Roborace TDP
cAyrton - BR.AI.M. Team

1st Werneck, Henrique Lima
UFABC Robotics
UFABC
Santo André, Brazil
werneck.henrique@outlook.com

***Abstract*—This paper describes the robotic project by the BR.AI.M. (BRazilian Artificial Intelligence Makers) team that led to the creation of the robot car that will be used in the RoboCar Race 2018.**

Keywords: robot car, artificial intelligence, autonomous vehicle, computer vision

I. Introduction

Robots and artificial intelligence are both evolving fast. Years ago, there was no such contact with robotic systems and working with autonomous machines was rare, even on places with easy access to advanced technology. Artificial intelligence is constantly being developed, always more powerful and impressive, improving along with the processing capabilities of the computers that run them. They are present everywhere and anyone is capable of coding an artificial intelligence. We are a long way from everyday robots in our daily lives, though, but the presence of these machines in our environment has increased and it is a promise for a better future. Among the most interesting subjects on robots, the autonomous vehicles, or robotic cars, draw the attention of even the most old-fashioned person that loves cars and driving.

The idea of having a car that can drive itself is simply too useful. The prospect of no more car accidents, being able to go to places without having to worry about traffic, or maybe going somewhere by yourself, even if you do not have a license. No need for drunk people to drive, fluid traffic even on peak hours. Unwasted money on all this problems could be reverted to the people in the shape of food or health care.

The Robocar Race offers a great opportunity for students to challenge themselves, get to know the technology, contribute to its growth, network and meet with people working on autonomous car and specially a great opportunity for learning.

The cAyrton project of the BR.AI.M. team was based on the graduation project of one of the members, modified after further testing and reimplementing the robot car systems.

II. Objectives

As the very first participation in a competition, the main objective was to present a complete functional, yet simple, solution for the task at hand: a robot car who could drive itself throughout the whole track.

The team enrolled at the Junior category, using a Raspberry Pi 3 to control the robot car. The car was made based on a ready-to-mount chassis, with the main focus of the project being the way it behaves on the track: the way it sees and analyses its environment and acts on its engines to steer and drive itself during the race.

It was not a major concern to improve lap times, nor a main focus to improve mechanical specifications for better race performance.

III. Robot Project

A. Physical Components

The project of the robot started by defining its physical parts. As the focus was making the car as simple as possible, it was decided to use a ready-to-mount platform, which came with the car chassis, engines and wheels. To connect the Raspberry Pi 3 with the engines of the car, it was used a L298N H-bridge. A simple webcam was used as the only car sensor, mounted on the back part of the car. Figure 1 shows the cAyrton assembled and ready.

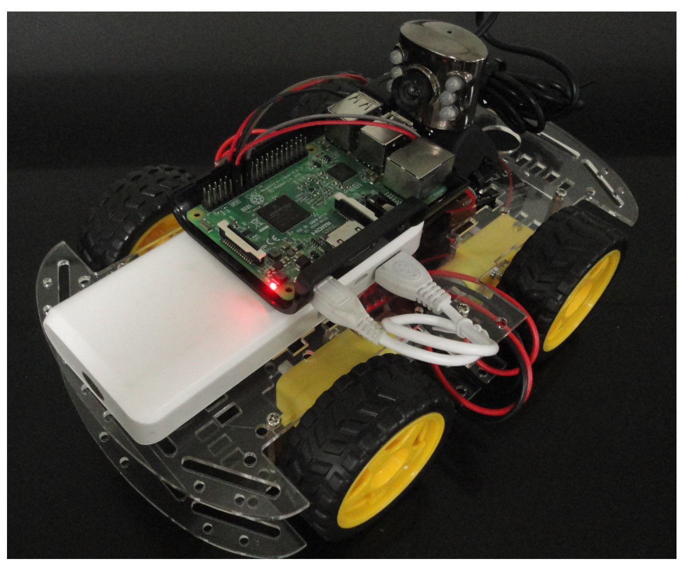

Fig. 1. cAyrton complete

B. Car Control System

The cAyrton, as defined by its physical structure, is a differential robot. Its control system was designed according to Figure 2, below:

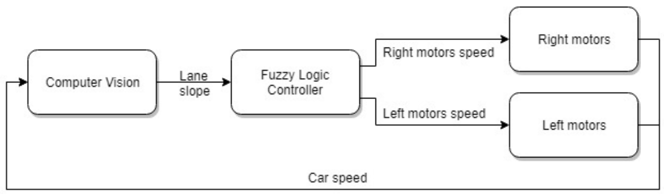

Fig. 2. cAyrton control system

The robot captures the view of the track with the camera, analysing the direction to which the track goes, based on the slopes of its left and right lanes. Then, the fuzzy artificial intelligence defines the speed of each motor, so that the robot keeps itself centered on the track. Finally, the motors are activated, the robot moves, and the loop is reset, with the cameras working on the current robot position once again.

All the vehicle's programs were programmed using Python. For the computer vision, the OpenCV library was used. For the artificial intelligence, it was developed a fuzzy controller, with the aid of the scikit-fuzzy library. Finally, the motors were fine controlled using the gpiozero library.

C. Computer Vision

The only sensor in the robot car is going to be its camera, as it is a requirement for the Junior category of the race. The vision system is solely based on the OpenCV library and its functions.

As stated above, the robot used for this race was based on the graduating project of one of the members. According to his project[2], the most efficient image treating technique for Canny edge detection tested on the robot car was applying a simple threshold to the camera capture, which among the tested functions (applying Canny edge detection to the original image capture, its black and white, simple and gaussian thresholds versions) proved to be the fastest, while also separating well the lane lines from the rest of the images captured.

As a first step to adapting the project for outdoors application, the same tests were run for the camera capturing the lane lines in the race track. Divergences regarding the environment lighting were expected, since the original tests were made in a laboratory with steady lighting. The team expected a change in the image treating algorithm, most probably using the gaussian thresholding treating, since it takes into account variations in the lighting from camera capture.

As for the information that would be captured from the camera, the vision system stayed the same: the lane lines were filtered from the rest of the image and their slopes were calculated. The final number passed to the fuzzy controller was the sum of the medium slopes of the right and left lanes in degrees, ranging from -90º to 90º.

D. Fuzzy Controller

The artificial intelligence that controls the robot car also kept being based on the original fuzzy logic applied to it. Originally, the fuzzyfication and the defuzzyfication functions had triangular shapes. The defuzzyfication method simply summed the result of the three possible lane orientations (straight, curve to the right and curve to left) for each motor based on the rules of the system.

Both the fuzzyfication and the defuzzyfication functions were changed. The new functions are gaussian distributions, which smooths the final outputs passed on to the motors, giving the robot car further control over its own pace.

The method for calculating the final output for each motor stayed the same, though. Simple sum of each defuzzyfication parameter for each motor's possible condition, based on the same set of rules.

The gaussian distribution kept the "perfect straight line condition" originally thought for the fuzzy artificial intelligence. It consists on the possibility of the car finding itself on a perfect straight line, where it should activate its motors at maximum speed.

IV. Robot Car Tests

A. Vision System

The vision system was first tested on the track from the car's view perspective, and only from the car perspective. Tests were run measuring image treating times for a fixed number of frames directly from a video, the same as the original work.

Simple thresholding still proved faster, but not trustable. The change in the environment lighting greatly decreased the quality of the treated image for edge detection, deeming it unusable for this application. Same was true for black and white, which depends heavily on the lighting. The gaussian thresholding was then chosen, since its characteristic to adapt the threshold value locally proved a good asset when using the camera in an outdoor testing ground. Applying the edge detection straight to the original image stills too noisy and error prone, so it was discarded.

The lane slope calculations were not adapted, since they seemed to work alright on the outdoor track.

B. Fuzzy Controller

The fuzzy controller system was tested with random inputs, much like the way it was tested on the original work. The logic worked for the first time, and the only thing that changed was the absolute speed that would change according to the new membership functions chosen for the artificial intelligence.

When testing the inputs, the "perfect straight", 0º (zero degree) input, return full throttle speed for both the right and the left motors. Same for the maximum values of right and left curves, 90º and -90º respectively.

Furthermore, the rest of the results of the simulated tests were always between the -1 to 1 range. This is important, once the motors can not be turned on at over 1 (100%), which express the maximum power output of the engines.

Finally, the last part of the process to recreate the original car in the form of the cAyrton was fixing the fundamental error that ocurred when it was taking curves. As soon as the car detected curves on the tracks limits, it would turn, even though it was not currently on the curve, but rather seeing it.

Commands to the motors processed by the cAyrton fuzzy controller are now added to a First In, First Out (FIFO) data structure. Basically, the robot has a list of commands to execute, and it must be executed in the order that the commands are produced. Thus, if the robot car is on a straight and sees a curve, it adds the command to take the turn to the list, but does not take the turn right away, since it has to complete the rest of the commands that are queued in front of the first curve command.

The commands in the FIFO structure are also cleared after the first curve command, so the car analyses the curve again from its new perspective.

C. Race Expectations

As stated before, it was not a concern to improve lap times. The team's main goal is for the robot to be able to go around the track, complete the whole course while detecting the lanes and acting on the camera's data with the precision correspondent to the calculated simulation of the fuzzy controller.

Other than that, provided the weather doesn't dramatically change after camera calibration, it is not expected errors to the image treating in a way that the robot car won't be able to recognize the lane markings.

ACKNOWLEDGMENTS

The BR.AI.M. team thanks UFABC Robotics and UFABC for providing the laboratory space and infrastructure of the university so we could conduct the researches and tests during the robot project and test phase.

Moreover, we would like to thank professor Luiz Celiberto for counselling and offering support when needed, and for taking much interest in our robot project for this race.

REFERENCES

[1]. IEEE. Manuscript Templates for Conference Proceedings. 2018. Available at: http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/IEEEtran_HOWTO.pdf.

[2]. Werneck, Henrique Lima. Técnicas de inteligência artificial e visão computacional para controle de carro robô. Santo André: UFABC, 2018.

[3]. Mordvintsev, Alexander; K, Abid. OpenCV-Python Tutorials Documentation. 2017. Available at: https://readthedocs.org/projects/opencv-python-tutroals/downloads/pdf/latest/.

[4]. scikit-fuzzy development team. scikit-fuzzy. Fuzzy Control Primer. Available at: https://pythonhosted.org/scikit-fuzzy/userguide/fuzzy_control_primer.html