TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 65, Issue II, June, 2022

DEVELOPMENT OF A VIRTUAL REALITY SIMULATOR FOR AN AUTONOMOUS VEHICLE

Florin COVACIU

Abstract: *In this paper it is presented the development of a virtual reality simulator that is created for an autonomous vehicle. This vehicle has to go through a maze, being controlled by artificial intelligence algorithms. The autonomous vehicle can detect the walls of the labyrinth by means of three sensors that are attached to the front and side of it. The fuzzy system was chosen for the artificial intelligence algorithm, as input this system uses the three proximity sensors and, as outputs, the algorithm has the value of the angle at which the wheels rotate in front of the autonomous vehicle and the vehicle speed, which slows down near the maze walls.*
Key words: *fuzzy system, vehicle driving, virtual reality, autonomous, unity.*

1. INTRODUCTION

A car without a driver circulating on the streets seems like an incredible idea, but thanks to artificial intelligence such a car can be created in the not too distant future. The autonomous vehicle is beneficial for our transport system, having a great potential in safety and traffic congestion. By replacing human beings with computers in recent years, the automotive and technology industries have made significant leaps in computerization over the past few years. Autonomous vehicles have been created by some companies, which can drive on their own on existing roads and can navigate many types of roads and environmental contexts with almost no direct human input [1].

The main advantage of autonomous driving is the ability of the vehicle to make predictions and communicate with infrastructure and other vehicles, in addition to replacing the human driver with physical and emotional limitations. The main goal now is to correctly identify, control and manage the input parameters that come from different sensors for a usable development of the real world operating environment and the state of the vehicle. Defining and developing models to find correlations between existing or future artificial intelligence scenarios and available physical signals can be a great challenge [2].

The capacity benefits of self-driving level 4 and 5 have been defined by the Society of Automotive Engineers, which are quite a few, especially in terms of mobility and accidents throughout life. To help achieve the highest levels of automation as safely as possible, designers are now looking to incorporate artificial intelligence capabilities, and this will change every aspect of the driving experience. Google was one of the pioneers who paved the way for the legalization of autonomous vehicles through the system of autonomous cars owned by this company.

The paper is organized as follow: Section 2 present related work. Section 3 presents creating the environment in virtual reality continuing with creating the algorithm for control using artificial intelligence in section 4, followed by conclusions and references.

2. RELATED WORKS

By using an advanced driving system without human intervention that can be driven alone, it is known as an autonomous car. In order to make

Received: 04.04.22; **Similarities**: 23.05.22: **Reviewed**: 24.05./02.06.22: **Accepted**:20.06.22.

driving decisions, the car control system will take into account the traffic conditions and the previous knowledge of the system. In the development of such a system for autonomous driving one of the most important challenges is the complex interaction between the environment and the car. Several approaches based on artificial intelligence have been simulated to help develop such a car control system. Various methods have been proposed such as fuzzy logic [3-4], genetic algorithm [5], Kalman filter [6] and Petri net [7] to solve this problem.

In [8] it has been proposed a fuzzy logic based self-driving car control system and its implementation into the JavaScript Racer game. The design of the car control system that uses artificial intelligence has been made so that the car can complete the race without leaving the road in a short time. Another fuzzy logic based autonomous vehicle control system has been designed for TORCS [9] where it was assumed that the track of the race and lanes are known in advance which separates them from real-word application. In research [10] was introduced the route control of an autonomous car by fuzzy logic that allows, automatically the avoidance of obstacles in unknown environments.

Rosique et. al, describe a systematic review of the perception systems and simulators for autonomous vehicles (AV). This study presents the principle of operation and electromagnetic spectrum used to operate the most common sensors used in perception systems (ultrasound, RADAR, LiDAR, cameras, IMU, GNSS, etc.). For this purpose, here are described simulators for model-based development, the main game engines that can be used for simulation, robotics simulators and simulators used especially for AV. As main game engines that can be used for simulation are mentioned Unity 3D, Unreal Engine, Blender, and Cry Engine. Also, in this study are studied the advantages and disadvantages of each type of sensor that limits its application. For example, the Lidar 3D sensor has a high spatial resolution and high accuracy that makes it the perfect element for navigating and mapping the environment [11]. Covaciu et. al, have created a virtual reality simulator in which it is used sensors and artificial intelligence to control a drone [12]. Also, in [13] the authors created a virtual reality simulator in which they use artificial intelligence to help a patient in the rehabilitation process.

3. CREATING THE ENVIRONMENT IN VIRTUAL REALITY

The environment used for virtual reality is Unity 3D 2018.3.5f1 version [14]. This environment is cross-platform game engine developed by Unity Technologies. Unity engine supports 27 platforms, and this engine can be used to create both 3D and 2D games as well as simulations for desktops and laptops, mobile device, smart TVs and home consoles. Unity environment is the creator of the world’s most widely used real-time 3D (RT3D) development platform, giving developers around the world the tools to create rich, interactive 2D, 3D, virtual reality, and augmented reality experiences.

The virtual reality development environment called Unity uses two programming languages, namely C Sharp (C#) and JavaScript. With OpenGL graphics API, The Unity environment on all platforms that is used offers incredible graphics. [15].

The first step in the virtual reality environment is adding a terrain on which objects are placed. In the next step this terrain has been populated (Figure 1), after that the labyrinth (Figure 2) was created. The autonomous vehicle must go through this labyrinth without striking his walls. After the labyrinth was designed, the autonomous vehicle has been introduced (Figure 3).

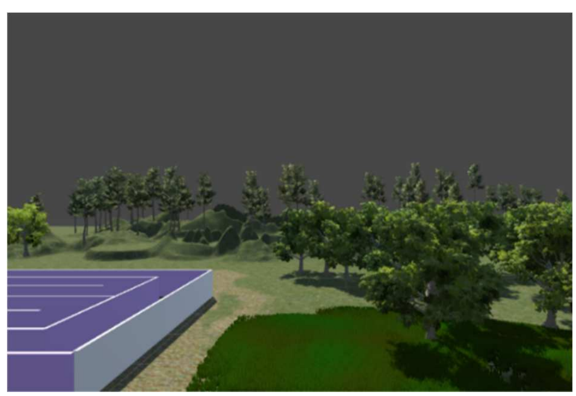

Fig. 1. Terrain population

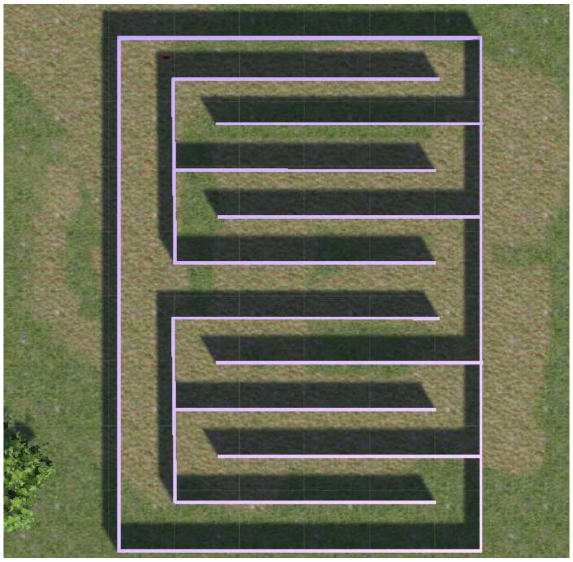

Fig. 2. Designing of the labyrinth

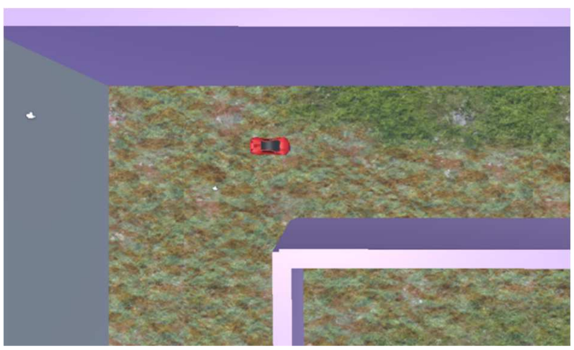

Fig. 3. Introducing of the vehicle into the labyrinth

In order to detect the walls three proximity sensors have been attached on this vehicle, namely: a sensor was attached to the front (Figure 4, 1) and two side sensors, respectively one sensor in the left side (Figure 4, 2) and one sensor in the right side (Figure 4, 3). To view the vehicle from multiple angles, three view cameras have been included, and with these cameras, vehicle can be seen from three angles (Figure 5), (Figure 6), (Figure 7).

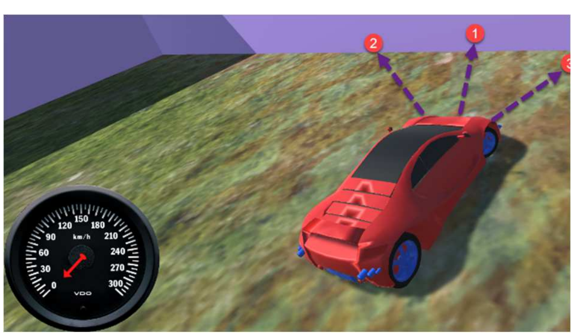

Fig. 4. Attaching the sensors

Fig. 5. The view camera 1

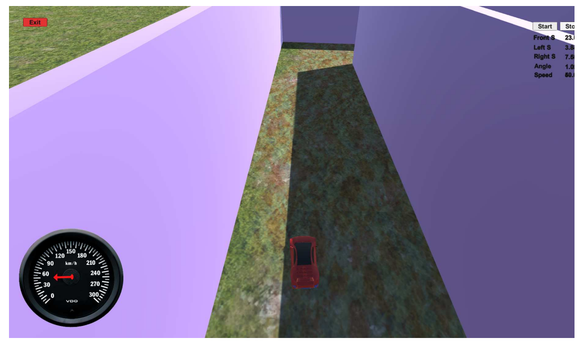

Fig. 6. The view camera 2

The view cameras of the autonomous vehicle it changes depending on the speed of the vehicle. After the virtual reality application has been opened, from the graphical user interface the "Start" button is clicked, the third viewing camera and the artificial intelligence algorithm being active.

From this moment, the vehicle begins to move in front, increasing its speed, the second viewing camera being activated, and through the sensors the vehicle detects the walls of the labyrinth so as not to hit it.

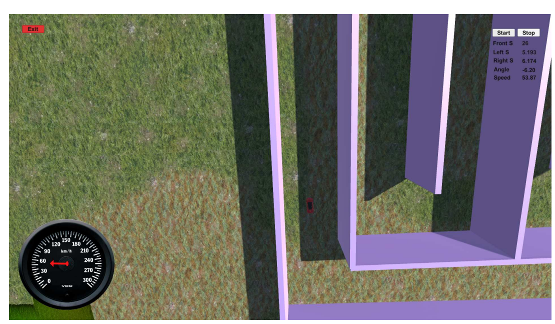

Fig. 7. The view camera 3

Near the end of the current corridor, the vehicle will have to enter the next corridor, and to perform this action the vehicle will slow down and turn, the first viewing camera being activated, these steps being repeated to go through the entire labyrinth. Pressing the "Stop" button while the vehicle traverses the labyrinth will stop the artificial intelligence algorithm, and the vehicle will hit the labyrinth wall, and to close the virtual reality application will press the "Exit" button on the user interface.

4. CREATING THE ALGORITHM USING ARTIFICIAL INTELLIGENCE

The fuzzy logic has been chosen to implement the artificial intelligence algorithm. Fuzzy logic systems have good outputs in response to inaccurate, distorted, incomplete and ambiguous inputs, being a method of reasoning similar with the human one. Fuzzy logic includes 0 and 1 as extreme cases of truth, but it also includes the different states of truth between them. The approach to fuzzy logic imitates how people make decisions, involving all intermediate possibilities between digital values Yes and No, such as: definitely Yes, maybe Yes, I can't say, possible Not, sure Not.

The solution used by fuzzy systems is to extend the classical sets by associating a function that returns a value between 0 and 1. Thereby the fuzzy sets have attached their membership functions which can be discrete or continuous. Fuzzy logic has a special value in automated control applications where it is difficult or impossible to develop a traditional control system.

The construction of the fuzzy logic system was done using an opensource framework implemented using C# language programing named AForce.NET which consists of several libraries like: AForge.Imaging (images processing), AForge.Neuro (neural networks), AForge.Genetic (genetic algorithms), AForge.Fuzzy (fuzzy logic), AForge.Robotics (robotics), AForge.MachineLearning (machine learning).

4.1 Creating the Control Algorithm for the Steering Angle of the Vehicle

In order to be able to rotate the front wheels of the vehicle at a certain left or right angle of rotation, an algorithm has been created in the fuzzy system. This fuzzy control system has three inputs because on front of the vehicle are attached three proximity sensors (a front sensor and two side sensors). Fuzzy sets corresponding to the lateral distances and the front for the three inputs were constructed as follows: two proximity sensors for determine the distances of the left and right side, which are attached to the front of the sides of the vehicle (Figure 8) and a proximity sensor input for the front distance attached to the front of the vehicle (Figure 9).

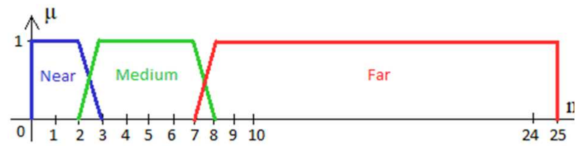

Fig. 8. Fuzzy sets for lateral distances

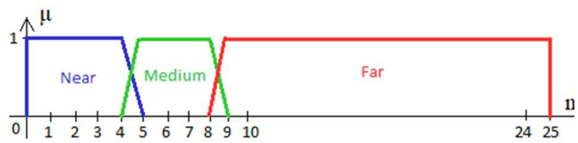

Fig. 9. Fuzzy sets for frontal distances

The three fuzzy variables which corresponding to the lateral distances are interpreted by the fuzzy system as is presented in Figure 9. If the lateral distance from the vehicle to the wall is between 0 and 2 meters, then the "Near" variable is consideration. The lateral distance from the vehicle to the wall is between 3 and 7 meters then the "Medium" variable is consideration, and if lateral distance from the vehicle to the wall is between 2 and 3, then either the "Near" variable or the "Medium" variable can be considered as probable. The lateral distance from the vehicle to the wall is between 8 and 25 meters then the "Far" variable is consideration, and if lateral distance from the vehicle to the wall is between 7 and 8, then either the "Medium" variable or the "Far" variable can be considered as probable. To remove the uncertainty when the lateral distance to the wall is between 2 and 3.

For output from the fuzzy system for the steering angle, we have the following sets as shown in Figure 10. The seven output fuzzy variables from Figure 10 are interpreted by the system as follows. Maximum turning to left ("VeryNegative") between the -60^o and -40^o angles. Middle turning to left ("Negative") between the -45^o and -5^o angles. Between the -

45° and -40° angles with probability may belong "VeryNegative" or "Negative" sets.

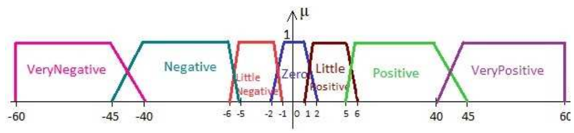

Fig. 10. Fuzzy sets for steering angle

4.2 Creating the Control Algorithm for the Speed of the Vehicle Displacement

When the vehicle is approaching to the walls of the labyrinth must reduce its speed in order to advance in good conditions, which is why a control algorithm fuzzy system has been developed. With three proximity sensors (Figure 4), the vehicle can sense the walls of the labyrinth. As output from the fuzzy system for the speed, we have the following sets as shown in Figure 11. The four output variables from Figure 11 are interpreted by the system as follows. Between -10 and -1 values is the negative speed, the vehicle can go back if it collides with the wall. Between values 1 and 10 is a low speed ("SmallSpeed"), and between values -1 and 1, the speed with probability may belong either to "NegativeSpeed" or "SmallSpeed" sets.

Between values 15 and 25 the speed is average ("MediumSpeed"), and between values 10 and 15, the speed with probability may belong either to "SmallSpeed" or "MediumSpeed". Between values 30 and 80 the speed is high ("GreatSpeed"), and between values 25 and 30, the speed with probability may belong either to "MediumSpeed" or "GreatSpeed" sets.

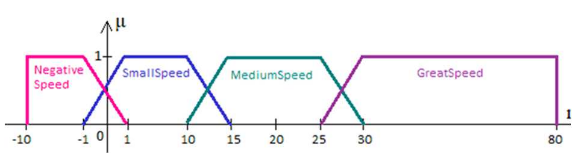

Fig. 11. Fuzzy sets for speed

5. CONCLUSION

The paper presents the use of artificial intelligence to control an autonomous vehicle in virtual reality. This autonomous vehicle must go through a labyrinth without hitting the walls of the labyrinth. For artificial intelligence a fuzzy system has been chosen. After the labyrinth has been developed in the virtual reality environment, the CAD model of the vehicle was introduced, attaching three proximity sensors to the front of the vehicle.

Using these sensors, the vehicle can detect distances from the walls of the labyrinth to navigate and avoid possible collisions. These distances obtained via the sensors are taken over by the intelligent module that returns the speed of the vehicle and its angle of travel. The AForge.NET framework developed in the C# programming language was used to implement the intelligent module. Among the packages of this framework, the following were used: AForge.Fuzzy for the development of the system based on fuzzy logic. Performed tests have validated the fuzzy system performances.

6. REFERENCES

[1] Daniel, J., Kockelman, K., *Preparing a Nation for Autonomous Vehicles: Opportunities, Barriers and Policy Recommendations*, Transportation Research Part A: Policy and Practice 2015, vol. 77, pp 167-181, DOI: 10.1016/j.tra.2015.04.003.

[2] Ilhan, I., *A Population Based Simulated Annealing Algorithm for Capacitated Vehicle Routing Problem*, Turkish Journal of Electrical Engineering and Computer Science 2020, vol. 28, pp. 1217-1235.

[3] Qaiyum, S., Aziz, I., Hasan, M. H., Khan, A. I., Almalawi, A., *Incremental Interval Type-2 Fuzzy Clustering of Data Streams using Single Pass Method*, Sensors 2020, vol. 20, no. 11, ISSN: 1424-8220.

[4] Panoiu, C., Panoiu, M., Muscalagiu, I., Iordan, A., *Visual Interactive Environment for Study the Power Electronics using PSCAD-EMTDC Simulation Program*, Computer Applications in Engineering Education 2010, vol. 18, no. 3, pp. 469-475, ISSN: 1061-3773.

[5] Tiwari, K. N., Waghmare, L. M., Krishnankutty, P., *Single Input Fuzzy Logic Controller Tuning for Steering Control of Autonomous Underwater Vehicle: Genetic Algorithm Approach*, Proceedings of Indian

Control Conference, Hyderabad, India, 4-6 January 2016; pp. 335-340, DOI: 10.1109/ INDIANCC.2016.7441156.

[6] Lin, M., Yoon, J., Kim, B., *Self-Driving Car Location Estimation Based on a Particle-Aided Unscented Kalman Filter*, Sensors 2020, Vol. 20, No. 9, pp. 1-16, ISSN: 1424-8220.

[7] López, J., Sánchez-Vilariño, P., Sanz, R., Paz, E., *Implementing Autonomous Driving Behaviours using a Message Driven Petri Net Framework*, Sensors 2020, Vol. 20, No, 2, pp. 1-26, ISSN: 1424-8220.

[8] Korkmaz, B., Etlik, U. B., Beke, A., Kumbasar, T., *Fuzzy Logic Based Self-Driving Racing Car Control System, Proceedings of the 6th International Conference on Control Engineering & Information Technology*, Istanbul, Turkey, 25-27 October 2018, DOI: 10.1109/ CEIT.2018.8751900.

[9] Armagan, E., Kumbasar, T., *A Fuzzy Logic Based Autonomous Vehicle Control System Design in the TORCS Environment*, Pamukkale University Journal of Engineering Sciences 2018, Vol. 24, no. 3, pp. 1-8, 1435-1442.

[10] Sordi Filho, A.L., de Oliveira, L.P., de Oliveira, A.S., Wehrmeister, M.A., *Simultaneous navigation, and mapping in an autonomous vehicle based on fuzzy logic*, Studies in Computational Intelligence 2017, 664, pp. 53-68.

[11] Rosique, F., Navarro, P. J., Fernández, C., Padilla, A., *A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research*, Sensors 2019, Vol. 19, no. 3, pp. 2-29, ISSN: 1424-8220.

[12] Covaciu, F., Iordan, A.E., *Control of a Drone in Virtual Reality Using MEMS Sensor Technology and Machine Learning*, micromachines 2022, Vol. 13, no 4, pp. 1-19, ISSN: 2072-666X.

[13] Covaciu, F., Pisla, A., Iordan, A. E., *Development of a Virtual Reality Simulator for an Intelligent Robotic System Used in Ankle Rehabilitation*, Sensors 2021, Vol. 21, no. 4, pp. 1-17, ISSN: 1424-8220.

[14] Thorn, A., Doran, J., Zucconi, A., Palacios, J., *Complete Unity 2018 Game Development*, Packt Publishing, UK, 2019, pp. 738, ISBN-13: 978-1789952865.

[15] Patil, P., Alvares, R., *Cross-platform Application Development using Unity Game Engine*, International Journal of Advance Research in Computer Science and Management Studies 2015, Vol. 3, no. 4, pp. 19-27, ISSN: 2321-7782.

Dezvoltarea unui simulator de realitate virtuală pentru un vehicul autonom

Rezumat: În această lucrare se prezintă dezvoltarea unui simulator de realitate virtuală care este creat pentru un autovehicul autonom. Acest autovehicul trebuie sa treacă printr-un labirint, fiind cotrolat prin intermediul algoritmilor de inteligență artificială. Autovehiculul autonom poate detecta peretii labirintului prin intermediul a trei senzori care sunt atașați în partea din față și lateral pe acesta. Pentru algoritmul de inteligentă artificială a fost ales sistemul fuzzy, ca intrare acest sistem foloseste trei senzori de proximitate și, ca ieșiri, algoritmul are valoarea unghiului la care sunt rotite roțile din fața vehiculului autonom și viteza vehiculului, care încetinește lângă pereții labirintului.

Florin COVACIU, PhD. Eng., Associate professor, Technical University of Cluj-Napoca, Department of Design Engineering and Robotics, Florin.Covaciu@muri.utcluj.ro, Phone: 0755566491, Muncii Blvd. No. 103-105, Cluj-Napoca 400641, ROMANIA.