Evolving TORCS Fuzzy driver using GA

No Author Given

No Institute Given

Abstract. When driving a car it is essential to take into account all possible factors;like in the TORCS simulated race game, the objective is not only to avoid collisions, but to go faster and safely to win the race. Fuzzy based controller were usually used to design autonomous drivers by computing target speed or steer or both of them.
The major drawback of these controllers , is that their membership function parameters are tuned by trial/error.
In this work, we propose tu use genetic algorithm approach to find the optimal combination of the MFs parameters so to design a robust driver for racing car in a simulated race.
The obtained drivers were compared and evaluated in practice and real races giving good results mainly in tracks that have many turning points.

Keywords: Videogames, Fuzzy Controller, TORCS, Steering control, Optimization, Genetic Algorithms

1 Introduction

Autonomous driving is a very interesting research topic supported by many vehicle manufacturers aiming to reduce fuel consumption and its efficient use, improve car safety and driver comfort and even to design real autonomous cars that can run in a desert and hostile environment.
The Open Racing Car Simulator (TORCS) is a realistic racing simulator with a sophisticated engine used for many standalone racing competition challenges each year. This fact, combined with the large gaming community and the ability to compare controllers have made TORCS the most used simulator in the field of autonomous driving [**?**].

Improved driving in TORCS was achieved using fuzzy logic in previous research where two fuzzy controllers were developed for steering (steer) calculation and the desired speed giving encouraging results but the major disadvantage of this approach is that the parameters of the membership functions of the controllers are defined by trial/error in the absence of experts to do so.

Since this problem of parameter determination can be considered as an optimization problem, evolutionary algorithms are well placed to solve this it.

In this work, we propose to apply the genetic algorithms to adjust the parameters of the fuzzy controller namely those of the membership functions for the two fuzzy controllers used in [**?**]. The considered approach requires the coding of chromosomes and the choice of a adequate cost function (fitness) due to the uncertainty and noisiness of the problem.

2 State of the Art

3 TORCS Description:

The Open Racing Car Simulator (TORCS) is an open source, modern, multi-player, modular and portable racing simulator that allows users to race against computer-controlled opponents. TORCS allows the user to control one of the robots through an input device such as keyboard, mouse or joystick [6].

Its high degree of modularity and portability make it ideal for artificial intelligence research. Indeed, a number of competitions and research-based papers have already appeared that make use of the TORCS engine. Because of these features, the simulator has become popular and interesting as it provides real-time driving simulation games. [2] The game allows different types of races from the practical single session to the championship [6].

Sensor	**Name**	**Range** (unit)	**Data type**
1	angle	[$-\pi$,$+\pi$]	Double
2	curLapTime	[0,+∞] (s)	Double
3	damage	[0,+∞)(point)	Double
4	distFromStart	[0,+∞) (m)	Double
5	distRaced	[0,+∞) (m)	Double
6	focus	[0,200] (m)	Double
7	fuel	[0,+∞) (l)	Double
8	gear	{-1,0,1,.. 6}g	Integer
9	lastLapTime	[0,+1] (s)	Double
10	opponents	[0,200] (m)	Double
10	racePos	{1,2,...,N}	Double
11	rpm	[0,+∞) (rpm)	Double
13	speedX	(-∞,+∞) (km/h)	Double
14	speedY	(-∞,+∞) (km/h)	Double
15	speedZ	(-∞,+∞) (km/h)	Double
16	track	[0,200]	Double
17	trackPos	(-∞,+∞)	Double
18	wheelSpinVel	[0,+∞) (rad/s)	Double
19	z	(-∞,+∞) (m)	Double

Table 1. Description of available sensors in TORCS [4].

Every TORCS driver bot is controlled by means of a set of actuators: the steering wheel 'Steer', the accelerator 'accel', the brake pedal and the gearbox. In addition, a meta-action is available to request a restart of the race to the server. Table 2 details the available actions/actuators and their representation.

Action	**Range** (unit)	**Data type**
Acceleration	[0,+1]	Double
Brake	[0,+1]	Double
Gear	-1..0..+6	Double
Steer	[-1,+1]	Double
Clutch	[-1,+1]	Double

Table 2. TORCS Actuators [5].

Hence, a controller is a program, which run inside TORCS, that automatically drives a car. It gets as input information about the current state of the car and its situation on the track. These collected data are used to compute actions to do in the next simulation tick; like steer, gear changes, acceleration or brake and clutch. A client may request a restart of the race by sending a special action on the server: Restart or shutdown [3].

4 Fuzzy Controller

The proposed controller has the same modular architecture as the simple driver, and has some common functions of this approach. However, the target speed and steering angle are computed by means of two modular and specialised fuzzy controllers, which consider five position sensors.

In the following sections, each sub-controller is described.

4.1 Fuzzy target speed sub-controller

The first proposed fuzzy sub-controller aims to estimate the optimal target speed of the car, both in straight parts and curves of the track, taking into account two criteria: moving as faster as possible and secure the car. This estimation is based on fuzzy rules, so two general cases are considered, following the simple driver approach:

This fuzzy controller has three input values and one output: the target speed.

The presented controller is a Mamdani-based fuzzy system [1] with trapezoidal Membership Functions (MF) for input variables, because it avoids sudden changes in input values. It considers three values among the 19 track sensors:

– Front = Track[9]: the front distance between the car and the border of the track (angle 0Âº).
– M5 = max (Track[8], Track[10]): the max distance to the track limits in an angle of +5Âº and -5Âº with respect to Front.
– M10 = max (Track[7], Track[11]): the max distance to the border in an angle of +10Âº and -10Âº.

Each input variable is represented by three membership functions: Low, Medium and High. The description of fuzzy inputs and output are represented in Table 3.

The base of rules has been composed modelling the behaviour of a human expert driver, refining them also. Thus, this set is designed so that if the frontal distance is maximal, then the target speed should be the maximum. Its value should be lower when the frontal distance is shorter. The fuzzy rules are listed below:

– IF Front is High THEN TargetSpeed is TS1
– IF Front is Medium THEN TargetSpeed is TS2
– IF Front is Low and M5 is High THEN TargetSpeed is TS3

Table 3. Fuzzy variables description.

Variable	Range	Name	MF	Low	Medium	High
Input	[0-100] m	Front	trapezoidal	[0-50]	[20-80]	[60-100]
Input	[0-100] m	M5	trapezoidal	[0-40]	[10-70]	[50-100]
Input	[0-100] m	M10	trapezoidal	[0-30]	[20-60]	[50-100]
Output	[0-200] m/s	TargetSpeed	singleton	/	/	/

– IF Front is Low and M5 is Medium THEN TargetSpeed is TS4
– IF Front is Low and M5 is Low and M10 is High THEN TargetSpeed is TS5
– IF Front is Low and M5 is Low and M10 is Medium THEN TargetSpeed is TS6
– IF Front is Low and M5 is Low and M10 is Low THEN TargetSpeed is TS7

In addition, a crisp rule is added to rule base to obtain a maximum value of the target speed when the three input variables are as big as possible:

– IF Front = MAXDISTSPEED or M5 = MAXDISTSPEED or M10 = MAXDISTSPEED THEN TargetSpeed = MAXSPEED

MAXDISTSPEED is the a longest possible value for the track sensors, and MAXSPEED, which is related to the car's properties, is the maximal car speed. For example in the case of car-trb1 model, MAXSPEED=300.

The output value is encoded by seven singletons TS1 to TS7, being respectively: 280, 240, 220, 180, 120, 60 and 30.

4.2 Fuzzy steering control sub-controller

In addition to the speed sub-controller, another fuzzy approach has been applied to control the steering, estimating and determining the target position of the car.

The architecture of this sub-controller is similar to the one shown in Figure **??**, but with the steering as output. The set of sensors considered are the same as in the speed case, described in Table 3.

Then, if the car is in a straight line, it will set as target position half width of the race track (central position of the lane). Whereas, if the car is near a right curve, it will approach the path leading to the right, with a space between the car and the border of the track to avoid the loss of control. The same approach is considered if the car is near a left curve.

In order to detect the curves, the controller focuses on the sensor values (M10, M5, and Front). So, if the value on Front sensor is the longest, there is a straight road; whereas if the values of M5 and M10 with positive angles (+5 and +10) are the longest, there is right curve; and the other way round.

The base of rules has been defined again modelling the behaviour of a human driver, so, for this controller is:

– IF Front is High THEN steer is S1

- IF Front is Medium AND M10 is High THEN steer is S2
- IF Front is Medium AND M10 is Medium AND M5 is Medium THEN steer is S2
- IF Front is Medium AND M10 is Medium AND M5 is Low THEN steer is S3
- IF Front is Low AND M10 is High THEN steer is S3
- IF Front is Low AND M10 is Medium AND M5 is Medium THEN steer is S4
- IF Front is Low AND M10 is Medium AND M5 is Low THEN steer is S4

The values for S1 to S4 are respectively: 0, 0.25, 0.5, and 1. When M10=Track[7] we will take negative values of the steer (sterr=-steer).

Once the controllers have been described, they will be tested and compared with the standard one y several races. The obtained results are presented in the following section.

5 Optimizing the fuzzy controller with GA

A triangular membership function is given by:

$$\mu_A(x) = \begin{cases} 0, & x \leq x_1 \\ \frac{x-x_1}{x_2-x_1}, & x_1 \leq x \leq x_2 \\ \frac{x_3-x}{x_3-x_2}, & x_2 \leq x \leq x_3 \\ 0, & x_3 \leq x \end{cases} \tag{1}$$

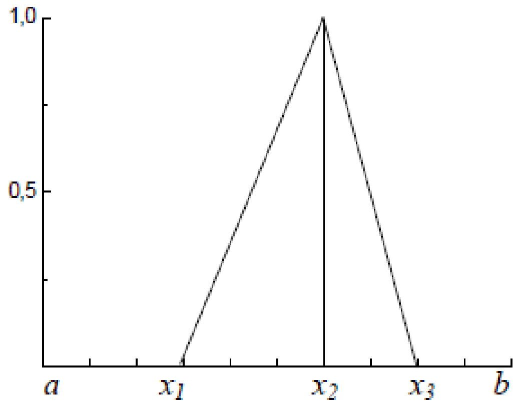
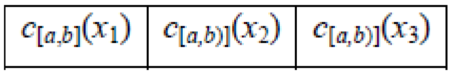


Fig. 1. Triangular MF

$$x_1 \leq x_2 \leq x_3 \tag{2}$$



$c_{[a,b]}(x_1)$	$c_{[a,b)]}(x_2)$	$c_{[a,b)]}(x_3)$

Fig. 2. Triangular MF parameters coding

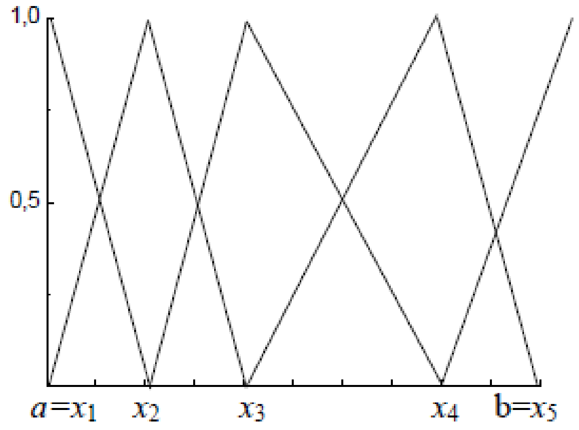


Fig. 3. Fuzzy partition with Triangular MFs

$$
\begin{aligned}
\mu_{A1}(x) &= \begin{cases} 1, & x \leq x_1 \\ \frac{x_2 - x}{x_2 - x_1}, & x_1 \leq x \leq x_2 \\ 0, & x > x_2 \end{cases} \\
\mu_{Ai}(x) &= \begin{cases} 0, & x \leq x_i \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x \leq x_i, n = 2, ..., i-1 \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i \leq x \leq x_{i+1} \\ 0, & x > x_{i+1} \end{cases} \\
\mu_{An}(x) &= \begin{cases} 0, & x \leq x_{n-1} \\ \frac{x - x_{n-1}}{x_n - x_{n-1}}, & x_{n-1} \leq x \leq x_n \\ 1, & x > x_n \end{cases}
\end{aligned} \tag{3}
$$

we have :

$$a = x_1 \leq x_2 \leq ... \leq x_n = b$$

$c_{[a,b]}(x_2)$	$\cdots$	$c_{[a,b]}(x_i)$	$\cdots$	$c_{[a,b]}(x_{n-1})$

Fig. 4. Codage des n points

Trapezoidal MFs

$$\mu_A(x) = \begin{cases} \frac{x-x_1}{x_2-x_1}, & x_1 \leq x \leq x_2 \\ 1, & x_2 \leq x \leq x_3 \\ \frac{x_4-x}{x_4-x_3}, & x_3 \leq x \leq x_4 \\ 0, & ailleurs \end{cases} \tag{4}$$

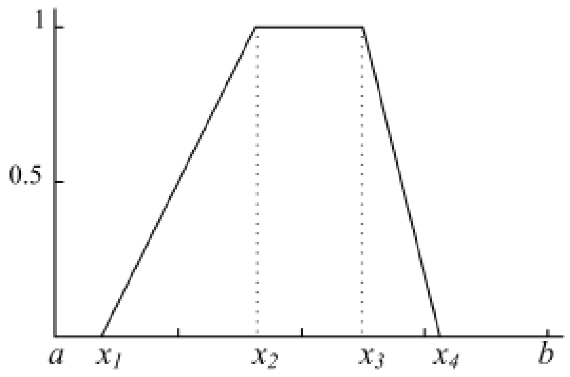


Fig. 5. Trapezoidal MFs

with

$$x_1 \leq x_2 \leq x_3 \leq x_4 \tag{5}$$

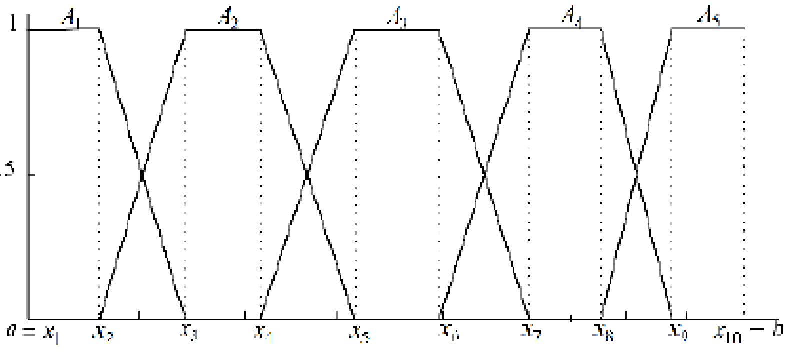
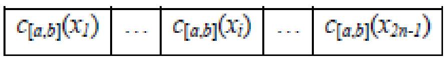


Fig. 6. Fuzzy partition with trapezoidal MFs

$$\mu_{A1}(x) = \begin{cases} 1, & x_1 \leq x \leq x_2 \\ \frac{x_3 - x}{x_3 - x_2}, & x_2 \leq x \leq x_3 \\ 0, & x > x_3 \end{cases}$$
$$\mu_{Ai}(x) = \begin{cases} 0, & x \leq x_{2i-2} \\ \frac{x - x_{2i-2}}{x_{2i-1} - x_{2i-2}}, & x_{2i-2} \leq x \leq x_{2i-1}, n = 2, ..., i-1 \\ 1, & x_{2i-1} \leq x \leq x_{2i} \\ \frac{x_{2i+1} - x}{x_{2i+1} - x_{2i}}, & x_{2i} \leq x \leq x_{2i+1} \\ 0, & x > x_{2i+1} \end{cases} \tag{6}$$
$$\mu_{An}(x) = \begin{cases} 0, & x \leq x_{2n-2} \\ \frac{x - x_{2n-2}}{x_{2n-1} - x_{2n-2}}, & x_{2n-2} \leq x \leq x_{2n-1} \\ 1, & x > x_{2n-1} \end{cases}$$

with:



$c_{[a,b]}(x_1)$	...	$c_{[a,b]}(x_i)$	...	$c_{[a,b]}(x_{2n-1})$

Fig. 7. Trapezoidal-shaped MFs coding

$$a = x_1 \leq x_2 \leq ... \leq x_{2n-1} \leq x_{2n} = b \tag{7}$$

5.1 Chromosomes

$0 = x_{11} \leq x_{12} \leq x_{13} \leq x_{14} \leq x_{15} \leq x_{16} = 100$

$0 = x_{21} \leq x_{22} \leq x_{23} \leq x_{24} \leq x_{25} \leq x_{26} = 100$
$0 = x_{31} \leq x_{32} \leq x_{33} \leq x_{34} \leq x_{35} \leq x_{36} = 100$

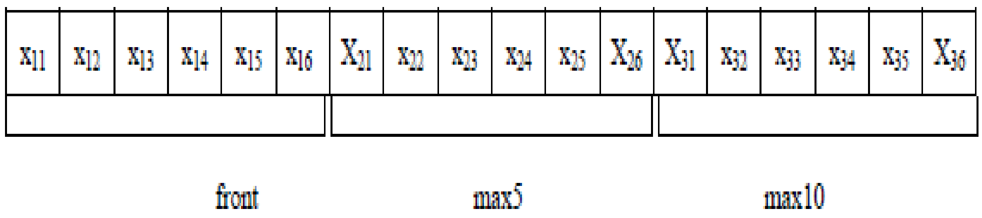


Fig. 8. Chromosome description

5.2 Fitness definition

fitness 1:

$$f_1 = Min\ damage + \ Min\ RaceTime \tag{8}$$

fitness 2:

$$f_2 = Min\ damage + \ Min\ RaceTime + \ Min\ \frac{1}{MaxSpeed} \tag{9}$$

6 Simulation Results

In this section we present all the experiments that we performed in order to measure the performance of our controller, called *AD (Automatic Driver)*. We first, describe the methodology we have used and next, we present the experimental results of the implementation of the fuzzy driver, including opponents and using special criteria in each case.

6.1 Simulation settings

TORCS provides several tracks to choose which have been designed in order to test the performance of the controllers on different difficulty circuits and on different types of roads. In our case, we chose the E-Track5 circuit

Table 4 presents the properties and description of selected track.

The aim of this selection is to test the value of AD in a wide set of situations, in order to check its potential as a global controller.

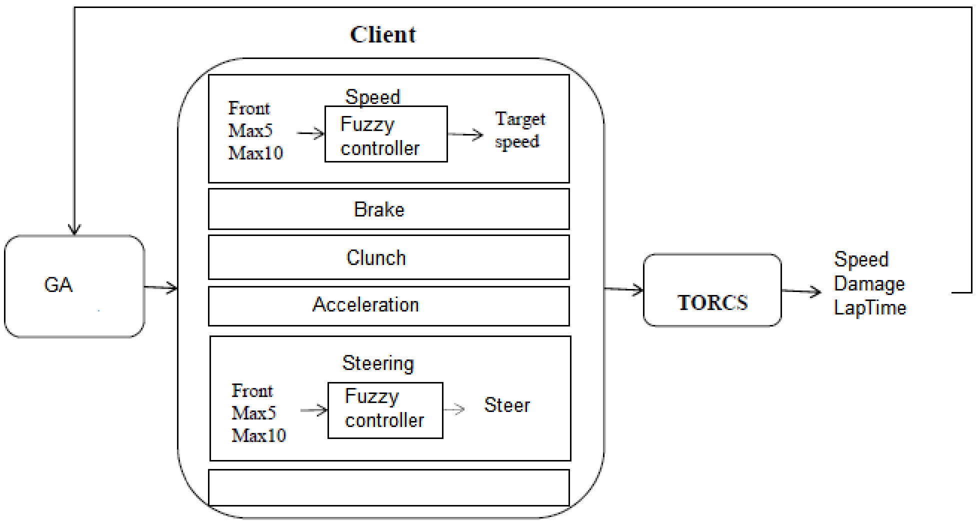


Fig. 9. Optimization of fuzzy controller flowchart

Table 4. Description of Selected track

Track name	E-Track5
Shape	
Track Type	Oval
Length	1621.73 m
Width	20.0 m

Cars settings. The car considered is *car1-tbr1* of SCR 1 Server, which is a NASCAR car part of the SCR Server team. It has a weight of 1150 KG, max fuel: 94 KG, length: 4.25m and width: 1.94m, with 300km/h as maximal speed. This is a fair car, i.e. not the fastest, but a quite fast. Anyway, the obtained results represents the quality of the controllers and could be extrapolated to any other car.

Controllers settings. We have tested the different sub-controller features in a separated way, in order to see which one has the highest influence on the success or failure of AD. Thus, we have considered the following bots in the experiments:

1. AD: Fuzzy controller .
- AGFC1: GA-Fuzzy controller with fitness 1 8,
- AGFC2: GA-Fuzzy controller with fitness 2 9,

Table 5. GA parameters

Population size	20
Generations	50
Crossover rateP_c	0.7
Mutation rate P_m	0.3

6.2 GA-Fuzzy controller in practice race

Table 6. Results of the three controllers in a 20 laps practice race

E-Track 5			
20 tours	**AD**	**AGFC1**	**AGFC2**
Best Time	29:70	30:03	29:50
Topspeed	209	216	216
Minspeed	168	148	182
Time Lastlap	29:79	30:03	29:50
Best lap	19/20	20/20	20/20
Damages	936	0	0
Lap	20/20	20/20	20/20
E-Road			
20 tours	**AD**	**AGFC1**	**AGFC2**
Best Time	02:31:71	02:26:72	02:26:54
Topspeed	206	205	208
Minspeed	30	39	37
Time Lastlap	03:12:79	02:42:26	02:33:36
Best lap	02/20	02/20	02/20
Damages	0	0	0
Lap	20/20	20/20	20/20

6.3 GA-Fuzzy controllers in a real race

7 Conclusions and Future Work

Table 7. Results of AGFC1 in a real race

E-track5	**AGFC1**	**berwin 10**	**bt 3**	**damned 2**	**inferno 5**	**contre tita 10**
Ranking	3/6	4/6	1/5	5/6	2/6	6/6
Temps total	02:30:74 +35:70	02:30:74 +1lap	02:30:74	02:30:74 +1lap	02:30:74 +12:13	02:30:74 +1lap
Best Lap	35:65	36:39	28:57	36:83	30:53	35:39
Maxspeed	196	202	231	192	226	202
Damages	0	0	0	603	0	471
Pit stops	0	0	0	0	0	0

Table 8. Results of AGFC2 in a real race

E-track5	**AGFC2**	**berwin 10**	**bt 3**	**damned 2**	**inferno 5**	**tita 10**
Ranking	2/6	4/6	1/6	6/6	3/6	5/6
Race Time	02:30:83 +03:99	02:30:83 +1lap	02:30:83	02:30:83 +1lap	02:30:83 +08:35	02:30:83 +1lap
Best Time	29:82	36:38	28:35	37:04	30:53	36:00
Maxspeed	214	202	230	188	226	204
Damages	0	0	343	1230	0	668
Pit stops	0	0	0	0	0	0

Table 9. Results of AD in a real race

E-track5	**AD**	**berwin 10**	**bt 3**	**damned 2**	**inferno 5**	**tita 10**
Ranking	6/6	5/6	1/6	4/6	2/5	3/5
Race Time	02:31:83 +5laps	02:31:83 +1lap	02:31:83	02:31:83 +1lap	02:31:83 +21:32	02:31:83 +33:43
Best Time	00:00	37:25	28:60	36:28	31:47	35:84
Maxspeed	111	202	230	189	225	202
Damages	10786	465	5	0	2394	0
Pit stops	0	0	0	0	0	0

References

1. Iancu, I.: A Mamdani Type Fuzzy Logic Controller, pp. 325–352. InTech (2012)
2. J.L., S., Chin, K., Jason, T., T.G.Tan, Rayner, A.: Evolving controllers for simulated car racing using differential evolution. Asia-Pacific Journal of Information Technology and Multimedia (2013)
3. Loiacono, D., Cardamone, L., Lanzi, P.: Simulated car racing championship competition. software manual. TORCS news (2013)
4. Loiacono, D., Cardamone, L., B, M., Lanzi, P.L.: The 2011 simulated car racing championship @ cig-2011. TORCS news (2011), `http://cig.dei.polimi.it/wpcontent/`
5. Onieva, E., Pelta, D., Godoy, J., Milanés, V., Rastelli, J.: An evolutionary tuned driving system for virtual car racing games: The autopia driver. International Journal of Intelligent Systems 27, 217–241 (2012)
6. Wyman, B., Espie, E., Guionneau, C., Dimitrakakis, C., Coulom, R., Sumner, A.: Torcs the open racing car simulator. http://www.torcs.org (2014)