A Fuzzy Logic Based Autonomous Vehicle Control System Design in the TORCS Environment

Ersin ARMAGAN, Tufan KUMBASAR

Department of Control and Automation Engineering, Istanbul Technical University, Istanbul, Turkey
{armagane, kumbasart}@itu.edu.tr

Abstract

In this study, a fuzzy logic based autonomous vehicle control system is designed and tested in The Open Racing Car Simulator (TORCS) environment. The aim of this study is that vehicle complete the race without to get any damage and to get out of the way. In this context, an intelligent control system composed of fuzzy logic and conventional control structures has been developed such that the racing car is able to compete the race autonomously. In this proposed structure, once the vehicle's gearshifts have been automated, a fuzzy logic based throttle/brake control system has been designed such that the racing car is capable to accelerate/decelerate in a realistic manner as well as to drive at desired velocity. The steering control problem is also handled to end up with a racing car that is capable to travel on the road even in the presence of sharp curves. In this context, we have designed a fuzzy logic based positioning system that uses the knowledge of the curvature ahead to determine an appropriate position. The game performance of the developed fuzzy logic systems can be observed from https://youtu.be/qOvEz3-PzRo.

1. Introduction

Nowadays, car-racing simulations are becoming more and more realistic. One of the most important of these simulations is The Open Racing Car Simulator (TORCS) [1]. The first factor of separation of TORCS from the other racing simulations is its realistic vehicle and track characteristics. The second factor is that artificial and computational intelligence methods can easily be tested, as it is an open source game environment. The source code of TORCS is licensed under the GPL and it runs on 32 and 64-bit architectures such as Linux, MacOS, Windows [1]. Tracks and cars have different characteristics in TORCS. Different friction coefficients, road widths and environmental factors exist for different tracks [6,7]. The racing cars have also different aerodynamic features, collision modelling and engine characteristics [3,4,6,7]. In literature, there are related works implemented artificial/computational intelligence methods in TORCS environment. In many of these works, noise reduction techniques of sensor data and machine learning techniques are considered. Control theoretical techniques have been also employed to design an autonomous car control system [2-10].

In this study, a fuzzy logic based autonomous vehicle control system is proposed and its performance is examined in the TORCS game environment. In the design of the intelligent system, we aimed to design an intelligent vehicle control system such that the racing car is capable to finish the race without any damage, to get out of the curvature and to move at desired position on the road. In order to accomplish such a goal, we developed an autonomous vehicle control system composed of PID and fuzzy logic control structures. Firstly, we have designed a simple but effective system to automate the gearshifts automatically with respect to engine rpm. Then, using expert knowledge, a Fuzzy Logic based Velocity Regulator (FLVR) has been designed such that the racing car is capable to accelerate/decelerate in a realistic manner as well as to drive at desired velocity. Finally, the steering control problem has been handled to position the car on the track on a desired trajectory, especially in presence of sharp curvature. In this context, Fuzzy Logic based Positioning Regulator (FLPR) is designed using expert knowledge to calculate the desired position of the vehicle on the road so performance of the vehicle is increased especially on the curvatures. In order to show the effectiveness of the developed fuzzy logic based autonomous vehicle control system various simulation results are presented. The game performance of the developed fuzzy logic systems can be observed from https://youtu.be/qOvEz3-PzRo.

The paper is organized as follows. Section 2 provides information about TORCS game. Section 3 presents the fuzzy logic based vehicle control system and its performance. Finally, the driven conclusions are summarized in Chapter 4.

2. TORCS Game Environment

As shown in Fig. 1, the TORCS provides a realistic game environment. The game provides different models of vehicles and several tracks, thus vehicles and tracks have their own realistic characteristics [3,4,6,7]. In this study, we have handled the “Motorway” and “Ruudskogen” tracks which have unique track characteristics. For an easy design and deployment, we have preferred to use the Matlab/Simulink TORCS driver [12]. In Table 1 and Table 2, feedback and input signals of the game environment are tabulated, respectively. The Simulink driver has a fixed sampling time of 0.02s [1], thus we have fixed the sampling time of the controllers to same value.

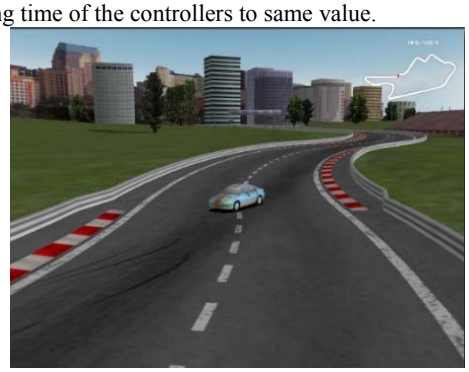

Fig. 1. A screenshot from TORCS game environment

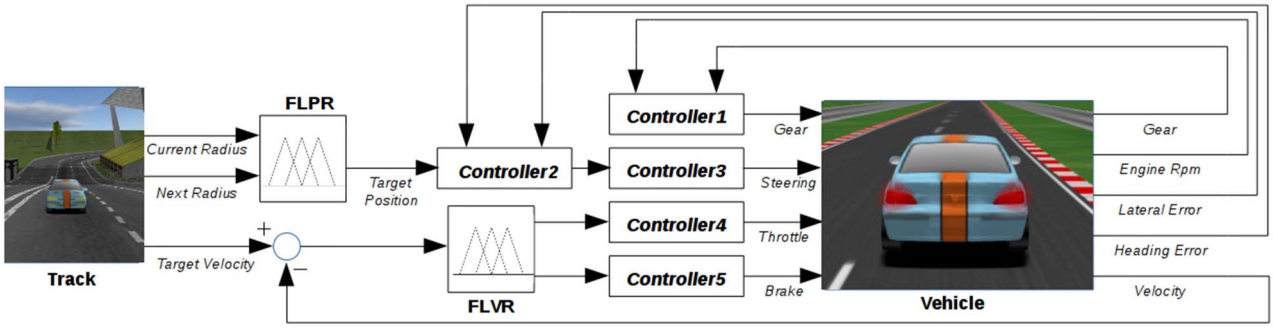

Fig. 2. The fuzzy logic based autonomous vehicle control system

Table 1. Feedback Signals from the TORCS game

Name	Name	Unit
Velocity	Velocity of the vehicle	m/s
Gear	Current gear of the vehicle	-
HeadingError	Error between car and track axis	rad
LateralError	Error between car and track center	m
RoadCurvature	Curvature of the road	m
EngineRPM	Engine rpm	rpm

Table 2. Input Signals to TORCS game

Name	Interval
Throttle	Between 0 and 1
Brake	Between 0 and 1
Steering	Between -1 and 1
Gear	Between -1 and 7

3. Fuzzy Logic Based Autonomous Vehicle Control System

In this section, we will present the internal structure and design of the proposed autonomous control system shown in Fig. 2.

3.1. Gear Control System

It is necessary to design gearshift mechanism to achieve a smooth acceleration/deceleration in the car speed [5,7,8]. Thus, we have developed a lookup table, presented in Table 3, with respect to vehicle dynamics.

Table 3. The Gearshift Lookup Table

Gear	Gear Down [rpm]	Gear Up [rpm]
Gear 0	-	1000
Gear 1	1000	5500
Gear 2	1500	6000
Gear 3	2000	6500
Gear 4	2500	6500
Gear 5	3000	6500
Gear 6	3200	-

3.2. Fuzzy Logic based Velocity Control System

In this subsection, we present the components and design of the velocity control system. In this context, we have firstly employed a constant throttle value to obtain velocity-throttle relationship for acceleration. The obtained velocity response is given in Fig.3 while the corresponding gear and rpm change is given in Fig. 4. It can be observed from Fig.4 that the gearshift mechanism works as desired.

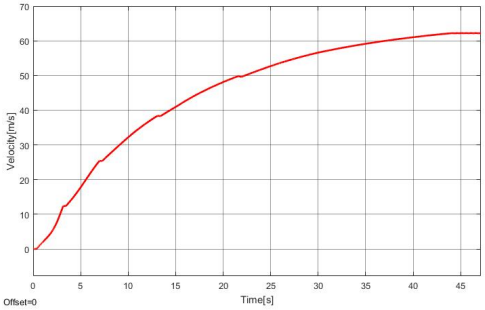

Fig. 3. Velocity graph for full throttle

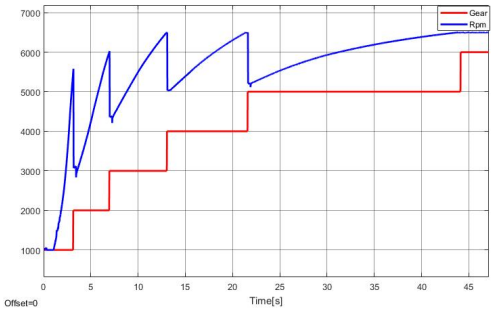

Fig. 4. Gear and rpm (x1000) change for full throttle

It can be observed from Fig.3, that the dynamics of the system show first order system characteristics. Thus, by using step response modelling techniques, we have obtained the following transfer function to represent the velocity dynamics.

$$G(z) = \frac{0.08843}{1 - 0.9986z^{-1}} \quad (1)$$

Then, by using this transfer function, we designed P controller which is $P(z) = 7.2761$. The resulting system response and control signal are shown in Fig. 5 and Fig. 6, respectively. It can be concluded that its performance is satisfactory. We have also handled the deceleration of speed, thus we designed a P type controller that regulates the brake input. In the game logic, we have observed that when the brake and throttle are employed simultaneously with the same values, the brake signal has a priority and thus we have designed a P controller with a smaller gain value ($P(z) = 1.2$) in comparison to the one for throttle.

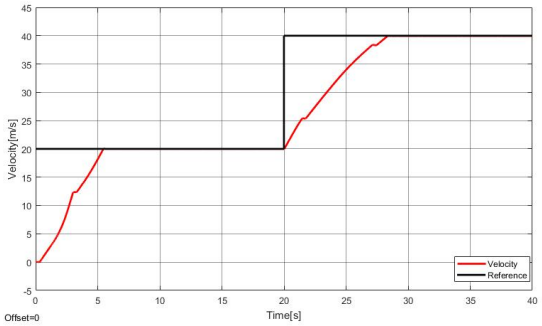

Fig. 5. Control System Performance

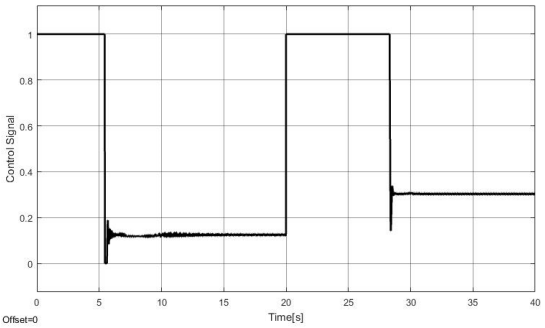

Fig. 6. Throttle control signal

For an efficient and smooth transition between the brake and throttle control system, we have used fuzzy logic based mechanism, namely the FLVR. The fuzzy inference uses the velocity error (the difference between the current speed and desired speed) and generates appropriate throttle and brake output values to be employed to the low-level controllers as shown in Fig.2. The FLVR is composed of five rules as tabulated in Table 4 and the corresponding membership functions (MFs) are given in Fig.7 which are define with linguistic values Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive Big (PB). The resulting input-output mappings of FLVR are given in Fig. 8.

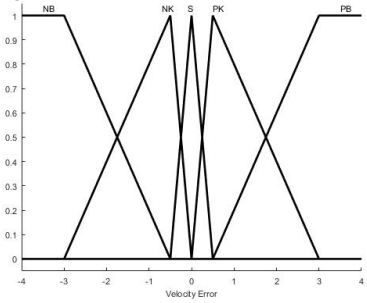

Fig. 7. The antecedent MFs of FLVR

Table 4. Rules

Input	Outputs	
Velocity Error	Throttle	Brake
NB	0	0.4
NS	0	0.1
Z	0	0
PS	0.2	0
PB	1	0

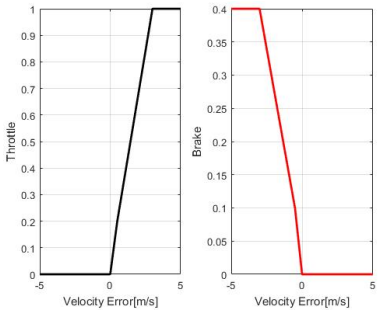

Fig. 8. Input-Output mappings of FLVR for throttle and brake

In order to show the efficiency of the FLVR mechanism, its performance is tested on "Motorway" track. The resulting step response and control signals for a varying speed trajectory different reference values are given in Fig. 9 and Fig. 10, respectively. It can be seen that the performance of the system is satisfactory.

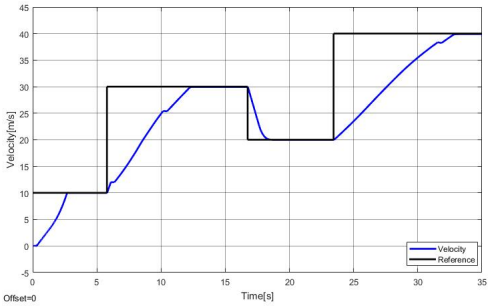

Fig. 9. Control System Performance with FLVR

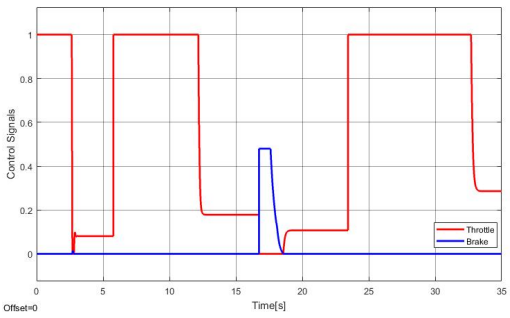

Fig. 10. The generated signals of the FLVR mechanism

3.3. Steering Control System

The design of the steering control system is usually accomplished in a cascade control structure where the heading and lateral (position) variables are controlled [3,5,7,8,12]. Thus, we will firstly design the heading control system and then one lateral control system. To design a controller for heading control, we examined the open loop system response and obtained the following transfer function.

$$G(z) = \frac{-0.00027z^2 - 0.00055z - 1.675e^{-05}}{z^2 - 1.968z + 0.9678} z^{-32} \qquad (2)$$

In this study, for the sake of simplicity, we preferred to simple controller that is defined as $P(z) = 1.5$. Then, to design position

(lateral) controller, the resulting closed loop transfer function is experimentally modelled using data collected from the game environment. The obtained transfer function is as follows:

$$G(z) = \frac{0.05403z + 0.03509}{z^2 - 1.27z + 0.2695} \quad (3)$$

For the above presented system, we have designed the following PD controller.

$$PD(z) = 0.0881 - 0.000881\frac{(z-1)}{(0.02z)} \quad (4)$$

Step response and control signal collected from the "Motorway" track are shown in Fig. 11 and Fig. 12, respectively. Here, we have employed a zero reference value to force the car to track the middle of the road.

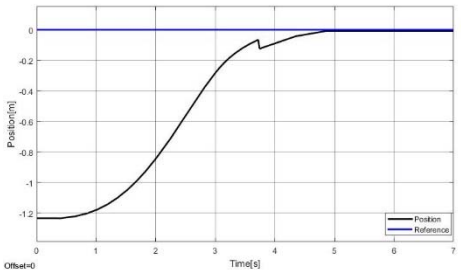

Fig. 11. Lateral error

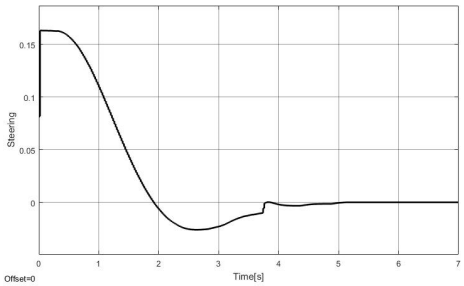

Fig. 12. Steering control signal

It has been observed that game performance of the PD controller is not satisfactory in the curvatures because of the nonlinear or unmodelled system dynamics on the system. Thus, we designed the following PID controller for a satisfactory system performance in curvatures.

$$PID(z) = 0.0881 + \frac{0.05(0.02z)}{z-1} - \frac{0.000881(z-1)}{(0.02z)} \quad (5)$$

However, an instant transition from the PD to the PID controller (or vice versa) might result with discontinuities and thus with stability issues. To overcome this problem, we employed a bumpless control transfer structure [13]. A control signal example is shown in Fig. 13 where it can be clearly seen the control signal does not inherit any discontinuities.

3.4. Fuzzy Logic based Position Regulation Mechanism

In order to enhance the car racing performance, it is necessary to detect curvatures (especially the sharp ones) ahead. To accomplish such a goal, we have defined a distance, which is proportional with velocity of the vehicle, to look ahead of the road and to detect the curvatures. This distance is defined as follows:

$$lookahead = const. + vehicle's\ velocity * coef. \quad (6)$$

In this context, we have modified the source code of the driver [11] to calculate the resulting value in real-time. The coefficients in Eq. (6) are set with respect to the vehicle that is used.

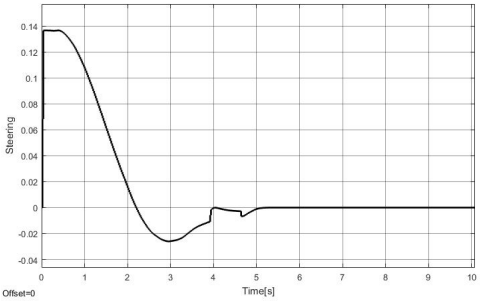

Fig. 13. Bumpless PD-PID steering control signal

In this study, we designed a FLPR mechanism to determine an appropriate reference position to be tracked by using the radius of current segment of track and next segment, which is calculated with lookahead value. Each input domain is defined with membership functions given in Fig. 14 which are defined with the following linguistic terms Negative (N), Zero (Z) and Positive (P). The rule base of the FLPR is given in Table 5 and the resulting control surface is given Fig. 15.

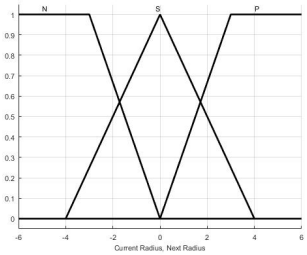

Fig. 14. The antecedent MFs of FLPR

Table 5. The rule table of the FLPR

		Next Radius		
		N	Z	P
Current Radius	N	-5	-2.5	0
	Z	-2.5	0	+2.5
	P	0	+2.5	+5

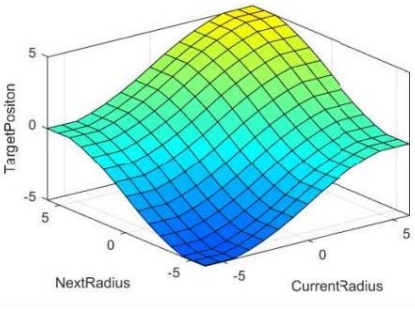

Fig. 15. Control surface of FLPR

It is worth to mention that, we have used also the detected curvature ahead to reduce the reference speed value to be employed. We have calculated the maximum allowable velocity (v_{ref}) on curvature by using the centripetal force relationship.

$$v_{ref} = \sqrt{rg\mu} \quad (7)$$

where g is the gravitational force, μ is the friction coefficient and r is the radius of the curvature. If the maximum allowable velocity is smaller than vehicle's velocity, the velocity is decreased.

4. Conclusions

In this study, we have presented a fuzzy logic based autonomous vehicle control system and examined its performance in TORCS game environment. As the main goal is to design an intelligent system such that racing car is able to compete the race autonomously, we have firstly modelled the car dynamics and then designed an intelligent control system composed of fuzzy logic and conventional control structures. In this context, we have firstly presented a fuzzy logic based throttle/brake control system has been designed such that the racing car is capable to accelerate/decelerate in a realistic manner as well as to drive at desired velocity. Then, fuzzy logic based positioning system is designed so that the car is capable to travel on the track, even in the presence of sharp curves. We have employed the proposed intelligent system to TORCS game via the Matlab/Simulink driver. The generated Simulink block diagram of the proposed intelligent system is given in Fig. 16. The resulting game performance of the developed fuzzy logic based autonomous vehicle control system can be observed from https://youtu.be/qOvEz3-PzRo.

5. Acknowledgement

This work was supported by Research Fund of the Istanbul Technical University. Project Number: 40988

6. References

[1] Wymann B., Espié E., Guionneau C., Dimitrakakis C., Coulom R., Sumner A., (2014). TORCS: The Open Racing Car Simulator, v1.3.6.

[2] Butz, M.V., Lönneker T.D., (2009). Optimized Sensory-motor Couplings plus Strategy Extensions for the TORCS Car Racing Challenge. *IEEE Symposium on Computational Intelligence and Games, pp.317-324.*

[3] Cardamone, L., Loiacono, D., Lanzi P.L., (2009). Learning Drivers for TORCS through Imitation Using Supervised Methods. *IEEE Symposium on Computational Intelligence and Games, pp.148-155.*

[4] *Loiacono, D., Prete, A., Lanzi, P.L., Cardamone, L., (2010). Learning to Overtake in TORCS Using Simple Reinforcement Learning. IEEE Congress on Evolutionary Computation (CEC'10).*

[5] *Lun, T.W., Leary, J., (2011). School of Environment and Technology MSC Automotive Engineering MEM47 Automotive Project. University of Brighton. http://itsuite.it.brighton.ac.uk/jjl/reports/torcs1.pdf.*

[6] *Muñoz, J., Gutierrez, G., Sanchis, A., (2009). Controller for TORCS created by imitation. IEEE Symposium on Computational Intelligence and Games, pp.271-278.*

[7] *Muñoz, J., Gutierrez, G., Sanchis, A., (2010). A human-like TORCS controller for the Simulated Car Racing Championship. IEEE Conference on Computational Intelligence and Games (CIG'10), pp.473-480.*

[8] *Onieva, E., Pelta, D.A., Alonso, J., Milanés, V., Pérez J., (2009). A Modular Parametric Architecture for the TORCS Racing Engine. IEEE Symposium on Computational Intelligence and Games, pp.256-262.*

[9] *Kemmerling, M., Preuss, M., (2010). Automatic Adaptation to Generated Content Via Car Setup Optimization in TORCS. IEEE Conference on Computational Intelligence and Games (CIG'10), pp.131-138.*

[10] *Preuss, M., Quadflieg, J., Rudolph, G., (2011). TORCS Sensor Noise Removal and Multi-objective Track Selection for Driving Style Adaptation. IEEE Conference on Computational Intelligence and Games (CIG'11), pp.337-344.*

[11] *Url-1<http://www.berniw.org/tutorials/robot/tutorial.html >, date retrieved 13.10.2016.*

[12] *McAree, O., (2014). VerifiableAutonomy/TORCSLink. https://github.com/Verifiable Autonomy/ TORCSLink.*

[13] *K. Åström, T. Hägglund, Advanced PID Control, ISA, Research Triangle Park, NC, August 2005.*

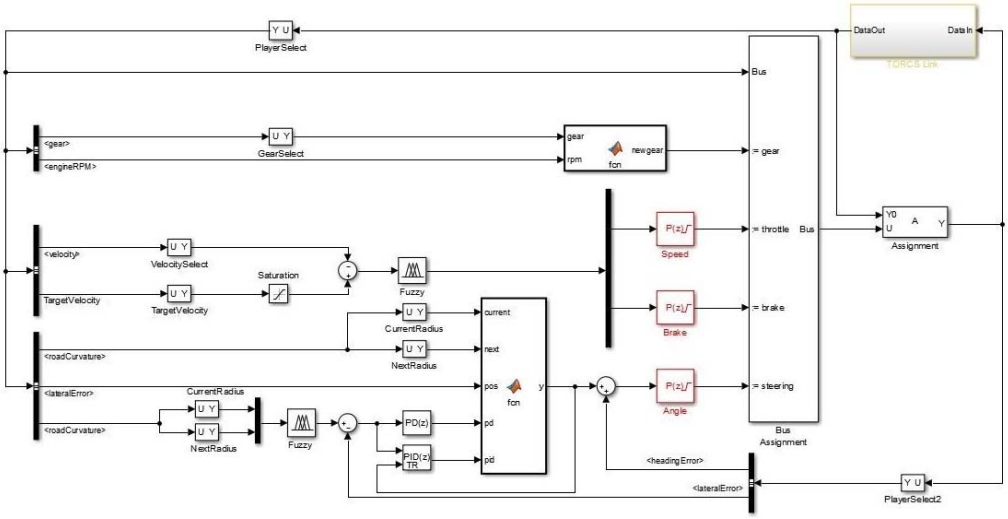

Fig. 16. Simulink driver of the designed fuzzy logic based autonomous vehicle control system