# **NEAR EAST UNIVERSITY**

# **Faculty of Engineering**

**Department of Computer Engineering** 

# **FUZZY CONTROL SYSTEM DESIGN**

**Graduation Project** COM-400

Student:

Serdar Yurdam (970353)

Supervisor: Asst. Prof. Dr Rahib Abiyev

Nicosia-2002

### **ACKNOWLEDGEMENTS**

"First, I would like to thank my supervisor Assistant Professor Dr Rahib Abiyev for his invaluable advice and belief in my work and myself over the course of this Graduation Project...

Second, I would like to express my gratitude to pharmacist Songül Erten and advocate Sabahat Özcan of the Yeni Doğa Pharmacy for the provided to me scholarship that made the work possible.

Third, I thank my family for their support during the preparation of this project.

Finally, I would also like to thank my friends Ulaş Taştan and Bülent Durukan for their advice and support."

#### ABSTRACT

The technological processes and problems are solved in such a way that the solutions are too complex and unpredicted. Because of that the deterministic models do not describe the solutions or processes enough. As a result controlling of the system becomes difficult. In these conditions using fuzzy technology bring us independent solutions for different models and adequately of the model.

The aim of thesis is to develop fuzzy control systems. To develop fuzzy control systems the structure and operation principle of fuzzy control system must be considered.

When analysing fuzzy control system the main problem is the synthesis of the fuzzy knowledge base for PD-like fuzzy controller. Processing mechanisms of fuzzy rules are described. Using max-min fuzzy processing of Zade the inference mechanism of fuzzy system must be realized.

The fuzzy controller for control temperature of technological processes is modelled.

Obtained results and simulation shows the efficiency of application of fuzzy technology for the industry.

# TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
CHAPTER ONE: THE STRUCTURE OF FUZZY CONTROLLER	2
<ul><li>1.1. Structure of General Fuzzy System</li><li>1.2. Structure of the PD-Like Fuzzy Controller</li></ul>	2 2
CHAPTER TWO: ALGORITHMS OF FUZZY CONTROLER	5
<ul><li>2.1. Fuzzification</li><li>2.2. Linguistic Variables</li><li>2.3. Rule Base</li><li>2.4. Inference Mechanism</li><li>2.5. Defuzzification</li></ul>	5 5 5 14 15
CHAPTER THREE: DEVELOPMENT OF FUZZY CONTROL SYSTEM	20
<ul> <li>3.1. Fuzzy control system architecture for inverted pendulum</li> <li>3.2. Choosing Fuzzy Controller Inputs and Outputs</li> <li>3.3. Linguistic Descriptions Of Knowledge</li> <li>3.3.1. Rules</li> <li>3.3.2. Rule-Bases</li> </ul>	20 21 22 24 25
<ul> <li>3.4. Fuzzy Quantification of Knowledge</li> <li>3.4.1. Membership Functions</li> <li>3.4.2. Fuzzification</li> <li>3.5. Matching: Determining Which Rules to Use</li> </ul>	26 26 31 31
<ul><li>3.6. Premise Quantification via Fuzzy Logic</li><li>3.7. Determining Which Rules Are On</li><li>3.8. Inference Step: Determining Conclusions</li></ul>	32 34 37
<ul> <li>3.9. Recommendation from One Rule</li> <li>3.10. Recommendation from Another Rule</li> <li>3.11. Converting decision into actions</li> <li>3.12 Combining Recommendations</li> </ul>	37 38 39 40

3.13. Other Ways to Compute and Combine Recommendations 3.14. Graphical Depiction of Fuzzy Decision Making	42 44
CHAPTER FOUR: FUZZY CONTROLLER FOR CONTROL TECHNOLOGICAL PROCESSES CONTROL	46
4.1. Development of Fuzzy Controller For Control of Temperature 4.2. Modeling of Fuzzy System for Control of Temperature	46 51
CONCLUSION	52
REFERENCES	53

#### INTRODUCTION

Fuzzy control is a control method based on fuzzy logic. Just as fuzzy logic can be described simply as "computing with words rather than numbers", fuzzy control can be described simply as "control with sentences rather than equations". A fuzzy controller can include empirical rules, and that is especially usefull in operator controlled plants.

Fuzzy controllers are used to control consumer products, such as washing machines, video cameras, and rice cookers, as well as industrial processes, such as cement kilns, underground trains, and robots.

In this project the development of fuzzy controller for technological processes is considered. The project consists of introduction, 4 chapters and conclusion.

Chapter One describes the structure of fuzzy controllers. The structure of general fuzzy systems, the functions of its main blocks are described. The structure of PD-like is described.

Chapter Two represents the algorithms of fuzzy controllers. The description of fuzzification, linguistic rules, their characteristics, fuzzy rules, inference mechanism and defuzzification are described.

Chapter Three describes the development of fuzzy controllers for technological process control. The developments of PD-like fuzzy controller and design example of fuzzy controller for inverted pendulum are presented.

Chapter Four describes the development of fuzzy controller for control of temperature. And the modeling of fuzzy system for control of temperature is given.

## **CHAPTER ONE: THE STRUCTURE OF FUZZY CONTROLLER**

#### 1.1.Structure Of General Fuzzy System

There are specific components characteristic of a fuzzy controller to support a design procedure, in the block diagram in Fig. 1, the controller is between a preprocessing block and a post-processing block. The following explains the diagram block by block.

Input signals entering to the preprocessing unit after scaling and performing some operation are enter to the fuzzification block. On the output of fuzzification block the fuzzy values of input signals are determined. Inference engine using these fuzzy input signal and rule base block mane decision. Obtained output signals after defuzzification are entered to the postprocessing unit , where the scaling of the output signal is carried out.



Figure 1.1. Blocks of a fuzzy controller

#### 1.2 Structure of PD-Like Fuzzy controller

The most simple fuzzy feedback control systems contain a fuzzy logic controller (FLC) in the form of a table of linguistic rules (or fuzzy relations matrix) and inputoutput interfaces. A linguistic rule consists of one or more premises and one or more consequences, f i. in the form:

IF (premises: a and b and c...) hold

THEN(consequences: x and y and z...) hold too.

A controller (see Fig. 1)presents and informationloop with:-an input signal g as an advising set-point(for example, a quality control);

-a comparatorwhich checks, if the emitted process output x is the correct reaction to the set-point g, and which emits himself an error signal e as an input to the decision element TLR, in order to report him, how much the process output x deviates from the preset value of g;

-a decision element TLR which emits for each value of e an output u which, on his side, becomes an input to a process with output x to be controlled.

A fuzzy logic controller is a synthesis of both, a controller's loop and a set of linguistic rules which are the content of the decision elementof the controller. The purpose of the input interface is to convert the non-fuzzy signals of error, either derivative(e''')or sum error(or both) into those input fuzzy sets which serve as premises in the correspondent linguistic rule of the FLC. The output fuzzy set(or the consequent of the linguistic rule)is converted by the output interface to the non-fuzzy control actionwhich is transferred to the input of an indistrial process.



Fig.1.2 A structure of a fuzzy controller

The transient performance demonstrated by these controllers as well as the noise immunity and robustness were essentially better then that of usual PID (Proportional, Integral, Differential) controllers. At the same time, the practical use of fuzzy control systems revealed the following problems:

**a.**there is not yet a satisfactory approach to the construction of input-output interfaces being sufficientlysupported by logical evidence;

**b**.there is no definitive agreement about how to proceed with an incomplete table of linguistic rule(TLR). Thus, no actualrule in the TLR can be applied to a concrete decision case, if the features of parameters p of this caseappear nowhere in the TLR as premises. Then, a new consequent c, as the missing term of a new rule r(p,c)must be introduced(this is done, for instance, by interviewing the human process operator). On the other hand , the broadened TLR demands an expensive study of the processand does not guarantee a desirable transient performance of the system in the case of a time variant process.

Moreover, the efficiency of fuzzy systems decends on the competence of the experts interviewed during the Knowladge elicitation process. Therefore, a wide application of single-loop fuzzy control systems is restiricted, because of their inability to cope with complex decision cases.

## **CHAPTER TWO: ALGORITHMS OF FUZZY CONTROLLER**

### 2.1 Fuzzification

The first block inside the controller is fuzzification, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each mie matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

#### 2.2 Linguistic Variables

To specify rules for the rule-base, the expert will use a "linguistic description"; hence, unliguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs. We will use "linguistic variables" (constant symbolic descriptions of what are in general time-varying quantities) to describe fuzzy system inputs and outputs. For our fuzzy system, linguistic variables denoted by  $i\sim$  are used to describe the inputs  $u_1$ . Similarly, linguistic variables denoted by  $5\sim$  are used to describe outputs Y1. For instance, an input to the fuzzy system may be described as  $\ddot{u}_1$ ="position error" or  $\ddot{u}_2$  =="velocity error," and an output from the fuzzy system may be  $\sim i \sim$ ="voltage in.

#### 2.3 Rule Base

The rules may use several variables both in the condition and the conclusion of the niles. The controllers can therefore be applied to both multi-input-multi-output (MIMO) problems and single-input-single-output (SISO) problems. The typical 5150 problem is to regulate a control signal based on an error signal. The controller may actually need both the error, the change in error, and the accumulated error as inputs, but we will call it single-loop control, because in principle all three are formed from the error measurement. To simplify, this section assumes that the control objective is to

regulate some process output around a prescribed setpoint or reference. The presentation is thus limited to single-loop control.

**Rule formats** Basically a linguistic controller contains rules in the if-then format, but they can be presented in different formats. In many systems, the rules are presented to the end-user in a format similar to the one below,

1. If error is Neg and change in error is Neg then output is NB

2. If error is Neg and change in error is Zero then output is NM

3. If error is Neg and change in error is Pos then output is Zero

4. If error is Zero and change in error is Neg then output is NM

5. If error is Zero and change in error is Zero then output is Zero

6. If error is Zero and change in error is Pos then output is PM

7. If error is **Pos** and change in error is **Neg** then output is **Zero** 

8. If error is **Pos** and change in error is **Zero** then output is **PM** 

9. If error is **Pos** and change in error is **Pos** then output is **PB** 

The names Zero, Pos, Neg are labels of fuzzy sets as well as NB, NM, PB and PM (negative big, negative medium, positive big, and positive medium respectively). The same set of rules could he presented in a relational format, a more compact representation.

Error	Change in error	Output
Neg	Pos	Zero
Neg	Zero	NM
Neg	Neg	NB
Zero	Pos	PM
Zero	Zero	Zero
Zero	Neg	NM
Pos	Pos	PB
Pos	Zero	PM
Pos	Neg	Zero

The top row is the heading. with the names of the variables. It is understood that the two leftmost columns are inputs, the rightmost is the output, and each row represents a rule. This format is perhaps better suited for an experienced user who wants to get an overview of the rule base quickly The relational format is certainly suited for

6

storing in a relational database. It should be emphasised, though. that the relational format implicitly assumes that the connective between the inputs is always logical **and** — or logical **or** for that matter as long as it is the same operation for all rules — and not a mixture of connectives. Incidentally, a fuzzy rule with an or combination of terms can be converted into an equivalent **and** combination of terms using laws of logic (DeMorgan's laws among others). A third format is the tabular linguistic format.

		Change in error			
		Neg	Zero	Pos	
	Neg	NB	NM	Zero	
Error	Zero	NM	Zero	PM	
	Pos	Zero	PM	PB	

This is even more compact. The input variables are laid out along the axes, and the output variable is inside the table. In case the table has an empty cell, it is an indication of a missing rule, and this format is useful for checking completeness. When the input variables are error and change in error, as they are here, that format is also called a linguistic phase plane. in case there are n > 2 input variables involved, the table grows to an n-dimensional array; rather user-un friendly.

To accommodate several outputs, a nested arrangement is conceivable. A rule with several outputs could also be broken down into several rules with one output. Lastly, a graphical format which shows the fuzzy membership curves is also possible (Fig. 3). This graphical user-interface can display the inference process better than the other formats, but takes more space on a monitor.

**Connectives** In mathematics, sentences are connected with the words and. or, if- then (or implies), and if and only if, or modifications with the word not. These five are called connectives. It also makes a difference how the connectives are implemented. The most prominent is probably multiplication for fuzzy **and** instead of minimum. So far most of the examples have only contained **and** operations, but a rule like "If error is very neg and not zero or change in error is zero then …" is also possible.

The connectives and and or are always defined in pairs, for example,

a and b = min (a. b) minimum a or b = max (a. b) maximum

a and b= a * b	algebraic product
$\mathbf{a} \text{ or } \mathbf{b} = \mathbf{a} + \mathbf{b} - \mathbf{a} * \mathbf{b}$	algebraic or probabilistic sum

or

There are other examples (e.g., Zimmermann. 1991, 31 32), but they are more complex.

**Modifiers** A linguistic modifier, is an operation that modifies the meaning of a term. For example, in the sentence "very close to 0". the word **very** modifies Close to 0 which is a fuzzy set. A modifier is thus an operation on a fuzzy set. The modifier **very** can be defined as squaring the subsequent membership function, that is

$$\operatorname{rerv} \mathbf{a} = \mathbf{a}^2 \tag{2}$$

(1)

Some examples of other modifiers are

extremely a = a<sup>3</sup> slightly a =a<sup>1/3</sup> somewhat a = moreorless a and not slightly a

A whole family of modifiers is generated by  $\mathbf{a}^p$  where p is any power between zero and infinity With  $p = \infty$  the modifier could be named **exactly**, because it would suppress all memberships lower than 1.0.

Universes Elements of a fuzzy set are taken from a universe oldiscourse oriust universe. The universe contains all elements that can come into consideration. Before designing the membership functions it is necessary to consider the universes for the inputs and outputs. Take for example the rule

If error is Neg and change in error is Pos then output is 0

Naturally, the membership functions for Neg and Pos must be defined for all possible values of error and change in error, and a standard universe may be convenient.

Another consideration is whether the input membership functions should be continuous or discrete. A continuous membership function is defined on a continuous universe by means of parameters. A discrete membership function is defined in terms of a vector with a finite number of elements. In the latter case it is necessary to specify the range of the universe and the value at each point. The choice between fine and coarse resolution is a trade off between accuracy, speed and space demands. The quantiser takes time to execute, and if this time is too precious, continuous membership functions will make the quantiser obsolete.

**Example 1 (standard universes)** Many authors and several commercial controllers use standard universes.

• The FL Smidth controller, for instance, uses the real number interval [-1,1].

• Authors of the earlier papers on fuzzy control used the integers in [-6.6].

• Another possibiliti' is the interval [-100, 100] corresponding to percentages of full scale.

• *Yet another is the integer range* [0, 4095] *corresponding to the output from a 12 bit analog to digital converter* 

• *A variant is* [-2047. 2048], where the interval is shifted in order to accommodate negative numbers.

The choice of datatypes may govern the choice of universe. For example, the voltage range [-5,5] could be represented as an integer range [-50,50], or as a floating point range [-5.0, 5.0], a signed byte datatype has an allowable integer range [-128, 127].

A way to exploit the range of the universes better is scaling. If a controller input mostly uses just one term, the scaling factor can be turned up such that the whole range is used. An advantage is that this allows a standard universe and it eliminates the need for adding more terms.

**Membership functions** Every element in the universe of discourse is a member of a fuzzy set to some grade, maybe even zero. The grade of membership for all its members describes a fuzzy set, such as Neg. In fuzzy sets elements are assigned a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the *support* of the fuzzy set. The function that ties a number to each element x of the universe is called the *membership function*  $\mu(x)$ .

The designer is inevitably faced with the question of how to build the term sets. There are two specific questions to consider:

(i) How does one determine the shape of the sets? and (ii) How many sets are necessary and sufficient? For example, the error in the position controller uses the family of terms

Neg, Zero, and Pos. According to fuzzy set theory the choice of the shape and width is subjective, but a few rules of thumb apply.

• A term set should be sufficiently wide to allow for noise in the measurement.

• A certain amount of overlap is desirable; otherwise the controller may run into poorly defined states, where it does not return a well defined output.

A preliminary answer to questions (i) and (ii) is that the necessary and sufficient number of sets in a family depends on the width of the sets, and vice versa. A solution could be to ask the process operators to enter their personal preferences for the membership curves; but operators also find it difficult to settle on particular curves. The merual for the TIL Shell product recommends the following (Hill Horstkotte &

The manual for the TIL SheII product recommends the following (Hill, Horstkotte & Teichrow, 1990).



Figure 2.2: Examples of membership functions. Read from top to bottom, left to right:
(a) s—function, (b) π— function, (c) z—function, (d-f) triangular versions, (g-i) trapezoidal versions, (j) flat π— function. (k) rectangle. (I) singleton.

• Start with triangular sets. All membership functions for a particular input or output should be symmetrical triangles of the same width. The leftmost and the rightmost should be shouldered ramps.

• The overlap should be at least 50%. The widths should initially be chosen so that each value of the universe is a member of at least two sets, except possibly for elements at the extreme ends. If, on the other hand, there is a gap between two sets no rules fire for values in the gap. Consequently the controller function is not defined.

Membership functions can be flat on the top, piece-wise linear and triangle shaped, rectangular, or ramps with horizontal shoulders. Fig. 2 shows some typical shapes of membership functions.

Strictly speaking, a fuzzy set A is a collection of ordered pairs

$$A = \{(x, \mu(x))\}$$

(3)

Item x belongs to the universe and  $\mu$  (x) is its grade of membership in A. A single pair (x,  $\mu$ (x)) is a fuzzy *singleton*; *singleton output* means replacing the fuzzy sets in the conclusion by numbers (scalars). For example

- 1. If error is Pos then output is 10 volts
- 2. If error is Zero then output is 0 volts
- 3. If error is Neg then output is -10 volts

There are at least three advantages to this:

- The computations are simpler;
- it is possible to drive the control signal to its extreme values; and
- it may actually be a more intuitive way to write rules.

The scalar can be a fuzzy set with the singleton placed in a proper position. For example 10 volts, would be equivalent to the fuzzy set (0,0,0,0,1) defined on the universe

(-10,-5,0,5,10) volts.

**Example 2 (membership functions)** Fuzzy controllers use a variety of membership functions. A common example of a function that produces a bell curve is based on the exponential function,

$$\boldsymbol{\mu} \left( \mathbf{x} \right) = \exp \left[ \frac{-\left( x - x_0 \right)^2}{2 \sigma^2} \right]$$
(4)

This is a standard Gaussian curve with a maximum value of 1 ,x is the independent variable on the universe, x to is the position of the peak relative to the universe, and  $\sigma$  is the standard deviation. Another definition which does not use the exponential is

$$\boldsymbol{\mu} \left( \mathbf{x} \right) = \left[ 1 + \left( \frac{x - x_0}{\sigma} \right)^2 \right]^{-1}$$
(5)

The FL Smidth controller uses the equation

$$\boldsymbol{\mu} \left( \mathbf{x} \right) = \mathbf{1} - \left( \frac{\sigma}{x - x_0} \right)^a$$
 (6)

The extra parameter a controls the gradient of the sloping sides. It is also possible to use other functions, for example the sigmoid known from neural networks. A cosine function can be used to generate a variety of membership functions. The s-curve can be implemented as

$$s(x_{l}, x_{r}, x) = \left\{ \begin{array}{ccc} 0 & x \prec x_{l} \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x - x_{r}}{x_{r} - x_{l}}\right) & x_{l} \leq x \leq x_{r} \\ 1 & x \succ x_{r} \end{array} \right\}$$
(7)

where  $x_1$  is the left breakpoint, and  $x_r$ , is the right breakpoint. The z-curve is just a reflection,



Figure2.3 :Graphical construction of the control signal in a fuzzy PD controller(generated in the Matlab Fuzzy Logic Toolbox).

$$z(x_{l}, x_{r}, x) = \left\{ \begin{array}{ccc} 1 & x \prec x_{l} \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x - x_{l}}{x_{r} - x_{l}} * \pi\right) & x_{l} \leq x \leq x_{r} \\ 0 & x \succ x_{r} \end{array} \right\}$$
(8)

Then the  $\pi$ -curve can be implemented as a combination of the s-curve and the z-curve, such that the peak is fiat over the interval  $[x_2, x_3]$ 

$$\pi(x_1, x_2, x_3, x_4, x) = \min((x_1, x_2, x), z(x_3, x_4, x))$$
(9)

#### 2.4 Inference Engine

Figures 2.3 and 2.4 are both a graphical construction of the algorithm in the core of the controller In Fig. 2.3. each of the nine rows refers to one rule. For example, the first row says that if the error is negative (row 1, column 1) and the *change in error* is negative (row 1, column 2) then the output should be negative big (row 1, column 3). The picture corresponds to the rule base in (2). The rules reflect the strategy that the control signal should be a combination of the reference error and the change in error, a fuzzy proportional-derivative controller. We shall refer to that figure in the following. The vertical lines on the first and second columns of the chart indicate the instances of the error and the change in error. For each rule, the inference engine looks up the membership values in the condition of the rule.

Aggregation The aggregation operation is used when calculating the degree of *fulfillment or firing strength*  $\alpha k$ . of the condition of a rule k. A rule, say rule 1, will generate a fuzzy membership value  $\mu$ e1 coming from the error and a membership value  $\mu$ cel coming from the *change in error* measurement. The aggregation is their combination,

$$\mu_{e1}$$
 and  $\mu_{ce1}$  (10)

Similarly for the other rules. Aggregation is equivalent to fuzzification, when there is only one input to the controller. Aggregation is sometimes also called *fulfilment* of the rule or *firing strength*.

Activation The activation of a rule is the deduction of the conclusion, possibly reduced by its firing strength. Thickened lines in the third column indicate the firing strength of each rule. Only the thickened part of the singletons are activated, and **min** or product (\*) is used as the activation operator. It makes no difference in this case, since the output membership functions are singletons, but in the general case of s—.  $\pi$ —, and z— functions in the third column, the multiplication scales the membership curves, thus preserving the initial shape, rather than clipping them as the **min** operation does. Both methods work well in general, although the multiplication results in a slightly smoother control signal. In Fig. 2.3, only rules four and five are active.

A rule k can be weighted a priori by a weighting factor  $\omega k \in [0,1]$ . which is its degree of confidence. In that case the firing strength is modified to

$$\alpha_{\mathbf{k}} = \boldsymbol{\omega}_{\mathbf{k}} * \alpha_{\mathbf{k}} \tag{11}$$

The degree of confidence is determined by the designer, or a learning program trying to adapt the rules to some input-output relationship.

Accumulation All activated conclusions are *accumulated*, using the **max** operation, to the final graph on the bottom right (Fig. 2.3). Alternatively, **sum** accumulation counts overlapping areas more than once (Fig. 2.4). Singleton output (Fig. 2.3) and **sum** accumulation results in the simple output

$$\alpha_{1} * S_{1} + \alpha_{2} * S_{2} + \dots + \alpha_{n} * S_{n}$$
(12)

The alpha's are the firing strengths from the *n* rules and  $s_1, ..., s_n$ , are the output singletons. Since this can be computed as a vector product, this type of inference is relatively fast in a matrix oriented language.

There could actually have been several conclusion sets. An example of a oneinput-two-outputs rule is "if  $e_a$  is **a** then  $o_1$  is **x** and  $o_2$  is **y**" The inference engine can treat two (or several) columns on the conclusion side in parallel by applying the firing strength to both conclusion sets. In practice, one would often implement this situation as two rules rather than one, that is, "If  $e_a$  is **a** then  $o_1$  is **x**", "If  $e_a$  is **a** then  $o_2$  is **y**".

#### 2.5 Defuzzification

The resulting fuzzy set (Fig. 2.3, bottom right; Fig. 2.4, extreme right) must be converted to a number that can be sent to the process as a control signal. This operation is called *defuzzification*, and in Fig. 2.4 the x-coordinate marked by a white, vertical dividing line becomes the control signal. The resulting fuzzy set is thus defuzzified into a crisp control signal. There are several defuzzification methods.

Centre of gravity (COG) The crisp output value u (white line in Fig.2.4) is the abscissa under the centre of gravity of the fuzzy set,

$$u = \frac{\sum_{i} \mu(x_{i}) x_{i}}{\sum_{i} \mu(x_{i})}$$
(13)

Here  $X_i$  is a running point in a discrete universe, and  $\mu$  ( $x_i$ ) is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals. It is a much used method although its computational complexity is relatively high. This method is also called *centroid of area*.

Centre of gravity method for singletons (COGS) If the membership functions of the conclusions are singletons (Fig. 2.3), the output value is

$$\mu = \frac{\sum_{i} \mu (s_{i}) s_{i}}{\sum_{i} \mu (s_{i})}$$
(14)

Here  $s_i$  is the position of singleton i in the universe. and  $\mu$  ( $s_i$ ) is equal to the firing strength  $\alpha_i$  of rule i. This method has a relatively good computational complexity and u is differentiable with respect to the singletons  $s_i$ , which is useful in neurofuzzy systems.

**Bisector of area (BOA)** This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. In the continuous case,

$$u = \left\{ x \middle| \int_{Min}^{x} \mu(x) dx = \int_{x}^{Max} \mu(x) dx \right\}$$
(15)

Here x is the running point in the universe,  $\mu(x)$  is its membership.

*Min* is the leftmost value of the universe, and *Max* is the rightmost value. Its computational complexity is relatively high, and it can be ambiguous. For example. if the fuzzy set consists of two singletons any point between the two would divide the area in two halves; consequently it is safer to say that in the discrete case. BOA is not defined.

Mean of maxima (MOM) An intuitive approach is to choose the point with the strongest possibility i.e. maximal membership. It may happen, though, that several such

points exist, and a common practice is to take the *mean of maxima* (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good.

Leftmost maximum (LM), and rightmost maximum (RM) Another possibility is to choose the leftmost maximum (LM), or the rightmost maximum (RM). In the case of a robot, for instance, it must choose between left or right to avoid an obstacle in front of it.



Figure2.4:One input, one output rule base with non-singleton output sets.

The defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small.

**Example 3(inference)** How is the inference in Fig. 8 implemented using discrete fuzzy sets?

Behind the scene all universes were divided into 201 points from -100 to 100. but for brevity, let us just use five points. Assume the universe u, common to all variables, is the vector



A cosine function can be used to generate a variety of membership functions. The scurve can be implemented as:

$$s(x, x, x) = \left\{ \begin{array}{l} 0 & x \prec x \\ 1 + -\cos(-\pi) & x \leq x \leq x \\ 1 & x \succ x \end{array} \right\}$$
(16)

where  $x_1$  is the left breakpoint, and  $x_r$  is the right breakpoint. The z-curve is just a reflection,

$$z(x_{l}, x_{r}, x) = \begin{cases} 1 & x \prec x_{l} \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x - x_{l}}{x_{r} - x_{l}} * \pi\right) & x_{l} \leq x \leq x_{r} \\ 0 & x \succ x_{r} \end{cases}$$
(17)

Then the  $\pi$ -curve (see for example Fig. 2.2 ( j )) can be implemented as a combination of the s-curve and the z-curve, such that the peak is flat over the interval [x2,x3]

$$\pi(x_1, x_2, x_3, x_4, x) = \min(s(x_1, x_2, x), z(x_3, x_4, x))$$
(18)

A family of terms is defined by means of the  $\pi$ —function, such that

$neg = \pi (-100, -100, -60, 10, u) =$	= 1	(	).95	0.05	0	0
zero = $\pi$ (-90, -20, 20, 90, u) =	0	0.6	31 1	0.6	1	0
$\mathbf{pos} = \pi (-10, 60, 100, 100, \mathbf{u}) =$	0	0	0.05	0.9	5	1

Above we inserted the whole vector **u** in place of the running point x; the result is thus a vector. The figure assumes that error = -50 (the unit is percentages of full range). This corresponds to the second position in the universe, and the first rule contributes with a membership neg(2) = 0.95. This firing strength is propagated to the conclusion side of the rule using **min**, such that the contribution from this rule is

 $0.95 \text{ min neg} = \boxed{0.95 \ 0.95 \ 0.05 \ 0 \ 0}$ 

The activation operation was **min** here. Apply the same pmcedure to the two remaining rules, and stack all three contributions on top of each other,

0.95	0.95	0.05	0	0
0	0.61	0.61	0.61	0
0	. 0	0	0	0

To find the accumulated output set, perform a max operation down each column. The result is the vector

0.95 min n	ieg =[	0.95	0.95	0.05	0	0
------------	--------	------	------	------	---	---

The centre of gravity method yields

$$u = \frac{\sum_{i} \mu (x_{i}) x_{i}}{\sum_{i} \mu (x_{i})}$$
(19)

$$=\frac{0.95*(-100)+0.95*(-50)+0.61*0+0.61*50+0*100}{0.95+0.95+0.61+0.61+0} = (20)$$

= -35.9

which is the control signal (before postprocessing).

(21)

## CHAPTER THREE: DEVELOPMENT OF FUZZY CONTROL SYSTEM

## 3.1 Fuzzy Control System Architecture for Inverted Pendulum

Consider each of the components of the fuzzy controller for a problem of balancing an inverted pendulum on a cart, as shown in Figure 3.1. Here, y denotes the angle that the pendulum makes with the vertical (in radians), 1 is the half-pendulum length (in meters), and u is the force input that moves the cart (in Newtons). The r is used to denote the desired angular position of the pendulum. The goal is to balance the pendulum in the upright position (i.e., r = 0) when it initially starts with some nonzero angle off the vertical (i.e., y<>0). This is a very simple and academic nonlinear control problem, and many good techniques already exist for its solution.



Fig 3.1 Inverted pendulm on a cart.

#### 3.2 Choosing Fuzzy Controller Inputs and Outputs

The fuzzy controller is to be designed to automate how a human expert who is successful at this task would control the system. First, the designers of the fuzzy controller tell what information human will use as inputs to the decision-making process. Suppose that for the inverted pendulum, the expert says that people will use

#### E(t)=r(t)-y(t)

and

#### de(t)/dt

as the variables on which to base decisions. Certainly, there are many other choices (e.g., the integral of the error e could also be used) but this choice makes good intuitive sense. Next, consider identify the controlled variable. For the inverted pendulum, for controled only the force that moves the cart, so the choice here is simple.

For more complex applications, the choice of the inputs to the controller and outputs of the controller (inputs to the plant) can be more difficult. If the designer believes that proper information is not available for making control decisions, people may have to invest in another sensor that can provide a measurement of another system variable. Alternatively, the designer may implement some filtering or other processing of the plant outputs.

Once the fuzzy controller inputs and outputs are chosen, it should determine what the reference inputs are. For the inverted pendulum, the choice of the reference input r = 0 is clear. In some situations, however, the designer may want to choose r as some nonzero constant to balance the pendulum in the off vertical position. To do this, the controller must maintain the cart at a constant velocity so that the pendulum will not fall.

After all the inputs and outputs are defined for the fuzzy controller, the fuzzy control system could be specify. The fuzzy control system for the inverted pendulum is shown in Figure 3.2. Let us consider a description of control process. The choice of the inputs and outputs of the controller places certain constraints on the remainder of the fuzzy control design process. If the proper information is not provided to the fuzzy controller, there will be little hope for being able to design a good rule-base or inference

21

mechanism. Moreover, even if the proper information is available to make control decisions, this will be of little use if the controller is not able to properly affect the process variables via the process inputs. It must be understood that the choice of the controller inputs and outputs is a fundamentally important part of the control design process.



Fig. 3.2 Fuzzy controller for an inverted pendulum on a cart.

#### 3.3 Linguistic Descriptions of Knowledge

Suppose that the human expert shown provides a description of how best to control the plant in some natural language (e.g., English). It consider to take this "linguistic" description and load it into the fuzzy controller, as indicated by the arrow in Figure 3.2

The linguistic description provided by the expert can generally be broken into several parts. There will be "linguistic variables" that describe each of the time-varying fuzzy controller inputs and outputs. For the inverted pendulum,

"error" describes e(t)
"change-in-error" describes de(t)/dtt;
"force" describes u (t )

The linguistic descriptions as short as possible (e.g., using "e(t)" as the linguistic variable for e(t)), yet accurate enough so that they adequately represent the variables

Suppose for the pendulum example that "error," "change-in-error," and "force" take on the following values:

"neglarge" "negsmall" "zero" "possmall" "poslarge"

Note that used "negsmall" as an abbreviation for "negative small in size" and so on for the other variables. Such abbreviations help keep the linguistic descriptions short yet precise. Here neg is negative, pos is positive. Every linguistic value nicely represent that the variable has a numeric quality.

The linguistic variables and values provide a language for the expert to express designer ideas about the control decision-making process in the context of the framework established by choosen of fuzzy controller inputs and outputs. Recall that for the inverted pendulum

r = 0 and e = r - y so that

and

$$de/dt = -dy/dt$$
.

e = -y

since dr/dt = 0. First, how the certain dynamic behaviors with linguistics can be quantified will studied.

For the inverted pendulum each of the following statements quantifies a different configuration of the pendulum :

- The statement "error is poslarge" can represent the situation where the pendulum is at a significant angle to the *left* of the vertical.
- The statement "error is negsmall" can represent the situation where the pendulum is just slightly to the right of the vertical, but not too close to the vertical to justify quantifying it as "zero" and not too far away to justify quantifying it as "neglarge."
- The statement "error is zero" can represent the situation where the pendulum is very near the vertical position (a linguistic quantification is not precise, hence we are willing to accept any value of the error around e(t) = 0 as being

quantified linguistically by "zero" since this can be considered a better quantification than "possmall" or "negsmall").

- The statement "error is poslarge **and** change-in-error is "possmall" can represent the situation where the pendulum is to the left of the vertical and, since dy/dt<0, the pendulum is moving *away* from the upright position (note that in this case the pendulum is moving counterclockwise).
- The statement "error is negsmall **and** change-in-error is possmall" can represent the situation where the pendulum is slightly to the right of the vertical and, since dy/dt<0, the pendulum is moving *toward* the upright position (note that in this case the pendulum is also moving counterclockwise).

#### 3.3.1 Rules

Next, consider the above linguistic quantification to specify a set of rules that captures the expert's knowledge about how to control the plant. In particular, for the inverted pendulum in the three positions shown in Figure 3.3, we have the following rules

# 1. If error is neglarge and change-in-error is neglarge Then force is poslarge

This rule quantifies the situation in Figure 3.3(a) where the pendulum has a large positive angle and is moving clockwise; hence it is clear that we should apply a strong positive force (to the right) so that we can try to start the pendulum moving in the proper direction.

# 2. If error is zero and change-in-error is possmall Then force is negsmall

This rule quantifies the situation in Figure 3.3(b) where the pendulum has nearly a zero angle with the vertical (a linguistic quantification of zero does not imply that e(t) = 0 exactly) and is moving counterclockwise; hence we should apply a small negative force (to the left) to counteract the movement so that it moves toward zero (a positive force could result in the pendulum overshooting the desired position).

## 3. If error is poslarge and change-in-error is negsmall Then force is negsmall

This rule quantifies the situation in Figure 3.3(c) where the pendulum is far to the left of the vertical and is moving clockwise; hence we should apply a small negative force

(to the left) to assist the movement, but not a big one since the pendulum is already moving in the proper direction.

Each of the three rules listed above is a "linguistic rule" since it is formed solely from linguistic variables and values. Since linguistic values are not precise representations of the



Figure 3.3) inverted pendulum in various positions

underlying quantities that they describe, linguistic rules are not precise either. They are simply abstract ideas about how to achieve good control that could mean somewhat different things to different people.

The general form of the linguistic rules listed above

is

#### If premise Then consequent

From the three rules listed above, the premises (which are sometimes called "antecedents") are associated with the fuzzy controller inputs and are on the left-handside of the rules. The consequents (sometimes called "actions") are associated with the fuzzy controller outputs and are on the right-hand-side of the rules.

#### 3.3.2 Rule-Bases

Using the above approach, the rules for the pendulum problem for all possible cases could be written. Note that since consider only specify a finite number of linguistic variables and linguistic values, there is only a finite number of possible rules. For the pendulum problem, with two inputs and five linguistic values for each of these, there are at most  $5^2 = 25$  possible rules.

A tabular representation of one possible set of rules for the inverted pendulum is shown in Table 1. Notice that the body of the table lists the linguistic-numeric consequents of the rules, and the left column and top row of the table contain the linguistic-numeric premise terms. Then, for instance, the (2, -1) position (where the "2" represents the row having "2" for a numeric-linguistic value and the "-1" represents the column having "-1" for a numeric-linguistic value) has a -1 ("negsmall") in the body of the table and represents the rule

If error is poslarge and change-in-error is negsmall Then force is negsmall which is rule 3 above. Table 1 represents abstract knowledge that the expert has about how to control the pendulum given the error and its derivative as inputs.

#### Table 1

force		Change-in-error e'				
u		NL	NS	Z	PS	PL
Error	NL	PL	PL	PL	PS	Z
e	NS	PL	PL	PS	Z	NS
	Z	PL	PS	Z	NS	NL
	PS	PS	Z	NS	NL	NL
	PL	Z	NS	NL	NL	NL

Notice the diagonal of zeros and viewing the body of the table as a matrix as seeing that it has a certain symmetry to it. This symmetry that emerges when the rules are tabulated is no accident and is actually a representation of abstract knowledge about how to control the pendulum; it arises due to a symmetry in the system's dynamics.

### 3.4 Fuzzy Quantification of Knowledge

#### 3.4.1 Membership Functions

The membership function quantifies, in a continuous manner, whether values of e(t) belong to (are members of) the set of values that are "possmall," and hence it quantifies the meaning



fig 3.4 ) membership function for linguistic value "possmall."

of the linguistic statement "error is possmall." This is why it is called a membership function. It is important to recognize that the membership function in Figure3.4 is only one possible definition of the meaning of "error is possmall"; also could use a bell-shaped function, a trapezoid, or many others.



a)Trapezoid



(d) Skewed triangle



Depending on the application and the designer (expert), many different choices of membership functions are possible.

A "crisp" (as contrasted to "fuzzy") quantification of "possmall" can also be specified, but via the membership function shown in Figure 3.6. This membership function is simply an alternative representation for the interval on the real line. and it indicates that this interval of numbers represents "possmall." Clearly, this characterization of crisp sets is simply another way to represent a normal interval (set) of real numbers.

Now the membership functions for all 15 linguistic values (five for each input and five for the output) of inverted pendulum example cab be specified. Figure 3.7 for one choice of membership functions.







29



Fig3.7) Membership functions for an inverted pendulum on a cart.

For the output u, the membership functions at the outermost edges cannot be saturated for the fuzzy system to be properly defined.

The rule-base of the fuzzy controller holds the linguistic variables, linguistic values, their associated membership functions, and the set of all linguistic rules (shown in Table 1), so the the description of the simple inverted pendulum is completed.

#### 3.4.2 Fuzzification

It is actually the case that for most fuzzy controllers the fuzzification block in Figure 1 can be ignored since this process is so simple. For now, the reader should simply think of the fuzzification process as the act of obtaining a value of an input variable (e.g., e(t)) and finding the numeric values of the membership function(s) that are defined for that variable. For example, if e(t) = n/4 and de(t)/dt = n/16, the fuzzification process amounts to finding the values of the input membership functions for these. In this case

$$\mu_{\text{possmal}}(e(t)) = 1$$

(with all others zero) and

$$\mu_{zero}(de(t)/dt) = \mu_{possmal}(de(t)/dt) = 0.5.$$

Some think of the membership function values as an "encoding" of the fuzzy controller numeric input values. The encoded information is then used in the fuzzy inference process that starts with "matching."

#### 3.5 Matching: Determining Which Rules to Use

- 1. The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. This "matching" process involves determining the certainty that each rule applies, and typically controller will more strongly take into account the recommendations of rules that who is more certain apply to the current situation.
- 2. The conclusions (what control actions to take) are determined using the rules that have been determined to apply at the current time. The conclusions are characterized with a fuzzy set (or sets) that represents the certainty that the input to the plant should take on various values.

#### 3.6 Premise Quantification via Fuzzy Logic

To perform inference it is needed to quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input. Consider Figure 9, where listed two terms from the premise of the rule

#### If error is zero and change-in-error is possmall Then force is negsmall

Above, quantified the meaning of the linguistic terms "error is zero" and "changein- error is possmall" via the membership functions shown in Figure 8 is quantified. Now search to quantify the linguistic premise "error is zero **and** change-in-error is possmall." Hence, the main item to focus on is how to quantify the logical "and" operation that combines the meaning of two linguistic terms. When the designer used to standard Boolean logic to combine these linguistic terms, since who quantified them more precisely with fuzzy sets (i.e., the membership functions), could used these.

To see how to quantify the "and" operation, begin by supposing that e(t)=n/8 and de(t)/dt=n/32, so that using Figure 3.7 (or Figure 3.8) let's see that



# $\mu_{zero}$ (e(t))=0.5 and $\mu_{possmal}$ (de/dt)=0.5

What, for these values of e(t) and de(t)/dt, is the certainty of the statement "error is zero and change-in-error is possmall"

that is the premise from the above rule? Denoted this certainty by  $\mu_{\text{premise}}$  There are actually several ways to define it:

- Minimum: Deine µ<sub>premise</sub>=min{0.5,0.25}=0.25, tht is, using the minimum of the two membership values.
- > **Product**: Deine  $\mu_{\text{premise}} = (0.5)(0.25) = 0.125$ , that is, using the product of the two membership values.

Notice that both ways of quantifying the "and" operation in the premise indicate that you can be no more certain about the conjunction of two statements than controller about the individual terms that make them up (note that  $0 \le \mu_{\text{premise}} \le 1$  for either case)..

While how to quantify the "and" operation for one value of e(t) and de(t)/dt is shown, if consider all possible e(t) and de(t)/dt values, it will obtained a multidimensional membership function  $\mu_{\text{premise}}(e(t), de(t)/dt)$  that is a function of e(t)and de(t)/dt for each rule. For example, if choosen the minimum operation to represent the "and" in the premise, then we get the multidimensional membership function  $\mu_{\text{premise}}(e(t), de(t)/dt)$ . Notice that if we pick values for e(t) and de(t)/dt, the value of the premise certainty  $\mu_{\text{premise}}(e(t), de(t)/dt)$  represents how certain we are that the rule

If error is zero and change-in-error is possmall Then force is negsmall

is applicable for specifying the force input to the plant. As e(t) and de(t)/dt change, the value of  $\mu_{\text{premise}}(e(t), de(t)/dt)$  changes according to Figure 3.9, and the controller become less or more certain of the applicability of this rule.



de(t)/dt, (rad/sec)

Fig.3.9 Membership function of the premise for a single rule.

# 3.7 Determining Which Rules Are On

Determining the applicability of each rule is called "matching." Consider that a rule is "on at time t" if its premise membership function  $\mu_{\text{premise}}(e(t), de(t)/dt) > 0$ . Hence, the inference mechanism seeks to determine which rules are on to find out which rules are relevant to the current situation. Consider, for the inverted pendulum example, how compute the rules that are on. Suppose that

# e(t)=0 and de(t)/dt = n/8-n/32(=0.294)

Figure 11 shows the membership functions for the inputs and indicates with thick black vertical lines the values above for e(t) and de(t)/dt. Notice that  $\mu_{zero}(e(t))=1$  but that the other membership functions for the e(t) input are all "off" (i.e., their values are zero). For the de(t)/dt input like that  $\mu_{zero}(de(t)/dt)=0.25$  and  $\mu_{possmal}(de(t)/dt)=0.75$  and that all the other membership functions are off. This implies that rules that have the premise terms

"error is zero"

"change-in-error is zero"

#### "change-in-error is possmall"

are on (all other rules have  $\mu_{premise}(e(t), de(t)/dt) = 0$ . So, which rules are these? Using Table 1 on, finded that the rules that are on are the following:

1. If error is zero and change-in-error is zero Then force is zero

2. If error is zero and change-in-error is possmall Then force is negsmall

Note that since for the pendulum example there are at most two membership functions over- lapping, the designer will never have more than four rules on at one time (this concept generalizes to many inputs). Actually, for this system the designer will either have one, two, or four rules on at any one time. To get only one rule on choose, for example, e(t) = 0 and de(t)/dt = n/8 so that only rule 2 above is on. What values would you choose for e(t) and de(t)/dt to get four rules on? For this system, to have exactly three rules on?





Fig 3.10.Input membeship functions with input values

It is useful to consider pictorially which rules are on. Consider Table 2, which is a copy of Table 1 with boxes drawn around the consequents of the rules that are on (notice that these

Table 2

force		Change-in-error e'						
u		NL NS Z PS PI						
Error	NL	PL	PL	PL	PS	Z		
e	NS	PL	PL	PS	Z	NS		
	Z	PL	PS	Z	NS	NL		
	PS	PS	Z	NS	NL	NL		
	PL	Z	NS	NL	NL	NL		

are the same two rules listed above). Notice that since  $e(t) = \theta(e(t)$  is directly in the middle between the membership functions for "possmall" and "negsmall") both these membership functions are off. If perturbed e(t) slightly positive (negative), then the controller would have the two rules below (above) the two highlighted ones on also.

# 3.8 Inference Step: Determining Conclusions

The consider how to determine which conclusions should be reached when the rules that are on are applied to deciding what the force input to the cart carrying the inverted pendulum should be. To do this, first consider the recommendations of each rule independently. Then later combine all the recommendations from all the rules to determine the force input to the cart.

# 3.9 Recommendation from One Rule

Consider the conclusion reached by the rule

#### If error is zero and change-in-error is zero Then force is zero

which for convenience the controller will refer to as "rule (1)." Using the minimum to represent the premise, we have

#### $\mu_{premise1} = min\{0.25, 1\} = 0.25$

(the notation  $\mu_{premisel}$  represents  $\mu_{premise}$  for rule (1)) so that it has 0.25 certain that this rule applies to the current situation. The rule indicates that if its premise is true then the action indicated by its consequent should be taken. For rule (1) the consequent is "force is zero" (this makes sense, for here the pendulum is balanced, so the controller should not apply any force since this would tend to move the pendulum away from the vertical). The membership function for this consequent is shown in Figure 3.11(a). The membership function for the conclusion rea\_ched by rule (1), which is denoted by  $\mu_l$ , is shown in Figure 3.11(b) and is given by

$$\mu_{I}(u) = \min\{0.25, \mu_{zero}(u)\}$$

this membership function defines the "implied fuzzy set"<sup>3</sup> for rule (1) (i.e., it is the conclusion that is implied by rule (1)).

Notice that the membership function  $\mu_1(u)$  is a function of u and that the minimum operation will generally "chop off the top" of the  $\mu_{zero}(u)$  membership function to produce  $\mu_1(ut)$ . For different values of e(t) and de(t)/dt there will be

different values of the premise certainty  $\mu_{premise}(e(t), de(t)/dt)$  for rule (1) and hence different functions  $\mu_1(u)$  obtained (i.e., it will chop off the top at different points).

That  $\mu_l(u)$  is in general a time-varying function that quantifies how certain rule (1) is that the force input u should take on certain values. It is most certain that the force input should lie in a region around zero (see Figure 3.11(b)), and it indicates that it is certain that the force input should not be too large in either the positive or negative direction-this makes sense if the designer consider the linguistic meaning of the rule. The membership function  $\mu_l(u)$  quantifies the conclusion reached by only rule (1) and only for the current e(t) and de(t)/dt. It is important that the reader be able to picture how the shape of the implied fuzzy set changes as the rule's premise certainty changes over time.



Fig3.11 (a) Consequent membership function and (b) implied fuzzy set with membership function  $\mu_1(u)$  for rule (1). Recall that the units for u(t) are Newtons (N).

#### 3.10 Recommendation From Another Rule

Consider the conclusion reached by the other rule that is on,

If error is zero and change-in-error is possmall Then force is negsmall which for convenience we will refer to as "rule (2)." Using the minimum to represent the premise, considering

#### $\mu_{premise2}(u) = min\{0.75, 1\} = 0.75$

so that by 0.75 rule is applied to the current situation. Notice that consider much more certain that rule (2) applies to the current situation than rule (1). For rule (2) the consequent is "force is negsmall" (this makes sense, for here the pendulum is perfectly balanced but is moving in the counterclockwise direction with a small velocity). The membership function for this consequent is shown in Figure 3.12 (a). The membership

function for the conclusion reached by rule (2), which is denoted by  $\mu_2(u)$ , is shown in Figure 3.12(b) (the shaded region) and is given by

## $\mu_2(u) = min\{0.75, \mu_{negsmall}(u)$

this membership function defines the implied fuzzy set for rule (2) (i.e., it is the conclusion that is reached by rule (2)). Once again, for different values of e(t) and de(t)/dt there will be different values of  $\mu_{premise2}(e(t), de(t)/dt)$  for rule (2) and hence different functions  $\mu_2(u)$  obtained. Rule (2) is quite certain that the control output (process input) should be a small negative value. This makes sense since if the pendulum has some counterclockwise velocity then the controller would want to apply a negative force (i.e., one to the left). As rule (2) has a premise membership function that has higher certainty than for rule (1).



This completes the operations of the inference mechanism in Figure 1. While the input to the inference process is the set of rules that are on, its output is the set of implied fuzzy sets that represent the conclusions reached by all the rules that are on. For this example, there are at most four conclusions reached since there are at most four rules on at any one time. (In fact, you could say that there are *always* four conclusions reached for our example, but that the implied fuzzy sets for some of the rules may have implied membership functions that are zero for all values.)

# 3.11 Converting Decisions into Actions

Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output (plant input). Some think of defuzzification as "decoding" the fuzzy set information produced by the inference process (i.e., the implied fuzzy sets) into numeric fuzzy controller outputs.

There are actually many approaches to defuzzification.

# 3.12 Combining Recommendations

Due to its popularity, first consider the "center of gravity" (COG) defuzzification method for combining the recommendations represented by the implied fuzzy sets from all the rules. Let b<sub>i</sub> denote the center of the membership function (i.e., where it reaches its peak for example) of the consequent of rule (i). For this example it has



Fig.3.13

as shown in Figure 3.13 Let  $\int \mu_i$ 

denote the area under the membership function  $\mu_i$ . The COG method computes  $u^{crisp}$  to be

$$u^{crisp} = \frac{\sum_{i} b_i \int \mu_i}{\sum_{i} \int \mu_i}$$
(1)

This is the classical formula for computing the center of gravity. In this case it is for com- puting the center of gravity of the implied fuzzy sets. Three items about Equation (1) are important to note:

1. Practically, the controller cannot have output membership functions that have infinite area since even though they may be "chopped off in the minimum operation for the implication (or scaled for the product operation) they can still end up with infinite area. This is the reason controller do not allow infinite area membership functions for the linguistic values for the controller output (e.g., we did not allow the

saturated membership functions at the outermost edges as controller had for the inputs shown in Figure 3.7).

- 2. The controller must be careful to define the input and output membership functions so that the sum in the denominator of Equation (1) is not equal to zero no matter what the inputs to the fuzzy controller are. Essentially, this means that controller must have some sort of conclusion for all possible control situations controller may encounter.
- 3. While at first glance it may not appear so,  $\int \mu_i$  is easy to compute for this example. For the case where controller have symmetric triangular output membership functions that peak at one and have a base width of w, simple geometry can be used to show that the area under a triangle "chopped off at a height of h (such as the ones in Figures3.11 and 3.12) is equal to

Given this, the computations needed to compute  $u^{crisp}$  are not too significant.

Seeing that the property of membership functions being symmetric for the output is important since in this case no matter whether the minimum or product is used to represent the implication, it will be the case that the center of the implied fuzzy set will be the same as the center of the consequent fuzzy set from which it is computed. If the output membership functions are not symmetric, then their centers, which are needed in the computation of the COG, will change depending on the membership value of the premise. This will result in the need to recompute the center at each time instant.

Using Equation (1) with Figure 3.13 it consider

$$u^{crisp} = \frac{(0)(4.375) + (-10)(9.375)}{4.375 + 9.375} = -6.81$$

as the input to the pendulum for the given e(t) and de(t)/dt (see fig3.13).

#### 3.13 Other Ways to Compute and Combine Recommendations

As another example, it is interesting to consider how to compute, by hand, the operations that the fuzzy controller takes when used the product to represent the implication or the "center-average" defuzzification method.

Consider Figure 3.14, where the controller have drawn the output membership functions for "negsmall" and "zero" as dotted lines. The implied fuzzy set from rule (1) is given by the membership function

$$\mu_{1}(u) = 0.25 \mu_{zero}(u)$$

shown in Figure 15 as the shaded triangle; and the implied fuzzy set for rule (2) is given by the membership function

$$\mu_2(u) = 0.75 \mu_{negsmall}(u)$$

shown in Figure 3.14 as the dark triangle. Notice that computation of the COG is easy since used 1/2 wh as the area for a triangle with base width w and height h. When the controller useproduct to represent the implication, which is obtain

$$u^{crisp} = \frac{(0)(2.5) + (-10)(7.5)}{2.5 + 7.5} = -7.5$$



fig3.14) implied fuzzy sets when the product is used to represent the implication

As another example of how to combine recommendations, it is consider the `center-average" method for defuzzification. For this method used :

$$u^{crisp} = \frac{\sum_{i} b_{i} \mu_{premise_{i}}}{\sum_{i} \mu_{premise_{i}}}$$
(2)

where to compute  $\mu_{premisei}$  used, for example, minimum. It called the "center-average" method since Equation (2) is a weighted average of the center values of the output membership function centers. Basically, the center-average method replaces the areas of the implied fuzzy sets that are used in COG with the values of  $\mu_{premisei}$ . This is a valid replacement since the area of the implied fuzzy set is generally proportional to  $\mu_{premisei}$  since  $\mu_{premisei}$  is used to chop the top off (minimum) or scale (product) the triangular output membership function when COG is used for our example. For the above example, it consider

$$u^{crisp} = \frac{(0)(0.5) + (-10)(0.75)}{0.25 + 0.75} = -7.5$$

which just happens to be the same value as above. Some like the center-average defuzzification method because the computations needed are simpler than for COG and because the output membership functions are easy to store since the only relevant information they provide is their center values (b; ) (i.e., their shape does not matter, just their center value).

Notice that while both values computed for the different inference and defuzzification methods provide reasonable command inputs to the plant, it is difficult to say which is best without further investigations (e.g., simulations or implementation). This ambiguity about how to define the fuzzy controller actually extends to the general case and also arises in the specification of all the other fuzzy controller components, as its discussed below. Some would call this "ambiguity" a design flexibility, but unfortunately there are not too many guidelines on how best to choose the inference strategy and defuzzification method, so such flexibility is of questionable value.

## 3.14 Graphical Depiction of Fuzzy Decision Making

In figure 3.15 the calculation of fuzzy controller output is given. Here, the controller used the minimum operator to represent the "and" in the premise and the implication and COG defuzzification. The reader is advised to study each step in this diagram to gain a fuller understanding of the operation of the fuzzy controller. To do this, develop a similar diagram for the case where the product operator is used to represent the "and" in the premise and the implication, and choose values of e(t) and de(t)/dt that will result in four rules being on. Then, repeat the process when center-average de(t)/dt defuzzification is used with either minimum or product used for the premise. Also, learn how to picture in your mind how the parameters of this graphical representation of the fuzzy controller operations change as the fuzzy controller inputs change.





## CHAPTER 4. FUZZY CONTROLLER FOR TECHNOLOGICAL PROCESSES CONTROL

# 4.1. Development of fuzzy controller for control of temperature

The aim of the initial oil refinery process is to obtain a wide fraction of light products, such as gasoline, kerosine and others. The main block of the initial oil refinery unit is a multicomponent rectifying tower K-2 (see Fig.1). The products of K-2 are cuts fr-1, fr-2 and fr-3 which are removed from the tower in the form of side-cut distillates to the strippers K-6, K-7 and K-9.



Fig.1. The sheme of the initial oil-refinery unit

The quality and quantity of the cuts are being controlled by the top temperature K-2 (x1) and by the temperatures of the side-cut distillates to the strippers K-6 (x2), K-7 (x3)and K-9 (x4). At the same time, the process parameters x1, i = 1,4 are controlled by the reflow rates to K-2 (u1), K-6 (u2), K-7 (u3) and K-9 (u4).

The structure of the control system is given in Fig.1. Here 1 is the electropneumatic transducer; 2 is the emf/current converter ; 3 is the A/D converter ; 4 is the D/A converter ; 5 is the active program of the multivariate fuzzy control system.

The fuzzy description of the process with respect to the control and disturbance channels is obtained in the form of linguistic rules. For example,

"IF  $U_i$  is negative medium AND  $X_j$  is negative medium THEN  $X_{ij}$  is zero, OR, "IF  $U_i$  is negative medium AND  $X_j$  is negative small THEN  $X_{ij}$  is negative small, OR

"IF  $U_i$  is negative medium AND  $X_j$  is zero THEN  $X_{ij}$  is negative medium", (see table 1),

		Xj				
		NM	NS	ZE	PS	PM
<b>u</b> 1	NM	ZE	NS	NM	NB	NVB
	NS	PS	ZE	NS	NM	NB
	ZE	PM	PS	ZE	NS	NM
	PS	PB	PM	PS	ZE	NS
	PM	PVB	PB	PM	PS	ZE
				Xij		

Table 1

NVB: negative very big PVB: positive very big here U<sub>i</sub>, Xj and Xij, i,j=1.4 are the process inputs, outputs and derivatives, respectively. The extended term-sets of cooresponding linguistic variables Ui and Xj contain five terms while the term-set of Xij contains nine terms. For example, terms of Ui are,

Negative medium (NM) = {( $u_1, \mu_{11}$  (u)) | $u \in U$ }

Negative small (NS)= {(U1,  $\mu_{12}$  (U))  $|u \in U$  }

Zero(ZE)= {( $u_1, \mu_{13}$  (u)) | $u \in U$  }

Positive small (PS)= {( $u_1, \mu_{14}$  (u)) | $u \in U$  }

Positive medium (PM)= {( $u_1, \mu_{15}$  (u)) | $u \in U$  }

Where

 $\mu_{1k}(\mathbf{u}) = \exp -g_{1k} \cdot |\mathbf{u} - \mathbf{S}_{1k}|$ 

And S<sub>1k</sub> are the elements of corresponding fuzzy sets with membership grade equal to unity. Obtained values of pairs (g,s) for all linguistic variables are given in the tables 2,3 and 4.

Ta	ble	2

	U <sub>1</sub>		IJ <sub>2</sub>		U3		UJ4			
	S	g	S	g	S	g	S	g		
NM	-0.15	33	-0.16	30	-0.18	29	-0.20	28		
NS	-0.08	25	-0.10	24	-0.11	23	-0.12	22		
ZE	0.00	25	0.00	24	0.00	23	0.00	22		
PS	0.08	25	0.10	24	0.11	23	0.12	22		
PM	0.15	33	0.16	30	0.18	29	0.20	28		

Га	bl	e	3

Table -	4	
---------	---	--

	f		0.04		x	
	S	g			S	g
NM	-5.0	0.4		NM	-4.0	2.6
NS	-3.0	0.5		NS	-2.0	4.0
ZE	0.0	1.0		ZE	0.0	10.
PS	3.0	0.5		PS	2.0	4.0
PM	5.0	0.4		PM	4.0	2.6

The investigations of the process revealed an essential dependence among temperatures of side-cut distillates (X<sub>2</sub> and X<sub>3</sub>) on the one hand and crude rate (f) and temperature of side-cut distillate to K-9 (X<sub>4</sub>) on the other hand. The letter circumstance forced to provide conditions of invariance and non-interaction while controlling the temperatures of side-cut distillates X  $i_{,j}$ =1,4. This system structure contains the following tables of linguistic rules.

TLR, i=1,4 are tables offuzzy feedback controllers;

TLR<sub>32</sub> and TLR<sub>23</sub> are tables of fuzzy compensators which provide conditions of non-interaction while controlling  $X_2$  and  $X_3$ ; TLR provides invariance of X4 relative to f; TLR<sub>ij</sub> and TLR<sub>f</sub> represent the fuzzy description of the process.

TLR<sub>32</sub>, TLR<sub>23</sub> and TLR<sub>f</sub> may be obtained using the described approach. At first, using TLR<sub>44</sub> and TLR<sub>f</sub> we developed the fuzzy relations matrices R<sub>44</sub> (U<sub>4</sub>, X<sub>4</sub>, X<sub>4</sub>) and R<sub>f</sub> (f, X<sub>4</sub>, X<sub>4</sub>). Further , on the base of invariance conditions the matrix of the compensator (R<sub>f</sub> (f, U<sub>4</sub>)) is calculated as a result of an inverse composition:

$$R_{f} (f, U_{4}) = \begin{bmatrix} 0,20 & 0,70 & 0,80 & 0,20 & 0,04 & 0,04 \\ 0,10 & 0,30 & 0,90 & 0,30 & 0,10 & 0,04 \\ 0,04 & 0,20 & 0,80 & 0,80 & 0,20 & 0,04 \\ 0,04 & 0,20 & 0,80 & 0,80 & 0,20 & 0,04 \\ 0,04 & 0,10 & 0,30 & 0,90 & 0,30 & 0,10 \\ 0,04 & 0,04 & 0,20 & 0,80 & 0,70 & 0,20 \end{bmatrix}$$

The pairs (g,s) of  $\mathbb{R}f$   $(f, U_4)$  are calculated, for instance, by:

$$\begin{split} F_1 &= NM = \{ 1.00/-5 + 0.45/-3 + 0.20/-1 + 0.09/1 + 0.04/3 + 0.02/5 \} \\ U_{f\,1} &= F_1 \text{ , } R_f = \{ 0.20/-0.2 + 0.70/-0.12 + 0.80/-0.04 + 0.30/0.04 + 0.20/0.12 + \\ &\quad + 0.04/0.2 \} \end{split}$$

All parameters (s,g) of TLR<sub>32</sub> and TLR<sub>23</sub> are calculated analogically (see Table 5).



	U23		U32		Uf	
	S	g	S	g	S	g
NM NS ZE PS PM	-0.07 -0.03 0.00 0.03 0.07	15 50 50 50 15	-0.06 -0.03 0.00 0.03 0.06	25 50 50 50 25	-0.05 -0.03 0.00 0.03 0.06	50 50 25 25 50

# 4.2) Modeling of Fuzzy system for Control of Temperature

Some results of the system's software simulation are shown in Fig.3; the efficiency of the constructed compensator is evident.



Fig.3. Results of software simulation

#### CONCLUSION

In a fuzzy controller the data passes through a preprocessing block, a controller, and a postprocessing block. Preprocessing consist of a linear or non-linear scaling as well as a quantisation in case the membership functions are discretised.

The analysis of some industrial and non-industrial processes show, that they are characterized with uncertainty of their functioning principle, fuzziness of information. In these condition the fuzzy system is effective mathematical tool for modeling and control both industrial and non-industrial processes.

The structure of fuzzy system for technological processes control is given. The functions of its main blocks- fuzzification, inference engine, rule base, defuzzification are described.

The development of fuzzy PD-like controller is performed. Using desired time response characteristic of system and fuzzy model of the processes the fuzzy knowledge base for this controller is developed. Defuzzification mechanizm is realised by using "center of gravity" algorithm.

The modeling of fuzzy system for control of temperature is carried out. In the result of simulation obtained characteristics of system show the efficiency of application of fuzzy controller in complicated process.

## REFERENCES

[1] Fuzzy Process Control and Knowledge Engineering in Petrochemical and Robotic Manufacturing, Refik Aliev, Fuad Aliev and Memetgasan Babae.

[2] Passino, K.M. and Yurkovich, S. (1998). Fuzzy Control, Addison Wesley Longman, Inc, Menlo Park, CA, USA.

[3] Leonid Reznik, Fuzzy Controllers, Victoria University of Technology, Melbourne, Australia.

[4] Hill, G. Horstkotte, E. and Teichrow, J. (1990). Fuzzy-C development system-users manual, Togai Infralogic, 30 Corporate Park, Irvine, CA 92714, USA.

[5] Driankov, D. Hellendoorn, H. and Reinfrank, M. (1996). An introduction to fuzzy control, second edn, Springer-Verlag, Berlin.

[6] Holmblad, L.P and O"stergaard, J.-J. (1982). Control of a cement kiln by fuzzy logic, in Gupta and Sanchez (eds), *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, pp. 389-399. (Reprint in: FLS Review No 67, FLS Automation A/S, Hoffdingsvej 77, DK-2500 Valby, Copenhagen, Denmark).

[7] IEC (1996). Programmable controllers: Part 7 fuzzy control programming. Technical Report IEC 1131, International Electrotechnical Commission.(Draft.).

[8] Lee, C. C.(1990). Fuzzy logic in control systems: Fuzzy logic controller, *IEEE, Trans. System, Man&Cybernetics* **20**(2):404-435.

[9] MIT (1995). &, CITE Literature and Products Database, MIT GmbH / ELITE, Promenade 9, D-52076 Aachen, Germany.

[10] K.M.Kerimov, R.H.Abiyev, T.G.Melikov. Fuzzy system for determination of oil-gas fields productivity / Geophysics News in Azerbaijan. Baku,N:1,1999.pp.13-15.

[11] Pedrycz, W. (1993). Fuzzy controller fuzzy system, second edn, Wiley and Sons, New York. Qiao, W. and Mizumoto, M. (1996). PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems* **78**:23-35.

[12] Siler, W. and Ying, H. (1989). Fuzzy control theory: The linear case, Fuzzy Sets and Systems 33: 275-290.

[13] Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control, IEEE *Trans.Systems,Man & Cybernetics 15*(1): 116-132.

53