Problem 3

Genetic programming refers to creating entire software programs (usually in the form of Lisp source code); and genetic algorithms refer to creating shorter pieces of code (represented as strings called chromosomes)

1. The structure is the main difference. A genetic algorithm is represented as a chromosome which is a binary string of 0 and 1 with fixed length, while a genetic programming is represented as a variable length parse tree structure of actions and values. The tree-based representation makes GP more flexible, but increase the complexity and is not efficient as well

2. From the implementation steps of these two algorithms, we can see: For GA, an offsping could be generated by crossover and then mutation, but GP the offspring could only be generated by using the operator selected from either crossover,reproduction or mutation

3. Genetic algorithm often produces invalid states, and Genetic program rarely produces invalid states.

4. For the applications, GA is used for the task of optimizing parameters for solutions when their structure is known, while GP is more often used to learn and discover both the contents and structures of solutions.

Problem 4

1. Design the fuzzy system

(1) Membership functions

According to the corresponding states of the distance from obstacle(D), angle wiith obstacle(A), speed(A), steering turn(ST), For the membership functions, I use the combination of triangular and trapezoid. The tables and figures below show how the membership functions are defined and generated:

D	State
0~5	Near
2.5~7.5	Far
5~10	Very Far

Table 1 Distance states

$$\mu N(x)\begin{cases} 1, 0\le x\le 2.5 \\ -0.4x+2, 2.5\le x\le 5 \end{cases} \quad \mu F(x)\begin{cases} 0.4x-1, 2.5\le x\le 5 \\ -0.4x+3, 5\le x\le 7.5 \end{cases} \quad \mu VF(x)\begin{cases} 0.4x-2, 5\le x\le 7.5 \\ 1, 7.5\le x\le 10 \end{cases}$$

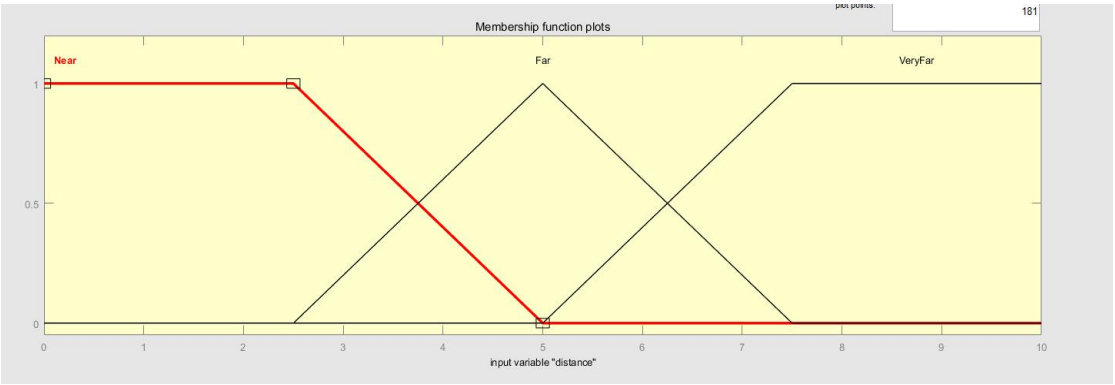


Figure 1 Distance Membership Function

A	State
0~45	Small
20~70	Medium
45~90	Large

Table 2 Angle states

$$\mu S(x)\begin{cases} 1, 0\le x\le 20 \\ -0.04x+1.8, 20\le x\le 45 \end{cases} \quad \mu M(x)\begin{cases} 0.04x-0.8, 20\le x\le 45 \\ -0.04x+2.8, 45\le x\le 70 \end{cases} \quad \mu L(x)\begin{cases} 0.04x-2.8, 45\le x\le 70 \\ 1, 70\le x\le 90 \end{cases}$$

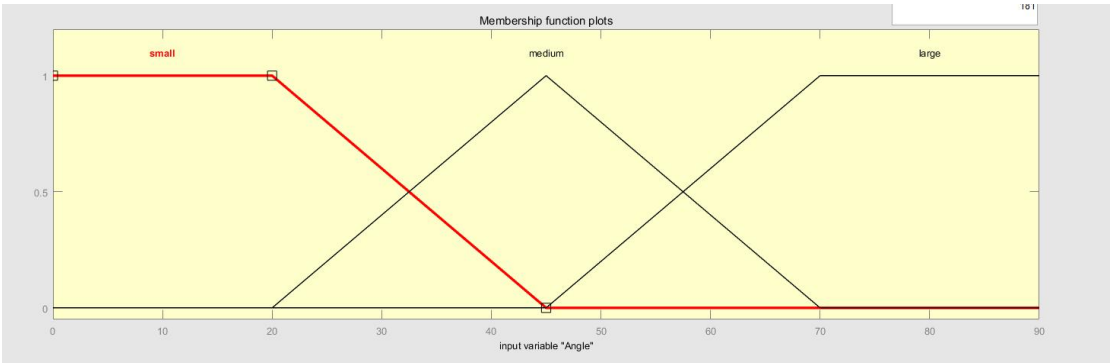


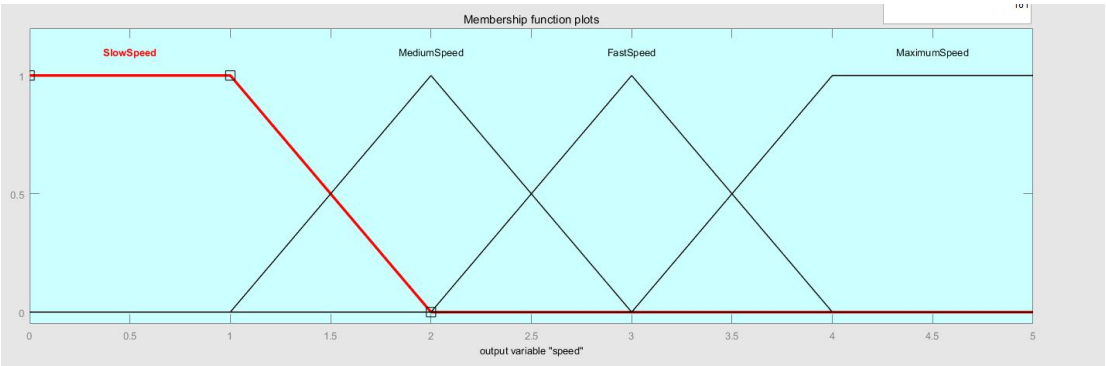
Figure 2 Angle Membership Function

S	State
0~2	Slow Speed
1~3	Medium Speed
2~4	Fast Speed
3~5	Maximum Speed

Table 3 Speed states

$$\mu SS(x)\begin{cases}1,0\le x\le 1\\ -x+1,1\le x\le 2\end{cases}\quad \mu MS(x)\begin{cases}x-1,1\le x\le 2\\ -x+3,2\le x\le 3\end{cases}\quad \mu FS(x)\begin{cases}x-2,2\le x\le 3\\ -x+4,3\le x\le 4\end{cases}\quad \mu MXS(x)\begin{cases}x-3,3\le x\le 4\\ 1,4\le x\le 5\end{cases}$$

Figure 3 Speed Membership Function



ST	State
0~45	Mild Turn
20~70	Sharp Turn
45~90	Very Sharp Turn

Table 4 Steering turn state

$$\mu MST(x)\begin{cases}1,0\le x\le 20\\ -0.04x+1.8,20\le x\le 45\end{cases}\quad \mu SST(x)\begin{cases}0.04x-0.8,20\le x\le 45\\ -0.04x+2.8,45\le x\le 70\end{cases}\quad \mu VST(x)\begin{cases}0.04x-2.8,45\le x\le 70\\ 1,70\le x\le 90\end{cases}$$

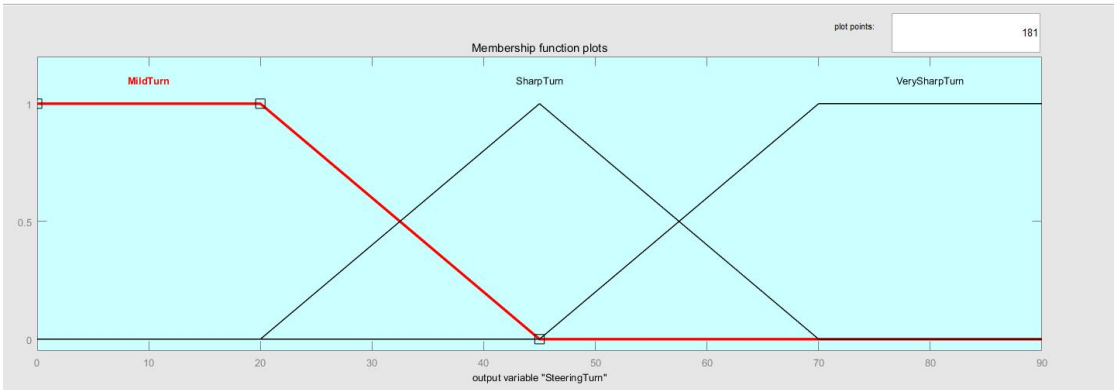


Figure 4 Steering turn membership function

(2) Rules

The rules required are:

(a) When an obstacle is detected, avoid it: does not matter which direction; left or right

(b)When there is no obstacle move forward

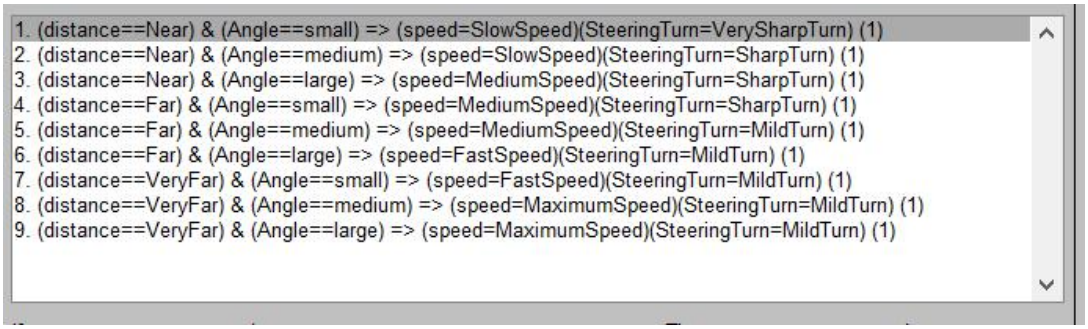
(c)Speed is reduced when turning away from an obstacle

(d)Speed is increased when cruising (when there is no obstacle)

Here we match these rules with the inputs(distance and angle from the obstacle) and outputs(speed and steering turn). For example, for rule(a), if the distance is near and angle is small, we can regard it as obstacle is detected, then we need to avoid it by make steering turn, since rule(c) also requires speed should be reduced when turning away, the speed should be slow; for rule(b), if the distance is very far, we can regard it as there is no obstacle detected, then we should move forward with mild turn, according to rule(d), we should also increase the speed.

Since there are three different states of input “distance” and three different states of input “angle”, we can define 9 rules according to the inputs and our requirements of outputs for our fuzzy control system.

The rules we made are shown as follows:



(3) Inferencing system

There are two main types of fuzzy inference systems:Mamdani and Sugeno FIS. Here I tried both to design the system.

(a) Mamdani inferencing system

The inference process of a Mamdani system can be described as following steps:

1. Input fuzzy sets. Here the inputs are distance and angle. Which are introduced in the former sections.

2. Apply fuzzy operation, here the operation is AND operator.

3. Apply implication method(min) in each rule.
4. Combine the fuzzy sets into a single fuzzy set by applying aggregation method (max).
5. Compute the final crisp output value by defuzzification.

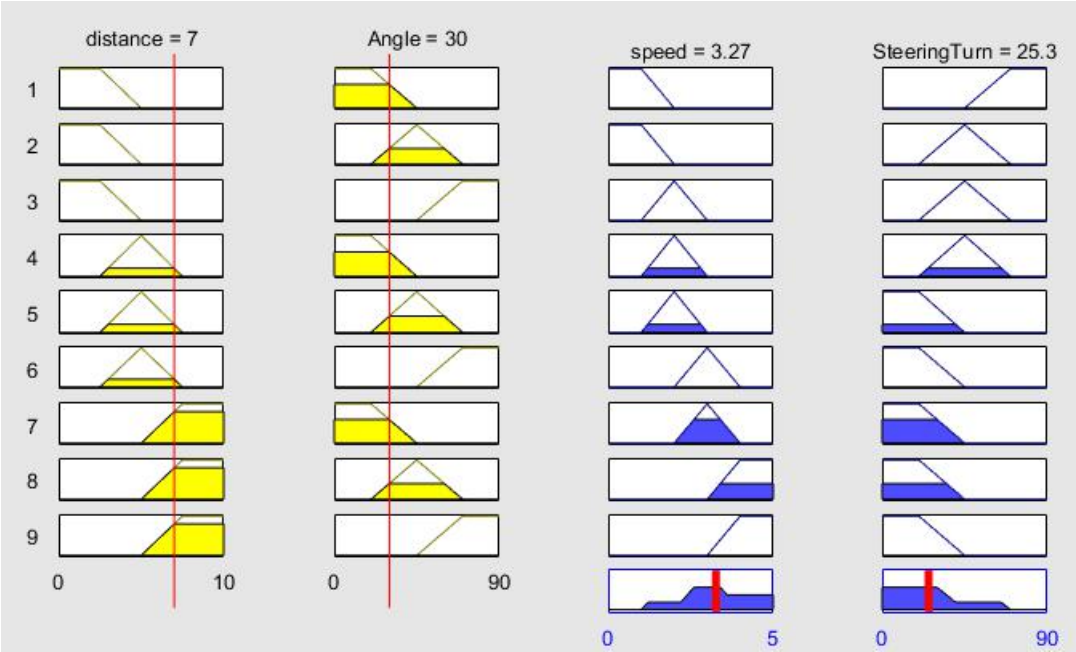


Figure 5 Mamdani inference steps

(b) Sugeno inferencing system

In Sugeno inferencing system, the output member functions are either constant or a linear function of the input values, which is different from the member functions defined in former sections. Since it is not easy to find the linear relation of the inputs values and outputs values, here I set the output memebership function as constants that represents for different output states. The membership functions of outputs are shown as follows:

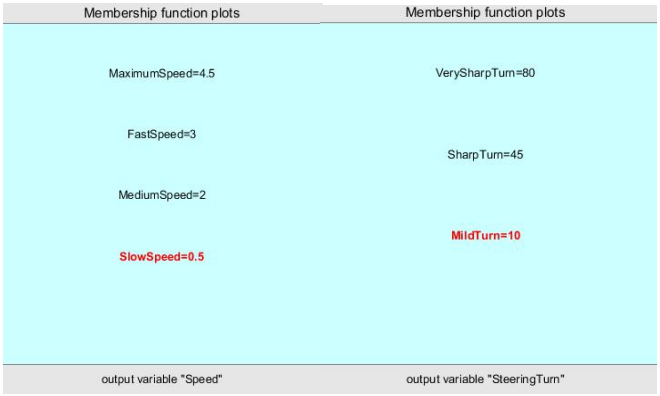


Figure 6 Output memeber functions for Sugeno

The fuzzy inference process for a Sugeno system can be described as following steps:

1. Input fuzzy sets. Here the inputs are distance and angle. Which are introduced in the former sections.

2. Apply fuzzy operation, here the operation is AND operator.

3. Apply implication method(prod) in each rule.The output of each rule is the weighted output level, which is the product

4 Combine the fuzzy sets into a single fuzzy set by applying aggregation method (sum). The final output of the system is the weighted average over all rule outputs:
5 Compute the final crisp output value by defuzzification.

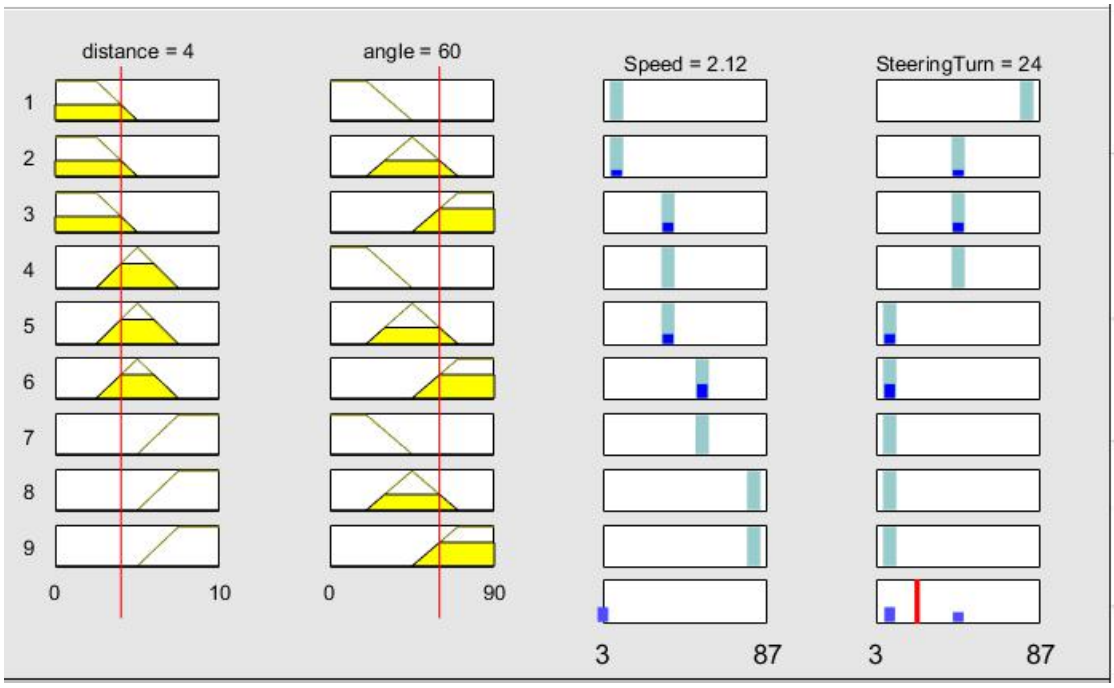


Figure 6 Sugeno inference steps

(4) Defuzzification method

For the defuzzification method, there are many options, such as:

Centroid: Centroid defuzzification returns the center of gravity of the fuzzy set along the x-axis; Bisector: The bisector method finds the vertical line that divides the fuzzy set into two sub-regions of equal area; MOM, SOM, and LOM stand for middle, smallest, and largest of maximum, respectively.

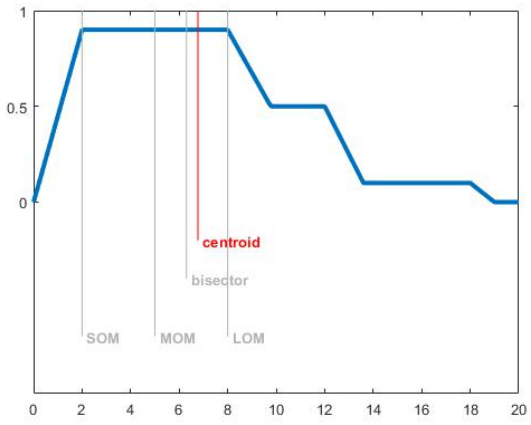


Figure 7 Different defuzzification methods
(cr: https://www.mathworks.com/help/fuzzy/defuzzification-methods.html)

2. Explain the choices

(1) Choice of Inferencing system

I choose the Mamdani inferencing system in this case. As was mentioned before, for Sugeno

inference system, we need to use a constant or a linear function of the input values. In Mamdani system, output membership is present and there is a distribution of output, while in Segeno system, no output memembership funciton is present and there is no distribution of output, only mathematical combination of the output and the rules strength. Aslo, Mamdani has expressive power and interpretable rule consequent, but Segeno is loss of interpretability. In this case, it is no given clear linear function of the input or corresponding constant value, , so it is not easy for the rules design. Generally, Mamdani can be used in both in MISO (Multiple Input and Single Output) and MIMO (Multiple Input and Multiple Output) systems, and Sugeno is better to be used only in MISO systems, Mamdani inference system is well suited to human input, Sugeno inference system is well suited to mathematically analysis. In this case, we are not aiming to mathematically analyze the problem, just to implement a fuzzy control system with given fuzzy quantities, above all, here Mamdani seems to be a better choice.

(2) Choice of Defuzzification

I choose centroid defuzzification method in this system. Apparently, it is not suitable to use LOM or SOM. As we can see in example(a) in the next section [distance,angle]=[3,30], if we use SOM, the output speed will be 0, which is too slow or even impossible to achieve; if we use LOM, the output steering turn will be 90, which is way too sharp and not reasonable as well. Centroid method gives an output of [speed, steering turn]=[1.1,54.8], bisector method gives an output of [1,59.4], while MOM gives an output of [0.675, 75.1]. It seems centroid and bisector gives similar outputs, and MOM gives a smaller speed and larger turn angle however it is not necessary for this input since if distance is 3 and angle is 30, it means the obstacle is not that close, so we do not need to slow down to 0.675 and make a very sharp turn of 75.1. So it seems the results given by centroid or bisector is more reasonable, and we can choose either of them, in general the centroid method is good enough.

3. Example

Figure 8 shows two designs, Mamdani and Sugeno systems, and we are going to introduce two examples implemented by each system.

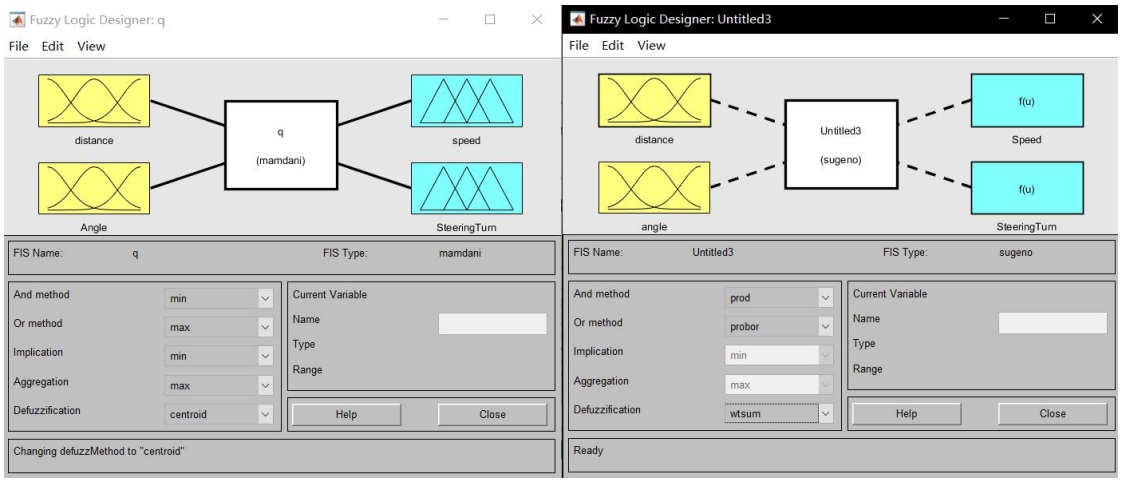


Figure 8 Design of Mamdani and Sugeno systems

(a) The first example is when input is [distance,angle]=[3,30], as we can see in figure 9, when

input distance from obstacle is 3, and angle with obstacle is 30, Mamdani system gives an output control command as: speed=1.1, steering turn=54.5. Sugeno system gives an output control command as speed = 0.8, steering turn =59. We can see if we assume there are obstacle is detected, both systems give a command that we need to slow down and turn away, it seems Sugeno system we designed gives a safer control command, it will make a sharper turn with slower speed.

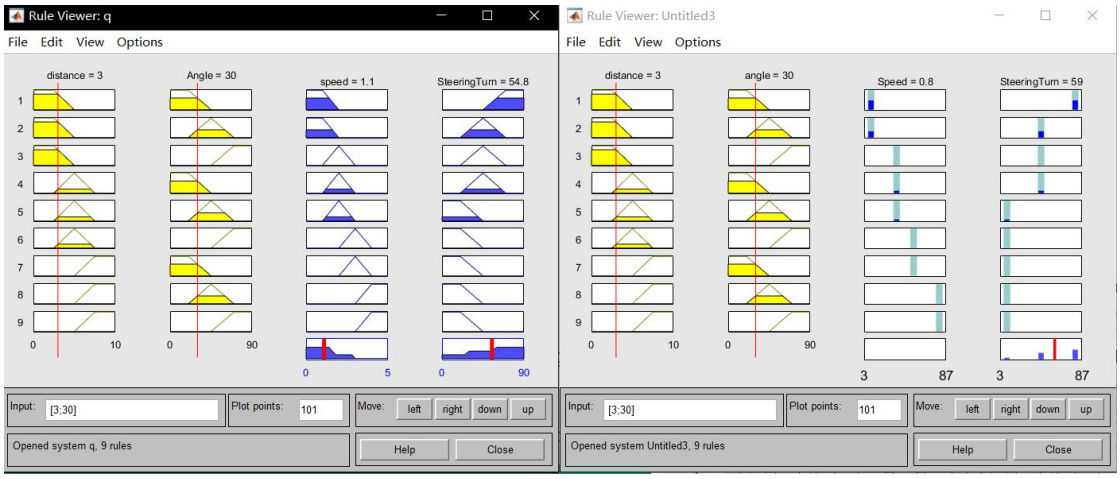


Figure 9 Example (a)

(b) The second example is when input is [6， 60],as we can see in figure 10, when input distance from obstacle is 6, and angle with obstacle is 60, Mamdani system gives an output control command as: speed=2.98, steering turn=18.8. Sugeno system gives an output control command as speed = 3.36, steering turn =10. It shows if we assume there no obstacle detected, i.e. the obstacle is rather far away and not in the detected range, both systems gives command that indicates we do not need to turn away that much and we can move on with faster speed.

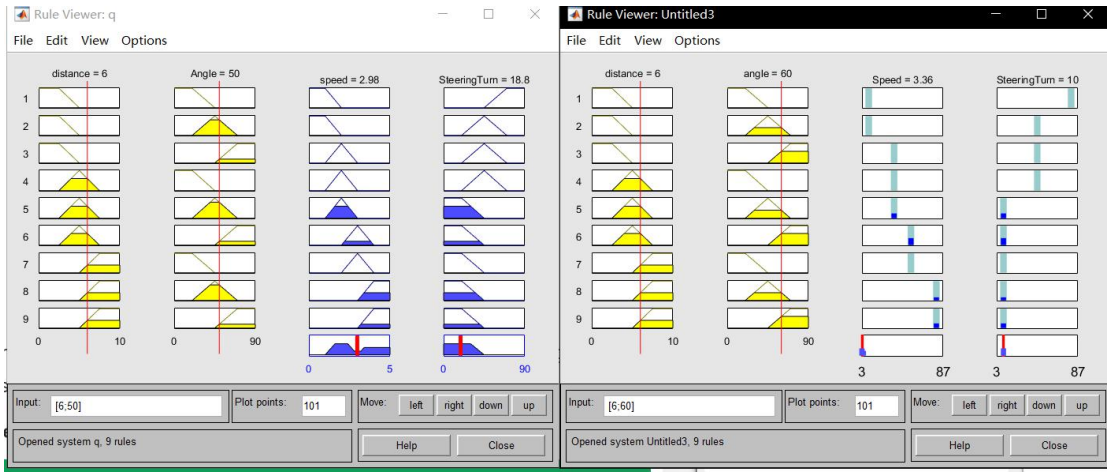


Figure 10 Example (b)