CS570 – Artificial Intelligence Project 3b

Chihsiang Wang

FUZZY LOGIC

By Chihsiang Wang

**Abstract**

This project uses the fuzzy logic algorithm to safely land a moon lander at the targeted location. In order to attain the correct outputs, there are four steps to implement the landing with fuzzy logic.

1. Obtain the inputs/percepts. For this project, the inputs will be height, velocity (Y), and position (X). Fuzzify the inputs and obtain the fuzzy set values (0~1) to determine how high, fast, and far.
2. Apply the fuzzy value to activate fuzzy rule.
3. Use defuzzification to determine the actual actions.This project picks the most active rule to decide what the actual action is and recieves the output.

The output burn and thrust will decrese the lander’s speed slower to less than 4.0. It will keep the position between -2.0~+2.0 in order to land safely. The results of this project indicate the moon lander has approximately a 70% chance of landing and approximately a 30% chance of failure.

**Fuzzy Set**

To fuzzify the data, the input will need a function in order to determine what the scale is. For this project the fuzzy values are between 0 ~ 1This will help to clearly measure how high, fast, and far for each input. The figure showing the height scale and the pseudo code is listed below:

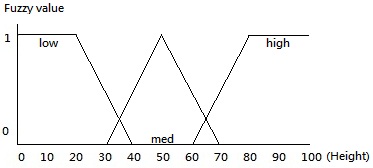
****

Figure 1

***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);***

***}***

After the execution of the function Get\_fuzzy\_height(), the program will store each low, medium, and high fuzzy numbers. These values will activate the fuzzy rules. Figure 2 and 3 are scales of speed and position of the moon lander. The output BURN needs to be calculated by the input height and speed, while the THRUST needs height and position. (NOTE: this setting is just a rough draft. It will later be compared with a model with more specific details.

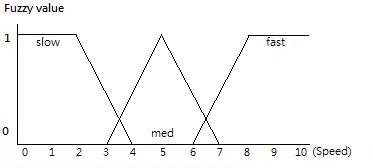
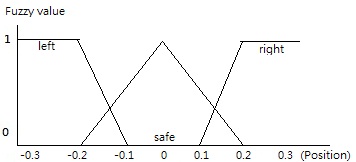
 

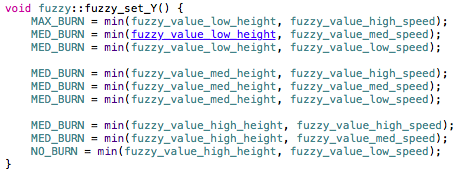
Figure 2. Figure 3.

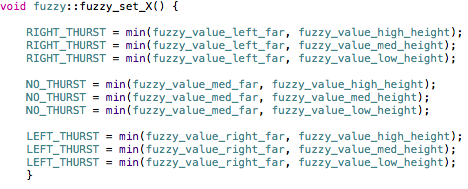
**Fuzzy Rules**

The program can activate the fuzzy rules after receiving the fuzzy values. There are a total of six different fuzzy values: height\_high, height\_med, height\_low, speed\_slow, speed\_med, speed\_fast. From the six fuzzy values, various value sets can be established, which are listed 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 is no “else if” statement in the fuzzy set. This is because every statement in the fuzzy set should be active (if the argument is true). To activate the fuzzy set, each fuzzy value must be given an action (e.g. MAX\_BURN, NO\_THRUST, etc). Also, the fuzzy logic usually uses max(value1, value2) instead of OR and min(value1, value2) for AND in order to decide each actions’ weights. An example of this fuzzy set is listed below:





**Defuzzification**

After each action provides a fuzzy value, there are several ways to defuzzify them. In this program, it picks the most active rule to be the output. For example, with burn, 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 specific, assuming 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 10000 times in each different setting. The sample output is listed 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 |  |
| Second test | 4897 / 5103 | 9167 / 833 | 5619 / 4381 | 412 / 9588 | 0 / 10000 |  |
| Percentage | 44% | 95% | 57% | 6% | 0% |  |

**Discussion**

The result of the fuzzy system with the first setting did not control the lander very well. When the acceleration was 1, it had a 67% chance of landing; but when the acceleration was 2 or 3, it rarely landed safely. In order to figure out why I was getting bad results, I tried to test the burn and thrust individually. What I found was that my fuzzy settings for the burn was really good. There was a 95% chance of landing successfully. The thrust, however, only had a 44% chance. I believe what caused this situation was that I didn’t input very specific fuzzy values (only left, med and right). As a result, there were only three kinds of values as outputs: -2.0, 0, and +2.0; which seems not to be enough to control the lander.

**Conclusion**

There are some 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”The table below 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 conclude, the fuzzy logic algorithm is really easy to understand and to implement. It, however, requires a stable environment, or it will always need adjusting in order to obtain the best results. With fuzzy logic, it is more of a “human given” artificial intelligence, because the programmer decides 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 easier to replicate but hard to set up, while fuzzy logic is easy to set up but needs a lot of adjustments to be effective.