Computational Intelligence - Project 2

Car Control

Nikolaos Konstas

Department of Electrical & Computer Engineering
Aristotle University of Thessaloniki

February 2024

Contents

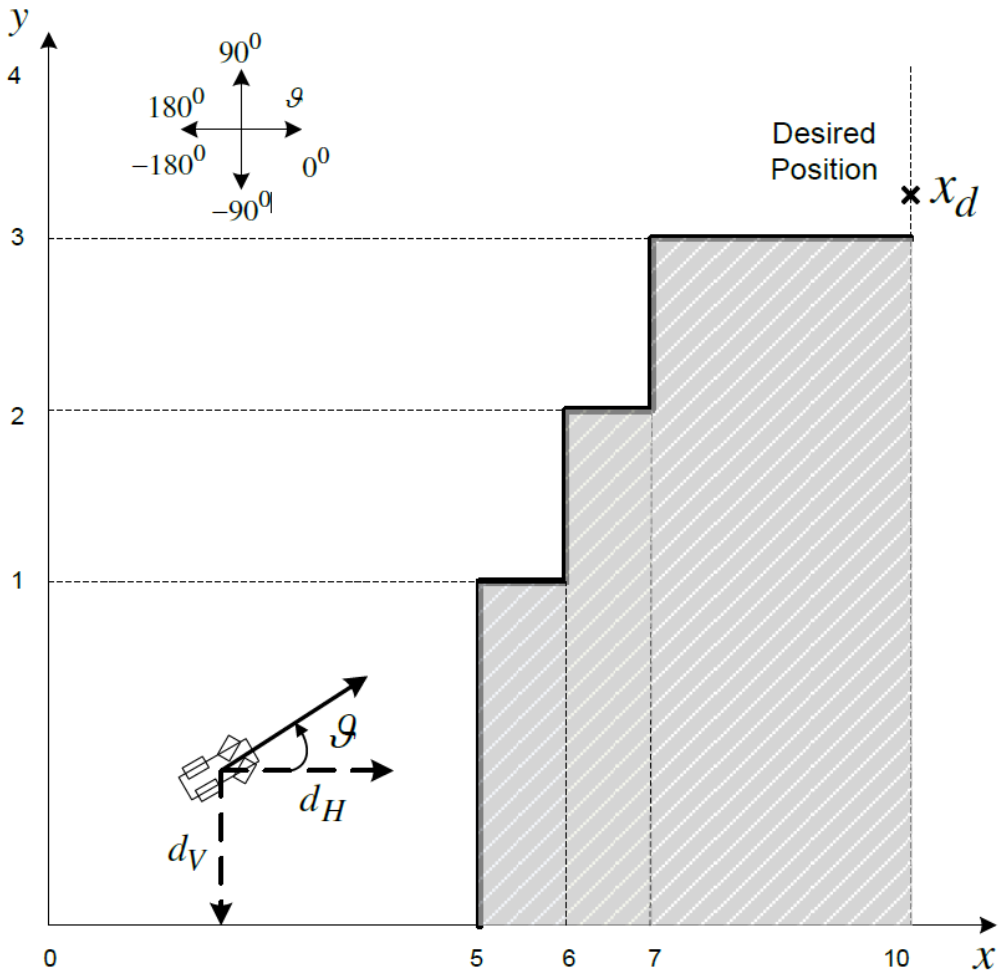
1 Introduction **2**

2 Rule Base **3**

3 Simulations **5**

1 Introduction

In this project we are asked to design a Fuzzy Controller, to control a vehicle, with the aim of avoiding obstacles. This process is shown in the figure below:



The purpose of the controller is to drive the vehicle to the desired location $(x_d, y_d) = (10, 3.2)$, without coming into contact with the obstacles shown in the figure. To achieve this, the controller, making use of the vehicle's sensors, receives at each sampling moment as input the vertical (d_V) and horizontal (d_H) distance from the obstacles, but also the angle θ of the direction of the vehicle. The output of the controller, with which it tries to steer the vehicle safely, is the change in the direction angle $\Delta\theta$.

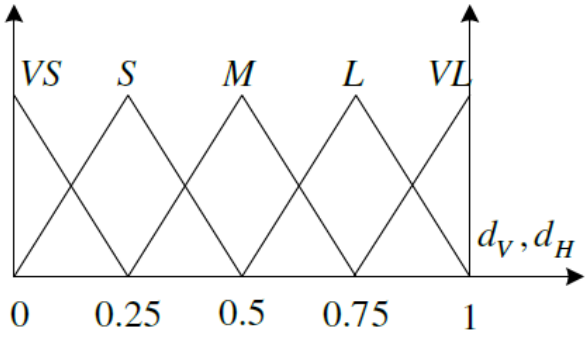
In the context of the project, we consider that the speed of the vehicle is constant and equal to $u = 0.05\, m/s$.

2 Rule Base

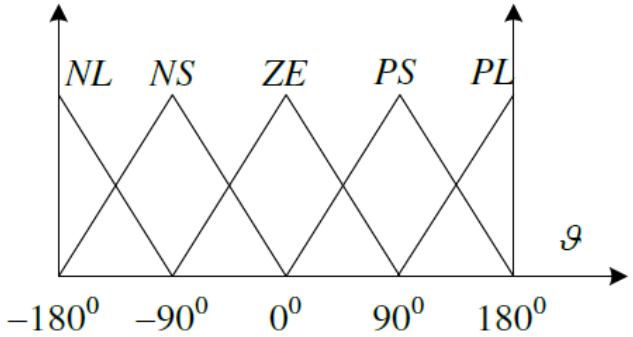
First we consider for the input and output variables of the Fuzzy Controller:

- $d_V = [0, 1]\ (m)$
- $d_H = [0, 1]\ (m)$
- $\theta = [-180°, 180°]$
- $d_V = [-130°, 130°]$

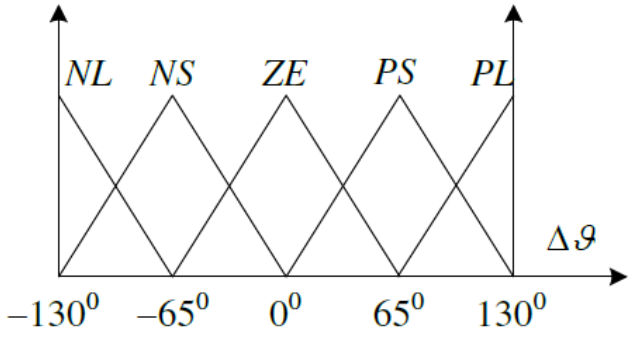
The fuzzy sets and participation functions of the variables are shown below:



Σχ. 2.



Σχ. 3.



Based on the above and after a trial & error process, we arrive at the following rule base:

	Rule
1	If d_h is VS and theta is NL then d_theta is NL
2	If d_h is VS and theta is NS then d_theta is PL
3	If d_h is VS and theta is ZR then d_theta is PL
4	If d_h is VS and theta is PL then d_theta is NS
5	If d_h is VS and theta is PS then d_theta is ZR
6	If d_h is S and theta is NL then d_theta is NL
7	If d_h is S and theta is NS then d_theta is PL
8	If d_h is S and theta is ZR then d_theta is PL
9	If d_h is S and theta is PS then d_theta is ZR
10	If d_h is S and theta is PL then d_theta is NS
11	If d_h is not S and theta is NL then d_theta is PL
12	If d_h is not S and theta is NS then d_theta is PS
13	If d_h is not S and theta is ZR then d_theta is ZR
14	If d_h is not S and theta is PS then d_theta is NS
15	If d_h is not S and theta is PL then d_theta is NL

Obviously, the above rules have a natural interpretation and have arisen with the following logic. Rules 1-5 are triggered when d_H $isVerySmall$ that is, when the vehicle is too close to an obstacle, and they try to avoid the impact by steering the vehicle in a safe direction. Rules 6-10 are triggered when d_H $isSmall$, i.e. a step ahead of the previous ones. The purpose of these rules is to steer the car towards the North, so as to avoid, if possible, getting too close to an obstacle. Finally, rules 10-15 are triggered when d_H $isnotSmall$, that is, when there are no obstacles at a short distance from the vehicle, and their purpose is to direct it towards the East.

With this combination of rules, as we will see below, all collisions are avoided and the vehicle approaches the target with satisfactory accuracy.

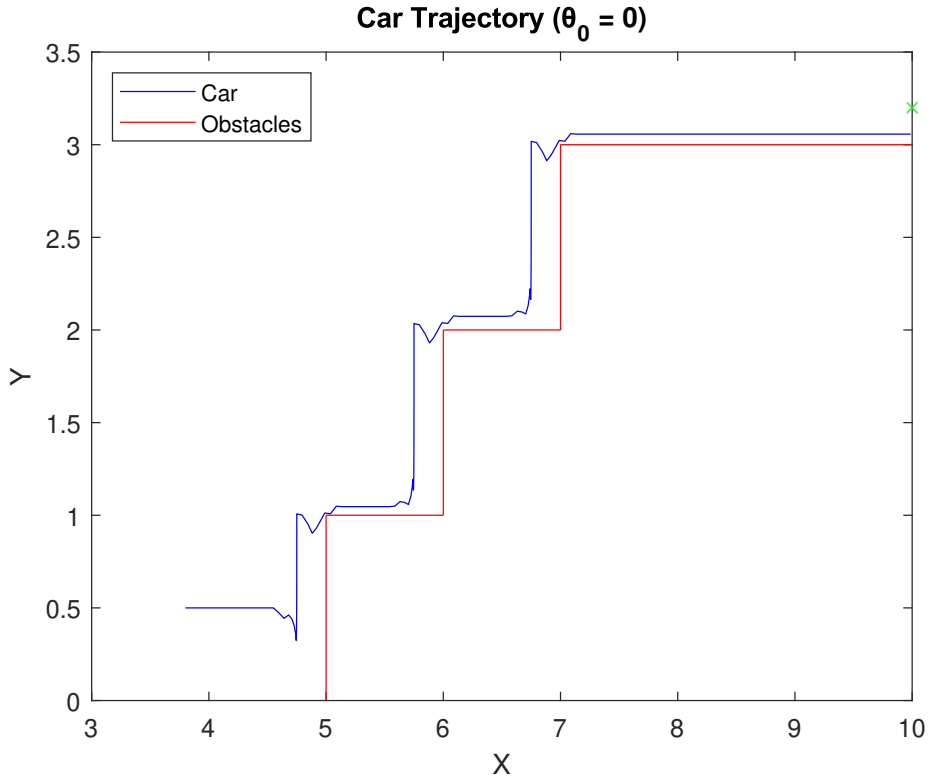
3 Simulations

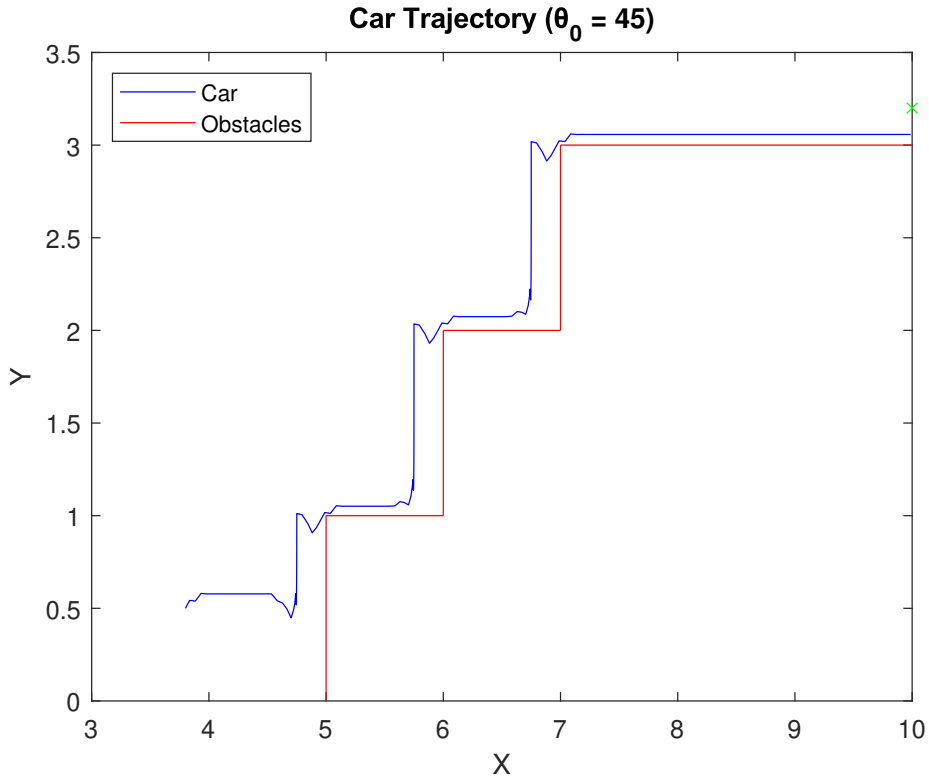
First, we created the Fuzzy Controller with the create_fis.m file, initializing the system, input/output variables, and participation functions, and then creating the rule base with the help of the fuzzyLogicDesigner GUI . The final form of the controller is stored in the file Car_FIS.fis.

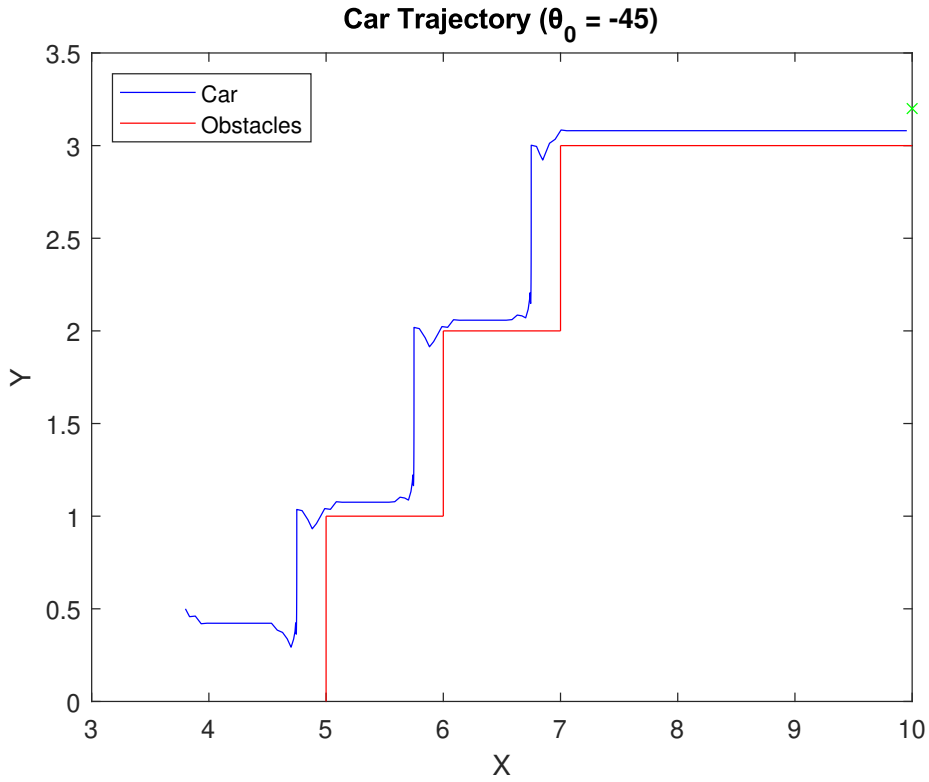
The system is simulated in the file Task2.m using an iterative while loop, which is executed until the distance from the desired position becomes sufficiently small.

Finally, the initial position of the vehicle is $(x_0, y_0) = (3.8, 0.5)$, while for the initial value of the direction of travel, 3 different values are tested, $\theta_0 = 0, \theta_0 = -45, \theta_0 = 45$.

Below are presented the resulting vehicle trajectories for the 3 different cases.




Car Trajectory (θ0 = 45)
Car
Obstacles
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
3
4
5
6
7
8
9
10


Car Trajectory (θ0 = -45)
Car
Obstacles
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
3
4
5
6
7
8
9
10

As can be seen from the diagrams above, the car successfully reaches the target for all 3 values of the initial orientation. Although it is quite close to the corners of the obstacles, it never comes into contact with them. In fact, we notice that apart from some differences in the initial path of the vehicle, due to the initial angle θ_0, in all 3 cases they have an almost identical trajectory. Therefore, the design of the Fuzzy Controller was successful.