CS570 - Artificial Intelligence Project 3b

Chihsiang Wang

# fuzzy logic

## By Chihsiang Wang

**Abstract**

This project is going to use the fuzzy logic algorithm to control a moon lander to land at the right position with a safe speed. For getting the right outputs there have four steps to implement the main idea with fuzzy logic:

1. Get the input/percepts, in this project the input will be height / Y-velocity / X-Position.

2. Fuzzify the input, and get the fuzzy set values (0~1) to determine how high / fast / far.

3. Apply the fuzzy value to activate fuzzy rule.

4. Defuzzification to determine actual actions, in this project picks the most active rule to decide what is the actual action, and gets the output.

The output burn and thrust will control the lander’s speed slower than 4.0, and keep the position between -2.0 ~ 2.0 to land safely. In this project the moon lander has about 70% chance to land successful and about 30% chance to be failed.

**Fuzzy Set**

To fuzzify the input will need a function to determine how the scale it is. In this project the fuzzy value are all between 0 ~ 1, it will be more clearly to measure how high / fast / far of each input. The figure (1) shows the height scale, and the pseudo code is list below:

***Void Get\_fuzzy\_value\_height (height) {***

***If (input < 20) fuzzy\_value\_low = 1;***

***If (input > 40) fuzzy\_value\_low = 0;***

***If (input >= 20 && input <= 40) fuzzy\_value\_low = algebra(input);***

***If (input > 30, input < 70) fuzzy\_value\_med = algebra(input); //different algebra***

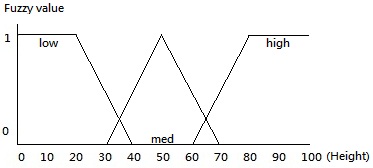
***If (input < 30 OR input > 70) fuzzy\_value\_med = 0;***

***If (input < 60) fuzzy\_value\_high = 0;***

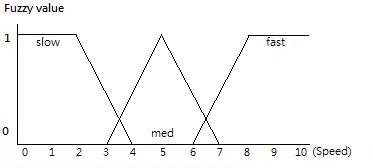
***If (input >80) fuzzy\_value\_high = 1;***

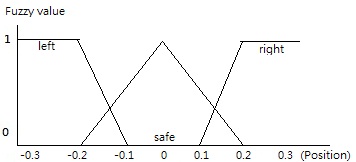
***If(input >=60 AND input <= 80) fuzzy\_value\_high = algebra(input);***

***}***

****(figure 1)

So after execute the function Get\_fuzzy\_height(), the program will store each (low/med/high) fuzzy numbers, and these values are going to active fuzzy rules. With the same idea, the figure 2 and 3 are showing how the scale of Speed and Position. In this project, the output BURN needs to calculate by the input height and speed, and the THRUST needs height and position. (NOTE: this setting is a very draft, later it will compare with a model, which has more specifically details)

****(figure 2)

****(figure 3)

**Fuzzy Rules**

The program can active fuzzy rules after got those fuzzy values, there have totally six different fuzzy values: height\_high, height\_med, height\_low, speed\_slow, speed\_med, speed\_fast, it can create various different sets. In this project uses these set lists below:

For BURN:

IF low height AND fast speed THEN MAX\_BURN;

IF low height AND med speed THEN MED\_BURN;

IF low height AND slow speed THEN MED\_BURN;

IF med height AND fast speed THEN MEX\_BURN;

IF med height AND med speed THEN MED\_BURN;

IF med height AND slow speed THEN MED\_BURN;

IF high height AND fast speed THEN MED\_BURN;

IF high height AND med speed THEN MED\_BURN;

IF high height AND slow speed THEN NO\_BURN;

For THRUST:

IF low height AND left THEN RIGHT\_ THRUST;

IF low height AND left THEN RIGHT\_ THRUST;

IF low height AND left THEN RIGHT\_ THRUST;

IF med height AND med THEN NO\_THRUST;

IF med height AND med THEN NO\_ THRUST;

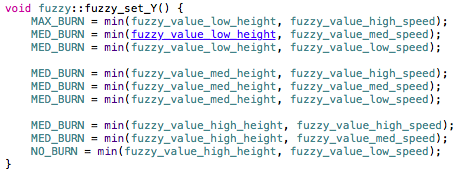
IF med height AND med THEN NO\_ THRUST;

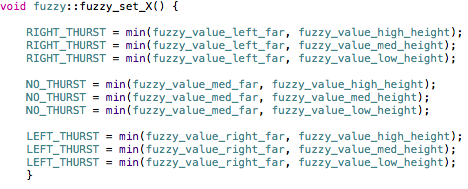
IF high height AND right THEN LEFT\_ THRUST;

IF high height AND right THEN LEFT\_ THRUST;

IF high height AND right THEN LEFT\_ THRUST;

Notice that there have no “else if “statement in the fuzzy set, because every statement in the fuzzy set should be active (if the argument is true). The method to active the fuzzy set is giving the fuzzy value to each actions (MAX\_BURN, NO\_THRUST…etc), in the fuzzy logic usually use max(value1, value2) to instead of OR, min(value1, value2) for AND to decide each actions’ weights. Using an example to describe the fuzzy set will be:





**Defuzzification**

After gives each actions a fuzzy value, there have several method to defuzzify these rules. In this program picks the most active rule to be the output. An example with burn, in the MAX\_BURN, MED\_BURN, NO\_BURN both has the weights (fuzzy value), in case the program will pick the highest weight set to execute, the pseudo code lists below:

***If MAX\_BURN > MED\_BURN and NO\_BURN then BURN = 3.0***

***If MED\_BURN > MAX\_BURN and NO\_BURN then BURN = 1.5***

***If NO\_BURN > MED\_BURN and MAX\_BURN then BURN = 0.0***

To be more specifically, assume that when the moon lander’s height is 10, the fuzzy\_value\_low is 1, fuzzy\_value\_med is 0, fuzzy\_value\_high is 0, to compare with the fuzzy value can active the fuzzy rule has weight as MAX\_BURN = 1, MED\_BURN = 0, NO\_BURN = 0, the last step will pick the most weight action MAX\_BURN, and give the output as 3.0 to control the lander every time to update the height.

**Result**

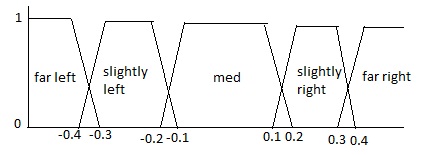
The program has been tested for 10000 times in each different setting, the sample output lists below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **(Success / Crash)** | Only Thrust  Acceleration 1 | Only Burn  Acceleration 1 | Thrust + Burn  Acceleration 1 | Thrust + Burn  Acceleration 2 | Thrust + Burn  Acceleration 3 | Extra values and sets |
| First test | 3933 / 6067 | 10000 / 0 | 5810 / 4190 | 867 / 9133 | 0 / 10000 | 7812 / 2188 |
| Second test | 4897 / 5103 | 9167 / 833 | 5619 / 4381 | 412 / 9588 | 0 / 10000 | 8194 / 1806 |
| % of success | 44% | 95% | 57% | 6% | 0% | 78% |

**Discussion**

The result of the first setting of fuzzy system was not really good to control the lander, when the acceleration is 1, it still have 67% chance to be successful, but when the acceleration gained to 2 and 3, it’s really rarely to land safely. For figuring out what is the reason to cause the bad result, I tried to test the burn and thrust individually, and I have noticed that my fuzzy sets for the burn is really good, it has 95% chance to land successfully, but for the thrust part, it only has 44%; I guessed what cause this situation is that I didn’t do very specifically for the fuzzy value (only left, med and right), and there only have three kinds of values as output are -0.2, 0, 0.2, it looks really not enough to control the location of the lander.

For a better result, I modified the thrust’s fuzzy set. This time I defined five different sections (figure 4): far left, slightly left, med, slight right, far right, with the five fuzzy sets: MAX\_RIGHT = 3.0, RIGHT = 1.5, NO\_BURN = 0, LEFT = -1.5, MAX\_LEFT = -3.0.



With this setting the average increased to 78% with acceleration 1, though it’s not a perfect system yet, but the incensement is obviously. But I also noticed that some cases of crashed because of out of fuel, I believe that for a perfect system, it will also need to be considered some extra outer effects as wind, fuel and acceleration as my fuzzy values and fuzzy sets.

**Conclusion**

There have a bunch of difference between the ANN and Fuzzy Logic. In the ANN, the A.I. agent has been told “how good of the decision it does, but not what decision is right”, and in the Fuzzy Logic has been told “what is the right decision”, in the figure shows more details about the difference between these two algorithm

|  |  |  |
| --- | --- | --- |
|  | ANN | Fuzzy Logic |
| Difficulty of implement | After set all settings done, will be easy to execute. | Need to give bunch of settings, before the program runs, need to give details that what actions need to do in different specifically situations. |
| Understandable | Hard to understand what is the code working for | Easy to understand the coding part, just IF/THEN statement |
| Environments changed | Just need to changed the inputs, because it learns how to reduce the error occur, can find the result easily | Need to change bunch of setting, include every inputs, fuzzy values, and fuzzy rules. |
| Time spend | In this project is longer then fuzzy logic algorithm. | Fast, but not sure with huge fuzzy sets. |

To conclusive, the fuzzy logic algorithm is really easy to understand and implement, but it requires with a stable environment, or it will always needs much hand works to adjust the system to get the best answer. In fuzzy logic, it’s more likely to be a “human given” artificial intelligence, because the programmer will decide all the logic to be a look up table, and the agent will pick the best choice as output. Both ANN and fuzzy logic can find the best answer if the agent has enough information, ANN is more repeatable but hard to set up, and fuzzy logic is easy to set up but needs lots of hand work and easier to get outer effect.