2 Fuzzy Computing

In the real word there exists much fuzzy knowledge, that is, knowledge which is vague, imprecise, uncertain, ambiguous, inexact, or probabilistic in nature.

Human can use such information because the human thinking and reasoning frequently involve fuzzy information, possibly originating from inherently inexact human concepts and matching of similar rather than identical experience.

The computing system, based upon classical set theory and two-valued logic, cannot give answers to some questions as a human does, because they do not have completely true answers.

We want the computing systems not only to give human-like answers but also to describe their reality levels. These levels need to be calculated using imprecision and the uncertainty of facts as well as rules that were applied.

2.1 Fuzzy sets

Fuzzy Logic is built on *The Fuzzy Set Theory* which was introduced to the world by Lotfi Zadeh in 1965 for the first time. The invention, or proposition, of *Fuzzy Sets* was motivated by the need to capture and represent the real world with its fuzzy data due to uncertainty. Uncertainty can be caused by imprecision in measurement due to imprecision of tools or other factors. Uncertainty can also be caused by vagueness in the language objects and situations. Lotfi Zadeh realized that the *Crisp Set Theory* is not capable of representing those descriptions and classifications in many cases. In fact, *Crisp Sets* do not provide adequate representation. We use linguistic variables often to describe, and maybe classify, physical objects and situations.

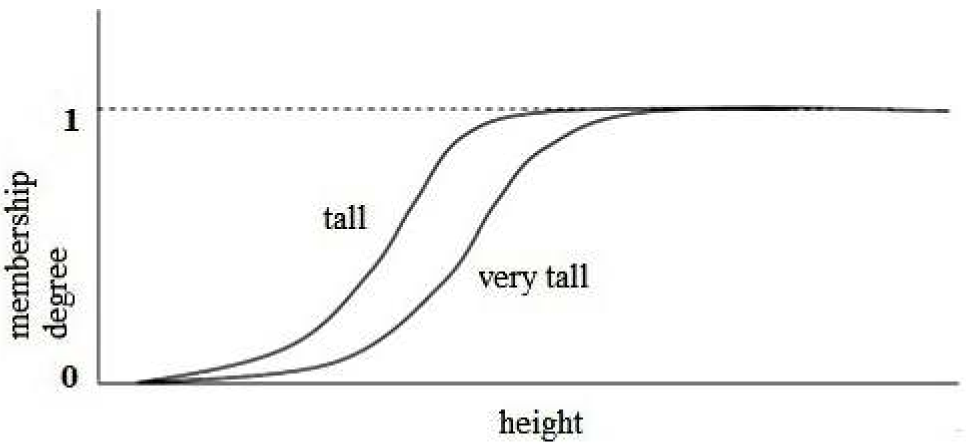

Figure 3: Fuzzy set representation.

Instead of avoiding or ignoring uncertainty, Lotfi Zadeh developed a set theory that captures this uncertainty. The goal was to develop a set theory and a resulting logic system that are capable of coping with the real world. Therefore, rather than defining Crisp Sets, where elements are either in or out of the set with the absolute certainty, Zadeh proposed the concept of a *Membership Function*. An element can be in the set with a degree of membership and out of the set with a degree of membership. Figure 3 illustrates the use of Fuzzy Sets to represent the notion of a tall person. It also shows how we can differentiate between the notions of tall and very tall, resulting in a more accurate model than the classical set theory.

2.1.1 Basic properties of fuzzy sets

Fuzzy set theory is primarily concerned with quantifying and reasoning using natural language in which many words have ambiguous meanings. It can also be thought of as an extension of the traditional crisp set, in which each element must either be in or not be in a set. Formally, the process by which individuals from a universal set X are determined to be either members or non-members of a crisp set can be defined by a *characteristic* or *discrimination function*. For a given crisp set A this function assigns a value $\mu_A(x)$ to every $x \in X$ such that

$$\mu_A(x) = \begin{cases} 1 & f \quad x \in A \\ 0 & f \quad x \notin A \end{cases}$$

Thus, the function maps elements of the universal set to the set containing 0 and 1 (Figure 4). This can be indicated by

$$\mu_A : \mathrm{X} \rightarrow \{0, 1\}$$

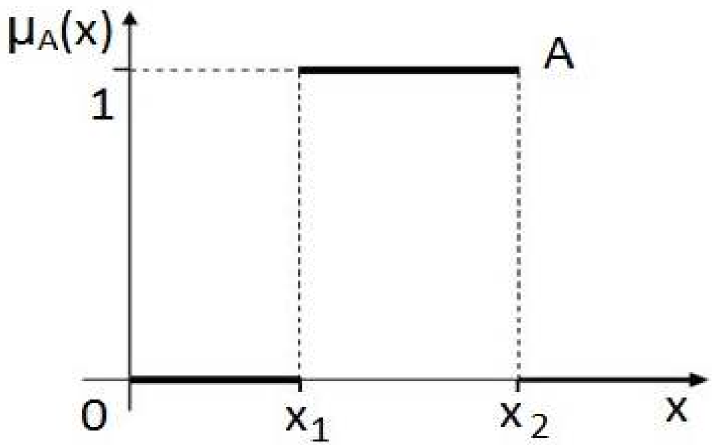

Figure 4: Graphical representation of crisp sets

This kind of function can be generalized such that the values assigned to the elements of the universal set fall within a specified range and are referred to as the membership grades of these elements in the set. Larger values denote higher degrees of the set membership. Such function is called a membership function μ_A by which a fuzzy set A is usually defined and represents the degree of membership of x in A. This function can be indicated by

$$\mu_A : X \rightarrow [0, 1],$$

where X refers to the universal set defined in a specific problem, and [0,1] denotes the interval of real numbers from 0 to 1, inclusively.

In the case of Crisp Sets, the members of a set are either out of *the* set, with the membership degree of zero, or in the set, with the value one being the degree of membership. Therefore, Crisp Sets $\subseteq$ Fuzzy Sets or in other words, Crisp Sets are Special cases of Fuzzy Sets.

There are four ways of representing fuzzy membership functions, namely, *graphical representation*, *tabular and list representation*, *geometric representation*, and *analytic representation*. Graphical representation is the most common in the literature. Figure 3 above is an example of the graphical representation of fuzzy membership functions. Tabular and list representations are used for finite sets. In this type of representation, each element of the set is paired with its degree of membership. Two different notations have been used in the literature for tabular and list representation. The following example illustrates the two notations for the same membership function.

$$\mu_A = \{ <x_1, 0.8>, <x_2, 0.3>, <x_3, 0.5>, <x_4, 0.9> \}$$

$$\mu_A = 0.8/x_1 + 0.3/x_2 + 0.5/x_3 + 0.9/x_4$$

The third method of representation is the geometric representation and is also used for representing finite sets. For a set that contains *n* elements, *n*-dimentional Euclidean space is formed and each element may be represented as a coordinate in that space. Finally analytical representation is another alternative to graphical representation in representing infinite sets, e.g., a set of real numbers. The following example illustrates both graphical and analytical representation of the same fuzzy function:

$$\mu(A) = \begin{cases} x-5 & \text{when } 5 \le x \le 6 \\ 7-x & \text{when } 6 \le x \le 7 \\ 0 & \text{otherwise} \end{cases}$$

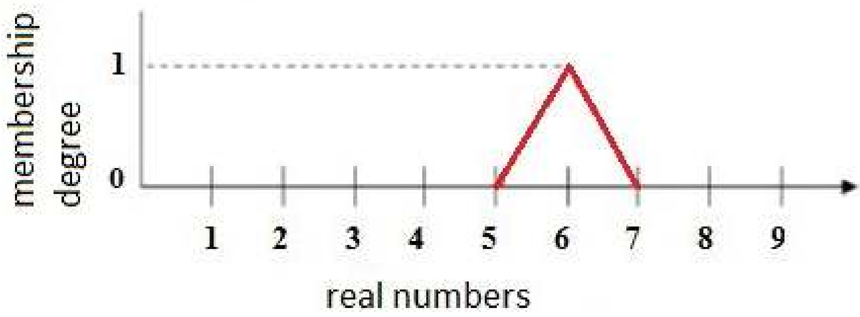

Figure 5: Graphical representation of the analytical representation given above

With the additional restriction that the membership function must capture an intuitive conception of a set of real numbers surrounding a given central real number, or interval of real numbers. In this context, the example above illustrates the concept of the fuzzy number "about six", "around six", or "approximately six".

Another very important property of fuzzy sets is the concept of α – *cut* (alpha cut). ∝-cuts reduce a fuzzy set into an extracted crisp set. The value α represents a membership degree, i.e. α ∈ [0, 1]. The α-cut of a fuzzy set *A* is the crisp set (*A* – α), i.e. the set of all elements whose membership degrees in *A* are ≥ α (Kohout 1999).

2.1.2 Basic properties of fuzzy sets

The basic operations on fuzzy sets (Figure 6) are Fuzzy Complement (Figure 7), Fuzzy Union (Figure 8), and Fuzzy Intersection (Figure 9). These operations are defined as follows:

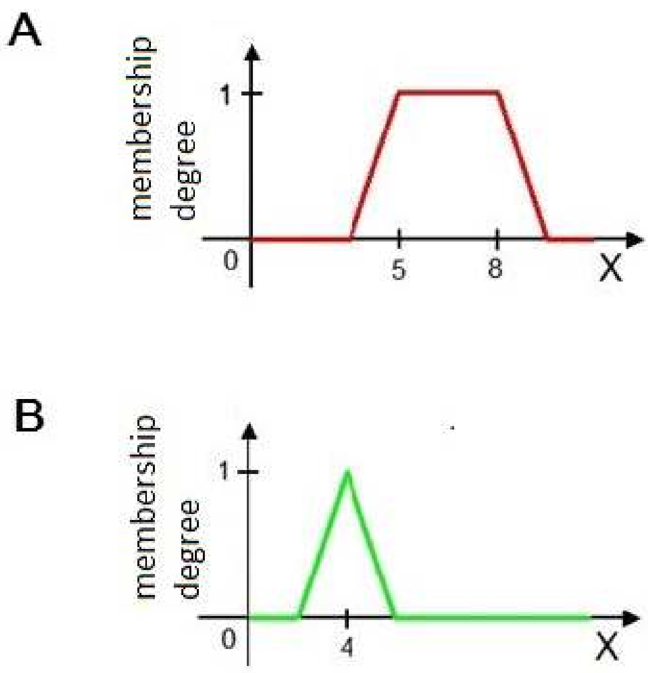

Figure 6: Fuzzy sets A and B

Fuzzy Complement of *A*: $\overline{A}(x): \mu_{\overline{A}} = 1 - \mu_A(x)$ for all $x \in X$

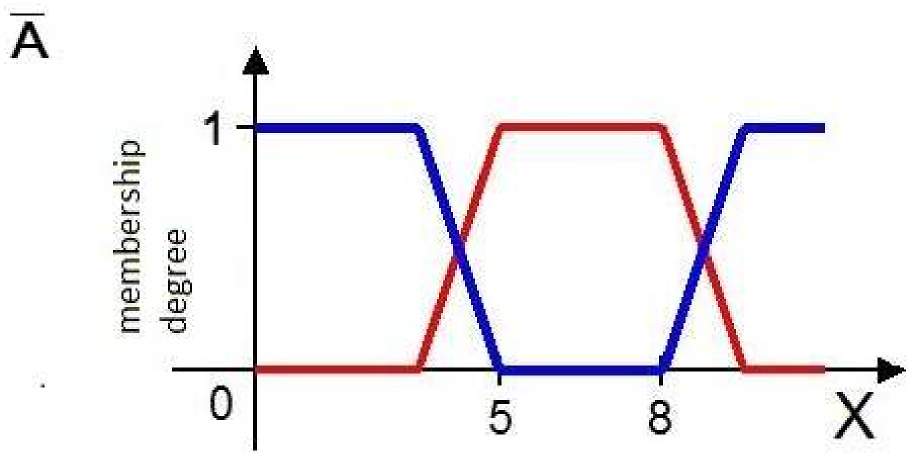

Figure 7: Fuzzy Complement

Fuzzy Union of A and B: $(A \cup B): \mu_{A \cup B} = \max\{\mu_A(x); \mu_B(x)\}$ for all $x \in X$

Notice that $A \cup \overline{A} \neq X$, which violates the *Law of Excluded Middle*.

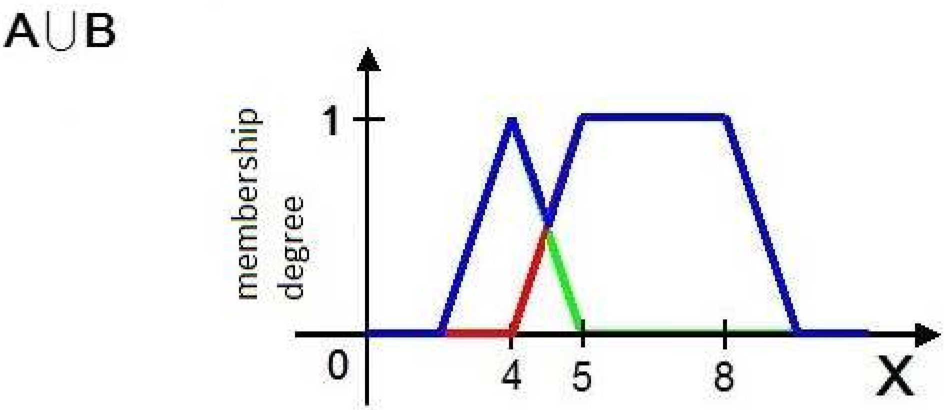

Figure 8: Fuzzy Union

Fuzzy Intersection of A and B: $(A \cap B): \mu_{A \cap B} = \min\{\mu_A(x); \mu_B(x)\}$ for all $x \in X$

Notice that $A \cap \overline{A} \neq \phi$, which violates the Law of Contradiction.

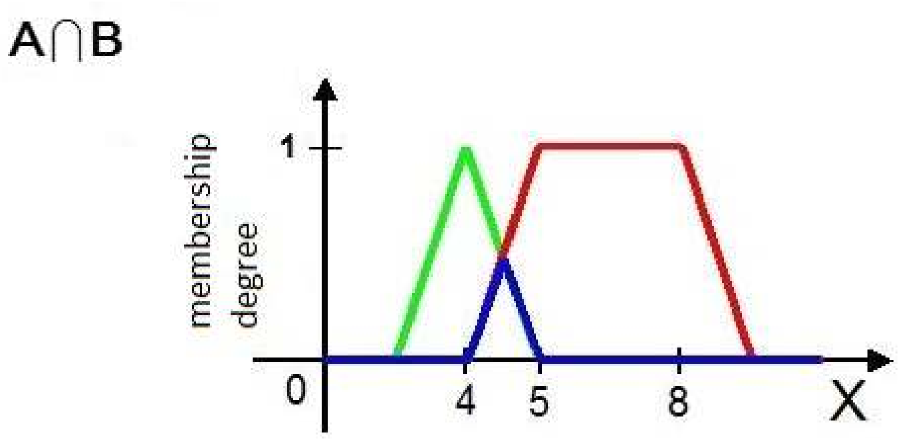

Figure 9: Fuzzy Intersection

2.1.3 Fuzzy arithmetic

Fuzzy Arithmetic uses arithmetic on closed intervals. The basic fuzzy arithmetic operations are defined as follows:

Addition: [a, b] + [c, d] = [a+ c, b + d]

Subtraction: [a, b] – [c, d] = [a – d, b – c]

Multiplication: [a,b].[c,d] = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]

Division: [a, b] / [c, d] = [a, b] . [1/d, 1/c] = [min(a/c, a/d, b/c, b/d),max(a/c, a/d, b/c, b/d)]

Figure 10 illustrates graphical representations of fuzzy addition and subtraction.

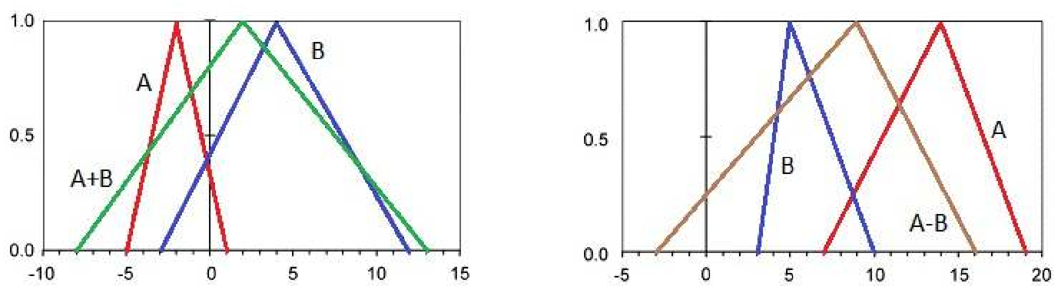

Figure 10: Example of fuzzy addition and subtraction

2.1.4 Fuzzy relations

Properties of fuzzy relations:

Fuzzy Relations were introduced to supersede classical crisp relations. Rather than just describing the full presence or full absence of association of elements of various sets in the case of crisp relations, Fuzzy Relations describe the degree of such association. This gives fuzzy relations the capability to capture the uncertainty and vagueness in relations between sets and elements of a set. Furthermore, it enables fuzzy relations to capture the broader concepts expressed in fuzzy linguistic terms when describing the relation between two or more sets. For example, when classical sets are used to describe the equality relation, it can only describe the concept "*x* is equal to *y*" with absolute certainty, i.e., if x is equal to *y* with unlimited precision, then *x* is related to *y*, otherwise *x* is not related to *y*, even if it was slightly different. Thus, it is not possible to describe the concept "*x* is approximately equal to *y*". Fuzzy Relations make the description of such a concept possible. Table 1 provides comparison of the special properties of Crisp and Fuzzy relations, E_x is the Equality Relation and O_x is the Empty Relation (Bandler and Kohout 1988). It is important to note here that the concept of local reflexivity was introduced for the first time in Crisp Relational Theory by Bandler and Kohout in 1977. The fast fuzzy relational algorithms that employ local reflexivity in fuzzy computing were introduced also by Bandler and Kohout in 1982.

Property	Crisp	Fuzzy
Covering	$\Leftrightarrow \forall i \in J, \exists j \in J \mid R_{ij} = 1$	$\Leftrightarrow \forall i \in J, \exists j \in J \mid R_{ij} = 1$
Locally reflexive	$\Leftrightarrow \forall i \in J, R_{ii} = \vee_j (R_{ij} \vee R_{ji})$	$\Leftrightarrow \forall i \in J, R_{ii} = \vee_j (R_{ij} \vee R_{ji})$
Reflexive	$\Leftrightarrow$ Covering and locally reflexive	$\Leftrightarrow$ Covering and locally reflexive
Transitive	$\Leftrightarrow R^2 \subseteq R$	$\Leftrightarrow R^2 \subseteq R$
Symmetric	$\Leftrightarrow R^T \subseteq R$	$\Leftrightarrow R^T \subseteq R$
Antisymmetric	$\Leftrightarrow R \cap R^T \subseteq E_x$	$\Leftrightarrow R_{ij} \wedge R_{ji} = 0$ if $j \neq i$
Strictly Antisymmetric	$\Leftrightarrow R \cap R^T = O_x$	$\Leftrightarrow \forall i, j \in J, R_{ij} \wedge R_{ji} = 0$

Table 1: Properties of Crisp vs. Fuzzy Relations

Representation of Fuzzy Relations

The most common methods of representing fuzzy relations are lists of n-tuples, formulas, matrices, mappings, and directed graphs. A list of n-tuples, i.e., ordered pairs, can be used to represent finite fuzzy relations. The tuple consists of a Cartesian product with its membership degree. When the membership has degree zero, the tuple is usually omitted. Suitable formulas are usually used to define infinite fuzzy relations, which involve n-dimensional Euclidean space, with $n \geq 2$. Matrices, or n-dimensional arrays, are the most common method to represent fuzzy relations. In this method, the entries of the matrix are the membership degrees associated with the n-tuple of the Cartesian product. The mapping of fuzzy relations is an extension of the mapping method of classical binary relations. For fuzzy relations, the connections of the mapping diagram are labelled with the membership degree. The same technique is used to extend the directed graph representation of classical relations to represent fuzzy relations.

Operations on fuzzy relations:

All the mathematical and logical operations on fuzzy sets explained above are also applicable to fuzzy relations. In addition, there are operations on fuzzy binary relations that do not apply to general fuzzy sets. Those operations are the inverse, the composition, and the BK-products of fuzzy relations.

The inverse of a fuzzy binary relation R on two sets X and Y is also a relation denoted by R^{-1} such that $xR^{-1}y = yRx$. Therefore, for any fuzzy binary relation, $(R^{-1})^{-1} = R$. When using matrix representation, the inverse can be obtained by generating the transpose of the original matrix, i.e., swapping the columns and the rows of the matrix as in the following example.

$$R = \begin{bmatrix} 0.5 & 1 & 0 \\ 1 & 0.8 & 0.2 \\ 0.7 & 0 & 0.3 \end{bmatrix} \qquad R^{-1} = \begin{bmatrix} 0.5 & 1 & 0.7 \\ 1 & 0.8 & 0 \\ 0 & 0.2 & 0.3 \end{bmatrix}$$

The composition of two fuzzy relations is defined as follows:

Let P be a fuzzy relation from X to Y and Q be a fuzzy relation from Y to Z such that the membership degree is defined by $P(x, y)$ and $Q(y, z)$. Then, a third fuzzy relation R from X to Z can be produced by the composition of P and Q, which is denoted as $(P \cdot Q)$. Fuzzy relation R is computed by the formula (Klir and Yuan 1995):

$$R(x,z) = (P \cdot Q)(x,z) = \max_{y \in Y} \{\min[P(x,y), Q(y,z)]\}$$

The idea of producing fuzzy relational composition was expanded by Bandler and Kohout in 1977 when they introduced, for the first time, special relational compositions called the Triangle and Square products (Bandler and Kohout 1987). The Triangle and Square products were named after their inventors and became known as BK-products. The BK-products of fuzzy relations proved to be very powerful not only as a mathematical tool for operations on fuzzy sets and fuzzy relations but also as a computational framework for fuzzy logic and fuzzy control. In addition to the set based definitions presented above, many valued logic operations are also implied and are defined as follows:

Circle product $(R \text{ o } S)_{ik} = \vee_j (R_{ij} \wedge S_{ik})$

Triangle sub-product $(R \triangleleft S)_{ik} = \wedge_j (R \rightarrow S_{ik})$

Triangle super-product $(R \triangleright S)_{ik} = \wedge_j (R \leftarrow S_{ik})$

Square product $(R - S)_{ik} = \wedge_j (R \equiv S_{ik})$

Where R_{ij} and S_{ik} represent the fuzzy degree of truth of the propositions $x_i R y_j$ and $y_j S z_k$, respectively (Kohout 2000).

BK-products have been applied, as a powerful computational tool, in many fields such as computer protection, artificial intelligence, medicine, information retrieval, handwriting classification, urban studies, investment, control, and most recently in quality of service and distributed networking (Moussa and Kohout 2001).

2.1.5 Fuzzy rules

Fuzzy rules are linguistic IF-THEN- constructions that have the general form “IF *A* THEN *B*” where *A* and *B* are (collections of) propositions containing linguistic variables. *A* is called the *premise* and *B* is the *consequence* of the rule. In effect, the use of linguistic variables and fuzzy IF-THEN- rules exploits the tolerance for imprecision and uncertainty. In this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

In a more explicit form, if there are *i* rules, each with *k* premises in a system, the *i-th* rule has the following form:

$$If\ a_1\ is\ A_{i,1}\ \Theta\ a_2\ is\ A_{i,2}\ \Theta\ \ldots\ \Theta\ a_k\ is\ A_{i,k}\ \text{then}\ B_i$$

In the above equation *a* represents the crisp inputs to the rule and *A* and *B* are linguistic variables. The operator Θ can be AND or OR or XOR.

Example:

If a HIGH flood is expected and the reservoir level is MEDIUM, then water release is HIGH.

Several rules constitute a *fuzzy rule-based system*.

Mamdani method

The most commonly used fuzzy inference technique is the so-called Mamdani method. Professor Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators in 1975.

Mamdani method runs in 4 steps:

1. Fuzzification of the input variables.
2. Rule evaluation.
3. Aggregation of the rule outputs.
4. Defuzzification.

Example:

Rule: 1

IF *funding* is *adequate* OR *staffing* is *small* THEN *risk* is *low*

Rule: 2

IF *funding* is *marginal* AND *staffing* is *large* THEN *risk* is *normal*

Rule: 3

IF *funding* is *inadequate* THEN *risk* is *high*

Note that *funding*, *staffing* and *risk* are linguistic variables, *inadequate*, *marginal* and *adequate* are linguistic values determined by fuzzy sets on the universe of discourses. Also, *small* and *large* are linguistic values determined by fuzzy sets.

STEP 1: Fuzzification

The first step is to take the crisp inputs, (let funding and staffing be x_1 and y_1), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets. The crisp input is a numerical input. For instance, let the expert determine a figure between 0–100 to represent funding and staffing, say 35% and 60%.

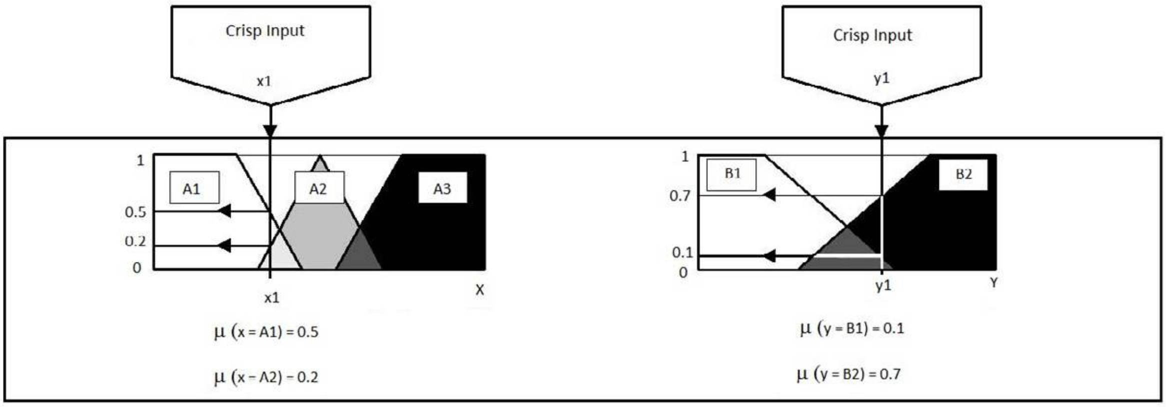

Figure 11: The first step is to take the crisp inputs (adapted from http://www.4c.ucc.ie).

STEP 2: Rule evaluation

The second step is to take the fuzzified inputs, $\mu_{(x=A1)} = 0.5$; $\mu_{(x=A2)} = 0.2$; $\mu_{(y=B1)} = 0.1$ and $\mu_{(y=B2)} = 0.7$, and apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (**AND** or **OR**) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function. To evaluate the disjunction of the rule antecedents, we use the **OR** fuzzy operation. Typically, using the fuzzy operation **union**:

$$(A \cup B): \mu_{A \cup B} = \max\{\mu_A(x); \mu_B(x)\}$$

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the **AND** fuzzy operation **intersection**:

$$(A \cap B): \mu_{A \cap B} = \min\{\mu_A(x); \mu_B(x)\}$$

Mamdami rule evaluation runs in the following steps (Figure 12):

- The result of the antecedent evaluation can be now applied to the membership function of the consequent.
- **Clipping** is a common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth. (Figure 13).
- **Scaling** is a better approach for preserving the shape of the fuzzy set. The original membership function of the rule consequent is adjusted by multiplying its membership degrees by the truth value of the rule antecedent. (Figure 13).

Since the top of the membership function is sliced, the clipped fuzzy set loses some information. However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify (in Step 4).

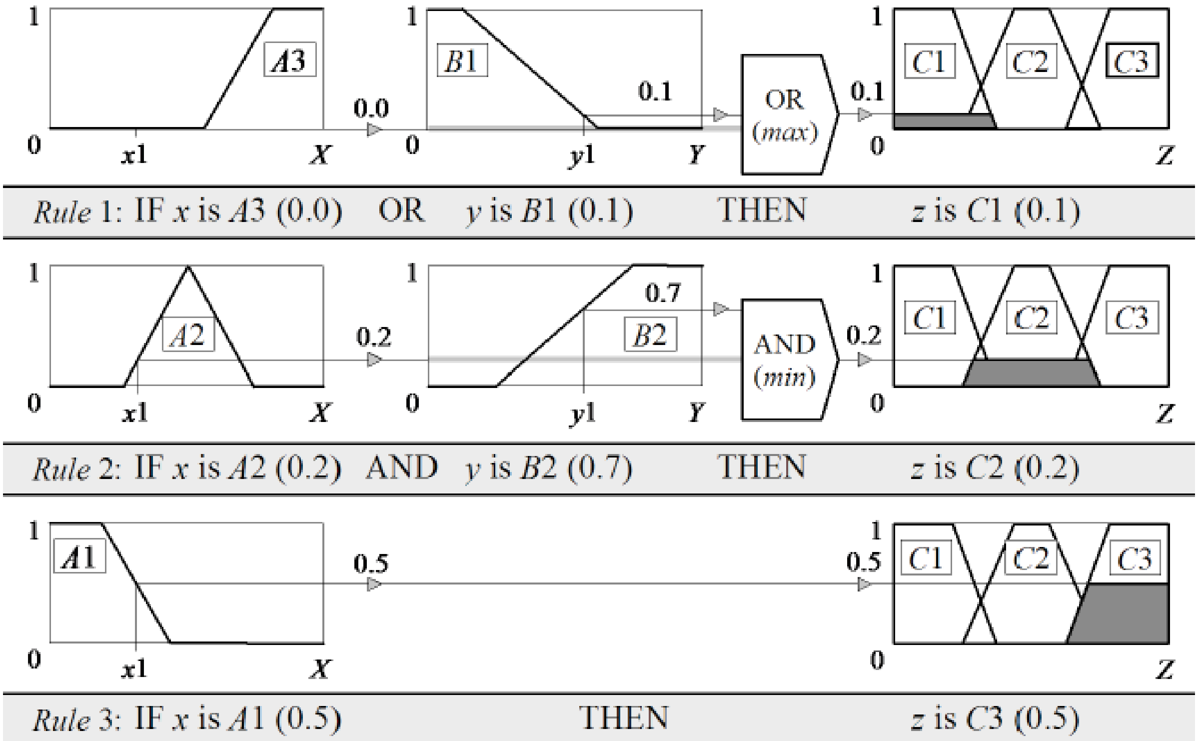

Figure 12: Mamdani rule evaluation (adapted from http://www.4c.ucc.ie).

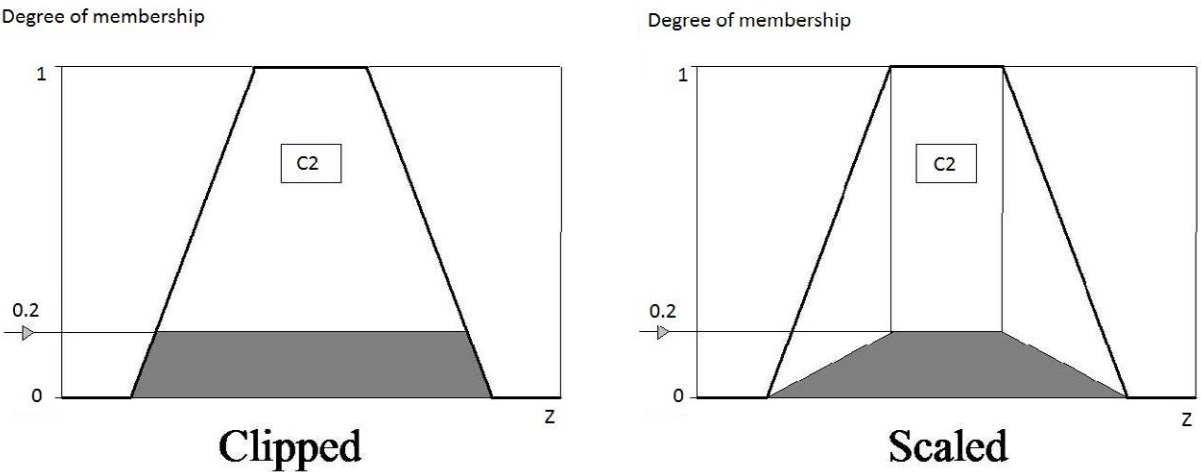

Figure 13: Clipping and scalling (adapted from http://www.4c.ucc.ie).

STEP 3: Aggregation of rule outputs (Figure 14)

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set. The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

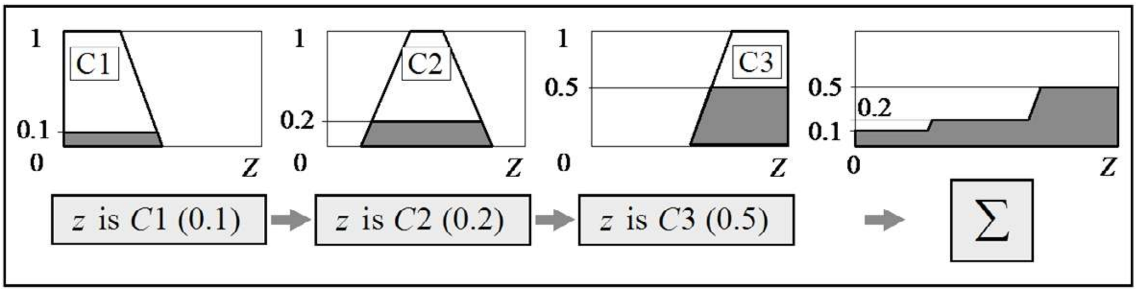

Figure 14: Aggregation of rule outputs (adapted from http://www.4c.ucc.ie).

STEP 4: Defuzzification

The last step in the fuzzy inference process is the **defuzzification**. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

There are several defuzzification methods, but probably the most popular one is the **centroid** technique. It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this **centre of gravity (COG)** can be expressed as (Figure 15):

$$COG = \frac{\int_a^b \mu_A(x)\,x\,dx}{\int_a^b \mu_A(x)\,dx}$$

- The centroid defuzzification method finds a point representing the centre of gravity of the fuzzy set, *A*, on the interval [*a*, *b*].
- A reasonable estimate can be obtained by calculating it over a sample of points.

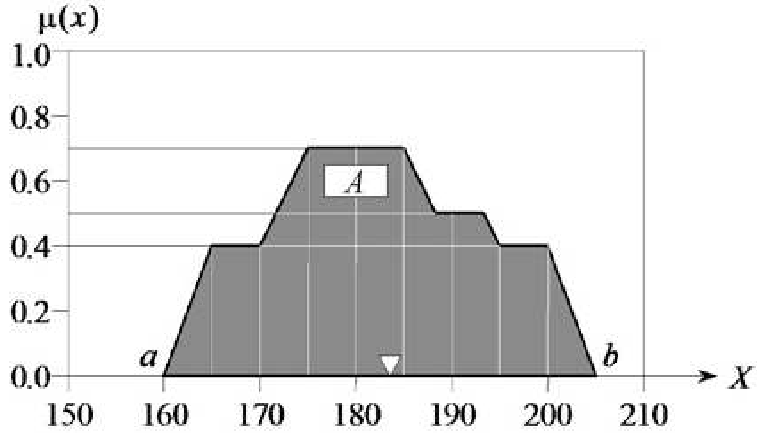

Figure 15: Centre of gravity (COG) (adapted from http://www.4c.ucc.ie).

Sugeno fuzzy interference

The Mamdani-style inference, as we have just seen, *requires* us to find the centroid of a two-dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient. Michio Sugeno suggested using a single spike, **a fuzzy singleton**, as the membership function of the rule consequent. This is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. Sugeno-style fuzzy inference is very similar to the Mamdani method. Sugeno changed only a rule consequent. He used a mathematical function of the input variable (instead of a fuzzy set).

Format of Sugeno-style fuzzy rule is the following. **IF** *x* is *A* **AND** y is *B* **THEN** *z* is *f* (*x, y*), where *x*, *y* and *z* are linguistic variables; *A* and *B* are fuzzy sets on universe of discourses *X* and *Y*, respectively; and *f* (*x, y*) is a mathematical function. The most commonly used zero-order Sugeno fuzzy model applies fuzzy rules whose **THEN** part takes the following form: **THEN** *z* is *k*, where *k* is a constant. In this case, the output of each fuzzy rule is constant. All consequent membership functions are represented by singleton spikes. The Sugeno-style rule evaluation is shown in Figure 16.

Example (cont.):

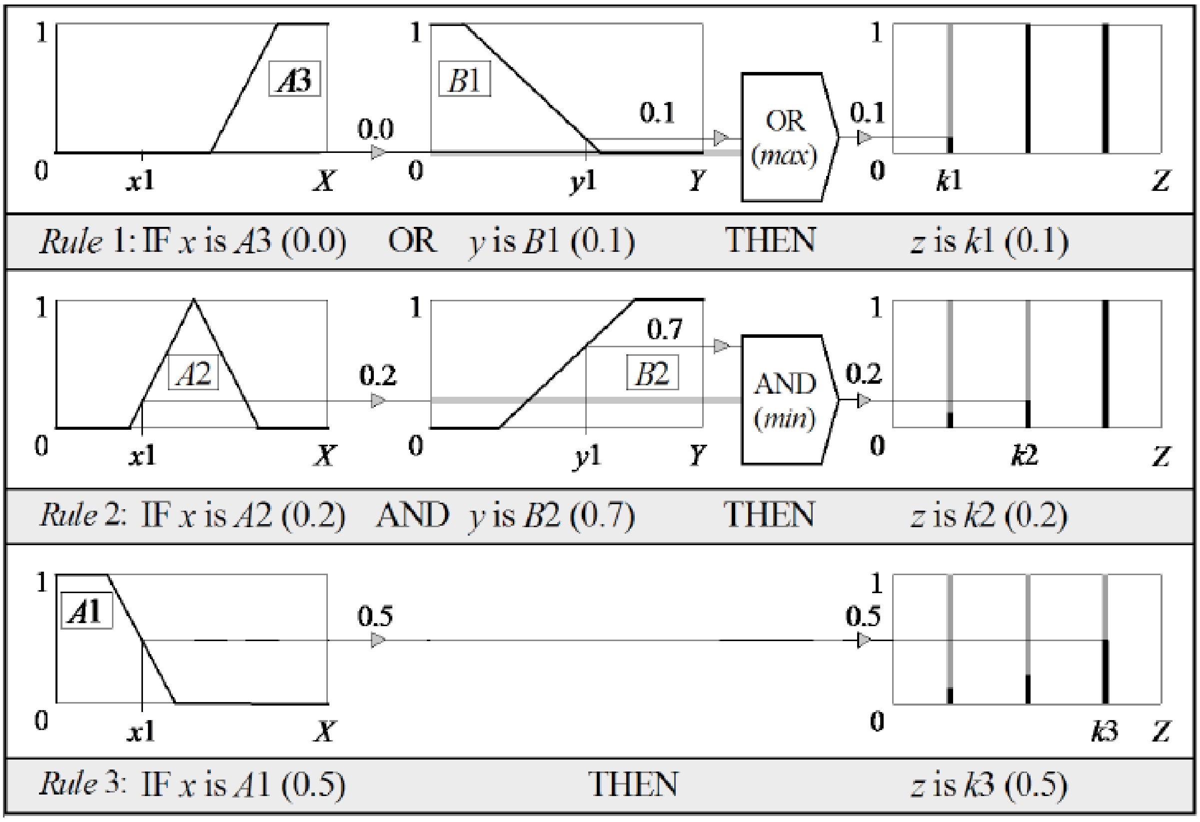

Figure 16 Sugeno-style rule evaluation (adapted from http://www.4c.ucc.ie).

The Sugeno-style aggregation is shown in Figure 17.

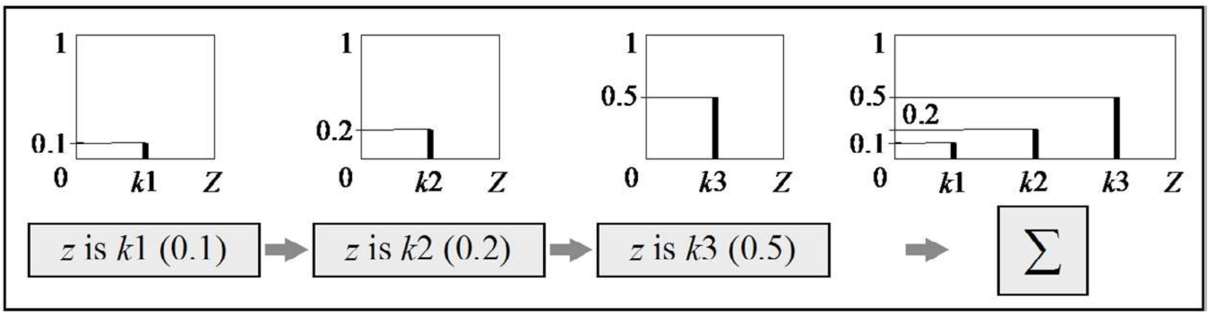

Figure 17: Sugeno-style aggregation (adapted from http://www.4c.ucc.ie).

We can find a weighted average (WA):

$$WA = \frac{(0.1 \times 20) + (0.2 \times 50) + (0.5 \times 80)}{0.1 + 0.2 + 0.5} = 65$$

The Sugeno-style defuzzification is shown in Figure 18.

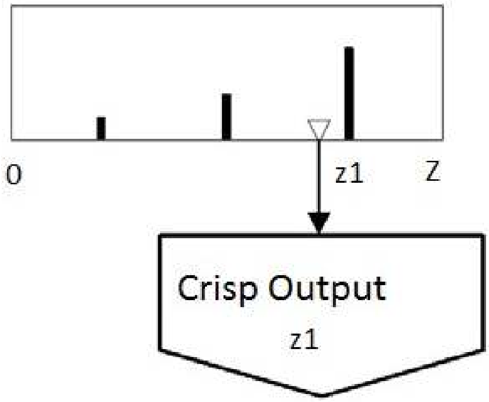

Figure 18: Sugeno-style defuzzification (adapted from http://www.4c.ucc.ie).

Mamdani or Sugeno? How to decide?

- The Mamdani method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. However, the Mamdani-type fuzzy inference entails a substantial computational burden.
- The Sugeno method is computationally efficient and works well with optimisation and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

2.2 Fuzzy control

Fuzzy control is considered to be the most successful area of application of the fuzzy set theory and fuzzy logic. Fuzzy controllers revolutionized the field of control engineering by their ability to perform process control by the utilization of human knowledge, thus enabling solutions to control problems for which mathematical models may not exist, or may be too difficult or computationally too expensive to construct.

2.2.1 Components of a fuzzy system

Figure 19 shows components of a fuzzy system. The input signals combined to the vector $x = (x_1, x_2, \ldots x_q)$ are crisp values, which are transformed into fuzzy sets in the fuzzification block. The output u comes out directly from the defuzzification block, which transforms an output fuzzy set back to a crisp value. The set of membership functions responsible for the transforming part and the rule base as the relational part contain as whole modelling information about the system, which is processed by the inference machine. This rule-based fuzzy system is the basis of a fuzzy controller.

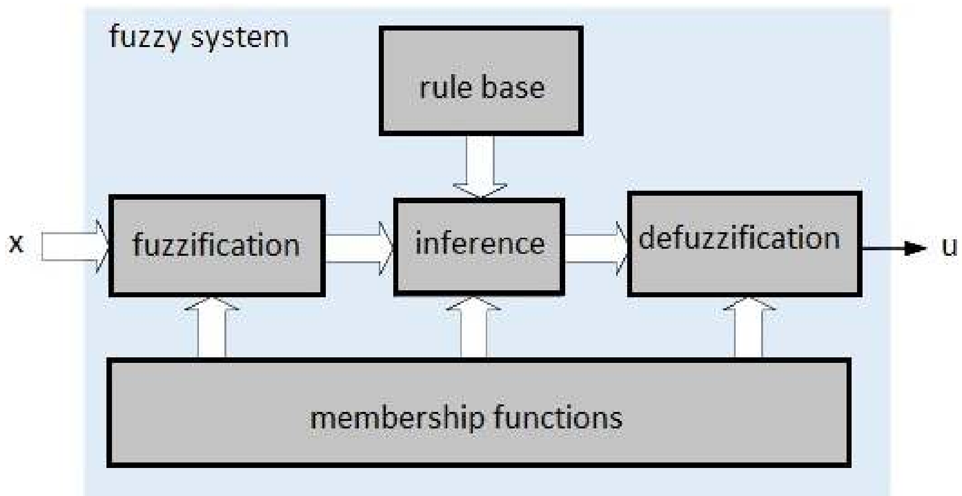

Figure 19: Components of a fuzzy system (adapted from http://www.atp.ruhr-uni-bochum.de).

2.2.2 Basic structure of a fuzzy controller

A fuzzy controller can be handled as a system that transmits information like a conventional controller with inputs containing information about the plant to be controlled and an output that is the manipulated variable. From outside, there is no vague information visible, both the input and output values are crisp values. The input values of a fuzzy controller consist of measured values from the plant that are either plant output values or plant states, or control errors derived from set-point values and controlled variables.

A control law represented in the form of a fuzzy system is a static control law. This means that the fuzzy rule-based representation of a fuzzy controller does not include any dynamics, which makes a fuzzy controller a static transfer element, like the standard state-feedback controller. In addition to this, a fuzzy controller is in general a fixed nonlinear static transfer element, which is due to those computational steps of its computational structure that have nonlinear properties. The computational structure of a fuzzy controller is described by presenting the computational steps involved. The computational structure of a fuzzy controller consists of three main steps as illustrated by the three blocks in Figure 20:

- signal conditioning and the input filter,
- the fuzzy system,
- signal conditioning and the output filter.

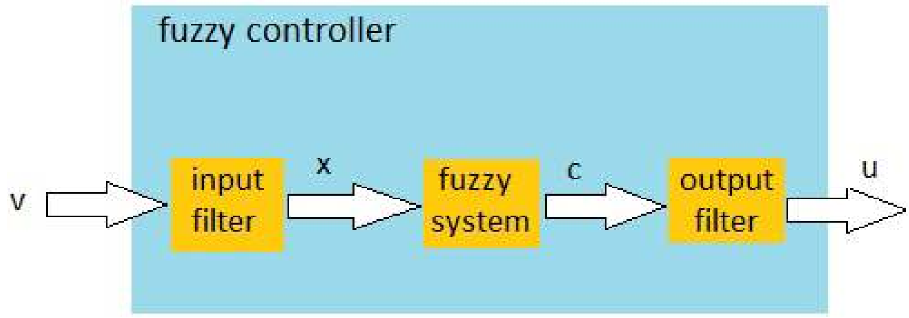

Figure 20: Basic structure of a fuzzy controller (adapted from http://www.atp.ruhr-uni-bochum.de).

Input and output filters implement signal conditioning. External input signals v must be scaled such that they can be fed as signals x into the fuzzification part of the fuzzy system. In many cases, the signals v are the control error e and its derivative is e'. In this case the input filter contains a differentiating element. Also other dynamical elements can be included in the input filter, e.g. integrators for the control error. Additionally, auxiliary signals from plant measurements may be used to represent plant states or disturbances acting on the plant. The design of this input filter depends on the application.

The fuzzy system contains a control strategy and consists of several components, see Figure 19. For example, a linguistic formulation of a proportional control strategy would be expressed by the following rules of the fuzzy system:

1. IF (control error positive) THEN (manipulated variable positive),
2. IF (control error zero) THEN (manipulated variable zero),
3. IF (control error negative) THEN (manipulated variable negative).

A proper rule base can be found either by asking experts or by evaluation of measurement data using data mining methods.

The output filter is used for adaptation of the crisp output from the fuzzy system concerning variable control. In principle, there are many possible dynamical and static operations. Often, the output of the fuzzy system describes an increment of the manipulated variable, and thus an integration of this increment must occur.

A typical Fuzzy controller consists of four modules:

The rule base, the inference engine, the fuzzification, and the defuzzification.

A typical Fuzzy Control algorithm would proceed as follows:

1. Obtaining information: To collect measurements of all relevant variables.
2. Fuzzification: To convert obtained measurements into appropriate fuzzy sets to capture uncertainties in the measurements.
3. Running the Inference Engine: To use fuzzified measurements to evaluate control rules in the rule base and select the set of possible actions.
4. Defuzzification: To convert the set of possible actions into a single numerical value.
5. The Loop: Go to step one.

Several defuzzification techniques have been proposed. The most common defuzzification methods are: the centre of gravity, the centre of maxima, the mean of maxima, and the root sum square.

Example:

The Figures below illustrate the notion of a simple fuzzy rule with one input and one output applied to the problem of an air motor speed controller for air conditioning. The Rules are given. Let us say the temperature is 22 degrees. This temperature is "right" to a degree of 0.6 and "warm" to a degree of 0.2 and it belongs to all others to a degree of zero. These rules are shown in Figure 21. The rule responses are shown in Figure 22 (thick lines).

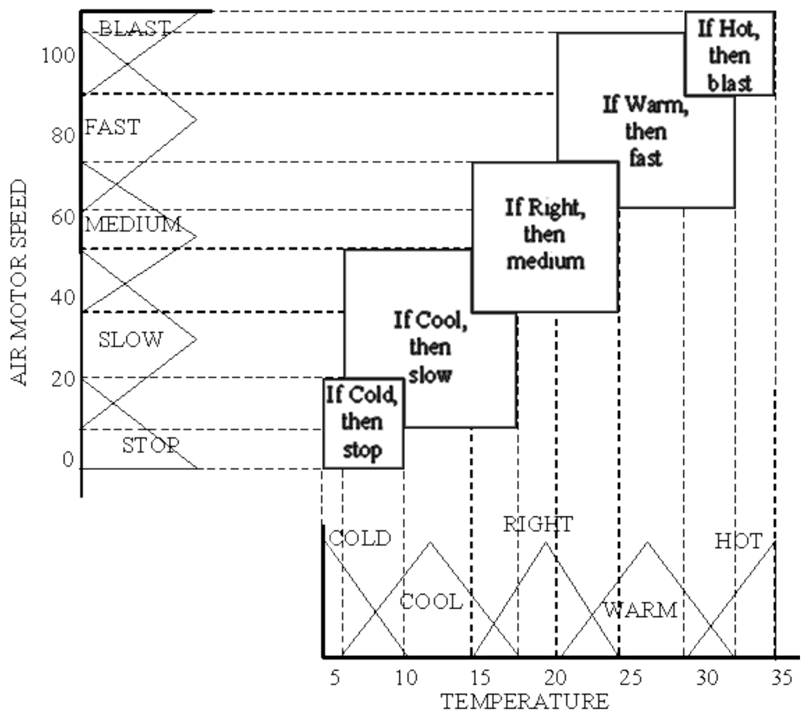

Figure 21: Air motor speed controller. Temperature (input) and speed (output) are fuzzy variables used in the set of rules (adapted from http://www.data-machine.nl/fuzzy1.htm).

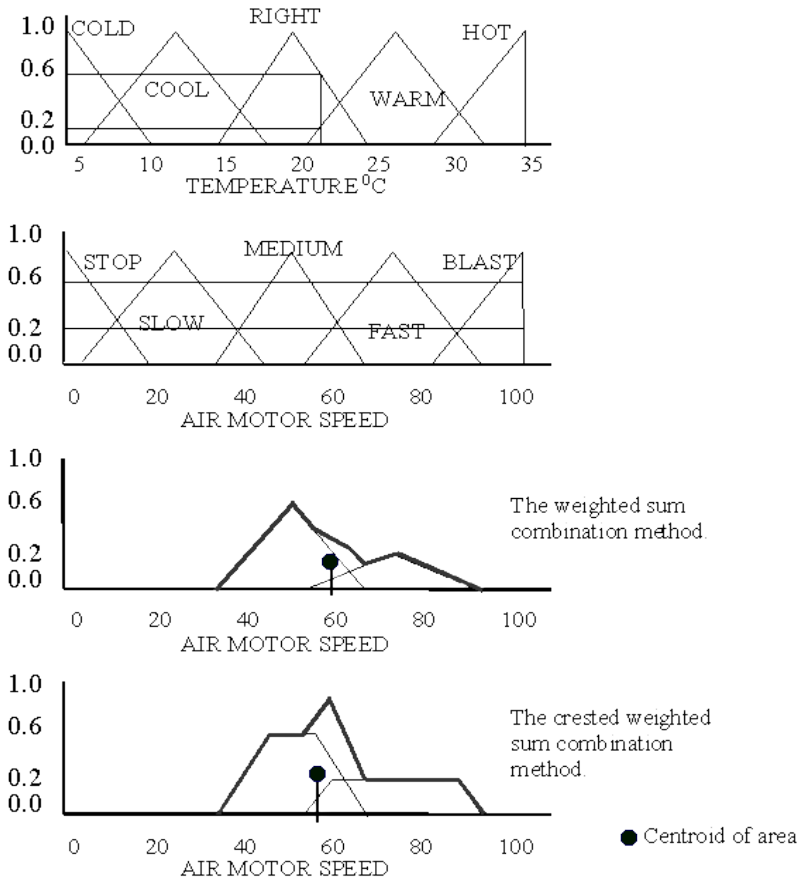

Figure 22: The temperature of 22 deg. "fires" two fuzzy rules. The resulting fuzzy value for air motor speed is defuzzified (adapted from http://www.data-machine.nl/fuzzy1.htm).

2.2.3 Representation using 2D characteristics

To provide the first approach for the design of membership functions for a fuzzy controller component, some prototype membership functions are introduced. For example, the following seven terms can be used to characterise the triangular shaped fuzzy sets according to Figure 23:

- NL negative large,
- NM negative medium,
- NS negative small
- AZ approximately zero
- PL positive large,
- PM positive medium,
- PS positive small.

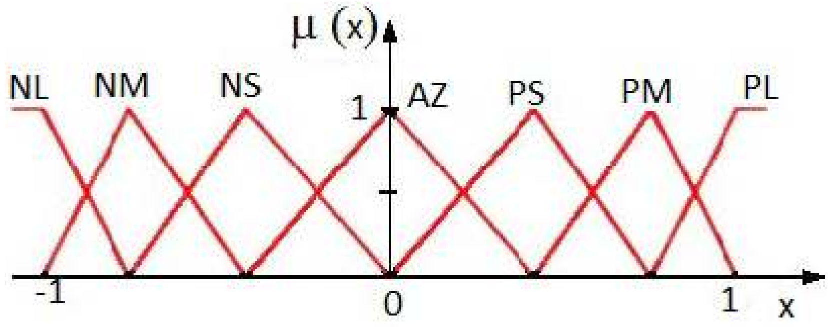

Figure 23: Prototype membership functions for a fuzzy set with seven linguistic terms (adapted from http://www.atp.ruhr-uni-bochum.de)

It is important to recognise that the fuzzy sets defined in Figure 23 and the seven linguistic terms are only a reasonable example. For various reasons, emerging from specific applications, other shapes of membership functions might be used over the given ranges. Moreover, different fuzzy sets may be defined for different variables. The prototype membership functions are usually chosen only as a preliminary candidate. They may later be modified by the designer.

For a fuzzy system using this kind of prototype with membership functions which overlaps, both for the premise and the conclusion the transfer characteristic is nonlinear, as shown in the following example.

Example:

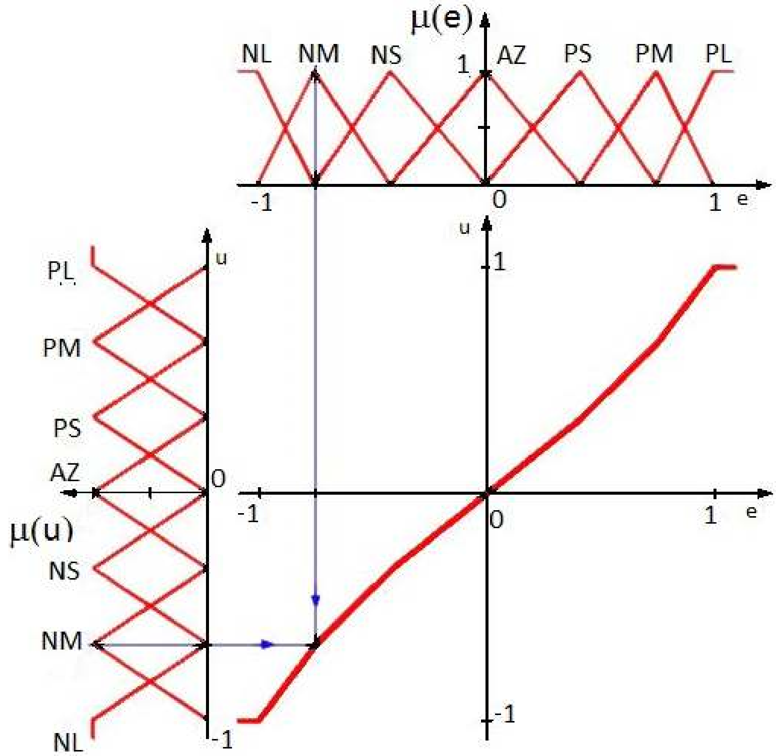

Figure 24: Nonlinear characteristic of a fuzzy controller (adapted from http://www.atp.ruhr-uni-bochum.de)

For a proportional fuzzy controller with the control error $e = w - y$ as input, with the manipulated variable u as output, with the rule base and with both membership functions of the form shown in Figure 23, the static nonlinear characteristic $u = u(e)$ is as shown in Figure 24.

1) IF e = NL THEN u = NL
2) IF e = NM THEN u = NM
3) IF e = NS THEN u = NS
4) IF e = AZ THEN u = AZ
5) IF e = PS THEN u = PS
6) IF e = PM THEN u = PM
7) IF e = PL THEN u = PL

2.2.4 Influence of the membership functions and rule base on the characteristic

The above example shows that a fuzzy controller is a nonlinear controller. The form of the characteristic depends only on the rule base and the membership functions of e and u. In the following discussions about the influence of membership functions the following assumptions for the fuzzy controller with the input signal e and the output signal u will be used:

- For AND connectives the *min* and for OR connectives the *max* operator will be used.
- The max/min inference will be used.
- The defuzzification will be performed by COG method with symmetrical membership functions at the margins.

Input and output values are normalised to the interval [-1, 1], and at first, only the following three linguistic terms NS (negative small), AZ (approximate zero) and PS (positive small) are considered. The rule base of a proportional fuzzy controller is the following:

1) IF e = NS THEN u = NS
2) IF e = AZ THEN u = AZ
3) IF e = PS THEN u = PS

The membership functions are shown in Figure 25a and 25b. The static characteristic (Figure 25c) is odd symmetrical about the origin due to the symmetry of the membership functions. Because of the different supports of the membership function for the fuzzy sets AZ of both functions, the characteristic is approximately piecewise linear and has three distinct levels. The membership functions of the input e have two overlaps in the intervals [-0.6, -0.4] and [0.4, 0.6] that correspond precisely with the ranges with the positive slope of the curve. The reason for this is just that two rules in these ranges are simultaneously active. On the other hand, in the non-overlapping ranges only one rule is active. The membership function of the output depends in this case only on the degree of relevance and thus the centre of gravity of the membership function remains constant.

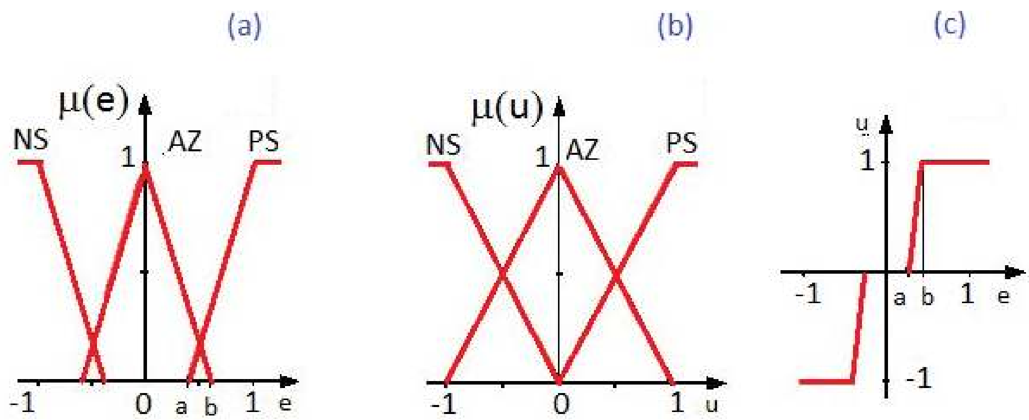

Figure 25: Membership functions and static characteristic of the fuzzy controller (adapted from http://www.atp.ruhr-uni-bochum.de)

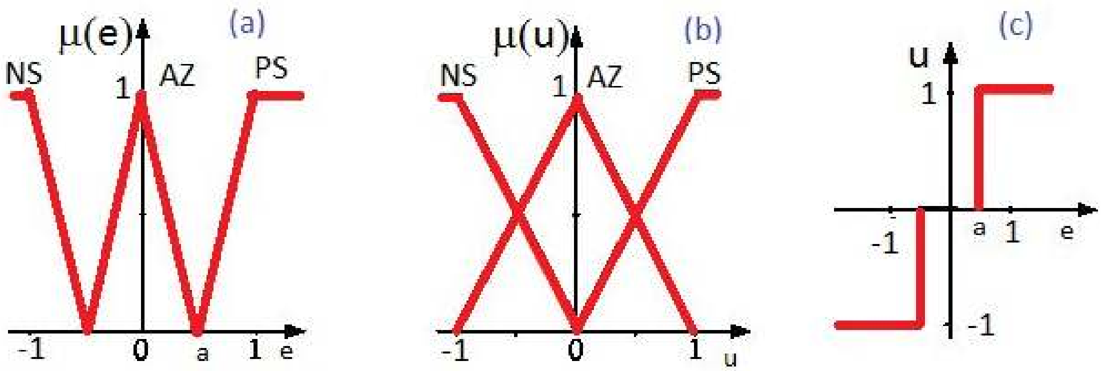

Figure 26: Influence of the (c) characteristic of a proportional fuzzy controller (a) without overlapping in the input membership functions and (b) with full overlapping in the output membership functions (adapted from http://www.atp.ruhr-uni-bochum.de)

If the number of linguistic terms for the input and output is increased, the characteristic is similar, but with more sections. The number of sections depends only on the number of linguistic terms and the width of the sections depends on the degree of overlapping. In the special case without overlapping in the input one obtains the characteristic of a three-level controller, as shown in Figure 26. In this case only one rule is active such that only the three crisp values -1, 0 and 1 are generated. Next, we consider varying the degree of overlap of the output membership functions. Figure 27 shows the case with full overlap on input and output, where the result is approximately a linear behaviour.

A modification of the output membership functions so that they do not overlap will cause the characteristic to become close to that of Figure 27, compare with Figure 28. Therefore one can establish the fact that the degree of overlap in the input membership functions has a strong influence on the static characteristic of a fuzzy controller. While small overlaps in the input membership functions generate step characteristics, with a higher degree of overlap the curves become smoother. The influence of overlap in the output membership functions has less effect on the characteristic. For a reduction of the support of the output membership functions the characteristic of Figure 29 is obtained, which does not differ significantly from that of Figure 27.

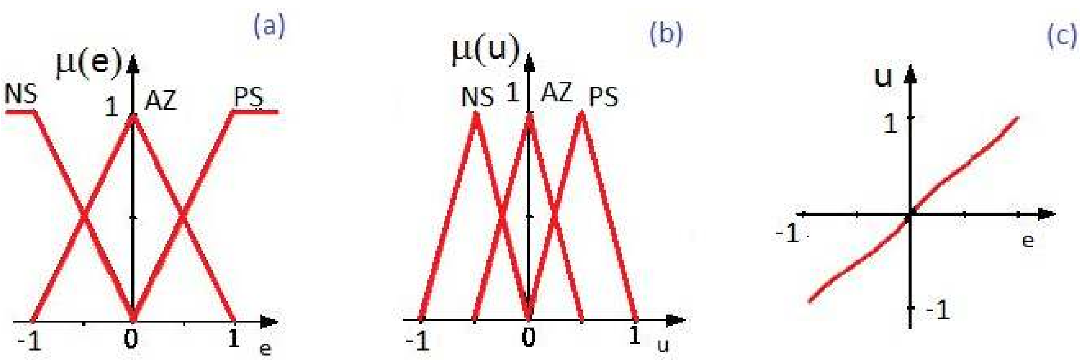

Figure 27: Influence on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and (b) full overlap in the output membership functions (adapted from http://www.atp.ruhr-uni-bochum.de)

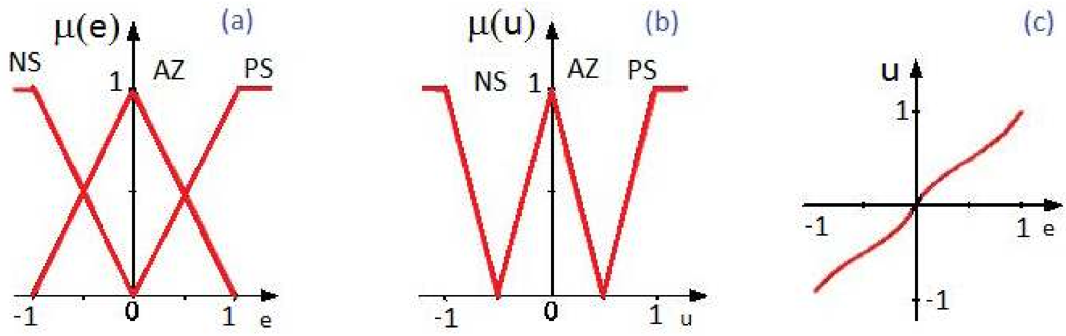

Figure 28: Influence on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and with (b) reduced support in the output membership functions (adapted from http://www.atp.ruhr-uni-bochum.de)

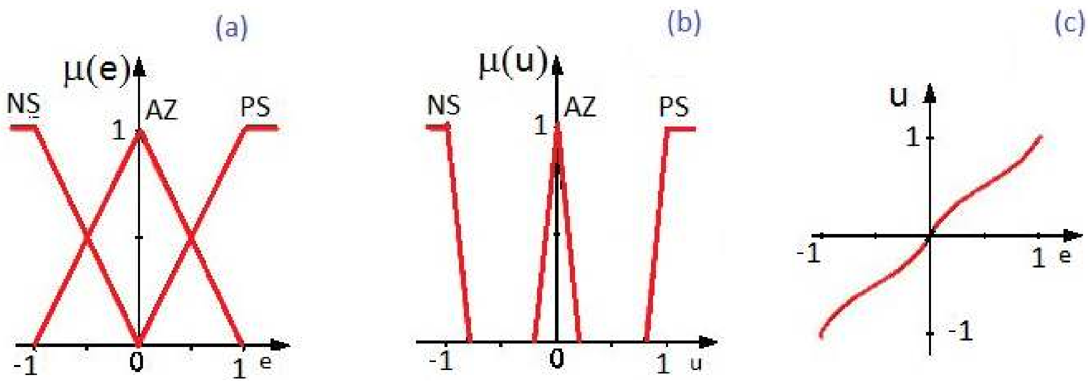

Figure 29: Influence on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and with (b) reduced support in the output membership functions (adapted from http://www.atp.ruhr-uni-bochum.de)

The size of the individual output membership function has a strong influence on the characteristic. Figure 30 shows the case for a very small support of the output membership function AZ, which generates a S-type characteristic with a high gain at the origin. Widening the support of the membership function AZ inverts the S-curve with a small gain at the origin, as shown in Figure 31. Thus, the form of the characteristic depends strongly on the support of the individual output membership function.

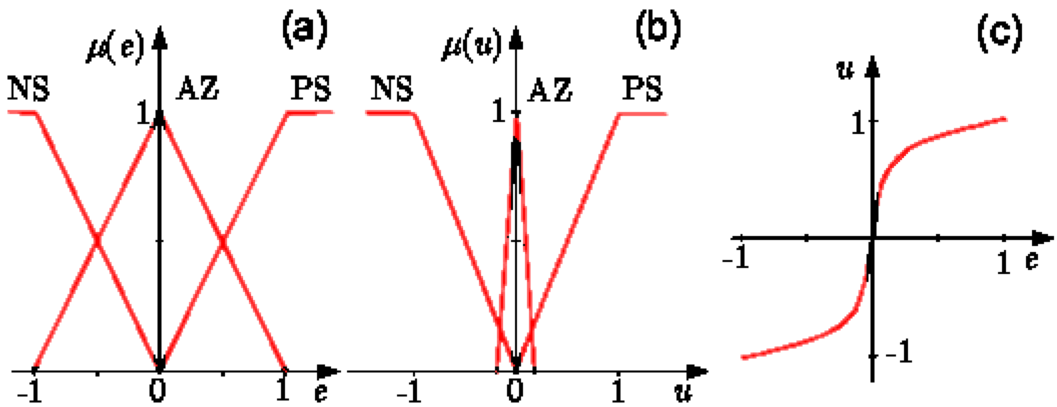

Figure 30: Influence on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and with (b) a small support in the output membership function AZ (adapted from http://www.atp.ruhr-uni-bochum.de)

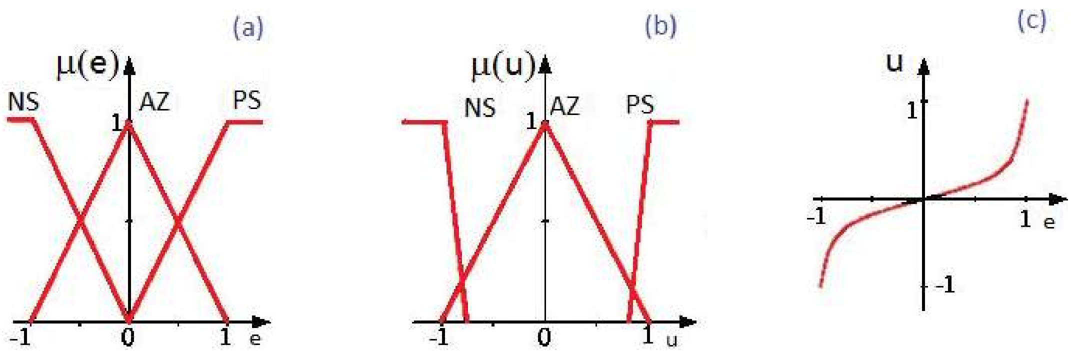

Figure 31: Influence on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and with (b) a large support in the output membership function AZ (adapted from http://www.atp.ruhr-uni-bochum.de)

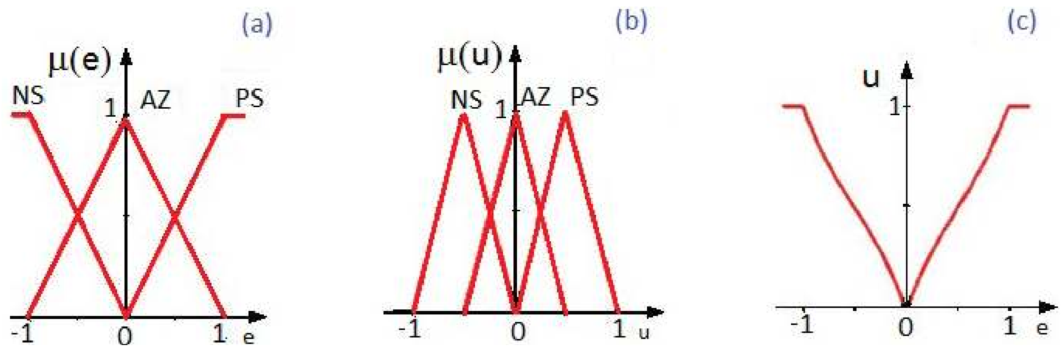

Figure 32: Influence on the rule base on the (c) characteristic of a proportional fuzzy controller with (a) full overlap in the input membership functions and (b) full overlap in the output membership functions (adapted from http://www.atp.ruhr-uni-bochum.de)

The effects of a modified rule base will be demonstrated by an example. The same full overlapping membership functions are used as in Figure 27a and 27 b. A modified rule base of the form

(1) IF e = NS THEN u = PS
(2) IF e = AZ THEN u = AZ
(3) IF e = PS THEN u = NS

will give a modulus-type of characteristic, as shown in Figure 32.

2.2.5 Representation using 3D characteristics

Up to now, fuzzy controllers with only one input and one output of the fuzzy system have been considered. The same arguments with respect to the degrees of freedom of a fuzzy system are also valid in the case of multiple inputs. A graphical representation of the characteristic is not as easy as in the 2D cases. Moreover, for the case of two inputs, a 3D representation is possible. For a fuzzy controller with a fuzzy system having two inputs e_1 and e_2 and one output u one gets a band of characteristics in a 2D discrete representation or a 3D representation with the output over the two inputs, as shown in Figure 33 for a PD-type of fuzzy controller.

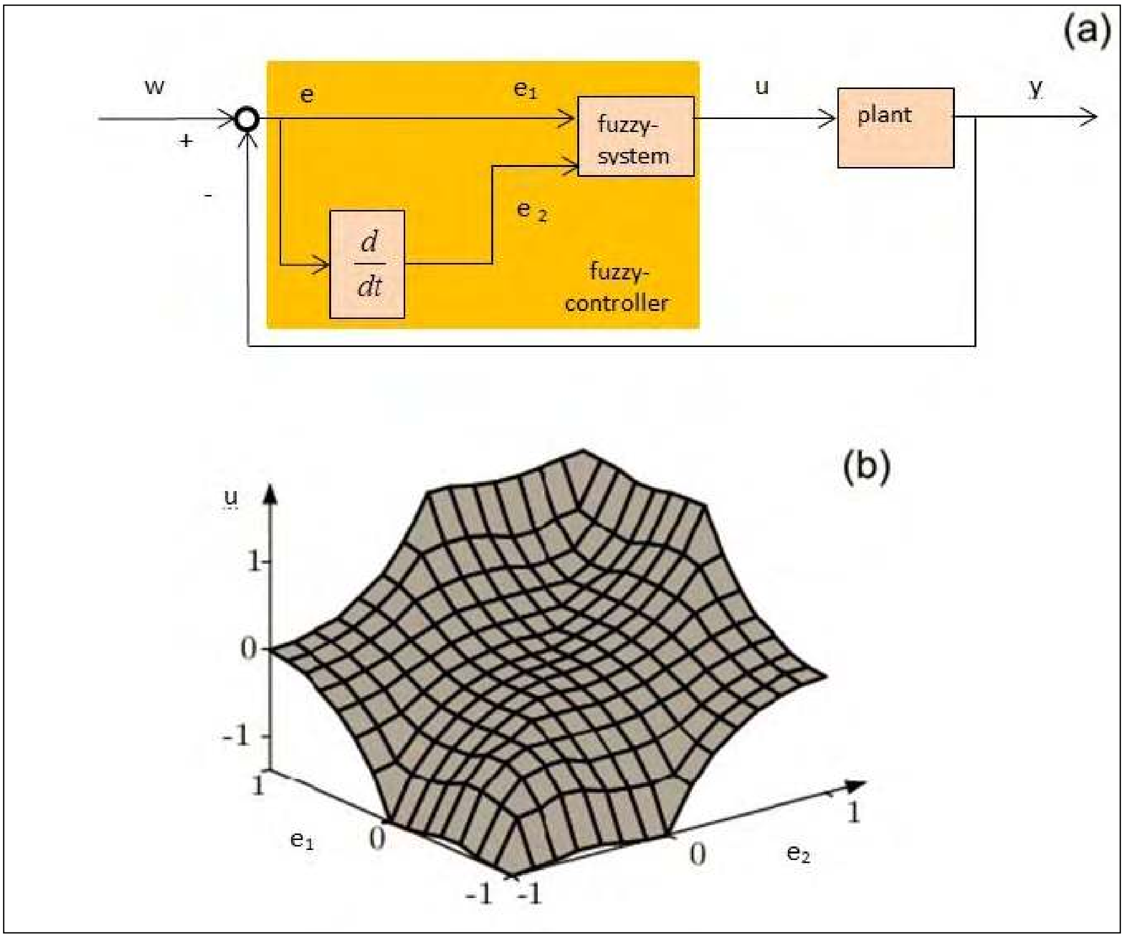

Figure 33: Control system with PD-type fuzzy controller: (a) block diagram and (b) 3D representation of the characteristics of the fuzzy system with e_1 = e and e_2 = e′ (adapted from http://www.atp.ruhr-uni-bochum.de)

For the case with more than two inputs, projections on the 3D space can be used to generate multiple 3D diagrams, but in general these representations have only a limited usefulness. A fuzzy controller is typically a classical controller using nonlinear characteristics. But the design and parameterisation is entirely different.

Example of a fuzzy control system

In the following the design and functioning of a fuzzy control system will be presented using the example of the portal-type loading crane shown in Figure 34.

Figure 34: View of a portal-type loading crane (adapted from http://www.atp.ruhr-uni-bochum.de)