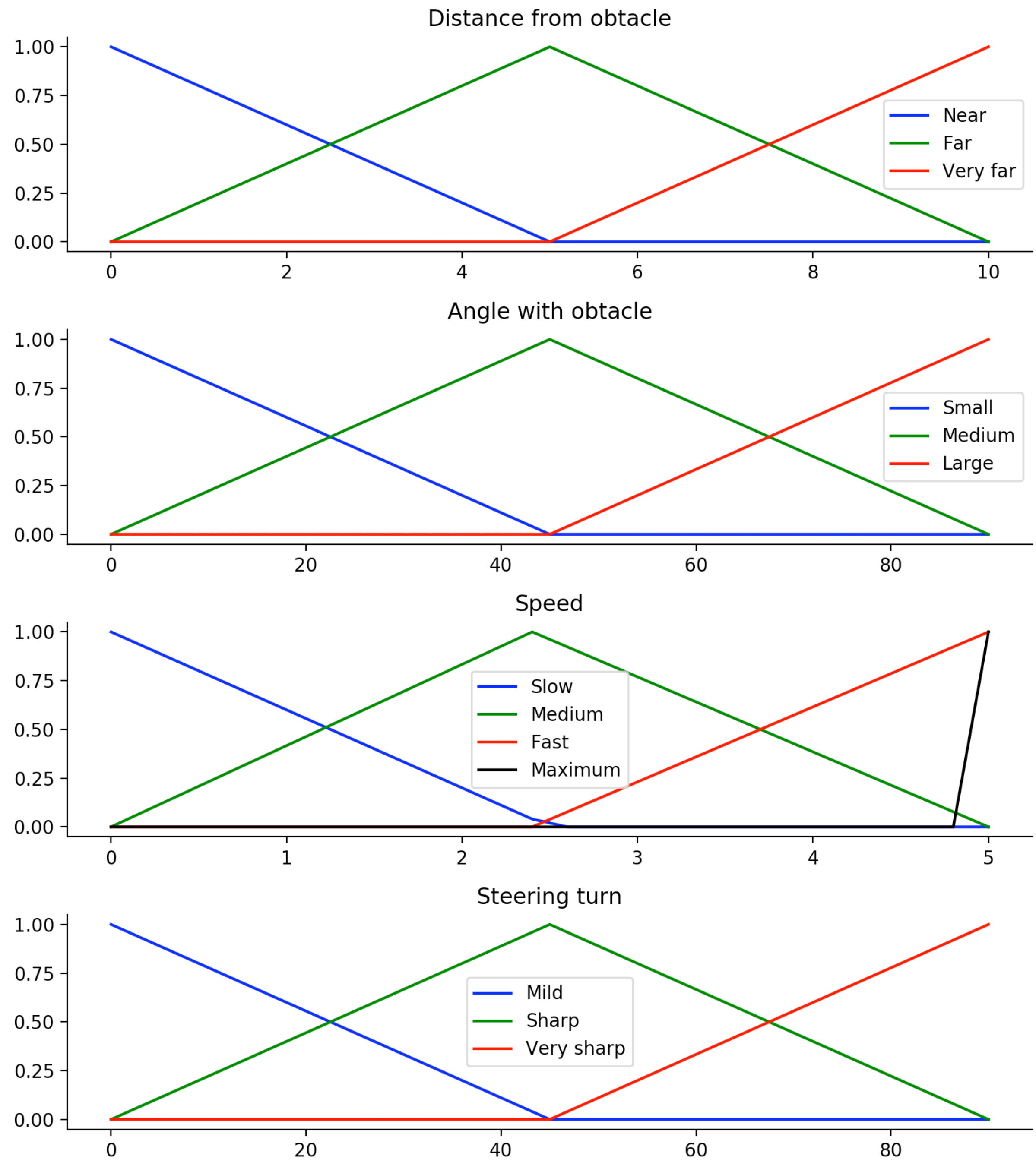
Problem 4

Design Steps:

a. Membership functions for each of the fuzzy quantities



b. Rule base
(Rule 1 to Rule 9 is for steering turn output, while Rule 10 to Rule 12 is for speed output)

Rule 1: if D is N, and A is S, then ST is VST
Rule 2: else if D is N, and A is M then ST is SST
Rule 3: else if D is N, and A is L then ST is MT

Rule 4: else if D is F, and A is S, then ST is SST
Rule 5: else if D is F, and A is M then ST is SST
Rule 6: else if D is F, and A is L then ST is MT

Rule 7: else if D is VF, and A is S, then ST is MT
Rule 8: else if D is VF, and A is M then ST is MT
Rule 9: else if D is VF, and A is L then ST is MT

Rule 10: if ST is MT, then S is FS
Rule 11: else if ST is SST, then S is MS
Rule 12: else if ST is VST, then S is SS

c. Inferencing system that will be used

For aggregation part, I use Mamdani inferencing system's aggregation part. Mamdani inferencing system uses the max-min operator for aggregation part. For the reason why I choose Mamdani is that, it is easy to implement the aggregation part; furthermore, for a typical fuzzy rule (If x is A and y is then z = f(x,y)) in a Sugeno fuzzy model has the form: Where A and B are fuzzy sets in the antecedent, while z= f(x, y) is a crisp. It is very convenient for me to use the fuzzy set as the consequence, instead of finding a crisp representation for the consequence.

For the defuzzification part, I use MOM (mean of maximum) to get the steering turn output and use MOM (mean of maximum) to get the speed output.

I give reason why I choose this defuzzification method by giving different inputs comparing the results with using centroid.

When I use centroid method for steering output defuzzification, centroid for speed output defuzzification. I give input (distance is 0.5, and angle with the obstacle is 1) and get the outputs are:

```
steering_turn output:  66.44386802241131
speed output:  2.2169746099681853
```

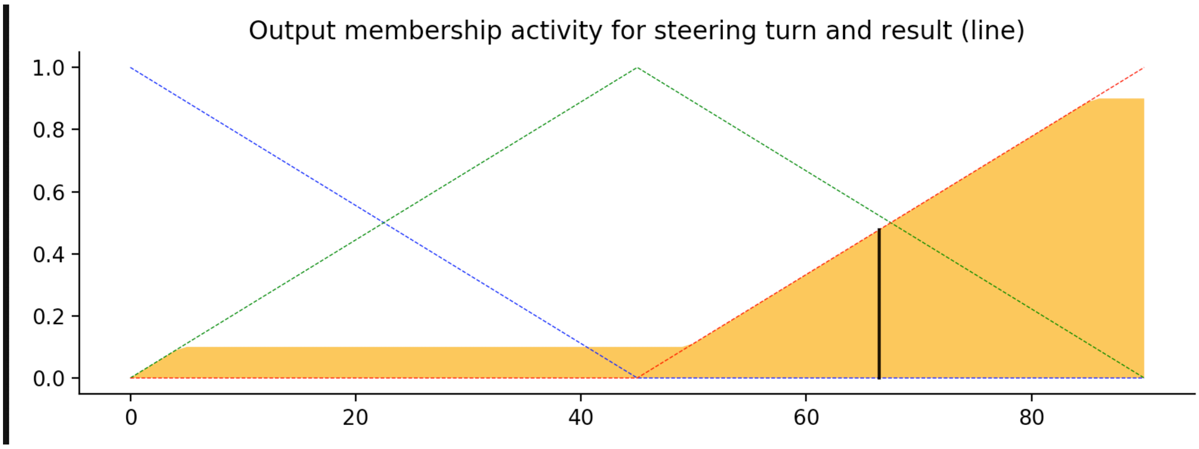


Figure 1

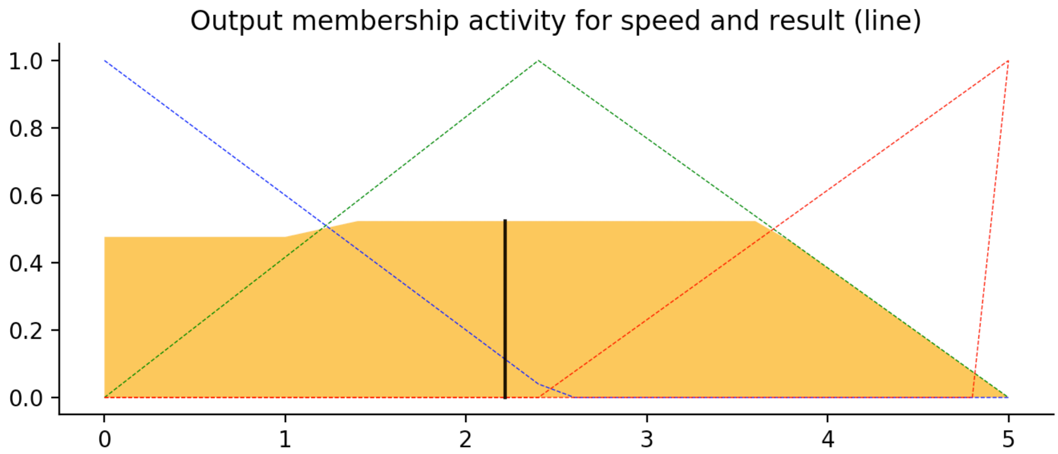


Figure 2

From this output membership function for steering turn, we can see that if we use MOM will get a more reasonable result for this input.

When I use MOM method for steering output defuzzification, centroid for speed membership function defuzzification, the outputs are:

```
steering_turn output:  88.0
speed output:  1.064983054553405
```

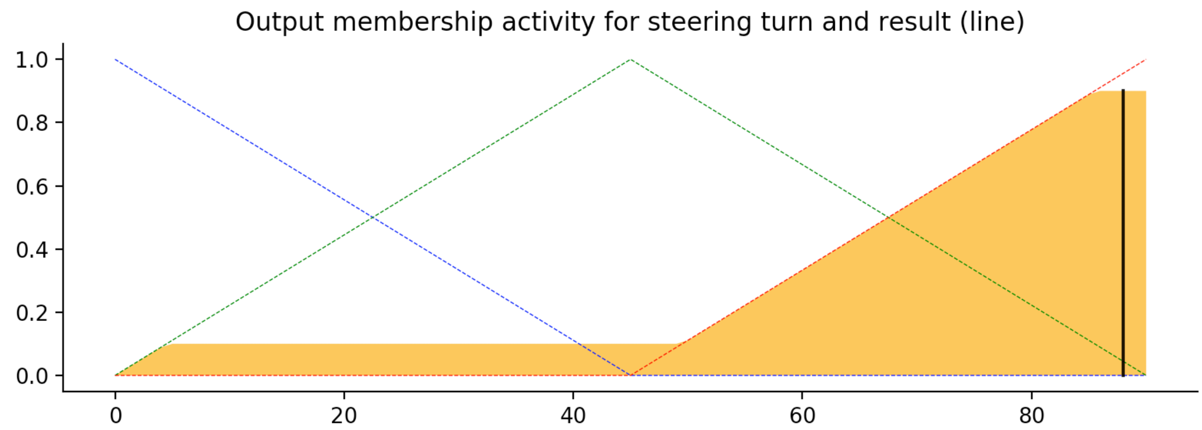


Figure 3

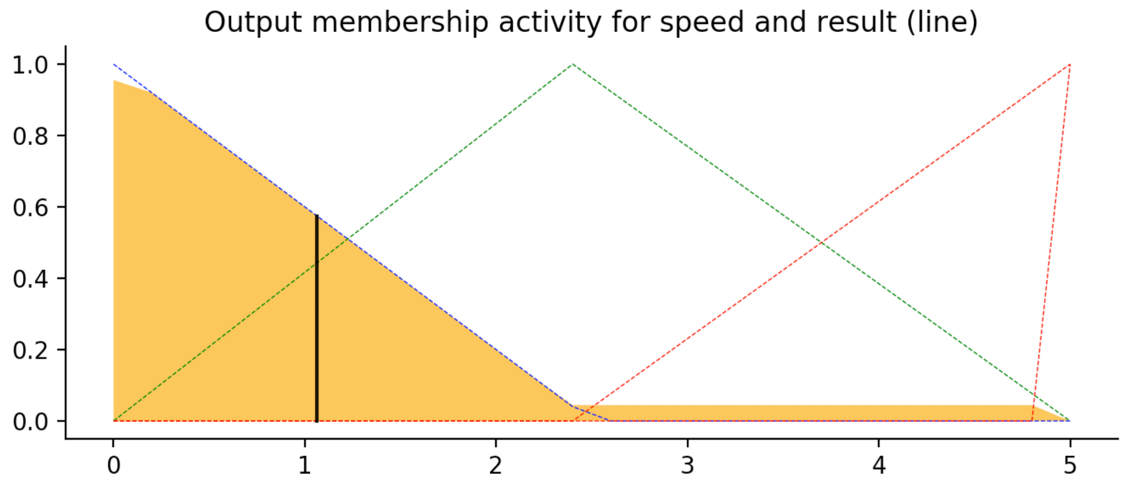


Figure 4

We can see that when we use MOM for the steering turn output defuzzification, the outputs for steering turn and speed are more reasonable; however, from the output membership function for speed, we can see that if we use centroid method for defuzzification, it is not that reasonable.

Then I change the inputs to explore the defuzzification for speed membership function.

When I use MOM method for steering output defuzzification, centroid for speed output defuzzification. I give input (distance is 9.5, and angle with the obstacle is 89) and get the outputs are:

```
steering_turn output:  2.0
speed output:  3.915266772319071
```

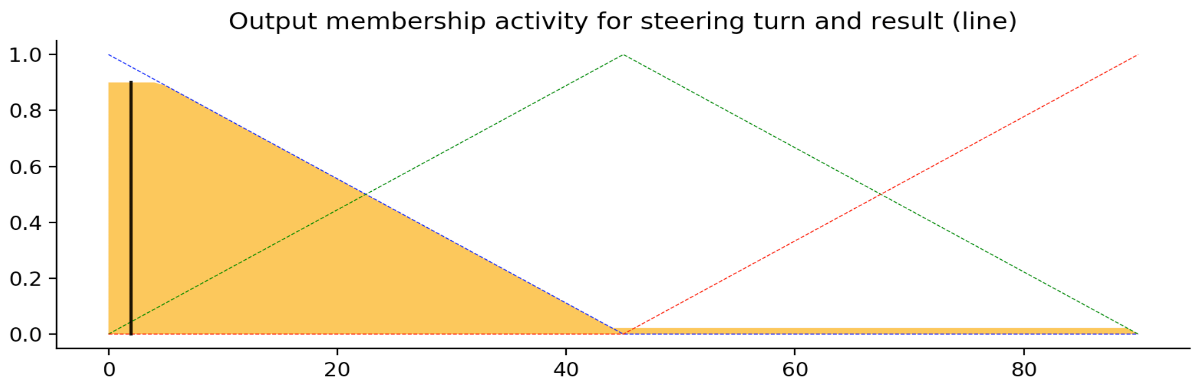


Figure 5

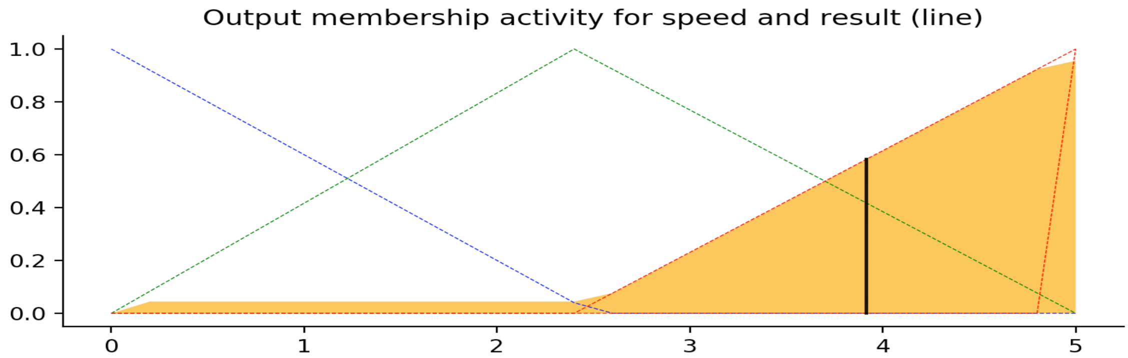


Figure 6

Then, When I use MOM method for steering output defuzzification, MOM for speed output defuzzification. I give input (distance is 9.5, and angle with the obstacle is 89) and get the outputs are:

```
steering_turn output:  2.0
speed output:  5.0
```

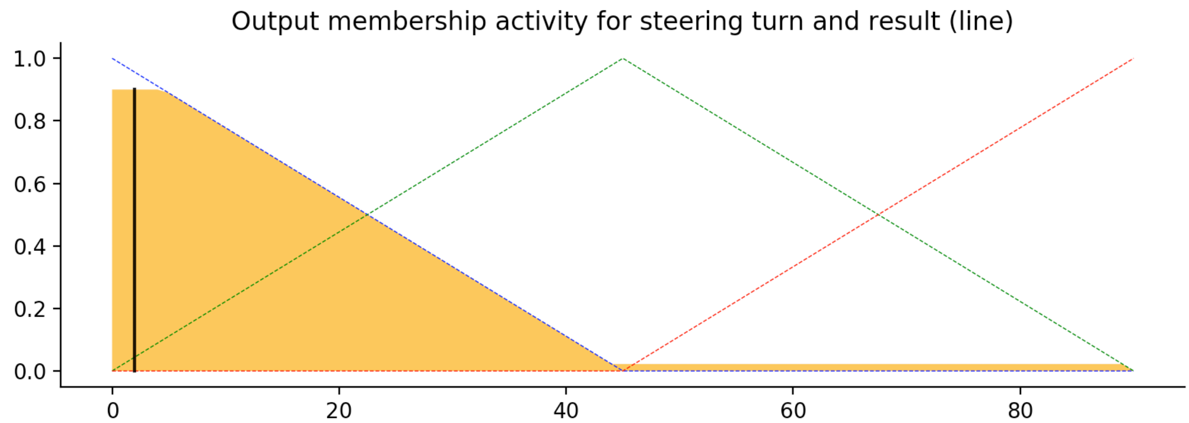


Figure 7

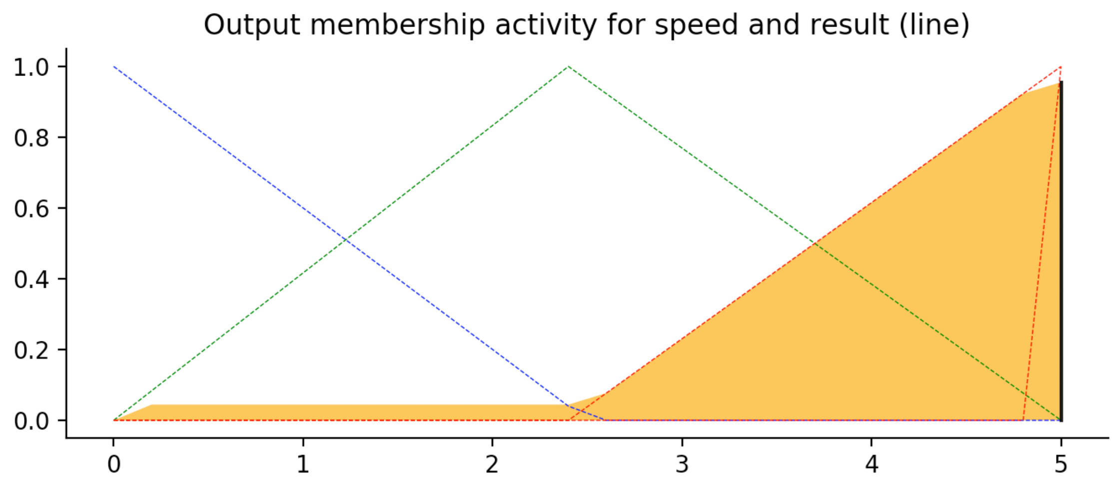


Figure 8

We can see that when we change the defuzzification method for speed output membership activity to MOM method, the output result is much more reasonable compared with centroid method.
Then compared with figure 3 and figure 5, we also can see that when I choose MOM, it will achieve a better and stable performance output compared with LOM and SOM method.

After these discussions based on different inputs, I choose MOM method for both steering turn and speed output defuzzification.

I will give three input examples and show the system output control command

1. Distance with obstacle is 2, and angle with the obstacle is 80

   ```
   steering_turn output:  9.0
   speed output:  4.800000000000001
   ```

   We can see in this situation, the robot is near from obstacle and the angle is large. Because the angle with the obstacle is large, the robot has no chance to have a collision with the obstacle. It is the same with that no obstacle in front of it. So it will increase it speed with mild steering turn.
2. Distance with obstacle is 5, and angle with the obstacle is 50

   ```
   steering_turn output:  45.0
   speed output:  2.4000000000000004
   ```

   We can see in this situation, the robot is relevant far away from obstacle and the angle is medium. If the robot does no action, it could cause small collision. So the output command is steering turn is sharp(more turn compared with mild), and the speed is medium (decrease speed relevant to fast speed) to help the robot to avoid the small collision smoothly.
3. Distance with obstacle is 8, and angle with the obstacle is 20

   ```
   steering_turn output:  10.0
   speed output:  4.800000000000001
   ```

We can see in this situation, the robot is very far away from obstacle. Although the angle is small, the robot has no chance to have a collision with the obstacle in short time; therefore, the robot will be increasing its speed with mild steering turn.

I will give an inference process for the third example. (we can see generate the same output as the output of my code)