Linguistic Variables

- Linguistic variable is an important concept in fuzzy logic and plays a key role in its applications, especially in the fuzzy expert system.
- Linguistic variable is a variable whose values are words in a natural language.
- For example, "speed" is a linguistic variable, which can take the values as "slow", "fast", "very fast" and so on.
- A fuzzy variable is characterized by \((X, U, R(X))\), \(X\) is the name of the variable; \(U\) is the universe of discourse; and \(R(X)\) is the fuzzy set of \(U\).
Linguistic Variables

- For example: $X = \text{"old"}$ with $U = \{10, 20, \ldots, 80\}$, and $R(X) = 0.1/20 + 0.2/30 + 0.4/40 + 0.5/50 + \ldots + 1/80$ is called a fuzzy membership of "old"

- Linguistic variable is a variable of higher order than fuzzy variable, and it take fuzzy variable as its values

- A linguistic variable is characterized by: $(x, T(x), U, M)$
  
  $x$: name of the variable
  
  $T(x)$: the term set of $x$, the set of names or linguistic values assigned to $x$, with each value is a fuzzy variable defined in $U$
  
  $M$: Semantic rule associate with each variable (membership)

Example

- $x$: "age" is defined as a linguistic variable

- $T(\text{age}) = \{\text{young, not young, very young, more or less old, old}\}$

- $U$: $U = \{0, 100\}$

- $M$, Defines the membership function of each fuzzy variable

  for example: $M(\text{young}) = \text{the fuzzy set for age below 25 years with membership of } \mu_{\text{young}}$
Linguistic Variables

Linguistic variables collect elements into similar groups where we can deal with less precisely and hence we can handle more complex systems.
Linguistic Variables

Concentration and Dilation of Fuzzy Variables

- Let $A$ be a linguistic value characterized by a fuzzy set with $\mu_A$, then $A^k$ is a modified version of the original linguistic value so that, $A^k = (\mu_A(x))^k$

- Concentration $\mu_{\text{con}}(A) = (\mu_A(x))^2$

- Dilation $\mu_{\text{dil}}(A) = (\mu_A(x))^{1/2}$

- This is called linguistic hedges, such as: very old; not old; and not......

- Hedges allow for the generation of fuzzy statements through mathematical calculations
Linguistic Variables

Concentration and Dilation: Example

- Consider the following two fuzzy variables "young" and "old" with their membership functions:
  - $\mu_{young}(x) = \frac{1}{1 + (x/20)^4}$; $\mu_{old}(x) = \frac{1}{1 + ((x-100)/20)^6}$
  - $x$ is the age for a given person

- Determine the membership function for the fuzzy variables "more or less old"; "not young not old"; "young but not too young"

- More or less old = $\{\frac{1}{1 + ((x-100)/20)^6}\}^{0.5}$

- Not young not old =
  $= \min\{1 - 1/ [1 + (x/20)^4]; 1 - 1/ [1 + ((x-100)/20)^6]\}$

- Young but not too young
  $= \min\{1/ [1 + (x/20)^4]; (1 - 1/ [1 + ((x-100)/20)^6])^{0.5}\}$
**Linguistic Variables**

Concentration and Dilation: Example

Membership Functions

**Rules of Thumb**

- Usually use Quadratic, triangular or trapezoidal fuzzy sets for continuous variables (triangular is most common)
- Quadratic membership function

\[
A(x) = \begin{cases} 
2 \frac{(x-a)}{(b-a)^2} & a < x < \frac{a+b}{2} \\
- \frac{2 (x-b)^2}{(b-a)^2} & \frac{a+b}{2} < x < b \\
- \frac{2 (x-c)^2}{(c-b)^2} & b < x < \frac{b+c}{2} \\
\frac{2 (x-c)}{(c-b)^2} & \frac{b+c}{2} < x < c \\
0 & \text{otherwise}
\end{cases}
\]
Membership Functions

Rules of Thumb

- Triangular or trapezoidal

Triangular Fuzzy Set

Trapezoidal Fuzzy Set

Degree of Membership (y)

Variable Values

1.0

0

a b c d

Degree of Membership (y)

Variable Values

1.0

0

a b' c d

Membership Functions

Rules of Thumb

- 3 to 7 fuzzy sets are usually used to describe each parameter
- Fuzzy sets should be narrow and more closely spaced near the desired operating point or in regions where a finer degree of control is needed
Membership Functions

Rules of Thumb

- Allow significant overlap between sets
- Often make the base of one set coincide the peak of the adjacent set

Fuzzy If-Then Rules

- As any other logic, the rules of inference in fuzzy logic govern the deduction of proposition \( q \) from a set of premises \( \{p_1, p_2, \ldots\} \). In fuzzy logic both premises and conclusions are allowed to be fuzzy proposition (describe actions to take under specific conditions)
- Fuzzy if-then rule take the form "IF \( x \) is \( A \) THEN \( y \) is \( B \)"
- \( A \) and \( B \) are linguistic values of fuzzy sets defined in \( X \) and \( Y \)
- "\( X \) is \( A \)”, called "antecedent” or "premise", "\( Y \) is \( B \)”, called "conclusion” or "consequence”
- Ex: if pressure is high, then volume is small
  - if tomato is red, then it is ripe
**Fuzzy If-Then Rules**

- Fuzzy rules are always represented in matrix format.
- If Speed is Moderate and Position is Centered then Change in speed is Faster.
- If Speed is Low and Position is Centered then Change in speed is Much Faster.

<table>
<thead>
<tr>
<th>Position</th>
<th>VS</th>
<th>S</th>
<th>M</th>
<th>F</th>
<th>MF</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>C</td>
<td>MF</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>MS</td>
</tr>
<tr>
<td>R</td>
<td>MF</td>
<td>F</td>
<td>F</td>
<td>Z</td>
<td>S</td>
</tr>
<tr>
<td>FR</td>
<td>F</td>
<td>F</td>
<td>Z</td>
<td>S</td>
<td>MS</td>
</tr>
</tbody>
</table>

**Partition universe of discourse for each input and output into fuzzy regions that overlap and completely cover their respective universes.**
Fuzzy If-Then Rules

Generating Fuzzy Rules

Use numeric data to generate 1 rule for each input output pair.

Ex: if \( x_1 \) is A and \( x_2 \) is B then \( z \) is C, where A, B, and C are fuzzy sets defined in the previous step.

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>c1</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>c2</td>
</tr>
</tbody>
</table>

For each input–output pair, assign data to the region with the highest membership.

\( a1 = 0.8 \) in \( B1 \) and \( 0.2 \) in \( B2 \) then \( B1 \)
\( b1 = 0.7 \) in \( S1 \) and \( 0.3 \) in \( S2 \) then \( S1 \)
\( c1 = 0.9 \) in \( C \) and \( 0.1 \) in \( B1 \) then \( C \)

Then rule becomes: if \( x_1 \) is \( B1 \) and \( x_2 \) is \( S1 \) then \( z \) is \( C \)
**Fuzzy If-Then Rules**

**Generating Fuzzy Rules**

- Assign a degree of truth (Strength) for each rule.
  
  \[
  \text{degree of truth} = \text{product of membership of data in each rule}
  \]

  Ex: rule 1 strength = 0.8 x 0.7 x 0.9 = 0.504

- In case of conflict, same premises give different results, keep the rule with the highest degree of truth

- Construct a rule matrix

```
<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Fuzzy If-Then Rules**

**Generating Fuzzy Rules: Example**

- A controller with three inputs. Each input is defined with 5 fuzzy sets. How many rules are required to describe the system?

  Number of rules 5 x 5 x 5 = 125 rules, these represents all the different combinations between all fuzzy sets

- If 7 fuzzy sets are needed then the number of rules = 7 x 7 x 7 = 343 rules
Fuzzy If-Then Rules

Generating Fuzzy Rules: Hierarchical Controller

- Hierarchical controller used to reduce the number of rules
- Rules are organized in layers with the output of one layer used as an input to the next layer
- Having a controller with three inputs $x_1$, $x_2$, and $x_3$ and one output $z$
- First layer rules is made between $x_1$ and $x_2$ fuzzy sets
  
  If $x_1$ is $A$ and $x_2$ is $B$ then $\lambda$ is $B_1$

---

Fuzzy If-Then Rules

Generating Fuzzy Rules: Hierarchical Controller

- Second layer rules is made between $\lambda$ and $x_3$
  
  If $\lambda$ is $B_1$ and $x_3$ is $D$ then $z$ is $C$
- Example:
  - A 2-level controller with three input variables each consists of 5 fuzzy sets, and one output variable
  - Number of rules required for first layer $5 \times 5 = 25$
  - Number of rules required for second layer $5 \times 5 = 25$
  - Total number of rules 50 compared to 125 rules for one layer controller
**Fuzzy If-Then Rules**

*Generating Fuzzy Rules: Hierarchical Controller*

- **Three input controller**
  - $x_1, x_2, x_3$ to Layer 1 Cont. to Layer 2 Cont.

- **Four input controller**
  - $x_1, x_2$ to Layer 1 Cont.1 to Layer 2 Cont.
  - $x_3, x_4$ to Layer 1 Cont.2 to Layer 2 Cont.