# A Short Fuzzy Logic Tutorial

April 8, 2010

The purpose of this tutorial is to give a brief information about fuzzy logic systems. The tutorial is prepared based on the studies [2] and [1]. For further information on fuzzy logic, the reader is directed to these studies.

A fuzzy logic system (FLS) can be defined as the nonlinear mapping of an input data set to a scalar output data [2]. A FLS consists of four main parts: fuzzifier, rules, inference engine, and defuzzifier. These components and the general architecture of a FLS is shown in Figure 1.

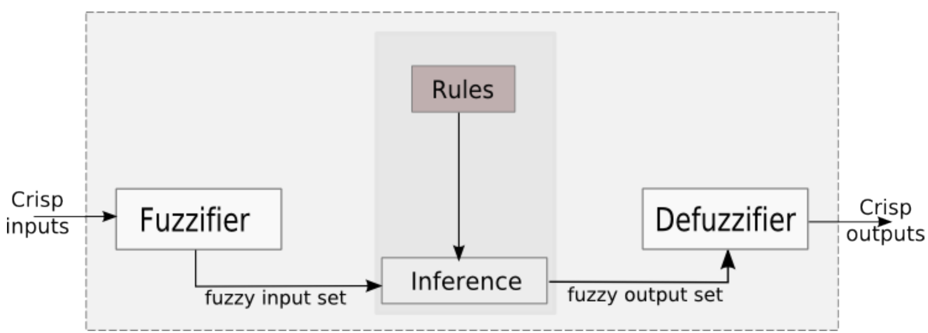


Figure 1: A Fuzzy Logic System.

The process of fuzzy logic is explained in Algorithm 1: Firstly, a crisp set of input data are gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the defuzzification step.

In order to exemplify the usage of a FLS, consider an air conditioner system controlled by a FLS (Figure 2). The system adjusts the temperature of the room according to the current temperature of the room and the target value. The fuzzy engine periodically compares the room temperature and the target temperature, and produces a command to heat or cool the room.

**Algorithm 1** Fuzzy logic algorithm

```
1. Define the linguistic variables and terms (initialization)
2. Construct the membership functions (initialization)
3. Construct the rule base (initialization)
4. Convert crisp input data to fuzzy values
   using the membership functions (fuzzification)
5. Evaluate the rules in the rule base (inference)
6. Combine the results of each rule (inference)
7. Convert the output data to non-fuzzy values (defuzzification)
```

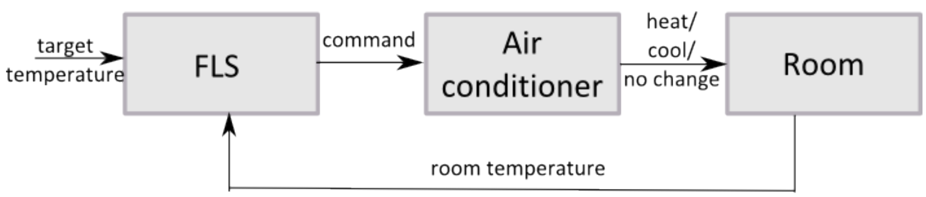


Figure 2: A Simple FLS to Control an Air Conditioner.

**Linguistic Variables**

Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable is generally decomposed into a set of linguistic terms.

*Example:* Consider the air conditioner in Figure 2. Let *temperature* (t) is the linguistic variable which represents the temperature of a room. To qualify the temperature, terms such as "hot" and "cold" are used in real life. These are the linguistic values of the temperature. Then, $T(t) = \{$*too-cold, cold, warm, hot, too-hot*$\}$ can be the set of decompositions for the linguistic variable temperature. Each member of this decomposition is called a linguistic term and can cover a portion of the overall values of the temperature.

**Membership Functions**

Membership functions are used in the fuzzification and defuzzification steps of a FLS, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function is used to quantify a linguistic term. For instance, in Figure 3, membership functions for the linguistic terms of temperature variable are plotted. Note that, an important characteristic of fuzzy logic is that a numerical value does not have to be fuzzified using only one membership function. In other words, a value can belong to multiple sets at the same time. For example, according to Figure 3, a temperature value can be considered as "cold" and "too-cold" at the same time, with different degree of memberships.

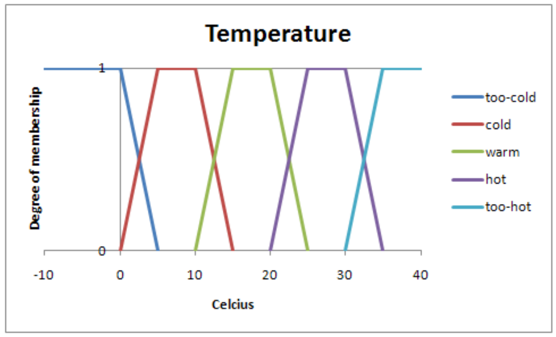


Figure 3: Membership Functions for *T(temperature)* = {*too-cold, cold, warm, hot, too-hot*}.

There are different forms of membership functions such as triangular, trapezoidal, piecewise linear, Gaussian, or singleton (Figure 4). The most common types of membership functions are triangular, trapezoidal, and Gaussian shapes. The type of the membership function can be context dependent and it is generally chosen arbitrarily according to the user experience [2].

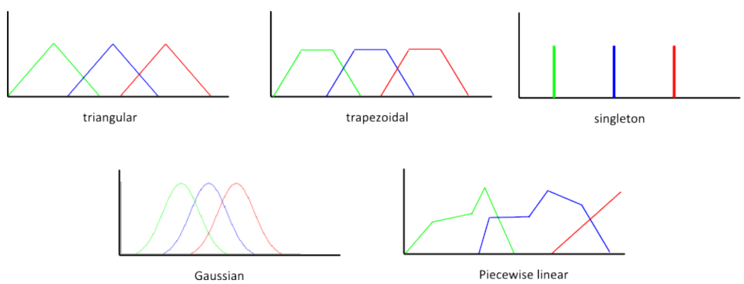


Figure 4: Different Types of Membership Functions.

**Fuzzy Rules**

In a FLS, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion. In Table 1, sample fuzzy rules for the air conditioner system in Figure 2 are listed. Table 2 shows the matrix representation of the fuzzy rules for the said FLS. Row captions in the matrix contain the values that current room *temperature* can take, column captions contain the values for *target* temperature, and each cell is the resulting *command* when the input variables take the values in that row and column. For instance, the cell (3, 4) in the matrix can be read as follows: *If temperature is cold and target is warm then command is heat.*

Table 1: Sample fuzzy rules for air conditioner system

| | Fuzzy Rules |
|---|---|
| 1. | IF (temperature is *cold* OR *too-cold*) AND (target is *warm*) THEN command is *heat* |
| 2. | IF (temperature is *hot* OR *too-hot*) AND (target is *warm*) THEN command is *cool* |
| 3. | IF (temperature is *warm*) AND (target is *warm*) THEN command is *no-change* |

Table 2: Fuzzy matrix example

| temperature/target | too-cold | cold | warm | hot | too-hot |
|---|---|---|---|---|---|
| **too-cold** | no-change | heat | heat | heat | heat |
| **cold** | cool | no-change | heat | heat | heat |
| **warm** | cool | cool | no-change | heat | heat |
| **hot** | cool | cool | cool | no-change | heat |
| **too-hot** | cool | cool | cool | cool | no-change |

**Fuzzy Set Operations**

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The operations on fuzzy sets are different than the operations on non-fuzzy sets. Let $\mu_A$ and $\mu_B$ are the membership functions for fuzzy sets $A$ and $B$. Table 3 contains possible fuzzy operations for OR and AND operators on these sets, comparatively. The mostly-used operations for OR and AND operators are *max* and *min*, respectively. For complement (NOT) operation, Eq. 1 is used for fuzzy sets.

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \tag{1}$$

Table 3: Fuzzy set operations

| OR (Union) | | AND (intersection) | |
|---|---|---|---|
| MAX | $Max\{\mu_A(x), \mu_B(x)\}$ | MIN | $Min\{\mu_A(x), \mu_B(x)\}$ |
| ASUM | $\mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$ | PROD | $\mu_A(x)\mu_B(x)$ |
| BSUM | $Min\{1, \mu_A(x) + \mu_B(x)\}$ | BDIF | $Max\{0, \mu_A(x) + \mu_B(x) - 1\}$ |

After evaluating the result of each rule, these results should be combined to obtain a final result. This process is called inference. The results of individual rules can be combined in different ways. Table 4 contains possible accumulation methods that are used to combine the results of individual rules. The maximum algorithm is generally used for accumulation.

Table 4: Accumulation methods

| Operation | Formula |
|---|---|
| Maximum | $Max\{\mu_A(x), \mu_B(x)\}$ |
| Bounded sum | $Min\{1, \mu_A(x) + \mu_B(x)\}$ |
| Normalized sum | $\frac{\mu_A(x)+\mu_B(x)}{Max\{1, Max\{\mu_A(x'), \mu_B(x')\}\}}$ |

**Defuzzification**

After the inference step, the overall result is a fuzzy value. This result should be defuzzified to obtain a final crisp output. This is the purpose of the defuzzifier component of a FLS. Defuzzification is performed according to the membership function of the output variable. For instance, assume that we have the result in Figure 5 at the end of the inference. In this figure, the shaded areas all belong to the fuzzy result. The purpose is to obtain a crisp value, represented with a dot in the figure, from this fuzzy result.

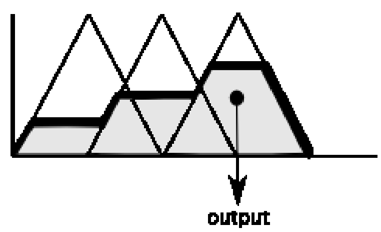


Figure 5: Defuzzification step of a FLS.

There are different algorithms for defuzzification too. The mostly-used algorithms are listed in Table 5. The meanings of the variables used in Table 5 are explained in Teble 6.

Table 5: Defuzzification algorithms [1]

| Operation | Formula |
|---|---|
| Center of Gravity | $U = \frac{\int_{min}^{max} u\,\mu(u)\,du}{\int_{min}^{max} \mu(u)\,du}$ |
| Center of Gravity for Singletons | $\frac{\sum_{i=1}^{p}[u_i\,\mu_i]}{\sum_{i=1}^{p}[\mu_i]}$ |
| Left Most Maximum | $U = inf(u')$, $\mu(u') = sup(\mu(u))$ |
| Right Most Maximum | $U = sup(u')$, $\mu(u') = sup(\mu(u))$ |

Table 6: The variables in Table 5

| **Variable** | **Meaning** |
|---|---|
| U | result of defuzzification |
| u | output variable |
| p | number of singletons |
| $\mu$ | membership function after accumulation |
| i | index |
| min | lower limit for defuzzification |
| max | upper limit for defuzzification |
| sup | largest value |
| inf | smallest value |

# References


[1] Fuzzy control programming. Technical report, International Electrotechnical Commision, 1997.

[2] J. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377, Mar 1995.