

CE1184A - Artificial Intelligence for Games

Table of Contents

Table of Contents.....	1
Introduction	2
Minimum System Requirements	2
Usage of the Application.....	2
Structure of the Project	3
Methodology.....	3
Results.....	5
Conclusion.....	6
References	6

Introduction

The project is an artificial intelligence (AI) that can be used in a game to control the steering of a racing car. There is a very basic graphical representation to visualize results as the main focus of the project is the AI. The AI technique used throughout the project is a Fuzzy Inference System (FIS) which was implemented using the Fuzzy Logic Library (FLL) made by Dmitry Kaluzhny.

Minimum System Requirements

- Windows operating system
- Microsoft .NET Framework 2.0 or later

Usage of the Application

To use the application the executable file "RacingGUI.exe" should be executed by double clicking it.

The left hand side of the application consists of an abstract top-down graphical representation of the racing track, the racing line and car. The black border defines the dimensions in which the car can move. Because the car is always moving at a constant velocity its vertical position on the track cannot be modified. The red line in the middle of the track represents the racing line. The blue rectangle with cyan smaller rectangles represents the car. The green dot in the center of the car represents its current position on the track, it can be thought of as the center of gravity within the car.

Clicking within the borders of the racing track will move the position of the car.

On the right hand side the values of the car are shown. This is its current distance from the line and its rate of change (RoC). These values can be manually adjusted as desired. The calculate button below allows a single calculation to be performed by the car's AI. The result of this calculation is then displayed as steering value.

The Test Data button can be used to show the results of the testing data that was used for the project. They are dynamically calculated as the form is generated.

The Step button allows the user to perform a single step or tick in the race. A step within the race uses the current Distance and RoC and calculates the steering value. It then uses the steering value to determine how much the car should move left (in case of negative steering value) or right (in case of positive steering value). This is done by multiplying the steering value by 0.01 which is the maximum Sideway Speed of the car.

The Race button can be used once the car is positioned as desired by the user and an appropriate update frequency (speed at which a race step or tick is performed expressed in

milliseconds) is chosen. Once clicked it will start the race car and perform a race step every update frequency until the car reaches the racing line.

A lower frequency results in the race car reaching its destination (the racing line) much faster.

Once a race is started the user can use the same button (which is now named STOP) to stop the race and halt the car instantly at its current position.

During a race the user can still manipulate the racing car's position by clicking within the racing track's boundaries.

Structure of the Project

The project consists of 3 components:

1. **Racing GUI:** A very basic graphical representation of the racing track, racing line and racing car with on the side the visualisation of the racing car's values and buttons to perform actions. This component is the executable file.
2. **Racing AI:** The artificial intelligence (AI) component of that car that acts as a plugin component and can be replaced by any other desired AI. This component is a (Dynamic Link Library) DLL.
3. **Fuzzy Logic Library (FLL):** An open source library created by Dmitry Kaluzhny that is used to create the Fuzzy Inference System (FIS). This component is also a DLL.

Each component only depends on one other component. The GUI depends on the AI and the AI depends on the FLL. Because of the way this project is set up it allows the replacement of components. For example the Racing AI DLL can be replaced by any other Racing AI DLL which could perform the AI totally different as long as the methods and classes exposed to the public are identical.

Methodology

The usage of the Fuzzy Logic Library (FLL) placed certain constraints on the project as this FLL has limitations of its own. The two main limitations are the type of membership functions (MF) that can be used and the defuzzification methods that are available within the FLL.

Due to these limitations the project only uses very basic MF shapes such as triangle and trapezoid and the only implemented defuzzification method, centroid.

The Fuzzy Inference System (FIS) has 2 inputs and 1 output.

The first input is the distance to the racing line (D), the second input is the rate of change of distance to the racing line (RoC) and the output variable is used for steering the car.

All the variables use the same MFs. There are 5 MFs with input values between -1 and 1 used in the FIS are: Big Left, Left, Zero, Right and Big Right. The MFs are carefully chosen so that

the they mirror each other.

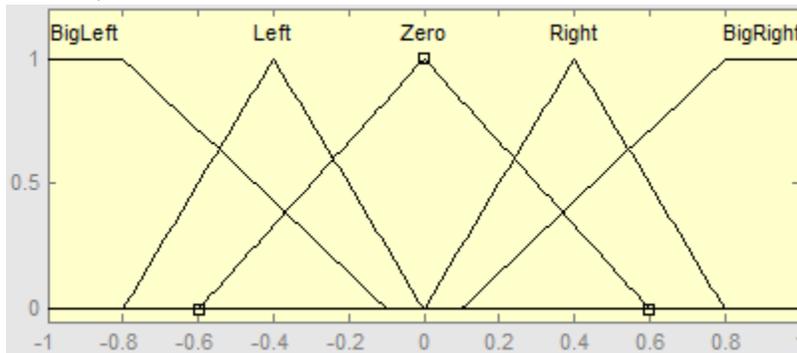


Figure 1: Membership Function Plot using MATLAB

The Big Left MF is a trapezoid defined by: [-1 -1 -0.8 -0.1]

The Left MF is a triangle defined by: [-0.8 -0.4 0]

The Zero MF is a triangle defined by: [-0.6 0 0.6]

The Right MF is a triangle defined by: [0 0.4 0.8]

The Big Right MF is a trapezoid defined by: [0.1 0.8 1 1]

There is a total of 25 rules used in the FIS. These rules for the FIS are kept simple and mirror each other as well. The following Fuzzy Associative Map (FAM) is used to visualize all the rules in the FIS.

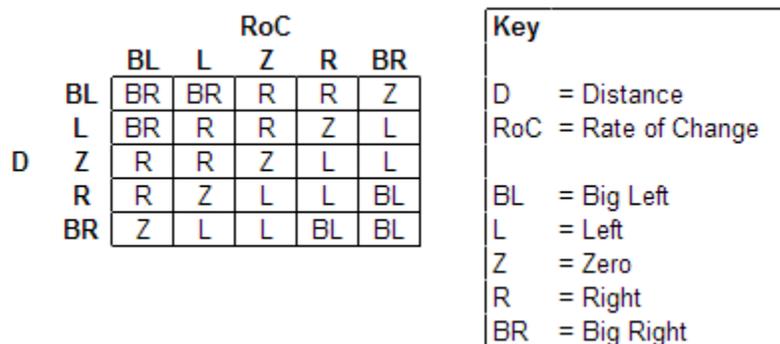


Figure 2: Fuzzy Associative Map

To evaluate the smoothness of these rules I've used MATLAB's surface tool which draws every possible output value for every possible combination of input values. The surface looks as next:

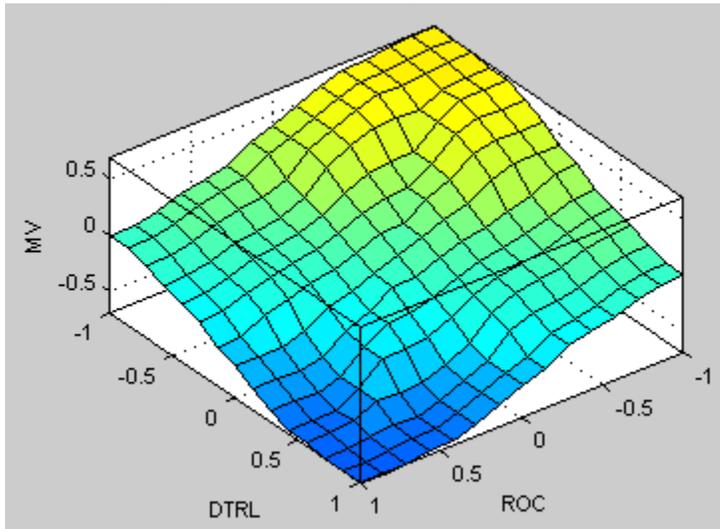


Figure 3: FIS Surface (generated by MATLAB)

Once the FIS was completed in MATLAB it was implemented into the application making use of the FLL.

In order to have the racing car function in a dynamic environment the output of the FIS was used to steer the car. The default and maximum steering value used is 0.01, this value is the maximum distance the car would move horizontally on the track. Because the FIS's output value ranges from -0.688 to 0.688, the maximum steering value is never steered as the output value acts as a percentage of steering on this maximum steering value to actually steer the car.

The formulas used to adjust the car's Distance and RoC are as following:

$$NewDistance = Distance + (SteeringValue * MaximumSpeed)$$

$$NewRoC = (NewDistance - Distance)$$

The maximum steering value is a 2 percentile of the tracks width. In further development this value can be adjusted to obtain different behavior for the cars but this was not worked out for this project and a fixed value was used instead.

Results

To test the AI a set of values which are 0.1 apart and range between -1 and 1 were used. The table below shows distance (D) in the rows and the Rate of Change (RoC) in the columns.

	-1	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
-1	0.688	0.688	0.688	0.675	0.659	0.64	0.643	0.593	0.532	0.465	0.4	0.4	0.274	0.209	0.177	0.146	0.105	0.056	0	0	0
-0.9	0.688	0.688	0.688	0.675	0.659	0.64	0.643	0.593	0.532	0.465	0.4	0.4	0.274	0.209	0.177	0.146	0.105	0.056	0	0	0
-0.8	0.688	0.688	0.688	0.675	0.659	0.64	0.643	0.593	0.532	0.465	0.4	0.4	0.274	0.209	0.177	0.146	0.105	0.056	0	0	0
-0.7	0.675	0.675	0.675	0.64	0.622	0.627	0.629	0.593	0.532	0.465	0.4	0.228	0.165	0.129	0.108	0.086	0.048	0	-0.056	-0.056	-0.056
-0.6	0.659	0.659	0.659	0.622	0.587	0.587	0.587	0.587	0.532	0.469	0.4	0.22	0.11	0.08	0.049	0.032	0	-0.048	-0.105	-0.105	-0.105
-0.5	0.64	0.64	0.64	0.627	0.587	0.422	0.422	0.422	0.404	0.335	0.264	0.166	0.104	0.051	0.024	0	-0.032	-0.086	-0.146	-0.146	-0.146
-0.4	0.643	0.643	0.643	0.629	0.587	0.422	0.314	0.311	0.308	0.264	0.202	0.124	0.064	0.03	0	-0.024	-0.049	-0.108	-0.177	-0.177	-0.177
-0.3	0.593	0.593	0.593	0.593	0.587	0.422	0.311	0.228	0.223	0.218	0.155	0.092	0.029	0	-0.03	-0.051	-0.08	-0.129	-0.209	-0.209	-0.209
-0.2	0.532	0.532	0.532	0.532	0.532	0.404	0.308	0.223	0.144	0.144	0.108	0.049	0	-0.029	-0.064	-0.104	-0.11	-0.165	-0.274	-0.274	-0.274
-0.1	0.465	0.465	0.465	0.465	0.469	0.335	0.264	0.218	0.144	0.056	0.056	0	-0.049	-0.092	-0.124	-0.166	-0.22	-0.228	-0.4	-0.4	-0.4
0	0.4	0.4	0.4	0.4	0.4	0.264	0.202	0.155	0.108	0.056	0	-0.056	-0.108	-0.155	-0.202	-0.264	-0.4	-0.4	-0.4	-0.4	-0.4
0.1	0.4	0.4	0.4	0.4	0.228	0.22	0.166	0.124	0.092	0.049	0	-0.056	-0.056	-0.144	-0.218	-0.264	-0.335	-0.469	-0.465	-0.465	-0.465
0.2	0.274	0.274	0.274	0.165	0.11	0.104	0.064	0.029	0	-0.049	-0.108	-0.144	-0.144	-0.223	-0.308	-0.404	-0.532	-0.532	-0.532	-0.532	-0.532
0.3	0.209	0.209	0.209	0.129	0.08	0.051	0.03	0	-0.029	-0.092	-0.155	-0.218	-0.223	-0.228	-0.311	-0.422	-0.587	-0.593	-0.593	-0.593	-0.593
0.4	0.177	0.177	0.177	0.108	0.049	0.024	0	-0.03	-0.064	-0.124	-0.202	-0.264	-0.308	-0.311	-0.314	-0.422	-0.587	-0.629	-0.643	-0.643	-0.643
0.5	0.146	0.146	0.146	0.086	0.032	0	-0.024	-0.051	-0.104	-0.166	-0.264	-0.335	-0.404	-0.422	-0.422	-0.422	-0.587	-0.627	-0.64	-0.64	-0.64
0.6	0.105	0.105	0.105	0.048	0	-0.032	-0.049	-0.08	-0.11	-0.22	-0.4	-0.469	-0.532	-0.587	-0.587	-0.587	-0.587	-0.622	-0.659	-0.659	-0.659
0.7	0.056	0.056	0.056	0	-0.048	-0.086	-0.108	-0.129	-0.165	-0.228	-0.4	-0.465	-0.532	-0.593	-0.629	-0.627	-0.622	-0.64	-0.675	-0.675	-0.675
0.8	0	0	0	-0.056	-0.105	-0.146	-0.177	-0.209	-0.274	-0.4	-0.4	-0.465	-0.532	-0.593	-0.643	-0.64	-0.659	-0.675	-0.688	-0.688	-0.688
0.9	0	0	0	-0.056	-0.105	-0.146	-0.177	-0.209	-0.274	-0.4	-0.4	-0.465	-0.532	-0.593	-0.643	-0.64	-0.659	-0.675	-0.688	-0.688	-0.688
1	0	0	0	-0.056	-0.105	-0.146	-0.177	-0.209	-0.274	-0.4	-0.4	-0.465	-0.532	-0.593	-0.643	-0.64	-0.659	-0.675	-0.688	-0.688	-0.688

Figure 3: Test Data Result Table

These values were generated through the Fuzzy Logic Library (FLL) that was used and are accessible from within the application.

There is very small difference (with a maximum of 0.006) discovered when comparing some of the test data results against the MATLAB results. The difference is likely due to a difference in rounding between the FLL and MATLAB. As the difference is very small it is being neglected here. Therefore the results of the FLL can be deemed as correct and the AI was implemented as designed within MATLAB.

Conclusion

The results of the project show that the Fuzzy Inference System (FIS) that was designed using MATLAB has been implemented correctly within an a certain error margin that is likely due to a difference in rounding.

The usage of a FIS suits perfectly for the goal of the project. With its 2 inputs and single output it gets close to what a racer would actually do when told to following a certain racing line.

References

Dmitry Kaluzhny, 2008, Fuzzy Logic Library for Microsoft .Net, Last accessed January 2014, <http://fuzzynet.sourceforge.net>