [\Pi(E|\ E_i) \wedge \Pi(DE|\ DE_i) \wedge U_i(u)] \end{aligned} \tag{2.5}$$

holding for each $u \in U$, which completes the proof.

Fuzzy set Λ_i has an interesting interpretation: its value of membership is simply a minimum possibility measure of the input fuzzy sets of error and change of error, taken with respect to the antacedent in the lth rule.

Thus, knowledge of the fuzzy relation is avoided, giving a large memory saving. Also this method of representation of fuzzy controller has an interesting property. The fuzzy control derives from the original control actions U_1, U_2, ..., U_N of the rules, where each of them 'activated' at a certain level matching the actual state of the process and the lth rule. In other words, ith rule contributes to the fuzzy control at a level equal to the value of the membership function Λ_i. Finally, the fuzzy control is a union of partial results.

The above analysis can be illustrated by a scheme in which both stages are clearly visuelised (Figure 2.4)

A simple illustration of haw this algorithm works is given in Figure 2.5. Here, the controller utilises one input and generates the fuzzy control; three control rules are available.

The input of fuzzy controller, say e, is matched to the linguidtic categories. This is done immediately, since the matching equals the membership function of the corresponding linguistic category, calculated at point e. The, as shown in the right of Figure 2.5, every control rule is activated at the level λ_i; the resulting fuzzy sets are also indicated. Summarising, a union of the fuzzy sets yields the result shown in Figure 2.5.

In the computation of fuzzy relation R, three methods are reported. For consistent presentstion, consider a controller with X as input abd U as output:

(i) Often used is the implicatiion method, modelled by the Cartesian product of X and U:

$$R(x, u) = \min(X(x), U(u))$$

which is the easiest way to calculate the fuzzy relation.

(ii) One can study the logic of the implication model, where $X \rightarrow U$ (X implies U) is equivalent to the union of the complement of X and U. Hence:

$$R(x, u) = \max[(1-X(x)), U(u)].$$

The main argument for this expression is its straightforward logical interpretation.

(iii) And the third form of implication was proposed by Zadeh []:

$$R(x, u) = \min[1, (1-X(x)+U(u))].$$

2.1.4. Examples of Applications of Fuzzy Controllers

Since 1974, when the first applications of fuzzy controllers were reported, there have been many advances. Now we have a number of valuable theoretical and practical concepts. On the practical aspects of fuzzy controllers, we refer to both experimental and industrial uses of process control.

Reference [] discusses fuzzy control of the activated-sludge process, in common use for treating sewage and waste water. This control is difficult because of:

(i) lack of relevant instrumentation,

(ii) poorly understood biological behaviour, and

(iii) poorly formalised control goals for the process model, which is itself hard to identify.

The results with the fuzzy controller were promising; simulation studies clearly indicated a suitable performance of the model.

Another example is the control of water purification [].

Pappis and Mamdani [] discuss the use of a fuzzy controller for traffic control at a single intersection of two one-way streets. When a performance index is stated as an avarage delay of vehicles, the fuzzy controller works better than the conventional vehicle actuated controller.

Experiments were performed by Lemke and Kickert [] on a warm-water process. Their main aim was to control the temperature of the water leaving atank for different temperature settings and a giwen flow of water. The process was subject to existing noises, asymmetries around the set point, and nonlinearities. Comparision a fuzzy controller with a PI digital controller optimally adjusted showed that the fuzzy controller had a faster response with the same eccuracy. Simplified versions of the fuzzy controller (fewer state variables) gave much faster step responses but larger oscillations around the set point.

An autopilot for dhips, designed with the aid of fuzzy sets, has been tested on a simulation model [].

Reference [] reports use of the fuzzy controller in rotary cement kilns. The fuzzy controller uses the knowledge of the operator of the kiln contained in the control protocol. Two operatin situations were studied:

- the kiln is stable as measured by kiln-drive torque, which shows small variations;
- the kiln-drive torque oscillates greatly: the kiln is unstable.

The fuzzy controller performs well in both modes. Fuzzy control works better than the operator []. In general, the fuzzy controller calls for control actions earlierthan the human being, and in smaller steps. It also reduces fuel consuption.

The fuzzy controller benefits automatic train operation []. Here, one needs to get some reasonable performance indices, such as running time, energy consumption, safety, comfort, stop gap, and traceability. These indices are difficult to handle in a formal mathematical model.

Many more examples of the use of fuzzy control, diverse control algorithms and a variety of industrial applications are described in [].