KHWARZIMIC
SCIENCE SOCIETY

# Fuzzy Logic Control Using Matlab
## *Part I*

Naveed Ahmed, FAST-NUCES, Lahore.

*This 2-day workshop was held under the auspices of the Khwarzimic Science Society at the University of Lahore in April 2003. Full details and copies of this document can be obtained from http://www.khwarzimic.org.*

**What is Fuzzy Logic?**
Fuzzy Logic is a convenient way to map input space to output space.

E.g., How much to tip at hotel? Input space is the quality of service and output space is the amount of tip.

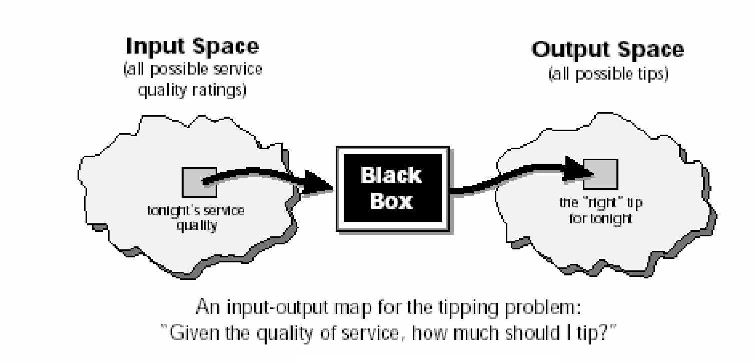

Input Space
(all possible service quality ratings)
Output Space
(all possible tips)
tonight's service quality
Black Box
the "right" tip for tonight
An input-output map for the tipping problem:
"Given the quality of service, how much should I tip?"

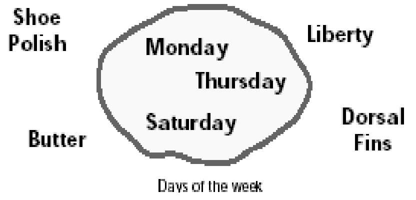
Figure. 1

**Things that can g o in Black Box**

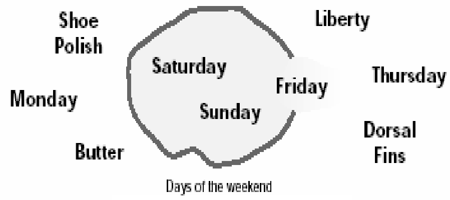
- Fuzzy Systems
- Linear Systems
- Expert Systems
- Neural Networks
- Differential Equations
- Interpolated multidimensional lookup tables

Fuzzy logic is often the best way because it is fast and cheap.

**Fuzzy Sets**
A fuzzy set is a set without any crisp, clearly defined boundaries. It can contain elements with only a partial degree of membership. Classical sets either wholly include given element or wholly discludes elements. E.g., set of days of the week. While example of a fuzzy set would be set of days that make up weekend.



a. Classical Set



b. Fuzzy Set

In fuzzy logic, truth of any statement becomes a matter of degree. Fuzzy reasoning gives us the ability to reply to a yes-no question with a not-quite-yes-or-no answer. This is the kind of thing that humans do all the time. Reasoning in fuzzy is just a matter of generalizing the familiar yes-no (Boolean) logic. If we give 'true' the numerical value of 1 and "false" the numerical value of 0, fuzzy logic also permits in-between values like 0.2 and

0.7345.

In the plot on the left in figure 4, notice that at midnight on Friday, just as the second hand sweeps past 12, the weekend-ness value jumps discontinuously from 0 to1. It doesn't really connect with our real-world experience of the weekend-ness. The plot on the right shows a smoothly varying curve that accounts for the fact that all of Friday and parts of Thursday to a small degree partake of the quality of weekend-ness and thus deserve partial membership in the fuzzy set of weekend moments. The curve that defines the weekend-ness of any instant in time is a function that maps the input space (time of week) to the output space (weekend-ness). Specifically it is known as a membership function.

**Membership Functions**

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse.

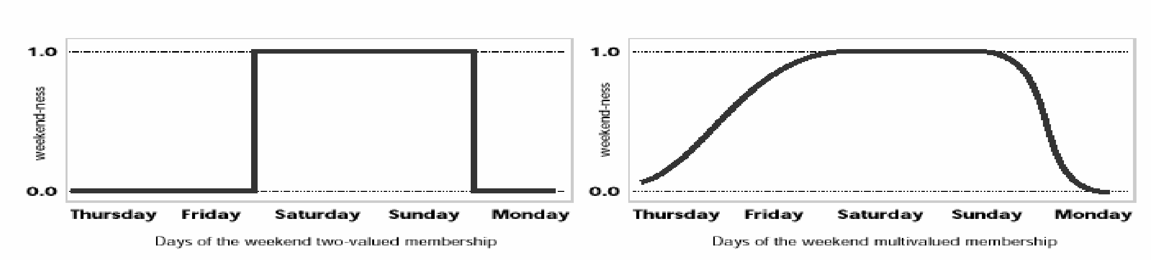


Figure. 2

The output-axis is a number known as the membership value between 0 and 1. The curve is known as a membership function and is often given the designation of µ.  E.g., of membership functions representing different seasons of the year are given in figure 4.

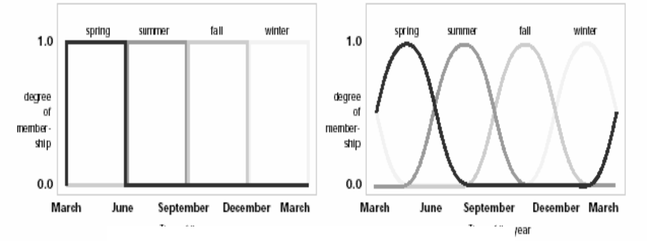


Figure. 3

The only condition a membership function must satisfy is that it must vary between 0 and 1.

A classical set might be described as

A= { x | x>6 }

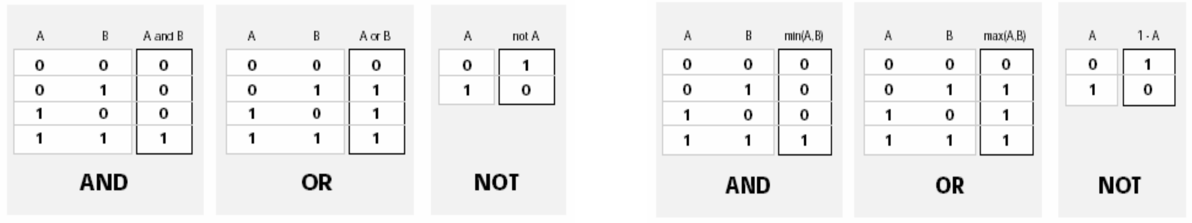
A fuzzy set is an extension of a classical set. If X is the universe of discourse and its elements are denoted by x, then the fuzzy set A in X is defined as a set of ordered pairs.

A= { x, $\mu_A$ (x) | x∈ X}

$\mu_A$ (x) is called the membership function (or MF) of x in A. The membership function maps each element of X to a membership value between 0 and 1. The fuzzy logic toolbox includes 11 built-in membership function types, triangular membership function, trapezoidal membership function, simple Gaussian curve and a two-sided composite of two different Gaussian curves, generalized bell membership function, sigmoidal membership function, difference between two sigmoidal functions and the product of two sigmoidal functions, polynomial based curves include Z, S and pi curves. Fuzzy Logic toolbox allows you to create your own membership functions also.

## Fuzzy Logic Operators

We know what’s so fuzzy about fuzzy logic, but what about the logic? Fuzzy logic is a superset of standard Boolean logic. If we keep the fuzzy values to the extremes of 1 (completely true) and 0 (completely false), standard logical operators will hold.



| A | B | A and B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

AND

| A | B | A or B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

OR

| A | not A |
|---|---|
| 0 | 1 |
| 1 | 0 |

NOT

| A | B | min(A,B) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

AND

| A | B | max(A,B) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

OR

| A | 1 - A |
|---|---|
| 0 | 1 |
| 1 | 0 |

NOT

Figure. 4

The input values can be real numbers between 0 and 1. What function will preserve the results of the classical logic truth table and also extend to all real numbers between 0 and 1.One answer is the min operation. We can replace the OR operation with the max function, so that A OR B becomes equivalent to max (A, B). Finally the operation NOT A becomes equivalent to the operation 1-A. Fuzzy intersection or conjunction (AND), fuzzy union or disjunction (OR), and fuzzy complement (NOT) can either be defined using the classical operators for these functions: AND=min, OR=max, and NOT= additive complement or using customized functions. Fuzzy logic toolbox uses the classical operator for the fuzzy complement, but the AND and OR operators can be easily customized if desired.

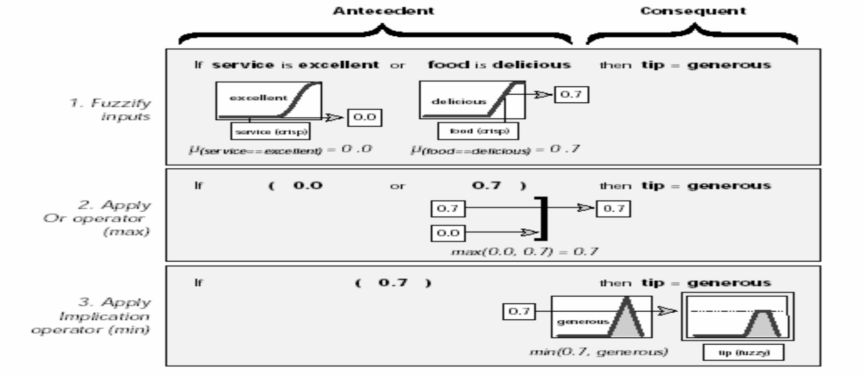

Antecedent
Consequent
If service is excellent or food is delicious then tip = generous
1. Fuzzify inputs
excellent
service (crisp)
0.0
delicious
food (crisp)
0.7
μ(service==excellent) = 0 .0
μ(food==delicious) = 0 .7
If ( 0.0 or 0.7 ) then tip = generous
2. Apply Or operator (max)
0.7
0.0
0.7
max(0.0, 0.7) = 0.7
If ( 0.7 ) then tip = generous
3. Apply Implication operator (min)
0.7
generous
min(0.7, generous)
tip (fuzzy)

Figure. 4

## If-Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. Conditional statements, if-then rules are the things that make fuzzy logic useful.

A single fuzzy if-then rule assumes the form:

*If x is A then y is B*

Where A and B are linguistic values defined by fuzzy sets on the ranges X and Y, respectively. The if-part of the rule *“x is A”* is called the antecedent or premise, while the then-part of the rule *“y is B”* is called the consequent or conclusion. An example of such a rule might be

*“If service is good then tip is average”*

Note that antecedent is an interpretation that returns a single number between 0 and 1, whereas the consequent is an assignment that assigns the entire fuzzy set B to the output variable y. A less confusing way of writing the rule would be

*“If service ==good then tip=average”*.

So the input to an if-then rule is the current value for the input variable (service) and the output is an entire fuzzy set (average). Interpreting an if-then rule involves distinct parts. First evaluating the antecedent (which involves fuzzifying the input and applying necessary fuzzy operators. Second, applying the results to the consequents (known as implication).

In the case of two-valued or binary logic, if the premise is true, then the conclusion is true. In case of fuzzy if-then rule if the antecedent is true to some degree of membership, then the consequent part is also true to that same degree.

*In binary logic: p ->q  (p and q are either true or false)*
*In fuzzy logic: 0.5 p ->0.5 q (partial antecedents apply partially)*

The antecedents of a rule can have multiple parts

*If sky is gray and wind is strong and barometer is falling then…*

In which case all the parts of the antecedent are calculated simultaneously and resolved to a single number using the fuzzy logical operators discussed previously. The consequent of a rule can also have multiple parts:

*If temperature is cold then hot water valve is open and cold-water valve is shut*

In which case all consequents are affected equally by the result of the antecedent. The consequent specifies a fuzzy set to be assigned to the output. The implication function then modifies the fuzzy set to the degree specified by the antecedent. The most common ways to modify the output fuzzy sets are truncation using the min function (where the fuzzy set is chopped off, as shown in Figure 4. part 3) or scaling using the prod function (where the output fuzzy set is squashed). Both are supported by fuzzy logic toolbox.

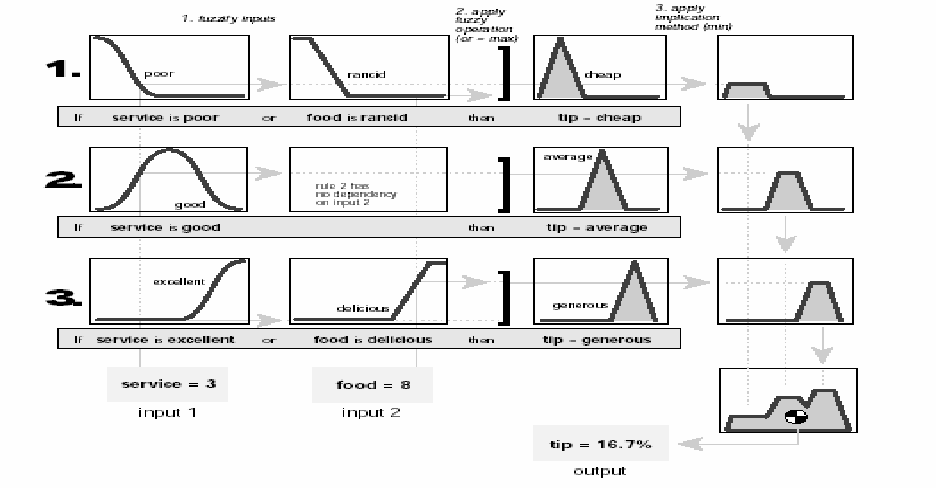
One rule by itself does not do much good. What are needed are two or more rules that can play off one another. The output of each rule is a fuzzy set, but in general we want the output for an entire collection of rules to be a single number.  How are all these fuzzy sets distilled into a single crisp result for the output variable? First the output fuzzy sets for each rule are aggregated into a single output fuzzy set. Then the resulting set is deffuzzified, or resolved to a single number.

**Fuzzy Inference Systems**

Fuzzy inference is the actual process of mapping from a given input to an output using fuzzy logic. The process involves all the pieces that we have discussed previously i.e., membership functions, fuzzy logic operators, and if-then rules.

Example:

We will see how everything fits together using the two-input (service, food) and one output(tip) three rule tipping problem.


1. fuzzify inputs
2. apply fuzzy operation (or = max)
3. apply implication method (min)
1.
poor
rancid
cheap
If service is poor or food is rancid then tip = cheap
2.
good
rule 2 has no dependency on input 2
average
If service is good then tip = average
3.
excellent
delicious
generous
If service is excellent or food is delicious then tip = generous
service = 3
input 1
food = 8
input 2
tip = 16.7%
output

Information flows from left to the right, from two inputs to a single output.

In the fuzzy logic toolbox, there are five parts of the fuzzy inference process:

1. **Fuzzification of the input variables**

   The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable and the output is a fuzzy degree of membership (always in interval between 0 and 1).

2. **Application of the fuzzy operator (AND or OR) in the antecedent**

If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function. Any number of well-defined methods can fill in for the AND operation or the OR operation. In fuzzy logic toolbox, two built-in AND methods are supported: min (minimum) and prod (product). Two built-in OR methods are also supported: max(maximum), and the probabilistic OR method probor.

3. **Implication from the antecedent to the consequent**
   The implication method is defined as the shaping of the consequent ( a fuzzy set) based on the antecedent ( a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication occurs for each rule. Two built-in methods are supported, min (minimum) which truncates the output fuzzy set, and prod (product) which scales the output fuzzy set.
4. **Aggregation of the consequents across the rules**
   Unify the outputs of each rule
5. **Defuzzification**
   Input for defuzzification phase is unified fuzzy set formed by aggregation of consequents and output is crisp number. If there are more than one output variables, final output for each variable is a crisp number. The most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. There are five built-in methods supported: centroid, bisector, middle of maximum ( the average of the maximum value of the output set), largest of maximum, and smallest of maximum.