Properties of fuzzy set – fuzzy logic control principles

Fuzzy sets can be considered as an extension and gross oversimplification of classical sets. It can be best understood in the context of set membership. Basically, it allows partial membership which means that it contains elements that have varying degrees of membership in the set.

Properties of Fuzzy Sets

Let us discuss the different properties of fuzzy sets.

Commutative Property

Having two fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, this property states −

$$\widetilde{A} \cup \widetilde{B} = \widetilde{B} \cup \widetilde{A}$$

$$\widetilde{A} \cap \widetilde{B} = \widetilde{B} \cap \widetilde{A}$$

Distributive Property

Having three fuzzy sets $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$, this property states −

$$\widetilde{A} \cup \left(\widetilde{B} \cap \widetilde{C}\right) = \left(\widetilde{A} \cup \widetilde{B}\right) \cap \left(\widetilde{A} \cup \widetilde{C}\right)$$

$$\widetilde{A} \cap \left(\widetilde{B} \cup \widetilde{C}\right) = \left(\widetilde{A} \cap \widetilde{B}\right) \cup \left(\widetilde{A} \cap \widetilde{C}\right)$$

Idempotency Property

For any fuzzy set $\widetilde{A}$, this property states −

$$\widetilde{A} \cup \widetilde{A} = \widetilde{A}$$

$$\widetilde{A} \cap \widetilde{A} = \widetilde{A}$$

Identity Property

For fuzzy set $\widetilde{A}$ and universal set U, this property states −

$$\widetilde{A} \cup \varphi = \widetilde{A}$$

$$\widetilde{A} \cap U = \widetilde{A}$$

$$\widetilde{A} \cap \varphi = \varphi$$

$$\widetilde{A} \cup U = U$$

Transitive Property

Having three fuzzy sets $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$, this property states −

$$If\ \widetilde{A} \subseteq \widetilde{B} \subseteq \widetilde{C},\ then\ \widetilde{A} \subseteq \widetilde{C}$$

Involution Property

For any fuzzy set $\widetilde{A}$, this property states −

$$\overline{\overline{\widetilde{A}}} = \widetilde{A}$$

De Morgan's Law

This law plays a crucial role in proving tautologies and contradiction. This law states −

$$\overline{\widetilde{A} \cap \widetilde{B}} = \overline{\widetilde{A}} \cup \overline{\widetilde{B}}$$

$$\overline{\widetilde{A} \cup \widetilde{B}} = \overline{\widetilde{A}} \cap \overline{\widetilde{B}}$$

Sugeno Fuzzy Model

The Sugeno FIS is quite similar to the Mamdani FIS. The primary difference is that the output consequence is not computed by clipping an output membership function at the rule strength. In fact, in the Sugeno FIS there is no output membership function at all. Instead the output is a crisp number computed by multiplying each input by a constant and then adding up the results. This is shown in Figure 9. "Rule strength" in this example is referred to as "degree of applicability" and the output is referred to as the "action". Also notice that there is no output distribution, only a "resulting action" which is

the mathematical combination of the rule strengths (degree of applicability) and the outputs (actions).

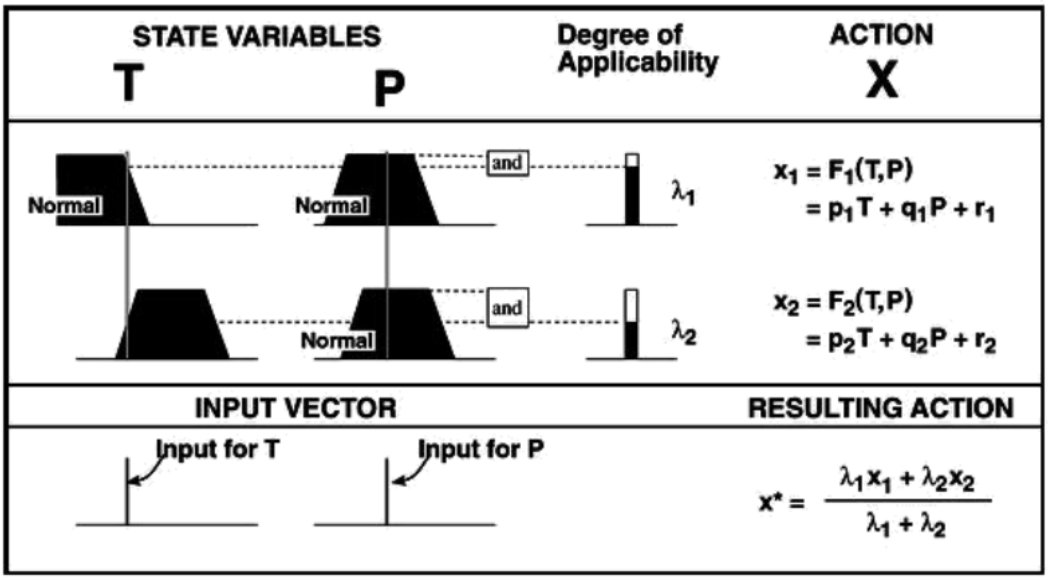


Fig 9. A two input, two rule Sugeno FIS (pn, qn, and rn are user-defined constants)

One of the large problems with the Sugeno FIS is that there is no good intuitive method for determining the coefficients, p, q, and r. Also, the Sugeno has only crisp outputs which may not be what is desired in a given HCI application. w hy then would you use a Sugeno FIS rather than a Mamdani FIS? The reason is that there are algorithms which can be used to automatically optimize the Sugeno FIS.

In classification, p and q can be chosen to be 0 and r can be chosen to be a number that corresponds to a particular class. For example, if we wanted to use the EMG from a person/persons forearm to classify which way his/her wrist was bending, we could assign the class "bend_inward" to have the value r = 1. we could assign the class "bend_outward" to have the value r=0. Finally, we could assign the class "no_bend" to have the value r=0.5.

Tsukamoto fuzzy model

In the *Tsukamoto fuzzy models*, the consequent of each fuzzy if-then rule is represented by a fuzzy set with a monotonical membership function, as shown in Figure 10. As a result, the inferred output of each rule is defined as a crisp value induced by the rule's firing strength. The overall output is taken as the weighted average of each rule's output. Figure 10 illustrates the reasoning procedure for a two-input two-rule system.

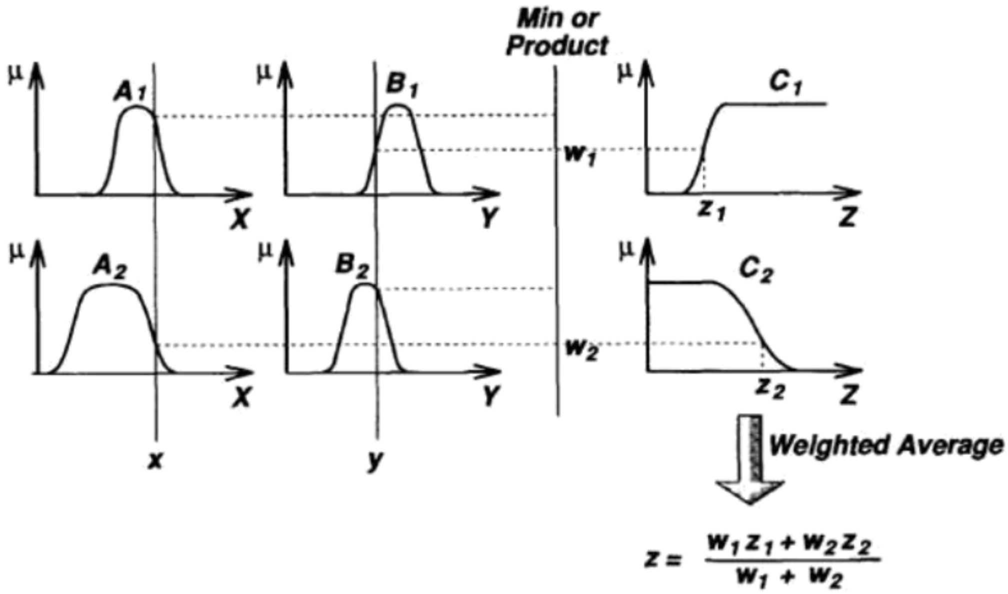


Fig 10. The Tsukamoto fuzzy model

Since each rule infers a crisp output, the Tsukamoto fuzzy model aggregate each rule's output by the method of weighted average and thus avoids the time-consuming process of defuzzification. However, the Tsukamoto fuzzy model is not used often since it is not as transparent as either the Mamdani or Sugeno fuzzy models. The following is a single-input example.

Example: Single-input Tsukamoto fuzzy model

An example of a single-input Tsukamoto fuzzy model can be expressed as:

$$\begin{cases} IF\ X\ is\ small\ then\ Y\ is\ C_1 \\ IF\ X\ is\ medium\ then\ Y\ is\ C_2 \\ IF\ X\ is\ large\ then\ Y\ is\ C_3 \end{cases}$$

w here the antecedent MFs for "small", "medium", and "large" are shown in Figure 11(a), and the consequent MFs for "C1", "C2", and "C3" are shown in Figure 11(b). The overall input-output

curve, as shown in Figure 11(d), is equal to $(\sum_{i=1}^{3} w_i f_i)/(\sum_{i=1}^{3} w_i)$, where *fi* is the output of each rule induced by the firing strength *Wi* and MF for *Ci*. If we plot each rule's output *fi* as a function of *x*, we obtain Figure 11(c), which is not quite obvious from the original rule base and MF plots.

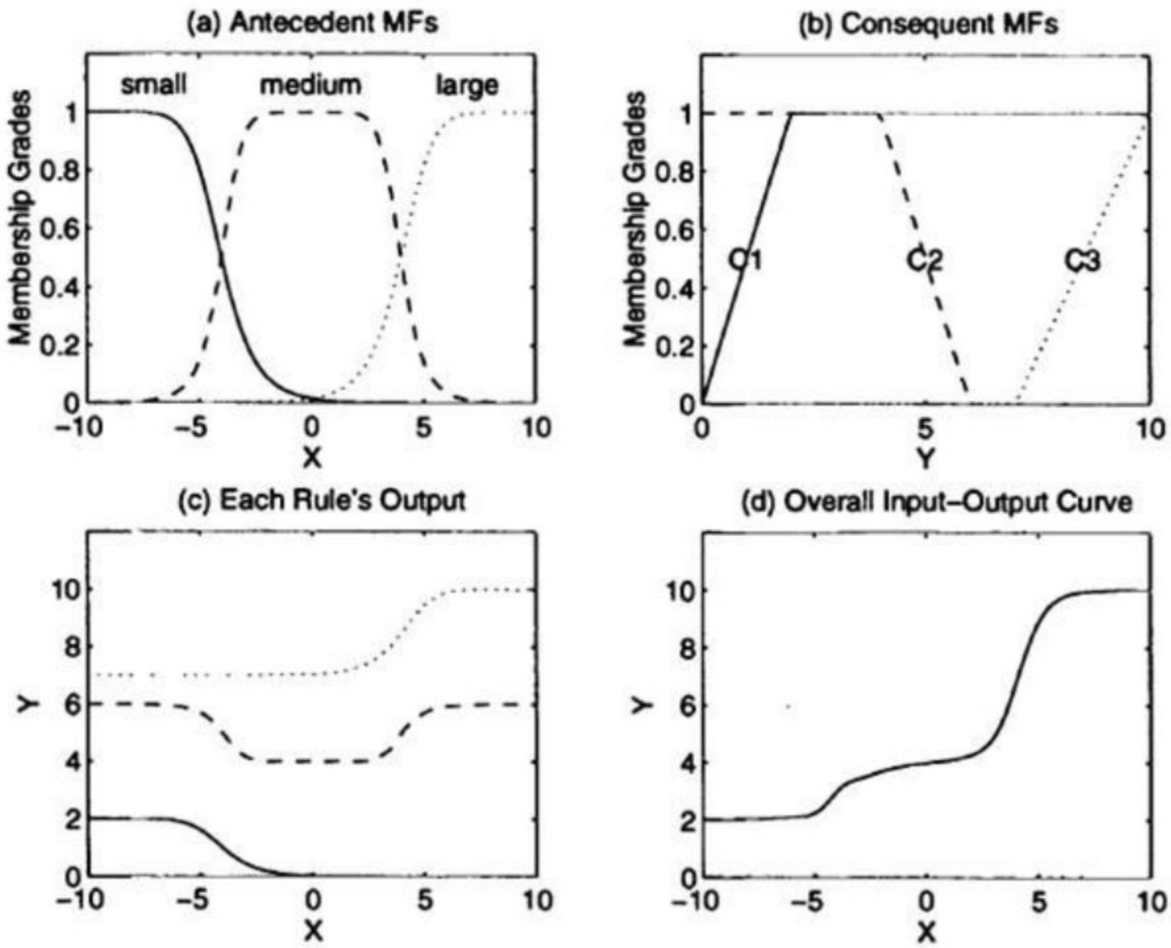


Fig 11. Single-input single output Tsukamoto fuzzy model: (a) antecedent MFs; (b) consequent MFs; (c) each rule's output curve; (d) overall input-output curve.

Since the reasoning mechanism of the Tsukamoto fuzzy model does not follow strictly the compositional rule of inference, the output is always crisp even when the inputs are fuzzy.

FUZZY LOGIC CONTROLLER

Assumptions in a fuzzy control system design

The assumptions made in the design of a fuzzy logic controller are:

- The plant is observable and controllable
- A body of knowledge exists for building rules, assigning membership values to the input and output variables
- A solution exists
- The control engineer is looking for a "good enough" solution, not necessarily the optimum one
- The controller will be designed to best of available knowledge and within an acceptable range of precision
- The problems of stability and optimality are still open problems in fuzzy controller design

Steps in the design of a fuzzy logic controller

The steps in the design of a simple fuzzy logic control system are as follows:

- Identify the variables of the plant (input and output)
- Partition the university of discourse of each variable into a number of fuzzy subsets, assigning each a linguistic label
- Assign or determine a membership function for each fuzzy subset
- Form the rule base
- Choose appropriate scaling factors for the input and output variables in order to normalize the variables to the [0,1] interval
- Fuzzify the inputs to the controller
- Use fuzzy approximate reasoning to infer the output contributed from each rule
- Aggregate the fuzzy outputs recommended by each rule
- Apply defuzzification to form a crisp output

Features of a simple Fuzzy Logic Control system

A simple fuzzy logic control system has the following features:

- Fixed and uniform input and output scaling factors
- Flat, single partition rule-base with fixed and non interactive rules
- Fixed membership functions

- Limited number of rules, which increase exponentially with the number of inputs
- Fixed knowledge
- Low-level control and no hierarchical rule structure

Why Should We Use Fuzzy Controllers?

• Very robust

• Can be easily modified

• Can use multiple inputs and outputs sources

• Much simpler than its predecessors (linear algebraic equations)

• Very quick and cheaper to implement

Constructing a Fuzzy Controller

5. Create the membership values (fuzzify).

6. Specify the rule table.

7. Determine your procedure for defuzzifying the result.

Fuzzy Logic Controller

The general structure of a Fuzzy Logic Controller (FLC) is as shown in fig. below:

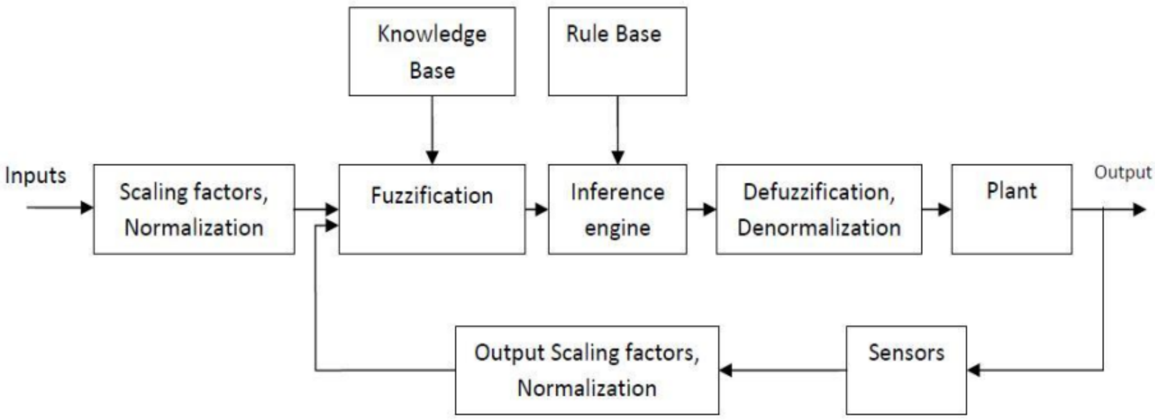


Fig. A fuzzy logic controller

Normalization:

Normalization maps the physical values of the current process state variables into a normalized university of discourse. It also maps the normalized value of the control output variable onto its physical domain. w hen a non-normalized domain is used then there is no need for this block.

Fuzzification:

This block performs the conversion of the crisp input variable to fuzzy variable. This is done so as to enable the proper functioning of fuzzy controller. The design parameter of this block is the choice of fuzzification strategy. The choice of the fuzzification strategy depends on the type of the inference engine or rule firing employed in the particular application. The rule firing is either composition based or individual rule-firing based.

Knowledge base:

The knowledge base of a fuzzy logic controller consists of data base and rule base.

Data Base:

The basic function of data base is to provide the necessary information for the proper functioning of the fuzzification module, the rule base and the defuzzification module. This information includes:

- Fuzzy sets (membership functions) representing the meaning of the linguistic values of the process and the control output variables. These concepts are subjectively defined based on experience and engineering judgment.
- Physical domains and their normalized counterparts together with the normalization or de-normalization (scaling) factors.

If the continuous domains of the process state and control output variables have been discretized

then the data base also contains information concerning the quantization look-up tables defining

the discretization policy.

Hence for continuous fuzzy sets, the data base consists of the information regarding the choice of membership function and choice of scaling factors. w hereas for discrete case, in addition to choice of membership function and scaling factors, information regarding quantization is also present.

Choice of membership function:

The choice of membership functions for the input and output variables are very important since it influences the firing of rules and in turn the control action. The shape, symmetry, cross points and width of the membership function all influence the final control action, so great care need to be exercised in choosing the membership function.

Choice of scaling factor

The use of normalized domains requires a scale transformation which maps the physical values of the process state variables into a normalized domain, this is called normalization. Furthermore, output de-normalization maps the normalized value of control output variables into their respective physical domains. These scale transformations are required both for continuous and discrete domains. The role of scaling factor is similar to the gain coefficients in a conventional controller, and affects the stability, oscillations and damping of the system, hence needs to be chosen with utmost care.

Rule Base

The basic function of the rule base is to represent in a structured way the control policy of an experienced process operator and/or control engineering in the form of a set of production rules such as:

If <process state> then <control output>

The if-part of the rule is called the rule-antecedent and is the description of a process state in terms of a logical combination of atomic propositions. The then-part of the rule is called the rule-consequent and is again a description of the control output in terms of a logical combination of fuzzy propositions. These propositions state the linguistic values which the control output variables take whenever the current process state matches the process state description in the rule-antecedent. A fuzzy rule is hence a conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control.

The design parameters involved in the construction of the rule base include:

- Choice of process state and control output variables
- Choice of contents of the rule-antecedent and the rule-consequent
- Choice of ranges of linguistic values for the process state and control output variables
- Derivation of a set of rules

Inference engine:

An inference engine basically decides the rules to be fired for particular inputs. There are
basically two types of approaches employed for their design. They are:

- Composition based inference and
- Individual-rule based inference

In composition based inference, first all the rules are combined to form one rule and then fired to get the output. w hereas in individual-rule based inference, first each rule is fired, resulting in many clipped fuzzy sets, which are then combined to get the overall fuzzy output. Usually the second type of inference is preferred as this is efficient and saves a lot of memory.

Defuzzification, Denormalization:

This block performs two operations, namely, defuzzification and denormalization. Defuzzification is the process of converting the fuzzy quantity back to crisp. Denormalization is the process of mapping the point-wise value of the control output onto its physical domain. The design parameter of this block is the choice of defuzzification method. The defuzzification procedure is chosen based on its continuity, disambiguity, plausibility and computational complexity.

Decision-making logic:

Decision making is a most important scientific, social, and economic endeavor. To be able to make consistent and correct choices is the essence of any decision process imbued with uncertainty. Decision making is usually weighed by the final outcome, which is not true with fuzzy processes as there is uncertainty in the event and its outcome. The problem in making decisions under uncertainty is that the bulk of information about the possible outcomes, about the value of new information, about the way the conditions change with time is typically vague, ambiguous and otherwise fuzzy. Some techniques for decision making under fuzzy environment are:

- Fuzzy synthetic evaluation
- Fuzzy ordering
- Preference and consensus
- Multi-objective decision making logic

Fuzzy synthetic evaluation

In this method, several individual elements and components of an evaluation are synthesized into an aggregate form. The elements can be either numeric or non-numeric. The evaluation is usually described in non-numeric terms as numeric evaluation is too complex and too unacceptable for fuzzy data. For example, when a professor grades a written exam, he might evaluate it from perspectives such as style, grammar, creativity, and so forth. The final grade on the paper may be linguistic instead of numeric, for example, excellent, very good, good, fair, poor, and unsatisfactory. After grading many exams, the professor may develop a relation by which he can assign a membership to the relations between the different perspectives, such as style and grammar, and the linguistic grades, such as fair, and excellent.

Fuzzy ordering

Decisions are sometimes based on the basis of rank. There is no ambiguity in ranking issues or actions that are crisp, but ranking fuzzy actions usually carries some ambiguity.

- If the uncertainty in ranking is random, probability density functions can be used to rank them.

• If the uncertainty in ranking arises due to imprecision, extension principle can be used to calculate the truth value of assertion, and hence can be ranked.

Preference and consensus

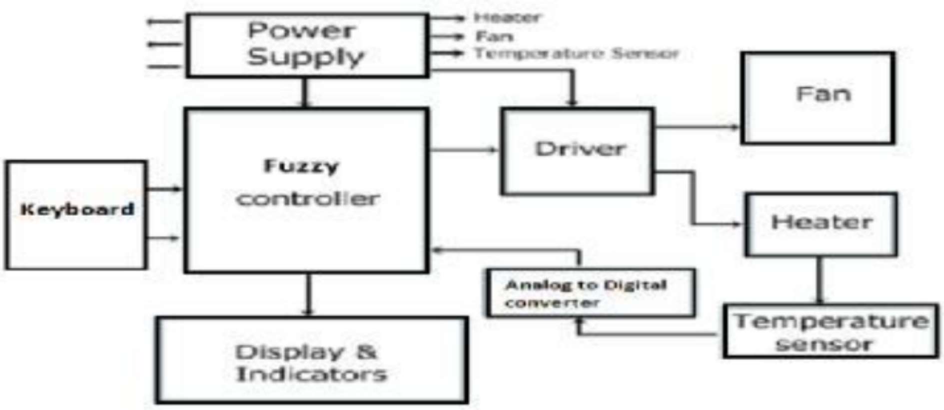
The goal of group decision making typically is to arrive at a consensus concerning a desired action or alternative from among those considered in the decision process. Consensus is usually taken to mean a unanimous agreement by all those in the group concerning their choice.

Multi objective decision making

Many simple decision processes are based on a single objective, such as minimizing cost, maximizing profit, minimizing runtime, and so forth. Often, however, decisions must be made in an environment where more than one objective function governs constraints on the problem. Two primary issues in multi objective decision making are to acquire meaningful information regarding the satisfaction of the objectives by the various choices and to rank or weigh the relative importance of the objectives.

Temperature Controller using Fuzzy Logic

Low cost temperature control using fuzzy logic system block diagram shown in the fig. In this system set point of the temperature is given by the operator using 4X4 keypad. LM35 temperature sensor sense the current temperature. Analog to digital converter convert analog value into digital value and give to the Fuzzy controller.



Controller calculates error between set point value and current value and consider as Input function of fuzzy logic. By fuzzification process controller calculate it membership. After in rule base and inference system output membership value calculated. Defuzzification process calculates actual value of PWM for heater and fan which is output of the temperature control system.

The process comprises of a heater, fan and a temperature sensor. The amount of current passing through the coil decides the temperature of the thin metal plate. Temperature detection of this metal plate can be done by dedicated temperature sensors. A fan is placed near to the heating mechanism. Amount of power delivered to both heater and fan can be controlled by passing a command through serial port via microcontroller. Now, microcontroller generate PWM (Pulse Width Modulation) signal for the MOSFET to deliver desired amount of power to fan and heater.

In order to exemplify the usage of a fuzzy logic system, consider a temperature control system controlled by a fuzzy logic controller. The temperature of the room can be adjusted by details like current temperature of the room and the target value by defined system. The comparison between the room temperature and the target temperature can be compared by fuzzy engine at certain period of time and produces a command of heating or cooling.

References

De Cock, Martine; Bodenhofer, Ulrich; Kerre, Etienne E. (1–4 October 2000). Modelling Linguistic Expressions Using Fuzzy Relations. Proceedings of the 6th International Conference on Soft Computing. Iizuka, Japan. pp. 353–360

Klaua, D. (1965) Über einen Ansatz zur mehrwertigen Mengenlehre. Monatsb. Deutsch. Akad. Wiss. Berlin 7, 859–876. A recent in-depth analysis of this paper has been provided by Gottwald, S. (2010). "An early approach toward graded identity and graded membership in set theory". Fuzzy Sets and Systems.

Liang LR, Lu S, Wang X, Lu Y, Mandal V, Patacsil D, Kumar D. FM-test: a fuzzy-set-theory-based approach to differential gene expression data analysis. BMC Bioinformatics. 2006