Autonomous Vehicle Control System as a Mobile Robot by Artificial Neural Network

Nazmul Haque, Md Hasnat Riaz
Departement of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Bangladesh

Article Info

Article history:

Received May 12, 2017
Revised Jul 13, 2017
Accepted Jul 27, 2017

Keyword:

Backpropagation algorithm
Mobile robot
Neural networks
Prediction from experience
Supervised learning

ABSTRACT

In this paper we have presented the artificial neural network controlled car in mobile robotics and intelligent car systems. The motion control architecture of the robot is presented with an importance on the support and directing units. This uses neural network methods and the values fundamental its design is drawn. A robust neural control system using a model of the process is also developed.

Corresponding Author:

Md Hasnat Riaz,
Department of Computer Science and Telecommunication Engineering,
Noakhali Science and Technology,
University, Noakhali-3814, Bangladesh.
Email: Hasnat.riaz90@gmail.com

1. INTRODUCTION

An autonomous car is a self-driving car or driver-free car which sense its environment and can navigate without human input. Autonomous vehicles are capable of sensing their surroundings. It is capable of fulfilling the human transportation capabilities as like as the traditional car. It interprets sensory information to recognize suitable navigation paths, as well as obstacles and relevant signage. Based on sensory input some autonomous vehicles update their maps. It allows the vehicles to keep track of their position even when conditions change or when they enter unknown environments [1].This paper we have presented a self-driving car that would be able to drive from a start location to a goal location in a normal driving environment. To achieve the goal a small robot car would find the path to the destination and drive along the path avoiding any obstacle along the way.

Researchers are working on the development of the technology of automatically driven vehicles from the early nineteen hundred. Some quasi-autonomous demonstration systems date back to the 1920s and the 1930s. Since the 1980s, when Mercedes-Benz and Bundeswehr University Munich built a driverless car through the EUREKA Prometheus Project. Significant advances have been made in both technology and law relevant to autonomous cars. Many well-known companies and research organizations have developed working prototype autonomous vehicles, including Mercedes-Benz, General Motors, Continental Automotive Systems, Autoliv Inc., Bosch, Nissan, Toyota, Audi, Volvo, Vislab from the University of Parma, Oxford University and Google.

In 2010, four driverless electric vans was successfully driven 8000 miles from Italy to China. The vehicles were developed in a research project financed by European Union funding, by Vislab of the University of Parma, Italy. In July 2013 Vislab world premiered BRAiVE. It moved autonomously on a mixed traffic route open to public traffic. As of 2013, four U.S. states have passed laws permitting autonomous cars: Nevada, Florida, California, and Michigan. In Europe, cities in Belgium, France, Italy and the UK are planning to operate transport systems for driverless cars. Also in Europe, Germany, Netherlands, and Spain have allowed testing autonomous car in traffic. Finland is planning on passing a law before year 2015 [2].

2. RESEARCH METHOD

The ALVINN system captures video frames and passes them to neural networks which have been trained by watching a human drive in similar environments. Then for the new video frame the trained neural network will predict how to steer to stay on the road ahead. We needed a system which could operate in two modes:

a. Record-The system captures video frames and the control input from a human driver and records them for later use to train the neural network.
b. Drive-Continuously captures video frames and passes them to a trained neural network which makes predictions about how to drive.

2.1. Design:

The system should be able to record video from the car. Then it should pass the frames to a neural network to control the car's steering. To do this we can mount an Android phone on the car to gather video frames and make neural network predictions locally on the device, hacking the car to be controlled from the onboard phone using an Arduino based Android ADK board, with data recorded and transferred to a computer for training. ADK boards require enough ability to keep the phone powered over USB and the weight of additional batteries would make this cheap and cheerful car struggle. Instead, I have chosen a design which barely modified any of the components involved. The system consists of:

a. Android phone-installed on the car, takes video frames of the road ahead using its built-in camera at ~15 fps. An application running on the phone connects to a server running on a laptop computer via wifi and streams 176x144 grayscale video frames across the connection.
b. Computer-"Driver" which is a little Java application acts as TCP server. It receives streamed image frames from the phone and a user interface allowing a human driver to control the car with the cursor keys or mouse. In record mode, disk stores the video frames, labeled with the current control input coming from the human driver. These labeled frames are used to train the neural networks in a separate environment on the computer. Trained parameters are saved out to files which are in turn read by the Driver application.
c. Arduino Uno-is a microcontroller connected via USB of a computer and simulate key presses on the car's radio controller PCB.
d. The Neural Network-The Neural Network can be trained using an octave program [3]. The below figure presents the architecture of the network that I have used.

Figures are presented center, as shown below and cited in the manuscript.

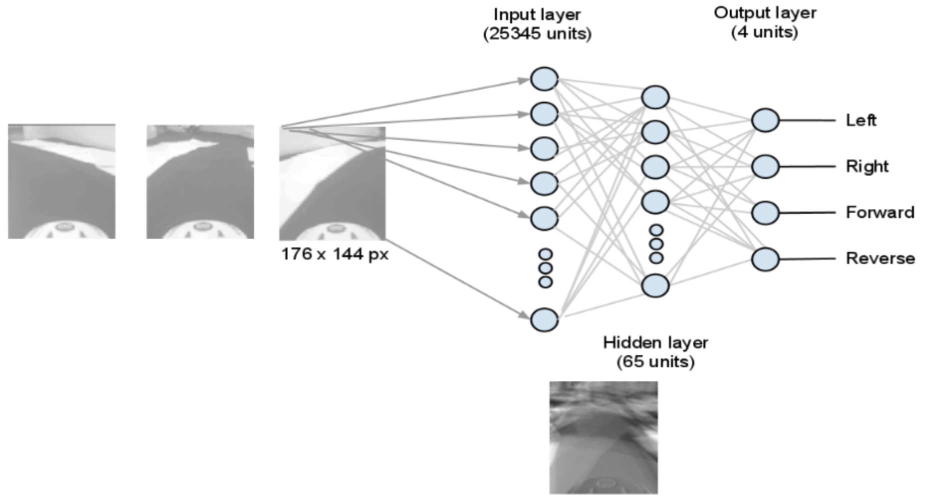

Figure 1. Neural network

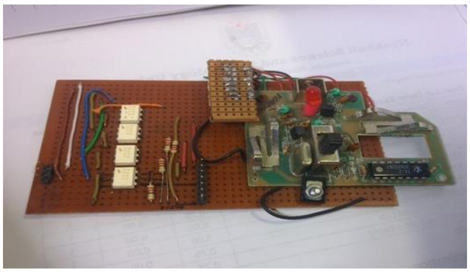

Figure 2. RC controller

Figure 3. Car model

Control Bit: Control Bytes are constructed in Hex like this:

Where X is the direction and Y is range of 16 different speed. The speed (Y) ranges from 0-F in hexadecimal, Table 1 shows the different drive commands. For example: If you want to drive left pretty darn quickly, just send a 0110. If you want to backup to the right slowly, send a 0101. All commands are sent as a 4 bit. Tables are presented center, as shown below and cited in the manuscript.

Table 1. Control Bit

Code	Move
0000	Stop
0100	Straight / Forward
1000	Straight / Backward
0010	Left / No Drive
0001	Right / No Drive
0110	Left / Forward
0101	Right / Forward
1010	Left / Backward
1001	Right / Backward

2.4. Road Model:

Here the road model is made of Back and White Paper. The edge and corner is designed perfectly and make sure the camera view angle is perfect. Almost any kind of shape road can be design for this Autonomous Car. Figures are presented center, as shown below and cited in the manuscript.

Figure 4. Road model

2.5. Development Phase:

The total development process of my Self-driving I-racer could be divided into 4 phases. Based on the 4 phases there are 4 prototypes:

a. Prototype 1: Towards the beginning of the project I used a totally different type of robotic car kit as my car. It was a line follow vehicle. The maneuvering system was totally different from traditional cars. It maneuvered like a rank, so it could turn to any direction staying in the same position. Later I moved on to a more car like robotic car kit, the I-racer. Figures are presented center, as shown below and cited in the manuscript.

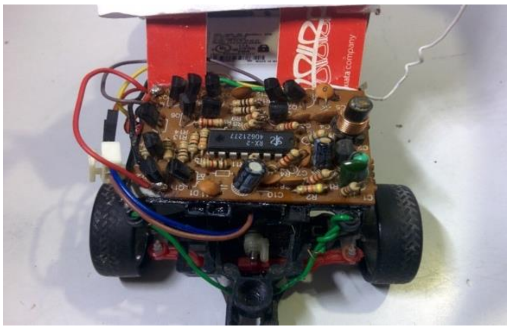

Figure 5. Prototype 1

b. Prototype 2: The setup of the prototype car used in this research is quite simple. In the earlier setup a small breadboard was mounted on top of the I-Racer. The breadboard was used as a platform as well as a means for connecting the battery and motors with wires. This prototype was used to detect obstacles on the pathway of the car.

c. Prototype 3: In the later setup an Android phone because of the availability of the integrated CAMERA, Wi-Fi, and Bluetooth and various sensors. The magnetic field created from the magnets in the motor interfered with the geomagnetic field sensor which resulted in faulty compass readings. This resulted in a more stable compass reading.

d. Prototype 4: The development of this phase is basically the combination of Prototype 2 and 3. With a few design changes I was able to mount both the obstacle detection module and the main processing module, the Android Phone, on top of the small chassis of a robotic car kit, the I-racer. It was observed that, putting the phone on top of the car vertically. This also saves up a lot of space for the other hardware, like the raspberry pi and its power supply unit, which are also placed vertically in order to save space. The design outlook is as the following images:

Figures are presented center, as shown below and cited in the manuscript.

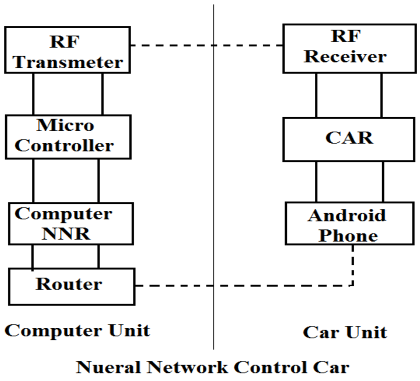

Figure 6. Nueral network control car

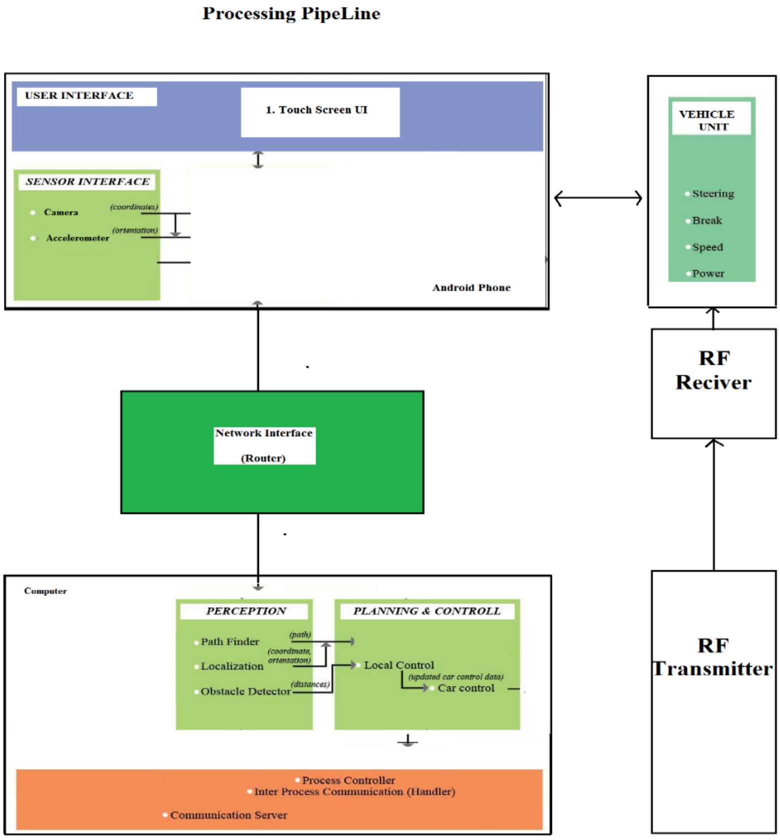

Figure 7. Processing pipeline

The overall work flow of the self-driven car could be represented sequentially like this:

a. First the computer software is open.
b. Android Phone and Computer is connected with IP Address which is Apps Input.
c. The app shows the current position of the car by its camera.
d. The car can reach its destination by iterating through each node, eventually leading to the destination.
e. There is an angle between two coordinates. This is called the ‘heading’.
f. The Car also has a heading. It is the direction it is going on.
g. The goal of the car is to try and maintain the heading between the current position and the next node.
h. While traveling the course, if the car detects any obstacle, its instant response is to move away towards the next best direction and avoid it.
i. So, though the main goal is to reach the destination, each node can be considered a current goal. Upon reaching the current goal, the next node becomes the goal; so recursively the car iterates through each current goal to the final goal.

2.6. Steps:

1. Find Current Position ‘A’: The global position of the car is determined by the Global Positioning System (GPS) of the Android Phone. Google Map is used to visualize the position. Google Maps API V2 has been used here. The current position is a coordinate consisting of a latitude and a longitude value.
2. Set Destination ‘B’: The destination can be set from the user interface, by clicking on a point on the map. The destination also denotes a coordinate.
3. Find Path from A to B: The path from A to B is determined by the Google Directions API. The path from A to B is actually a set of coordinates that form a line, thus the path. These coordinates could be considered nodes.
4. Drive from A to B (Control): The control system could be divided into 2 categories:
   a) Global Control: Figure 8, the priority is to reach destination. So, the actions taken during this process focused on achieving that. One of the key component is calculating the heading. Once the path is set, the car starts its journey from the first node, which is the current location of the car. The immediate next node is the current goal for the car. The heading direction between the car’s

current position and the current goal is calculated frequently as the car moves towards it. This is its desired heading. The car has its own heading direction. It constantly tries to keep its own heading towards the desired heading by steering towards it. If it is driving in the right direction, eventually it will reach the current goal. Once the current goal is reached, the next node becomes the current goal and the car drives towards it. This way all nodes are traversed and the final goal is reached. Figures are presented center, as shown below and cited in the manuscript.

Figure 9 the steering angle of the I-Racer is fixed, the car overshoots. Meaning, by the time it reaches the desired heading angle and straightens the wheels it may have gone slightly off track. And this keeps happening from time to time, making the car sway a little left and right. So the traveled path isn't always a straight like. This problem could be solved by gradually decreasing the steering angle as the car's heading gets closer to the desired heading. But in order to do that a car of which the steering wheel can be controlled by the degree is needed.

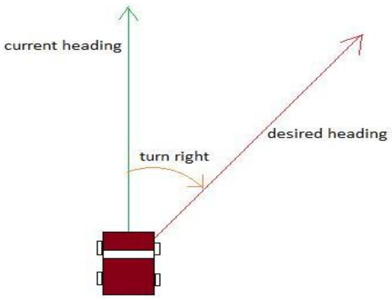

Figure 8. Global control

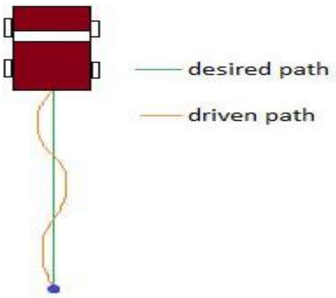

Figure 9. Wavy track

b) Local Control: During the Local Control phase, the car is focused on avoiding obstacles. It can override the Global control command, even if it means to get away from the destination; once obstacle avoided, it goes back to course of Global control. The local control includes the Obstacle Detection System.

2.7. Algorithm

The AI for the self-driving car was implemented on Java platform. An android app was made to interface between the phone and the I-Racer. The driving algorithm of the car can be represented by the following processing pipeline and flow chart as shown in Figure 10:

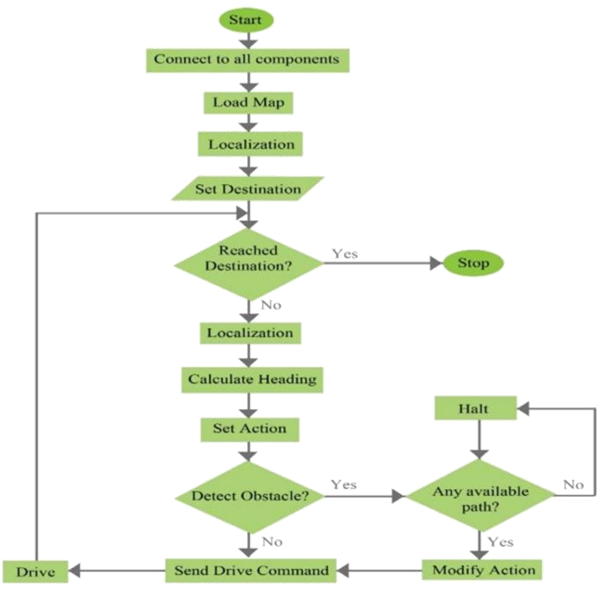

Figure 10. Algorithm flowchart

3. RESULTS AND ANALYSIS

Several test drives have been conducted to find the output of the program. Which led to a lot of bug fixes. The output of the program depended on a few external things, like- electromagnetic field from the motor of the I-Racer, the GPS signal quality, the phone's operator signal. Because of these thing the output fluctuated from time to time, causing the car to drive in a wrong way; that's why outdoor testing didn't give any satisfactory result, other than the fact that for the initial time while it is on course, the car is able to steer towards the desired heading keeping it on track. And the wavy track was noted.

The following image shows the inaccurate GPS readings. The desired path is marked by the red lines. The orange marker marks the starting location, and the blue marker marks the destination. When manually travelled down the path, the GPS signals fluctuated a lot. The green markers show the GPS track that was obtained.Indoor test drives have also been conducted. The target was to test the obstacle detection functionality of the car. It was able to move freely indoors, avoiding obstacles, even coming out of corners.

Though fixed sensors handled the job of obstacle detection pretty well, the car still had some blind spots. The total 360o surrounding was not covered. This can be overcome by having a rotating sonar sensor, or more expensive sensors that take a rotational 360o scan of the environment. Storing this surrounding information would help create a map of the environment the car is in, thus provide better Localization. And with more precise sensing of the velocity and displacement of the car's movement, the car could be able to perform more precise driving while Global Control is uncertain. Google car uses a very expensive Lidar to achieve better Localization. It senses the surrounding and stores all the data creating a map. So, the next time the car comes down the same road, it knows how to react.

4. CONCLUSION

The aim of my work was to design an autonomous vehicle driving system, to develop the artificial intelligence that would be able to make a driver less automobile safe enough to drive on the road. The goal was to develop a program that would run in an android operated environment and will be able to control the vehicle, thereby taking the responsibilities of the driver, providing a more manageable control over it. And a swarm of these autonomous vehicles could just be the solution to our traffic problems and traffic jam, reduce the number of road accidents and nevertheless save the time that people engage themselves while driving for some other task.

REFERENCES

[1] Autonomous car. (2015, August 18). In Wikipedia, The Free Encyclopedia. Available: https://en.wikipedia.org/w/index.php?title=Autonomous_car&oldid=676696225
[2] History of autonomous car. (2015, August 6). In Wikipedia, The Free Encyclopedia. Available: https://en.wikipedia.org/w/index.php?title=History_of_autonomous_car&oldid=674883593
[3] David Singleton (2013, January 11). How I built a neural network controlled self-driving (RC) car. Retrieve from: http://blog.davidsingleton.org/nnrccar

BIOGRAPHIES OF AUTHORS

Nazmul Haque
Undergraduate student
Department of Computer Science and Telecommunication Engineering
Noakhali Science and Technology University,
Noakhali 3814, Bangladesh.
Email: nazmul.cste07@gmail.com

Hasnat Riaz
Lecturer
Department of Computer Science and Telecommunication Engineering
Noakhali Science and Technology University,
Noakhali 3814, Bangladesh.
Email: hasnat.riaz90@gmail.com