Fuzzy Logic

IMDL
{By Kevin Harrelson formerly at Harris Corporation now at Agilent}

Fuzzy Logic: What is it???

• It does not involve putting fake fur on robots.

• It is a different way of looking at the world.

• It is a superset of Boolean logic!

• It deals with “shades of gray!”
A Better Method to Deal With the Real World

• Not just “True” and “False.”
• Takes on a range of values
  – True
  – Mostly True
  – Half True
  – Kind of True
  – False
• Values range from 0 to 1.
  – Including decimal values (0.2, 0.7, etc.)

Why?
Without Fuzzy Logic

```c
#include <confusing.h>

void bladder();
{
    if ( read_sensor(3.14159) > sqrt(42) )
    {
        do_something_confusing( make_noise(12) );
    }

    junk[max(my_IQ, my_shoe_size)] = peek(0x0f00);
    four_score[7] = "years ago";
}
```

With Fuzzy Logic

```

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>Straight</td>
<td>Straight</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>Straight</td>
<td>Straight</td>
</tr>
</tbody>
</table>
```
Now, Let’s see how this works...

Fuzzy Logic Process

```
“Crisp” Input ─── Fuzzification ─── “Fuzzy” Input

Fuzzy Logic
-or-
F.A.M.

“Fuzzy” Output ─── De-Fuzzification ─── “Crisp” Output
```
The First Step...

Fuzzification

How tall is Kevin?

- Very Tall?
- Tall?
- Average?
- Short?
- Very Short?
How tall is Kevin?

- Very Tall (7 feet)?
- Tall (6 feet)?
- Average (5 feet)?
- Short (4 feet)?
- Very Short (3 feet)?

Fuzzification Rules

- Very Short
- Short
- Average
- Tall
- Very Tall
Some Examples:

If you are 5 feet:
- Very tall - 0%
- Tall - 0%
- Average - 100%
- Short - 0%
- Very Short - 0%

*Same as Boolean logic (so far…)*

Some Examples:

If you are 5½ feet:
- Very tall - 0%
- Tall - 50%
- Average - 50%
- Short - 0%
- Very Short - 0%

*Not Boolean logic (Whoa. Cool!)*
How tall is Kevin?

Kevin is 6’ 2”

- Very Tall - 16%
- Tall - 84%
- Average - 0%
- Short - 0%
- Very Short - 0%

Fuzzy Representation

- All fuzzy variables are theoretically represented as a number between 0 and 1.
- The fuzzy number can be represented on a computer as a number between 0 and 255.
Some Hints

- Fuzzy values are **NOT** probabilities.
- HOWEVER, it might help to think of them as probability values.

The Second Step...

Fuzzy Logic & the FAM
Fuzzy Operators: AND

FAND(\(A,B\)) - Fuzzy AND = min(\(A,B\))
FAND( 100, 30 ) = 30
FAND( 20, 250 ) = 20
FAND( 1, 0 ) = 0 -- Just like boolean logic
FAND( 1, 1 ) = 1 -- Geeeee. This too!

Fuzzy Operators: OR

FOR(\(A,B\)) - Fuzzy OR = max(\(A,B\))
FOR( 100, 30 ) = 100
FOR( 20, 250 ) = 250
FOR( 1, 0 ) = 1 -- Just like boolean logic
FOR( 0, 0 ) = 0 -- Geeeee. This too!
Fuzzy Operators: NOT

FNOT( \( A \)) - Fuzzy NOT = 100\% - A

(100\% defined as 255)

FNOT( 100 ) = 155
FNOT( 250 ) = 5
FNOT( 255 ) = 0
FNOT( 0 ) = 255

- See the similarity to Boolean logic???

Fuzzy Associative Memory
(FAM)

The Next Step
Fuzzy Associative Memory

- It is a Fuzzy Truth Table
- Shows all possible outputs for all possible inputs
- Easy to create!

FAM Example

FUZZY-BOT
First, the sensors

Sharp Sensor Mappings:
- Nothing = 80
- Very Far = 100
- Far = 120
- Near = 130 -- Note: non-linear spacing
- Very Near = 140

Second, the Motors

Direction Output Mappings:
- Hard Left = -100
- Left = -20
- Straight = 0
- Right = 20
- Hard Right = 100
Lastly, the FAM (rule table)

<table>
<thead>
<tr>
<th>Left Sensor</th>
<th>VN</th>
<th>N</th>
<th>F</th>
<th>VF</th>
<th>VVF</th>
</tr>
</thead>
<tbody>
<tr>
<td>VN</td>
<td>HL</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>N</td>
<td>HL</td>
<td>L</td>
<td>HR</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>F</td>
<td>HL</td>
<td>HL</td>
<td>L</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>VF</td>
<td>HL</td>
<td>L</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>VVF</td>
<td>HL</td>
<td>L</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

• V=Very
• N=Near
• F=Far
• N=Near
• H=Hard
• L=Left
• R=Right
• S=Straight

FAM Operation

• AND the associated inputs…
• OR the result with the result for that output group.
FUZZ-BOT Example

- **Left Sensor**
  - Very Near = 80%
  - Near = 20%

- **Right Sensor**
  - Near = 30%
  - Far = 70%

<table>
<thead>
<tr>
<th>Left Sensor</th>
<th>VN 80%</th>
<th>N 20%</th>
<th>F 0%</th>
<th>VF 0%</th>
<th>VVF 0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>VN 80%</td>
<td>HL 0%</td>
<td>HR 30%</td>
<td>HR 70%</td>
<td>HR 0%</td>
<td>HR 0%</td>
</tr>
<tr>
<td>N 20%</td>
<td>HL 0%</td>
<td>L 70%</td>
<td>HR 0%</td>
<td>R 30%</td>
<td>R 0%</td>
</tr>
<tr>
<td>F 0%</td>
<td>HL 0%</td>
<td>HL 0%</td>
<td>L 70%</td>
<td>S 0%</td>
<td>S 0%</td>
</tr>
<tr>
<td>VF 0%</td>
<td>HL 0%</td>
<td>L 0%</td>
<td>S 70%</td>
<td>S 0%</td>
<td>S 0%</td>
</tr>
<tr>
<td>VVF 0%</td>
<td>HL 0%</td>
<td>L 0%</td>
<td>S 0%</td>
<td>S 0%</td>
<td>S 0%</td>
</tr>
</tbody>
</table>

HL = 0%
L = 20%
S = 0%
R = 0%
HR = 30% OR 70% OR 20% = 70%

NOTE:
0+20+0+0+70 = 100%
Can We Simplify This???

Removing the FAM

Simplifying the Table

1) Group the common Outputs (similar to K-Maps)

2) For each block:
   (each value OR’d together) AND
   (each value OR’d together)

3) OR the output of each block together
### FUZZ-BOT Example

- **HL** = (RVN \(\text{AND}\) (LVN \(\text{OR}\) LN \(\text{OR}\) LF \(\text{OR}\) LVF \(\text{OR}\) LVVF)) \(\text{OR}\) (LF \(\text{AND}\) RN)
- **L** = (LN \(\text{AND}\) RN) \(\text{OR}\) (LF \(\text{AND}\) RF) \(\text{OR}\) ((LVF \(\text{OR}\) LVVF) \(\text{AND}\) RN)
- **S** = ((LVF \(\text{OR}\) LVVF) \(\text{AND}\) (RF \(\text{OR}\) RVF \(\text{OR}\) RVVF)) \(\text{OR}\) (LF \(\text{AND}\) (RVF \(\text{OR}\) RVVF))
- **R** and **HR** are left as an exercise to the student.
The Final Chapter...

De-Fuzzification

Defuzzification: Two Methods

1) Winner Take All

2) Weighted Average
**Winner Take All**

- Output “Hard Right” = 70%
- It is the winner!
- Output = **100** (from output mapping)
- Looses some of the smoothness of fuzzy logic.

**Direction Output Mappings**
- Hard Left = -100
- Left = -20
- Straight = 0
- Right = 20
- Hard Right = 100

**Output of FAM**
- HL = 0%
- L = 20%
- S = 0%
- R = 0%
- HR = 70%

---

**Weighted Average**

- Output “Hard Right” = 70%
- Output “Left” = 20%
- \((70 \times 100 + 20 \times -20) / (70 + 20)\)
- Output = **73.3**

**Direction Output Mappings**
- Hard Left = -100
- Left = -20
- Straight = 0
- Right = 20
- Hard Right = 100

**Output of FAM**
- HL = 0%
- L = 20%
- S = 0%
- R = 0%
- HR = 70%
Any Questions?