Fuzzy Systems - Car Control (G)

Nikolaos Giakoumoglou 9043

24/08/2020

Abstract

The purpose of this assignment is to design a fuzzy controller in order to avoid obstacles. The vehcile should drive itself safely to the desired position (x_d, y_d). Special sensors provide information about the distances from the obstacles in horizontal (d_H) and vertical position (d_V). The vehicle's velocity is constant and equal to $u = 0.05$ m/sec. Given the vertical and horizontal distances from the obstacles and the angle of the velocity θ, the FLC should decide about the change of the angle $\Delta\theta$ so as for the car to move to the desired position. Script written in `MATLAB 2019a`.

Contents

1 Design of the FLC

1.1 Preparation

First of all we created the function `get_distances.m` to calculate the distances dH,dV, given the coordinates of the car. The distances belong to $[0, 1]$ so any distance grater than 1 is limited to 1.
The next step is to create the FIS. The code of the FIS lies to `fis_model_initial.m` and saves the model to `fis_model_initial.fis`. For the model we took into consideration the following

- Implication Rule $\rightarrow$ Mamdani
- ALSO $\rightarrow$ max
- Composition $\rightarrow$ max-min
- Defuzzifier $\rightarrow$ COA (centroid)

The inputs are the following

- $dH \in [0, 1]$
- $dV \in [0, 1]$
- $\theta \in [-180, 180]$ (referred as theta)

And the output is

- $\Delta\theta \in [-130, 130]$ (referred as Dtheta)

1.2 Membership Functions

The initial configuration of the membership functions is 3 triangle for each input and output as following

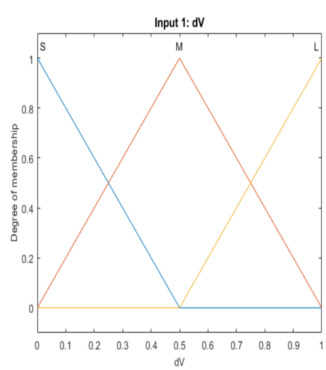

Figure 1: Membership function of dV

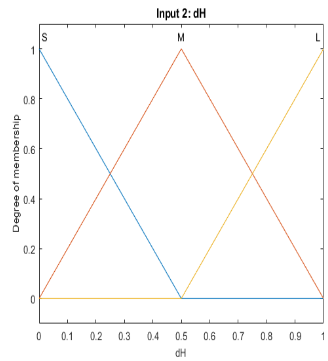

Figure 2: Membership function of dH

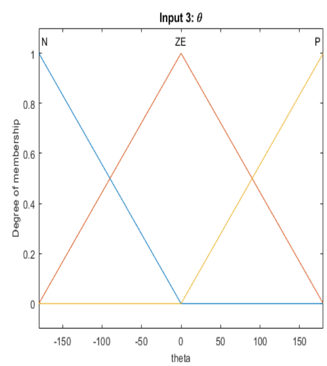

Figure 3: Membership function of θ

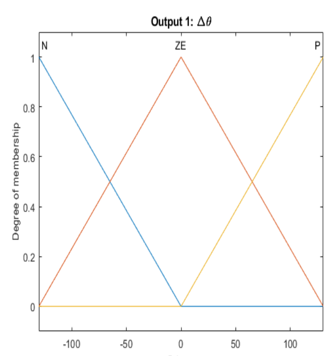

Figure 4: Membership function of $\Delta\theta$

1.3 Rule Base

The rule base is based on the experience and is initialized as following with $3^3 = 27$ rules

dH is S		θ		
		N	**ZE**	**P**
dV	**S**	P	P	P
	M	P	P	N
	L	P	P	N

Table 1: Rule Base where dH is S

dH is M		θ		
		N	**ZE**	**P**
dV	**S**	P	P	N
	M	P	ZE	N
	L	P	ZE	N

Table 2: Rule Base where dH is M

dH is L		θ		
		N	**ZE**	**P**
dV	**S**	P	P	N
	M	P	ZE	N
	L	P	ZE	N

Table 3: Rule Base where dH is L

1.4 Simulation

We are ready to model the movement of the car.
In the functiom `main.m`, we set the initial position $(x_0, y_0) = (9.1, 4.3)$ and the desired $(x_d, y_d) = (15, 7.2)$, the velocity $u = 0.05$ m/s and the starting values of $\theta = \{0, 45, 90\}$. Then we make a call to `simulate.m` which is the code that simulates the car's route in the plain. In that function, we iterate for every θ and while the car is not out of bounds which are set arbitarily to $0 < x < 15$ and $0 < y < 10$. The change of the angle $\Delta\theta$ is determined through the function $evalfis$. The new position is based on the rule

$$x(t) = x(t-1) + u \cdot cos(\theta)$$

$$y(t) = y(t-1) + u \cdot sin(\theta)$$

$$\theta(t) = \theta(t-1) + \Delta\theta$$

The only thing left is to plot the movement of the car vs the obstacles in a single plot with some information about the error in the x and y axe and the euclidean error of the final position to the desired.

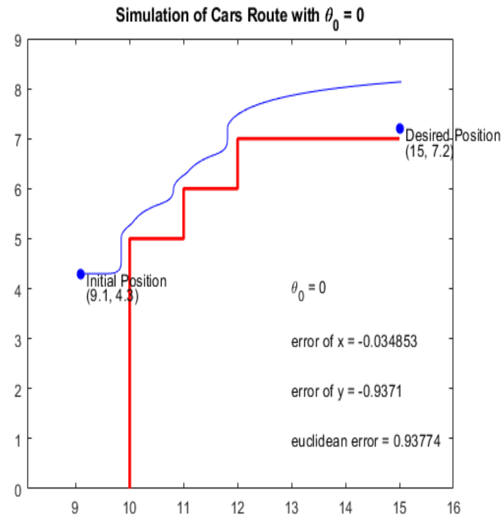

Figure 5: Simulation of car's movement for $\theta = 0$

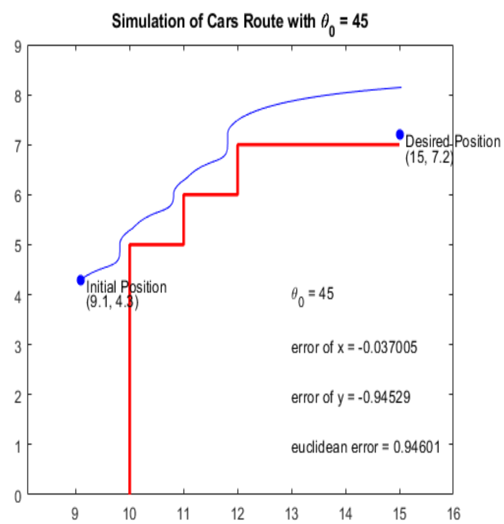

Figure 6: Simulation of car's movement for $\theta = 45$

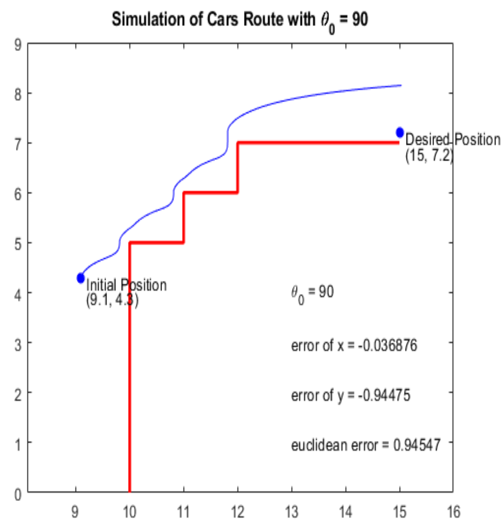

Figure 7: Simulation of car's movement for $\theta = 90$

With the initial configuration, the car fails to reach the desired position since the error is significant. We observe that in case the car reaches a corner, it moves up and then to the right to avoid hitting to the obstacle. This is the result of the Rule Base we have set. However, the car fails to turn soon and moves away from the desired position. Therefore further improvement is mandatory.

2 Improved Design of the FLC

In order to improve the accuracy of final position, we changed the universe of discourge of $\Delta\theta$ to $[-150, 150]$. We also slightly changed the membership functions of dH, dV and θ. However, the most significant impact to the car's accuracy was the change of $\Delta\theta$'s membership function where we switched to trapezoidal. After trial and error we found the values where the euclidean error seems to have a minimal.

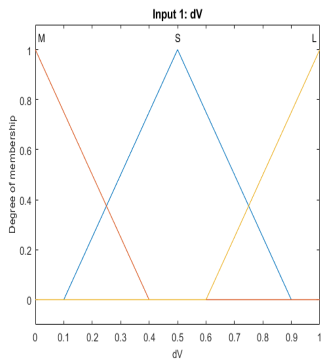

Figure 8: Membership function of dV

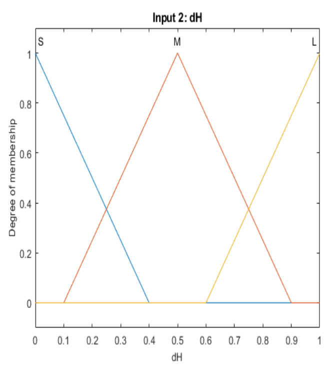

Figure 9: Membership function of dH

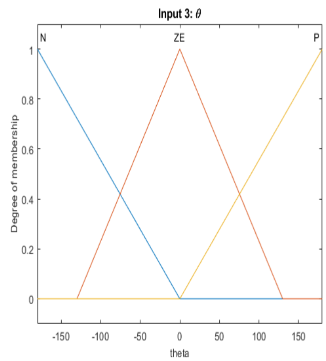

Figure 10: Membership function of θ

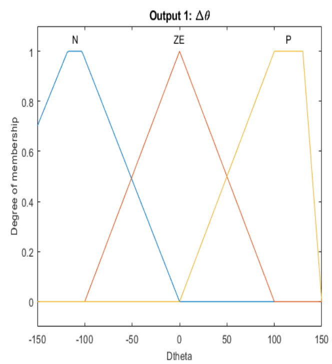

Figure 11: Membership function of $\Delta\theta$

The changes can be found in `fis_model_improved.m`. which are saved to `fis_model_improved.fis`. Simulating the new FIS model we get the following figures:

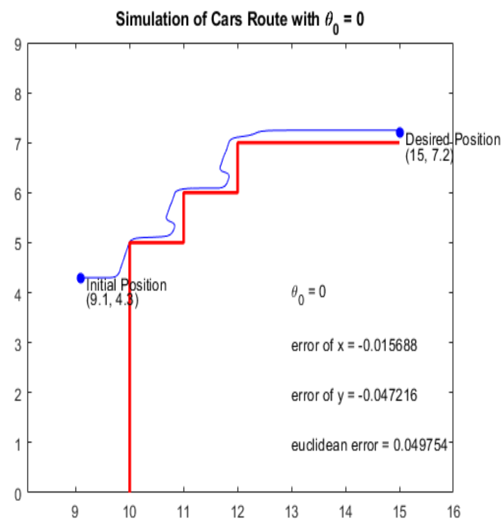

Figure 12: Simulation of car's movement for $\theta = 0$

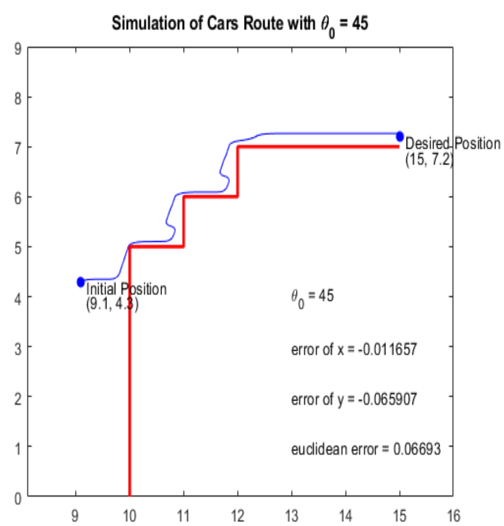

Figure 13: Simulation of car's movement for $\theta = 45$

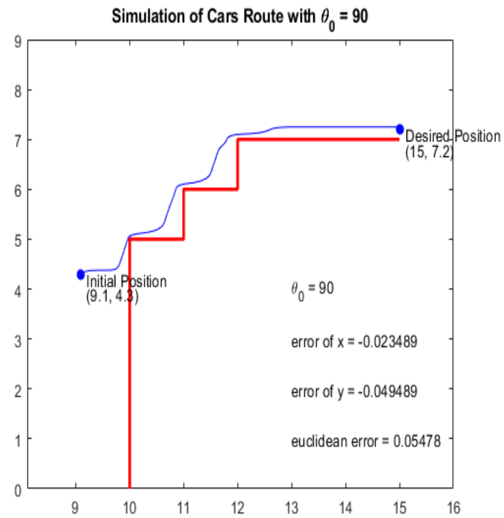

Figure 14: Simulation of car's movement for $\theta = 90$

The improvement is obvious. When the car moves horizontically, it should stay on its way unless it finds a wall ahead. In the latter case it starts drifting away in order to turn smoothly and keep going straight ahead afterwards.

3 Comparison: Initial VS Improved

To clarify the improvement, the combined plots are shown below

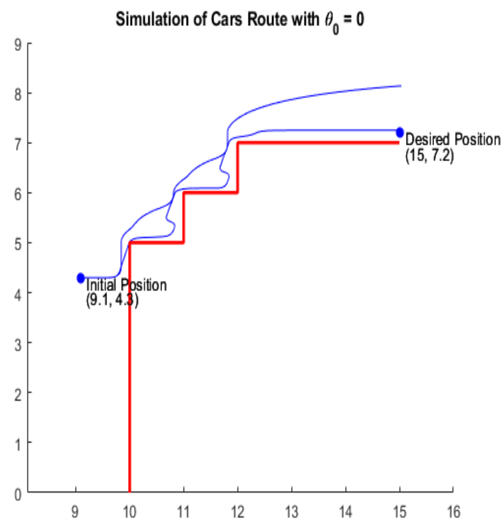

Figure 15: Simulation of car's movement for $\theta = 0$

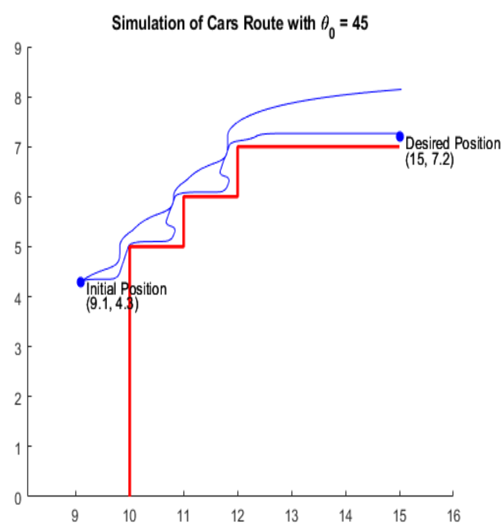

Figure 16: Simulation of car's movement for $\theta = 45$

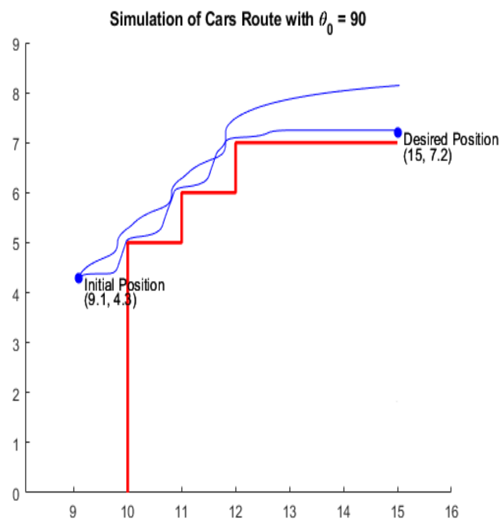

Figure 17: Simulation of car's movement for $\theta = 90$

List of Figures

List of Tables