EMS702P Statistical Thinking and Applied Machine Learning

Week 5.1-5.3 – Fuzzy Rule Based Systems

Jun Chen

Introduction to Statistics and Machine Learning

Edition: v1.1

Table of Contents

1 Fuzzy Rule Based Systems

In any type of fuzzy logic systems, fuzzy rules using linguistic variables have to be built. These represent the 'knowledge' about the process under investigation. There is more than one type of fuzzy rule-based systems, but the two most popular ones are:

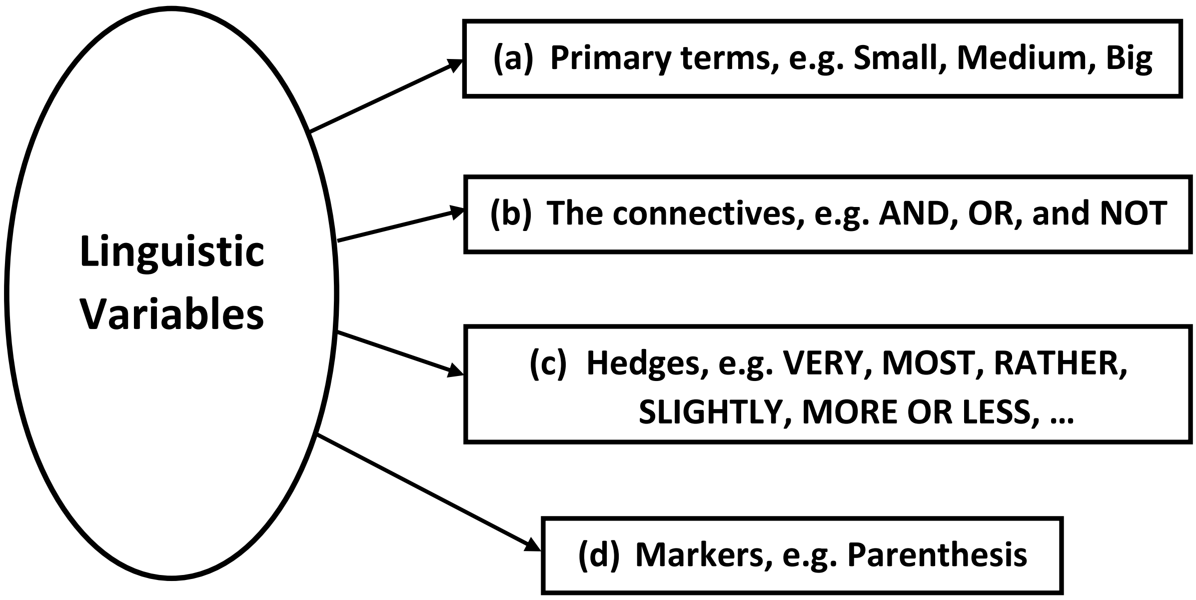
- Mamdani-type fuzzy rule base
- Sugeno-type fuzzy rule base

1.1 Linguistic Variables and Hedges

If fuzzy logic operates with **words** and terms of **natural language**, why all these mathematical formulas?

Because fuzzy sets theory allows for modelling of terms of natural language with the help of **linguistic variables**.

Linguistic variables consist of the following:



For (a), (b) and (d), consider the following examples.

$$Small(but(not(too\ small)))$$

For the linguistic variables, ‘Slow’, ‘Medium’ and ‘Fast’, given in the graph below, express ‘Slow OR Medium’, ‘Slow AND NOT (Fast)’.

For (c), hedges are applied on primary terms to realise 4 (strictly speaking 3) processes.

- Intensification or Concentration
- Dilation
- Fuzzification

Example 1

The operator ‘very’ is a concentration (intensification) operator, which can be defined as

$$Very(u) = u^2$$

Determine the fuzzy set $Very\big(Very(u)\big)$, and illustrate it graphically.

Solution

Example 2

Consider a fuzzy set of 'short pencil' A as:

$$A = \left\{ \frac{0.20}{Pencil\ 1} + \frac{0.5}{Pencil\ 2} + \frac{1}{Pencil\ 3} + \frac{1}{Pencil\ 4} + \frac{0.9}{Pencil\ 5} \right\}$$

Determine the fuzzy set 'Very short pencils'. Now define a new linguistic hedge which 'fuzzifies' as: $hedge1(u) = u^{\frac{1}{2}}$, determine the fuzzy set 'hedge1 pencil'.

Illustrate these two fuzzy sets graphically.

What is the linguistic meaning of 'hedge1'?

Solution

We can use **more than one hedge**, for example

- *Almost very fast but generally below* $100\ km/hr$
- *Close to 100 m but not very high* $(but \equiv and)$

As a summary, linguistic variables and hedges allow us to construct mathematical models for expressions of natural language. These models can then be used to write and process rules and other objects.

1.2 Fuzzy Rules Processing

1.2.1 Mamdani-type fuzzy rules processing

Professor Ebrahim Mamdani (former QMUL Academic) controlled the plant by constructing **fuzzy rules** or **fuzzy conditional statements** of the following form:

$$\textbf{IF } Pressure\ Error\ (PE) \text{ is } Negative\ Big\ (NB) \textbf{ THEN } Heat\ Change\ (HC) \text{ is } Positive\ Big\ (PB)$$

Where $Error = SetPoint - Measured\ Output$

But Mamdani also increased the number of inputs and added '*Change in pressure error (CPE)*' and '*Change in speed error (CSE)*'. These additional inputs add more degrees of freedom to the fuzzy system. *CPE* and *CSE* are indeed the **derivative term** in Mamdani's controller.

Rule-based control was used before Mamdani's controller, fuzzy logic control appear to be so innovative as Mamdani used fuzzy theory to calculate the output according to the set of rules that he formulated. The process is called '**inference mechanism**' or '**inference engine**'.

Mathematically, a rule 'a la Mamdani' would be formulated as follows:

$$\textbf{IF } A_{i1}(x_1), A_{i2}(x_2), \dots, A_{im}(x_m) \textbf{ THEN } y \text{ is } B_i$$

Hence, in the case of the steam-boiler combination used by Mamdani, a typical rule would read:

$$\textbf{IF}\ Pressure\ Error\ (PE)\ \text{is}\ 'NB'\text{and}\ \ Change\ in\ Pressure\ Error\ (CPE)\ \text{is}\ 'N\text{B}'$$

$$\textbf{THEN}\ Heat\ Change\ (HC)\ \text{is}\ Positive\ Big\ (PB)$$

Where,

$$Pressure\ error = \ X_1, Change\ in\ Pressure\ error = X_2$$

$$A_{i1}(x_1) = NB, A_{i2}(x_2) = NB, B_i = PB$$

Linguistic Variables

A linguistic variable is generally defined by the following set:

$$< X_i, LX_i, Z_i, A(X_i) >$$

X_i: Symbolic name of linguistic variable, e.g. pressure, temperature, height, etc.

LX_i: Linguistic value, e.g. NB, PB, PM, etc.

Z_i: Physical domain where the linguistic value is defined (discrete or continuous).

$A(X_i)$: A function that gives an interpretation of a linguistic value, $A(X)$: $LX_i \rightarrow \mu_{LX}(x)$.

Having defined our linguistic variable, we can define a set of ‘m’ rules as:

$$\textbf{IF}\ X_1\ is\ LX_1(k)\ and\ \dots and\ X_n\ is\ LX_n(k)\ \textbf{THEN}\ u\ \text{is}\ LU(k)$$

$$k = 1, \dots, m$$

As another example, see the following rule:

$$\textbf{IF}\ interest\ \text{is}\ low\ \text{and}\ \ price\ \text{is}\ low\ \textbf{THEN}\ invest\ a\ large\ amount\ of\ money$$

There are two techniques of inference mechanism, i.e. **analytical** and **graphical** (rule firing).

Analytical Technique of Inference

Example 3

Consider a fuzzy algorithm relating *yaw error* ε, and *rudder control movement* δ, consisting of only two fuzzy rules (or fuzzy conditional statements):

a) $\textbf{IF}\ \varepsilon\ much\ greater\ than\ \ 10°\text{C}\ \textbf{THEN}\ \delta\ greatly\ reduced$

b) $\textbf{IF}\ \varepsilon\ about\ 15°\text{C}\ \textbf{THEN}\ \delta\ slightly\ reduced$

Where,

$$\varepsilon\ much\ greater\ than\ \ 10°\text{C} = \left\{\frac{0}{10} + \frac{0.2}{12.5} + \frac{0.5}{15} + \frac{0.8}{17.5} + \frac{1}{20}\right\}$$

$$\varepsilon\ about\ \ 15°\text{C} = \left\{\frac{0}{10} + \frac{0.6}{12.5} + \frac{1}{15} + \frac{0.6}{17.5} + \frac{0.1}{20}\right\}$$

and

$$\delta\ greatly\ reduced = \left\{\frac{1}{-10} + \frac{0.8}{-7.5} + \frac{0.4}{-5} + \frac{0.1}{-2.5} + \frac{0}{0}\right\}$$

$$\delta\ slightly\ reduced = \left\{\frac{0}{-10} + \frac{0.1}{-7.5} + \frac{0.5}{-5} + \frac{1}{-2.5} + \frac{0.6}{0}\right\}$$

What is the change in rudder angle required if '*yaw error is about* $20°\text{C}$'?

Where

$$yaw\ error\ is\ about\ 20°\text{C} = A = \left\{\frac{0}{10} + \frac{0.1}{12.5} + \frac{0.4}{15} + \frac{0.8}{17.5} + \frac{1}{20}\right\}$$

Solution

Step 1: Calculate the relational matrix for each rule

Step 2: Calculate the overall relational matrix

Step 3: Use the composition rule of inference to find the change in rudder angle given the yaw error

Step 4: Defuzzify the output membership function to obtain the crisp value of the change in rudder angle

The obvious **drawbacks** associated with the analytical method is it is computationally expensive, especially when the number of rules + number of antecedents and consequents increases (multivariable case). In such a case, we will need the approach of '**firing the rules**', which can be illustrated graphically through the following example.

Graphical Technique of Inference

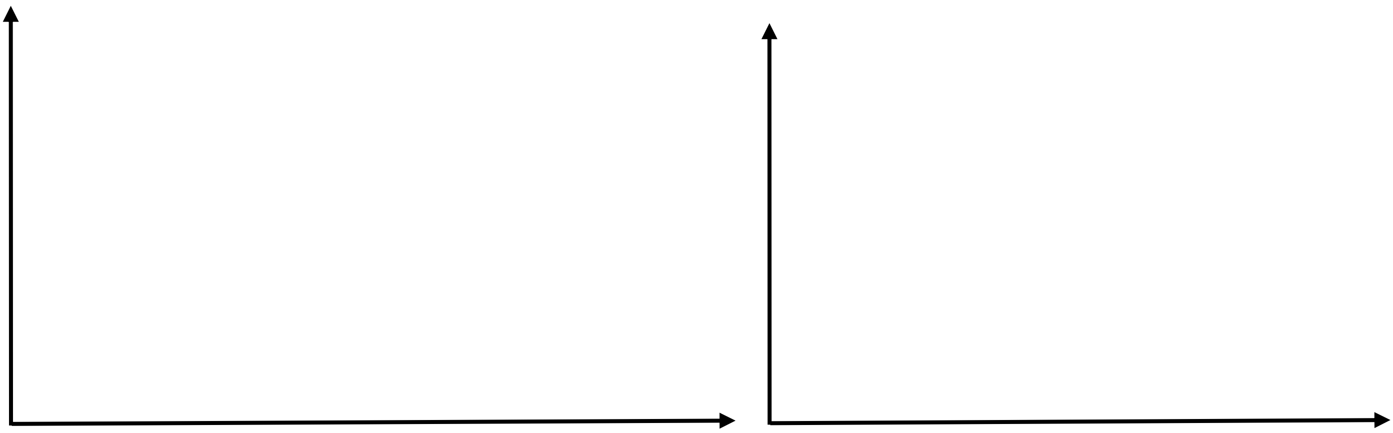
Example 4

Consider the following set of rules:

1) $\textbf{IF}$ $pressure$ is NB $\textbf{THEN}$ $time$ is $short$
2) $\textbf{IF}$ $pressure$ is NS $\textbf{THEN}$ $time$ is $short$
3) $\textbf{IF}$ $pressure$ is $Zero$ $\textbf{THEN}$ $time$ is $average$
4) $\textbf{IF}$ $pressure$ is PS $\textbf{THEN}$ $time$ is $long$
5) $\textbf{IF}$ $pressure$ is PB $\textbf{THEN}$ $time$ is $long$

What would be the fuzzy output if a crisp value of pressure of $-22\ Kpa$ is recorded?

Solution



The reason we need to fire the rules is to **execute the mapping between the fuzzy input and fuzzy output**, in other words we need to transform the membership degree from the antecedent part to the consequent part.

In the about case we have two fired rules, i.e. (short, 0.80) and (short, 0.05), we choose (short, 0.80) as we use **OR** across different fired rules.

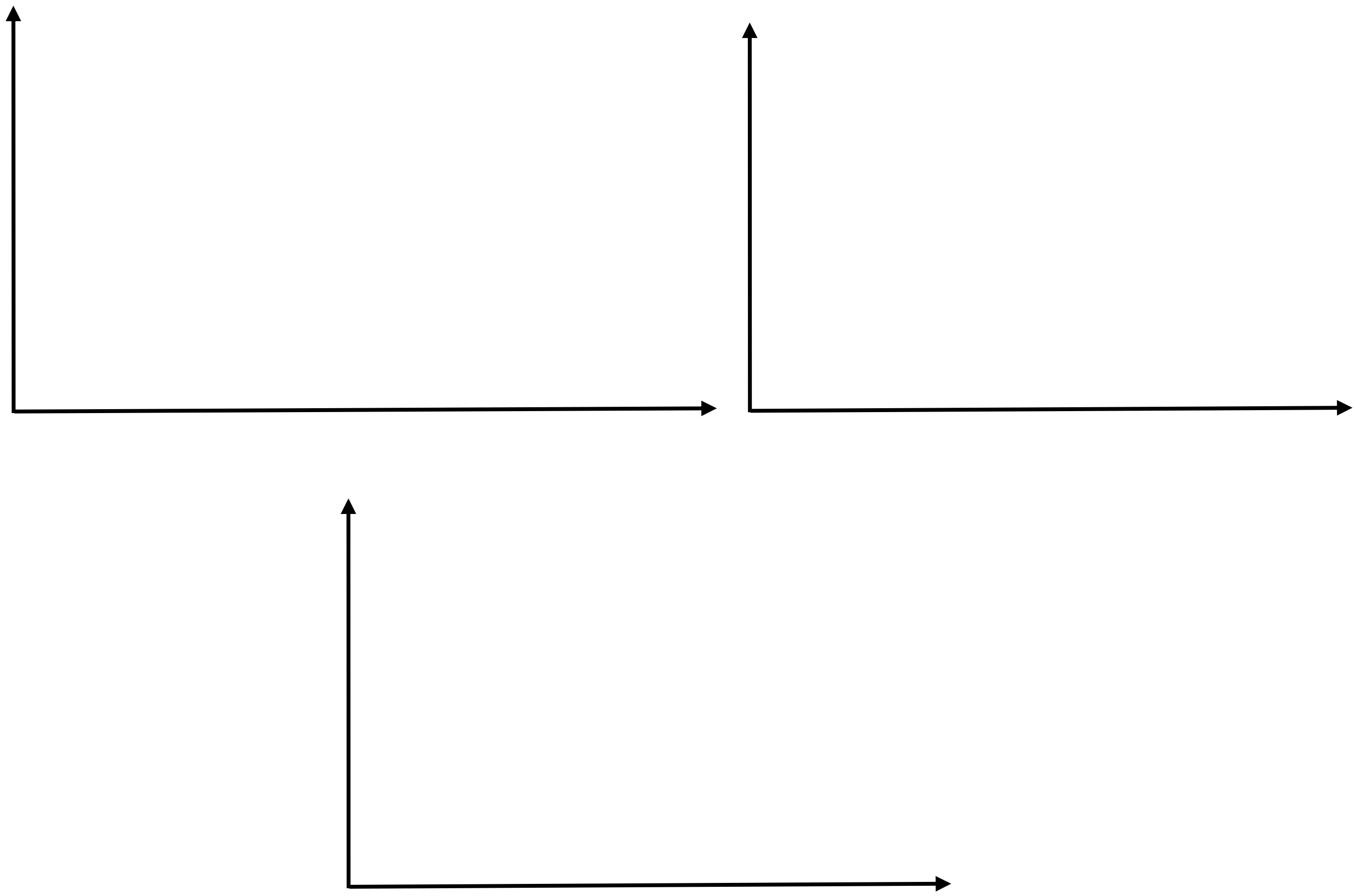
Example 5

Consider the following set of rules:

1) **IF** $pressure$ is NB and $temperature$ is $high$ **THEN** $time$ is $short$
2) **IF** $pressure$ is NS and $temperature$ is $Mdium$ **THEN** $time$ is $Average$
3) **IF** $pressure$ is $Zero$ and $temperature$ is $high$ **THEN** $time$ is $short$

What would be the fuzzy output if a crisp value of pressure of $-22\ Kpa$ and temperature of 22℃ are recorded?

Solution



1.2.2 Sugeno-type fuzzy rules processing

Takagi and Sugeno (1985) have proposed a type of fuzzy system where the system output instead of being a linguistic term, is a numerical function of the input variables. They replaced Mamdani's **consequent part** only by following function 'f' such that $y = f(x_1, x_2, \dots, x_n)$.

Hence, a rule 'a la Sugeno' would read as follows:

$$\mathbf{IF}\ X_1\ is\ LX_1(k)\ and \dots and\ X_n\ is\ LX_n(k)\ \mathbf{THEN}\ u = f_k(x_1, x_2, \dots, x_n)$$

It is noted that 'f' can be either linear or quadratic.

As an example, see the following rule:

$$\textbf{IF } pressure \text{ is } NB \text{ and } temperature \text{ is } high \textbf{ THEN } time = 0.3 \cdot pressure + 0.5 \cdot temperature$$

Where, 0.30 and 0.50 simply express a certain dependency of time on pressure and temperature.

The inference of Sugeno-type of fuzzy rule-based system is simply a weighted sum average of the consequent part as follows.

$$y = \frac{\sum_j^k (A^j(X_1) \cap \ldots \cap A^j(X_n)) f_j(x_1, x_2, \ldots, x_3)}{\sum_j^k (A^j(X_1) \cap \ldots \cap A^j(X_n)))}$$

1.2.3 Comparison of Mamdani's processing to Sugeno's fuzzy processing

Mamdani-type of fuzzy processing	Sugeno-type of fuzzy processing
Consequent parts are fuzzy	Consequent parts are singletons or mathematical functions of them
• Easily understandable by a human expert • Rules simpler to formulate • Proposed earlier and commonly used	• More effective computationally • More convenient in mathematical analysis and in systems analysis • Guaranteed continuity of output surface

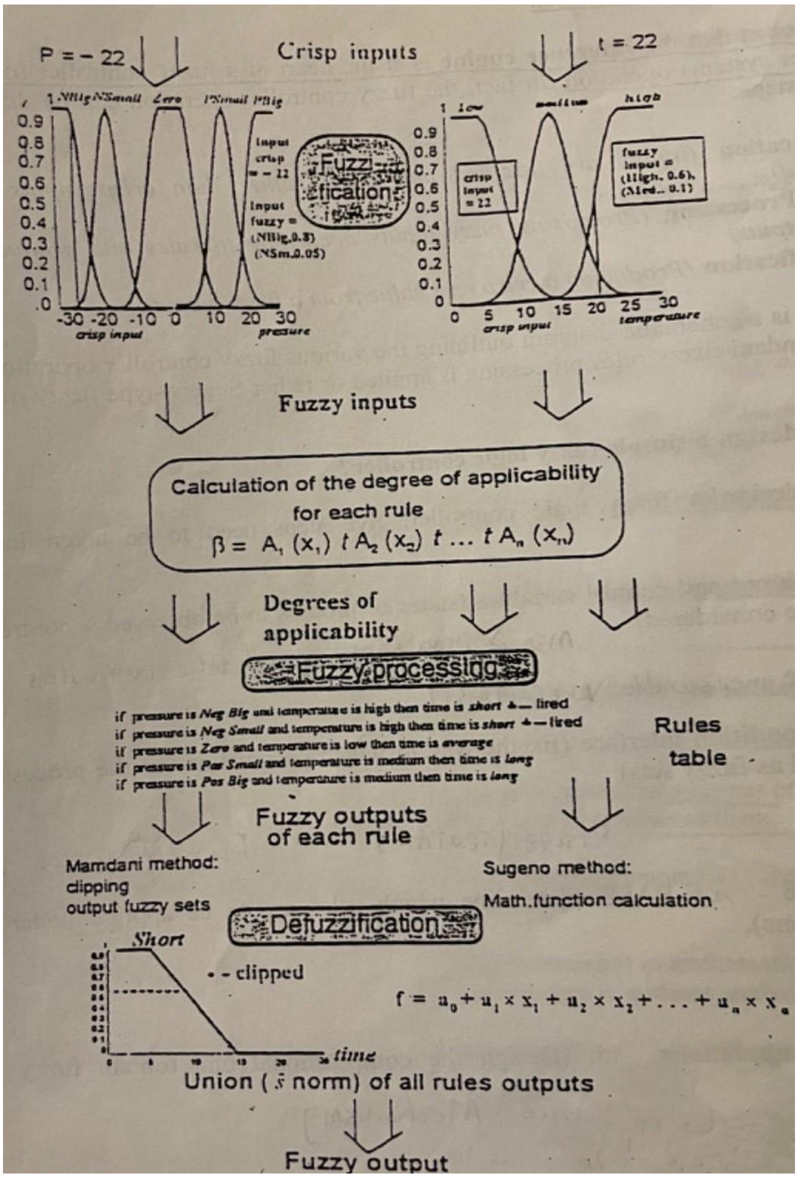
2 Fuzzy Rule Based System Operation

We have seen that the **inference engine** is at the heart of any fuzzy rule based system operation. In fact, fuzzy rule based system operation can be divided into three steps:

- **Fuzzification** (inputs are fuzzified and fuzzy rather than crisp inputs are obtained).
- **Fuzzy Processing** (processing fuzzy inputs according to rules and producing fuzzy outputs).
- **Defuzzification** (producing a crisp real value from a fuzzy output).

2.1 How to Design a Simple Fuzzy Rule Based System

The figure below is a schematic diagram outlining the various fuzzy rule based systems operations, whether Mamdani fuzzy rule processing is applied or rather Sugeno-type fuzzy rules processing.



In order to design a fuzzy rule based system, five steps need to be taken into consideration:

Step 1: Define the input and output variables (features/states of the system/process to be observed + outputs/control actions to be considered).

Step 2: Define the condition interface (fix the way in which observations of the system/process are expressed as fuzzy sets).

Step 3: Design the rule base (determine which rules are to be applied under which conditions)

Step 4: Design the computational unit (design the computational unit to obtain fuzzy outputs).

Step 5: Determine rules according to which fuzzy statements can be transformed into crisp control actions.

Step 3 is at the heart of the fuzzy rule base design! To design the rule base, those rules can be obtained through:

- an expert: his/her understanding of the system/process under investigation.
- historical data: rules will be automatically elicited/learnt from data.

We will first look into how to building a fuzzy rule based system through experts in this lecture, and leave the rule elicitation to Week 11.

2.2 An Example for Building a Fuzzy Expert System

A fuzzy vacuum cleaner should regulate the force of sucking dust from a surface being cleaned.

- The input of this fuzzy vacuum cleaner should consider the amount of dust on the surface which can be
    - Very dirty
    - Dirty
    - Rather dirty
    - Almost clean
    - Clean
- The force of sucking dust can be described as linguistic variables with values of
    - Very strong
    - Strong
    - Ordinary
    - Weak
    - Very weak

The fuzzy vacuum cleaner should be able to change the force depending on how dirty the surface is. Hence, as the first attempt, we may have the following fuzzy rule based system as the fuzzy vacuum cleaner.

$$\begin{array}{llll} \boldsymbol{IF} & surface\ is\ \boldsymbol{very\ dirty} & \boldsymbol{THEN} & force\ is\ \boldsymbol{very\ strong} \\ \boldsymbol{IF} & surface\ is\ \boldsymbol{dirty} & \boldsymbol{THEN} & force\ is\ \boldsymbol{strong} \\ \boldsymbol{IF} & surface\ is\ \boldsymbol{rather\ dirty} & \boldsymbol{THEN} & force\ is\ \boldsymbol{ordinary} \\ \boldsymbol{IF} & surface\ is\ \boldsymbol{almost\ clean} & \boldsymbol{THEN} & force\ is\ \boldsymbol{weak} \\ \boldsymbol{IF} & surface\ is\ \boldsymbol{clean} & \boldsymbol{THEN} & force\ is\ \boldsymbol{very\ weak} \end{array}$$

Example 6

Can this fuzzy vacuum cleaner be improved by adding extra knowledge of vacuum cleaning process?

Solution

Rules Table for a Fuzzy Vacuum Cleaner					
	Clean	**Almost Clean**	**Rather Dirty**	**Dirty**	**Very Dirty**

A fuzzy vacuum cleaner like the above has been implemented firstly in a Japanese manufactured vacuum cleaner. Similar systems like this has been widely used in context-aware smart products, like iRobot (https://www.irobot.com/) that allows users to hack the system for further development (see [2]).

Example 7

Would this controller work on a different surface, like linen fabric surface? How is this vacuum cleaner implemented practically?

Solution

3 Fuzzy Rule Based System as a Fuzzy Controller

Rule Based control was used before Mamdani's controller. What makes fuzzy logic control so innovative lies in the way all those rules were processed. We need fuzzy control as

- It differs significantly from conventional control.
- It does not use mathematical models of the process under investigation.
- It is able to comprehend instructions and generate strategies based on verbal communication.
- Control engineers can relate to it better than they would do with conventional methods.

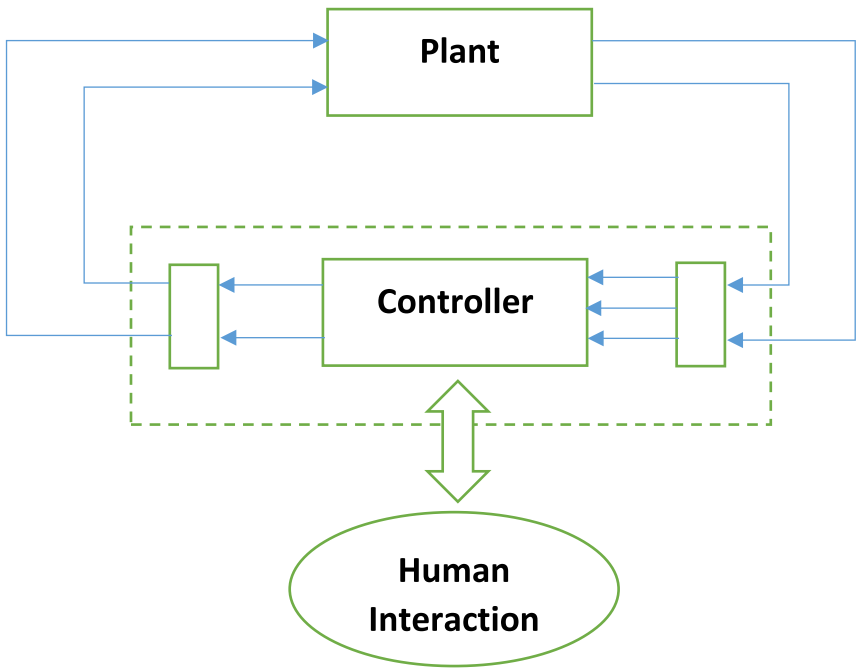
The above does not imply that people have been applying the wrong numeric model. They have been using mathematical models which can describe "at best" the process under investigation. However, systems/processes can be characterised by

- Uncertainty
- Non-linearities
- Unmodelled dynamics

In this case, mathematical models will be ineffective. In addition, fuzzy logic control came at a crucial time where dissatisfaction with conventional control was growing. Fuzzy Logic Control can be said to

- Be able to translate imprecise/vague knowledge of human experts.
- Be simple and easy to implement.
- Be fully supported by software design and hardware implementation.
- Be able to generate smooth control.
- Be able to produce robust control behaviour.
- Be able to solve the curse of dimensionality.
- Be able to control unstable systems.

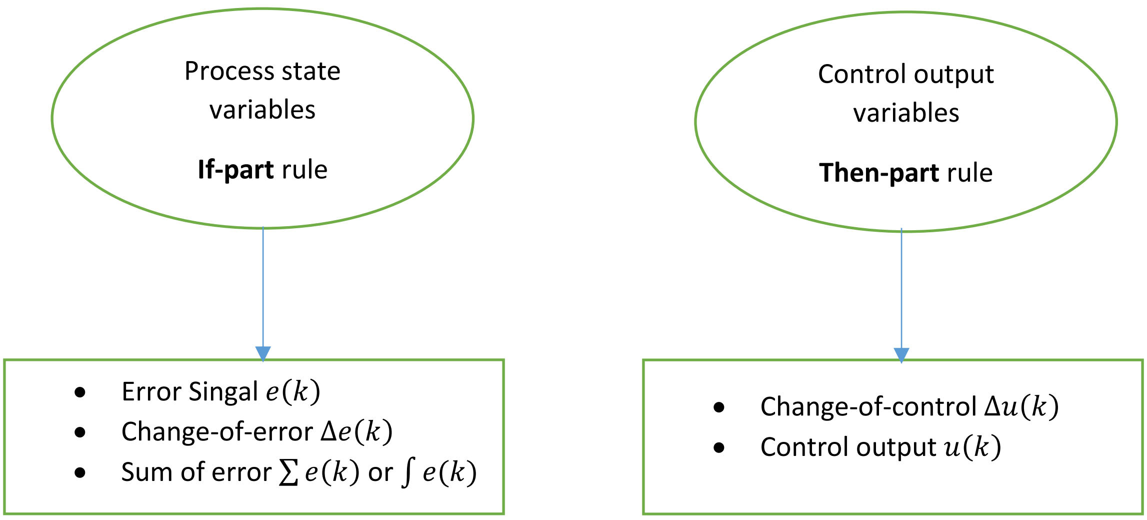
E. H. Mamdani constructed the first fuzzy controller (see below and [3], [4]) to control a plant comprising a steam-engine and boiler combinations.



The plant characteristics feature highly non-linearity (different behaviour at various operating points; a conventional DDC has to be re-tuned all the time to reach the “best possible” performance) and presence of time-delay.

In order to design a PID-like fuzzy controller it is necessary to choose the input-output variables and the rules of the controller properly.

If a decision has already been made to design a P-, PD-, PI- or PID-like controller, it implies automatically a choice of the process state(s) and control output variables as well as the content of the rule-**antecedent** and the rule-**consequent** parts of each rule.



Where

$$e(k) = \omega - y(k)$$

$$\Delta e(k) = e(k) - e(k-1) = y(k-1) - y(k)$$

$$\Delta u(k) = u(k) - u(k-1)$$

$$k: sample\ number$$

$$y(k): outputvalue\ at\ sample\ k$$

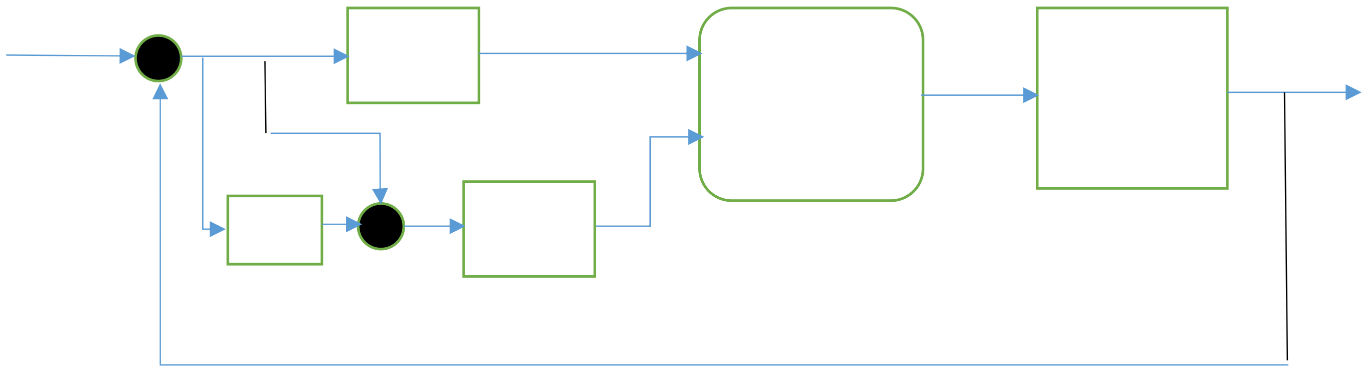
$$\omega: set\ point\ value$$

3.1 Design a PD-like Fuzzy Controller

In the case of a PD-controller, the controller output can be written as follows

$$u(k) = K_p e(k) + K_D \Delta e(k)$$

Where K_p and K_D are the proportional and derivative relating to the controller. A control system built around this controller's expression can be represented by the following schematic diagram.



Let us describe the above question with the help of rules:

The fuzzy controller should be able to calculate the control signal u given a pair of values e and Δe (just like the conventional PD controller.

The PD-like fuzzy controller consists of rules and a symbolic description of each rule is given as follows:

$$\boldsymbol{IF}\ e(k)\ is < \boldsymbol{property\ symbol} >\ and\ \Delta e(k)\ is < \boldsymbol{property\ symbol} >$$

$$\boldsymbol{THEN}\ u(k)\ is < \boldsymbol{property\ symbol} >$$

$< property\ symbol >= linguistic\ value$ (e.g. High, Positive Big, etc.)

Example of such a rule is

$\boldsymbol{IF}\ e(k)\ is$ *"Postive Big" and Δe(k) is "Negative Big"* ***THEN*** *u is "Negative Small"*

Note:

- $e(k) < 0 \rightarrow$ output is above the set point (overshoot).
- $\Delta e(k) < 0 \rightarrow y(k-1) - y(k) < 0 \rightarrow y(k-1) < y(k) \rightarrow y(k)$ has increased.
- $\Delta e(k) = 0 \; and \; e(k) = 0 \rightarrow y(k)$ is about the set point and has not changed from its previous position.

3.2 Rules Table Notation

Let us consider the following convenient way of writing down a set of rules for two antecedents and one consequent:

$e(k)$ \ $\Delta e(k)$	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Group 1:

Group 2:

Group 3:

Group 4:

Group 5:

Example 8

Do all PD-like fuzzy controllers have the same configuration as that of the above table?

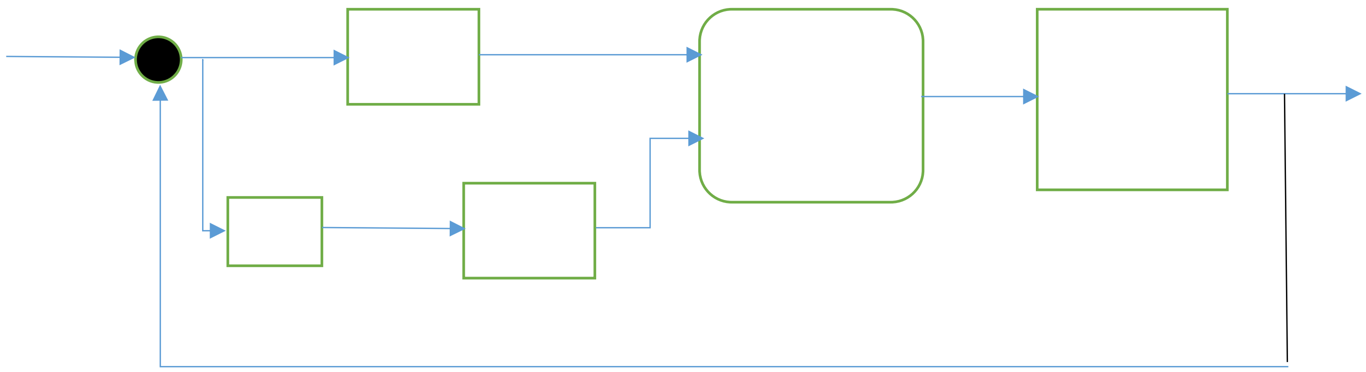
Solution

3.3 Design a PI-like Fuzzy Controller

In the case of a PI-controller the controller output can be written as follows:

$$u(t) = K_p e(t) + K_I \int e(t)\,dt$$

Where K_p and K_I are the proportional and integral relating to the controller. A control system built around this controller's expression can be represented by the following schematic diagram.



If we follow the same procedure as the one in Section 3.2, we would be formulating rules according to ***error*** and ***sum-of-error*** (or ***integral of error***); this can prove to be a difficult exercise, because we will inevitably generate a relatively large universe of discourse for *sum-of-error*.

To solve this problem, let us change the above equation into the following:

$$\frac{du(t)}{dt} = K_p \frac{de(t)}{dt} + K_I e(t)$$

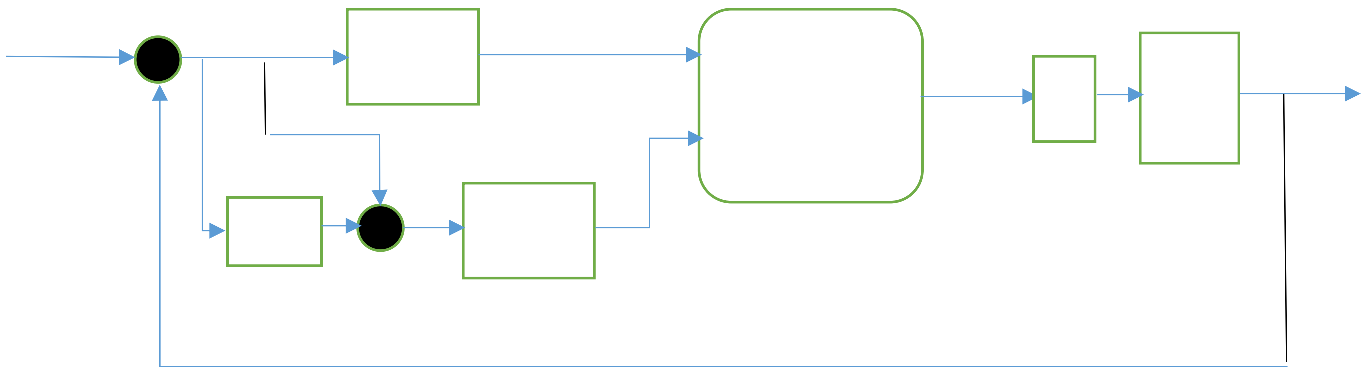
Hence,

$$\Delta u(k) = u(k) - u(k-1) = K_p \Delta e(k) + K_I e(k)$$

$$u(k) = u(k-1) + K_p \Delta e(k) + K_I e(k)$$

The above equation means that **the controller gives a change of control action rather than the control action itself**.

Hence, the following diagram is obtained instead:



Note:

- As far as the rules are concerned, the table in Section 3.2 can still be used but the controller gives ***change-of-control***.
- The PI-like fuzzy controller consists of rules and a symbolic description of each rule is given as follows:

$$\boldsymbol{IF}\ e(k)\ is < \boldsymbol{property\ symbol} > and\ \Delta e(k)\ is < \boldsymbol{property\ symbol} >$$

$$\boldsymbol{THEN}\ \Delta u(k)\ is < \boldsymbol{property\ symbol} >$$

Example 9

Consider the following minors changes made to the table in Section 3.2, what will be the implications of such changes?

$\Delta e(k)$ c $e(k)$	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	**NS->NM**	ZE	**PS->PM**	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

$\Delta e(k)$ $e(k)$	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	**ZE->PS**	PS	PM	PB	PB	PB	PB

Solution

Note: One should not always use error and change-of-error. Instead the states of the system can be introduced in the rule specification process.

4 Fuzzy Rule Based System Parameter Choice

A fuzzy rule based system design process includes the same steps as any other design process, i.e.,

- Initial choice of structure and parameters.
- Subsequent adjustment based on the analysis.

The first step possesses a high degree of subjectivity. As a result, this step will require a high effort in order to be effectively implemented. Hence, the better the initial choice is, the simpler the second step will be.

4.1 Scaling Factors Choice

A fuzzy rule based system includes fuzzification and defuzzification. The membership functions which describes the different values of the linguistic variables are applied during both stages (fuzzification and defuzzification).

In the choice of membership functions, the definition universe of discourse is of prime importance. We can identify two major problems:

- The range covered by the universe of discourse is too small

- The range covered by the universe of discourse is too large

But, the correct choice would cover the whole range without going over, i.e.,

However, in general it is not easy to give general recommendations, although one successful method is being applied which is scaling or normalisation of the universe of discourse of input and output variables, i.e. apply the range $[-1, +1]$ for inputs and outputs. Hence, in building a fuzzy rule based system, one should include both the input and output scaling factors.

As far as tuning the fuzzy controller is concerned, let us assign a few priorities:

- The output scaling factor has the most influence on stability and oscillation tendency – **Priority 1**

- The input scaling factors have the most influence on the basic sensitivity of the controller – **Priority 2**
- The shape and location of the input and output membership functions may influence the behaviour of the controlled system in different areas of the state space – **Priority 3**

The experience with PID helps. However, comparison is qualitative rather than quantitative.

4.2 Membership Function Choice

In the membership function choice, one has to consider the following parameters:

- The number of classes
- The position of the various membership functions
- The width of the membership function
- The peak of the membership function
- The shape of the membership function

Example 10

Let us consider the following membership functions:

Is this representation good or bad?

Solution

4.3 Fuzzy Rule Formulation

In relation to rule formulation one question can be asked – how do we find the rules practically?

- Redeveloping, manuals, operation instructions, and any other documents available into a set of useful rules.
- Interrogation of experienced experts using a carefully organised questionnaire.
- Rules can also be formulated by observing how a skilful operator operates/controls the system/process.

There are other methods which are based on measured data, for instance:

- Synthesis a set of rules based on the model of the system/process.
- Using a learning mechanism that will derive/elicit the rules online or offline.

Usually, rules are formulated one by one; after this one has to analysis the whole set of rules. In this analysis, one needs to determine if the set of rules is **complete**, **consistent** and **continuous**.

- A set of rules is complete if any combination of input values results in an appropriate value. ***Any combination of inputs should fire at least one rule***.
- A set of rules is consistent if it does not contain contradictions.

  e.g.,

$\boldsymbol{IF}\ e(k)\ is\ "Zero"\ and\ \Delta e(k)\ is\ "Zero"\ \boldsymbol{THEN}\ u\ is\ "Zero"$

$\boldsymbol{IF}\ e(k)\ is\ "Zero"\ and\ \Delta e(k)\ is\ "Zero"\ \boldsymbol{THEN}\ u\ is\ "Negative\ Big"$

One rule must be excluded.

- A set of rules is continuous if it does not have neighbouring rules with output fuzzy sets that have an empty intersection (see Example 9), i.e., if small changes in inputs result in jump over a neighbouring class, the rules set is discontinuous.

4.4 Choice of the Defuzzification Procedure

Defuzzification is essential in Mamdani-type fuzzy rule based systems but is not required in Sugeno-type fuzzy rule based systems. There are more than 5 defuzzification methods, but only two received much attention:

- Centre (centroid) of Area/Gravity method
- Mean of Maxima method

The reason to have so many methods is because none of them have proved to be more advantageous over one another.

The criteria for comparison should be

- Continuity – whether a small change in the input generates a large change in the output.
- Un-ambiguity – whether it works in any situation.
- Plausibility – produces an output that lies approximately in the middle of the support of the resulting membership function and has a high degree of membership.
- Computational complexity.

Example 11

The following table can be produced with reference to the two most popular defuzzification methods. Fill in the empty cells.

Comparison of Two Defuzzification Methods		
	Centre of Area Method	**Mean of Maxima Method**
Continuity		
Unambiguity		
Plausibility		
Computational Complexity		

5 Further Readings

[1] Mendel, J.M., 1995. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, *83*(3), pp.345-377.

[2] Liu, A., Teo, I., Chen, D., Lu, S., Wuest, T., Zhang, Z. and Tao, F., 2019. Biologically inspired design of context-aware smart products. *Engineering*, *5*(4), pp.637-645.

[3] Lilly, J.H., 2011. *Fuzzy control and identification*. John Wiley & Sons.

[4] Passino, K.M., Yurkovich, S. and Reinfrank, M., 1998. *Fuzzy control* (Vol. 42, pp. 15-21). Reading, MA: Addison-wesley.