AUTONOMOUS VEHICLE CONTROL USING IMAGE PROCESSING

A Final Year Project Report

Presented to

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

Department of Mechanical Engineering

NUST

ISLAMABAD, PAKISTAN

In Partial Fulfillment

of the Requirements for the Degree of

Bachelors of Mechanical Engineering

---

by

Muhammad Arqab

Muhammad Aibak Arshad

June 2018

EXAMINATION COMMITTEE

We hereby recommend that the final year project report prepared under our supervision by:

Muhammad Arqab	NUST – 2014 – 32445
Muhammad Aibak Arshad	NUST – 2014 – 32313

Titled: "**Autonomous Vehicle Control using Image Processing**" be accepted in partial fulfillment of the requirements for the award of **Bachelors of Mechanical Engineering** degree.

Supervisor: Dr. Omer Gilani	______________ Dated:
Committee Member: Dr. Mohsin Jamil	______________ Dated:
Committee Member: Dr. Emad Uddin	______________ Dated:

______________________ ___________

(Head of Department) (Date)

COUNTERSIGNED

Dated: ________________ _________________________

(Dean / Principal)

ABSTRACT

The manufacturing and development of Autonomous Vehicles paves the way for a sustainable environment through minimization of carbonic emissions by increasing vehicle performance and efficiency through effective speed and steering control. The integration of autonomous vehicle concept with alternative power sources e.g. electric cars is now the pivotal research aspect for sustainable future development. Hence, this project serves as an initial benchmark for potential future research at the National University of Sciences and Technology (NUST), Pakistan.

Furthermore, autonomous vehicle control will help in reducing the number of road accidents occurring due to drivers' fatigue and negligence. The project is to be implemented in limited resource to provide a low-cost solution. Hence this research will provide an optimum performance solution, enabling more cost to be spent on the efficiency. This aspect is more important in developing countries, and will pave the way for future development and sustainability.

ACKNOWLEDGEMENTS

We thank God Almighty, for being able to successfully complete the project in the allocated timeframe.

We would like to express our deepest appreciation to our project supervisor, Dr. Omer Gilani, who continued to support us throughout the duration of the project, and without whose guidance this endeavor would have not been possible.

We also thank our committee members, Dr. Mohsin Jamil and Dr. Emad Uddin, for their time and support, and for their valuable guidance whenever we needed it. We appreciate the efforts of Dr. Abdur Rehman, who as the project coordinator was always willing to provide genuine help and support and timely address our concern and queries.

Lastly, we thank and appreciate the department faculty, especially Dr. Hasan Sajid and Dr. Sami Ur Rehman Shah, who were always willing to address our queries regarding the project and provide their feedback. We appreciate the efforts of our diligent course instructors, who enabled and inspired us to pursue knowledge and skills necessary for the project.

ORIGINALITY REPORT

FYP FINAL

ORIGINALITY REPORT

10%	5%	8%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Coventry University Student Paper	1%
2	"4th European Conference of the International Federation for Medical and Biological Engineering", Springer Nature, 2009 Publication	<1%
3	S M Veres, L Molnar, N K Lincoln, C P Morice. "Autonomous vehicle control systems — a review of decision making", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2011 Publication	<1%
4	B.A. Guvenc, L. Guvenc. "Robust two degree-of-freedom add-on controller design for automatic steering", IEEE Transactions on Control Systems Technology, 2002 Publication	<1%
5	Khalid Isa. "Experimental Studies on Dynamics Performance of Lateral and Longitudinal	<1%

Control for Autonomous Vehicle Using Image Processing", 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, 07/2008
Publication

6	hal.inria.fr Internet Source	<1%
7	oa.upm.es Internet Source	<1%
8	Coşkun, Fuat, Özgür Tuncer, Elif Karslıgil, and Levent Güvenç. "Vision System Based Lane Keeping Assistance Evaluated in a Hardware-in-the-Loop Simulator", ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis Volume 3, 2010. Publication	<1%
9	journals.sagepub.com Internet Source	<1%
10	Springer Series in Perception Engineering, 1992. Publication	<1%
11	digital.lib.washington.edu Internet Source	<1%
12	www.ifac-papersonline.net Internet Source	<1%

13	Li, Dongjie, Weibin Rong, Lining Sun, Bo You, Yu Zou, and Wanzhe Xiao. "Fuzzy Control and Connected Region Marking Algorithm-Based SEM Nanomanipulation", Mathematical Problems in Engineering, 2012. Publication	<1%
14	Submitted to University of Greenwich Student Paper	<1%
15	www.scriptiebank.be Internet Source	<1%
16	Submitted to Eastern Mediterranean University Student Paper	<1%
17	staff.polito.it Internet Source	<1%
18	Enrique Onieva, Javier Alonso, Joshue Perez, Vicente Milanes, Teresa de Pedro. "Autonomous car fuzzy control modeled by iterative genetic algorithms", 2009 IEEE International Conference on Fuzzy Systems, 2009 Publication	<1%
19	O.D. Altan, H.K. Patnaik, R.P. Roesser. "Computer architecture and implementation of vision-based real-time lane sensing", Proceedings of the Intelligent Vehicles `92 Symposium, 1990	<1%

Publication

20	www.path.berkeley.edu Internet Source	<1%
21	ulsrl.org Internet Source	<1%
22	Submitted to SASTRA University Student Paper	<1%
23	Gupta, Neerja, Abhishek Singhal, J. K. Rai, and Rohit Kumar. "Path tracking of Automated Guided Vehicle", 2014 Seventh International Conference on Contemporary Computing (IC3), 2014. Publication	<1%
24	Evangelia Lampiri. "Sensor anomaly detection and recovery in a nonlinear autonomous ground vehicle model", 2017 11th Asian Control Conference (ASCC), 2017 Publication	<1%
25	Deshpande, David P., Ernest L. Hall, Juha Röning, Kovid Mathur, and Ernest Hall. "", Intelligent Robots and Computer Vision XXVII Algorithms and Techniques, 2010. Publication	<1%
26	Boissonade, F.M.. "Methods for recording the jaw-opening reflex to tooth-pulp stimulation in awake cats", Journal of Neuroscience Methods,	<1%

199106
Publication

27	Yafei Wang, Binh Minh Nguyen, Hiroshi Fujimoto, Yoichi Hori. "Multi-rate Kalman Filter Design for Electric Vehicles Control based on Onboard Vision System with Uneven Time Delay", IFAC Proceedings Volumes, 2013 Publication	<1%
28	Submitted to University of East London Student Paper	<1%
29	Submitted to Harper Adams University College Student Paper	<1%
30	Kazuhiro Nakano, Ken-ichi Takizawa, Yasuo Ohtsuka. "The Application of Neural Network Systems to the Evaluation of the Quality of Fruits", IFAC Proceedings Volumes, 1995 Publication	<1%
31	Cruz Tolentino, Jos Alberto, Alejandro Jarillo, Luis Enrique Ramos Velasco, and Omar Arturo Domnguez Ramrez. "Wavelet PID and Wavenet PID: Theory and Applications", PID Controller Design Approaches - Theory Tuning and Application to Frontier Areas, 2012. Publication	<1%
32	Gavaskar Vincent, T. Sasipraba. "An efficient routing algorithm for improving the QoS in Internet", INTERACT-2010, 2010	<1%

Publication

33	iosrjournals.org Internet Source	<1 %
34	dariopiga.com Internet Source	<1 %
35	www.science.gov Internet Source	<1 %
36	doras.dcu.ie Internet Source	<1 %
37	eprints.qut.edu.au Internet Source	<1 %
38	hal-polytechnique.archives-ouvertes.fr Internet Source	<1 %
39	Yu, Wei, Emmanuel Collins, and Oscar Chuy. "Dynamic Modeling and Power Modeling of Robotic Skid-Steered Wheeled Vehicles", Mobile Robots - Current Trends, 2011. Publication	<1 %
40	Submitted to University of Central Lancashire Student Paper	<1 %
41	Tsung-Ying Sun. "The robust and fast approach for vision-based shadowy road boundary detection", Proceedings 2005 IEEE Intelligent Transportation Systems 2005, 2005 Publication	<1 %

42	M. Murayama. "Computer-controlled Ntsc Single-chep Ls1 For Color Tv", IEEE 1993 International Conference on Consumer Electronics Digest of Technical Papers, 1993 Publication	<1%
43	A Kilin, P Bozek, Yury Karavaev, A Klekovkin, V Shestakov. "Experimental investigations of a highly maneuverable mobile omniwheel robot", International Journal of Advanced Robotic Systems, 2017 Publication	<1%
44	Zhao, Pan, Jiajia Chen, Yan Song, Xiang Tao, Tiejuan Xu, and Tao Mei. "Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID", International Journal of Advanced Robotic Systems, 2012. Publication	<1%
45	Kafrouni, Elie, and Mohammed Chadli. "Application to Vehicles", Command-Control for Real-Time Systems Chadli/Command-Control for Real-Time Systems, 2013. Publication	<1%
46	Onieva, E.. "Automatic lateral control for unmanned vehicles via genetic algorithms", Applied Soft Computing Journal, 201101 Publication	<1%
47	Alexander Meystel. "Multiresolutional	

	Representation and Behavior Generation: How Does It Affect the Performance of and Planning for Intelligent Systems", Planning in Intelligent Systems, 03/03/2006 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Vehicle autonomy has long since been an area of wide-scale research, with the limitless potential of development of advanced control mechanism in order to reduce driving accidents and increase vehicle drive efficiency. Taking into account driver's safety and comfort, the design and implementation of an autonomous control system on a practical level involves careful deliberation of implemented methodology. Although autonomous vehicles without humans are also widespread, for example Autonomous Ground Vehicles (AGVs) which can traverse a specific route without human intervention [1], much of the focus of research in this area is on vehicles for carrying passengers. In this regard, there has been considerable research in the area of Automated Highways Systems (AHS), which is a vehicle and road-based system designed for a vehicle to function autonomously on highways [1]. Some existing system incorporates cruise control, where the driver does not have to adjust the speed of the car. The system relieves the driver of this, by the use of a feedback loop controller for keeping the speed of the vehicle constant. Henceforth, there has been a considerable focus in development towards the design of an advanced cruise control, capable of automating the steering control in addition to the speed control.

An autonomous vehicle is an intelligent unit, capable of environmental perception, vehicle positioning and orientation, and based on the sensory feedback, able to determine the optimum trajectory in terms of intelligent decision making. The ANSI general reference architecture [2] defines intelligent systems in terms of four basic aspects, namely *World Model*, *Behavior Generation*, *Value Judgment* and *Sensory Processing*. The ideal information flow from an environment incorporating the four aspects defines the role of an intelligent system, as described in Figure 1.

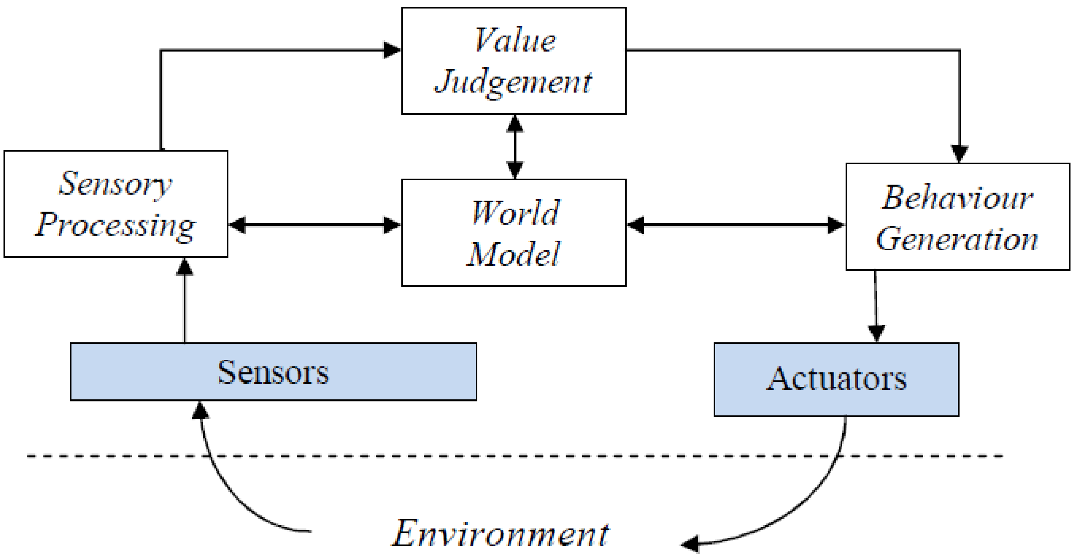

Figure 1: ANSI general reference structure for intelligent systems [2]

Image processing has been one of the pivotal strategies towards the development of autonomous vehicle control systems [2,4,5], with neural networks as one of the most widely used among them [13,14,15]. A simple neural network based control system has been the pivotal area of research in this area.

1.1 – MOTIVATION OF THE PROJECT

Over the years, due to a considerable increase in the number of vehicles, road accidents become more common. Over the last 4 years, the total number of road accidents in Pakistan have increased by almost 1000 from 2014 to 2016, and currently continue to increase [3]. Driver's safety has been one of the major concerns towards the development of safe driving systems all over the world, and autonomous vehicle control has been one of the fields which are very promising in this regard.

Furthermore, as the number of vehicles on the road continue to increase, so does the traffic congestion and blockage. Due to a limited number of routes and poor route planning, added by the effect of inefficient driving, the traffic blockage is a major pertaining issue. The carbonic emissions from the stationary traffic are harmful for the environment. One of the underlying principles of autonomous vehicle control is the efficiency of the driving system. Independent of human input, the autonomous control coupled with navigation system can offer decent route planning to avoid traffic congestions. Furthermore in case of one, the vehicle output drive can be automatically adjusted when the vehicle is not moving, so as to minimize the carbonic emissions from the vehicle when stationary. Much of the research on autonomous vehicle control is prevalent on electric vehicles, providing the incentive for the people to purchase electric vehicles over conventional IC Engine vehicles – a step towards the reduction in environmental pollution.

Autonomous Vehicle Control also has a definite advantage in the utilization of driving time towards some more productive activity. The concept carries great appeal for the professional workers and businessmen on tight schedule, who can utilize the time, but within the vehicle, saved from driving in work-related activities and hence increasing the average work output and taking a step towards a more productive community. In this regard, concepts such as the use of the vehicle as an office or mobile platform are under study for viability. Another advantage which autonomous vehicle control offers is for the elderly and disabled – people who have difficulty driving or have limitation in motor reflexes can therefore safely travel.

1.2 – PROJECT OUTLINE

The project is aimed towards the development of a multi-sensor based vehicle navigation system, trained and tested under different environmental conditions and implemented as a neural network image processing system on a prototype model using OpenCV and Raspberry PI. The system will enable the vehicle to achieve speed control, obstacle detection and collision avoidance and stopping on stop signs and traffic lights accordingly.

As the aim of the project is to develop a low-cost solution for an electric vehicle, the algorithm will then be implemented, with a suitable steering control strategy, on a prototype model electric vehicle. The prototype will incorporate the use of Raspberry-PI controller for the utilization of neural network image processing and ARDUINO board for vehicle control.

A control system model is also developed for the simulation of the autonomous control strategy for a real car. The steering system of the model vehicle will consist of a differential Ackerman steering system, controlled by the steering control algorithm using the steering angle and position parameters which are obtained from the image processing algorithm. Complete discussion on the type of steering control and obstacle detection strategies, and why this method was recommended for real-life environment, is discussed in the *Methodology* Section, in Chapter 3. In order to optimize the performance of the system, a classical PID control will be incorporated. The PID control parameters will be adjusted for the optimum control output and efficiency. Figure 2 provides an outlook towards the complete strategy for the development of the autonomous vehicle control.

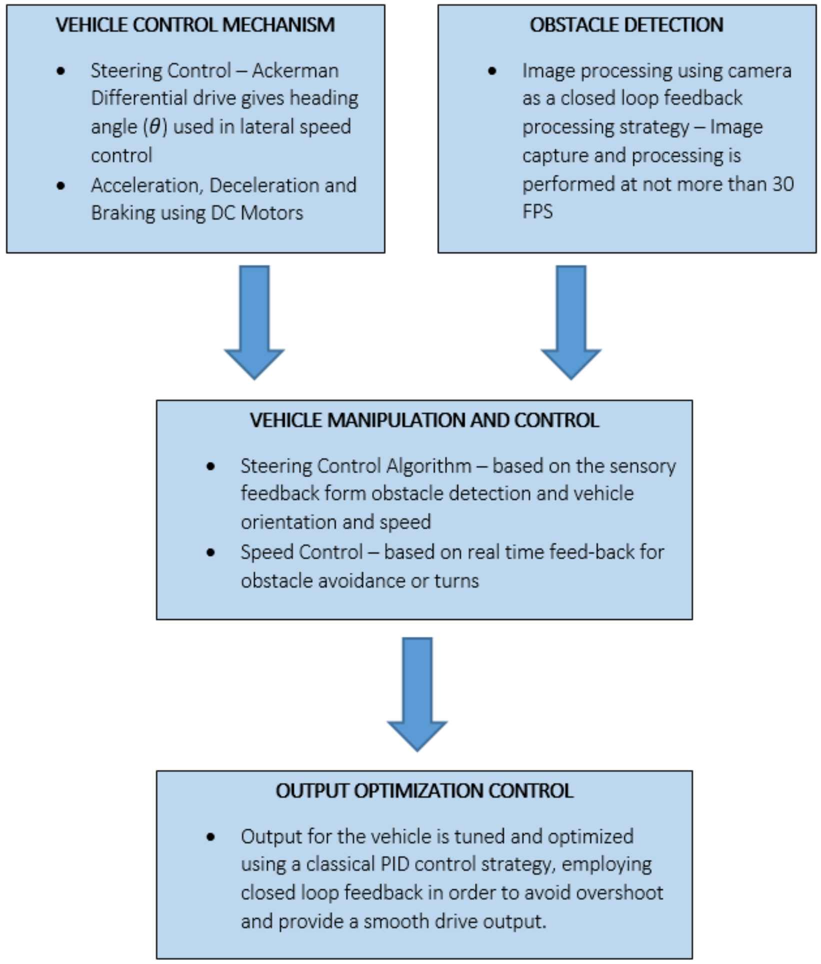

Figure 2: Autonomous Vehicle Control implementation strategy for real-life cars

The prototype model will be tested on the basis of vehicle control dynamics. The system is expected to function smoothly and provide a stable solution for low velocity drives.

Chapter 1 is the introduction to the project, and explains the motivation and general outline of the project. Chapter 2 consists of a detailed literature review, and lists the different adopted methodologies and similar existing works in literature. Chapter 3 defines the adopted methodology and the mathematical modeling for real cars, as well as that which was implemented for the prototype model, and highlights its advantages over other strategies. Chapter 4 discusses in detail the different cases designed to judge the performance of the developed prototype model, the parameters upon which the system is evaluated and the results of the evaluation. Chapter 5 is an evaluation of the developed prototype system, with possible future recommendations for improvement of the system.

CHAPTER 2: LITERATURE REVIEW

2.1 – AUTONOMOUS CONTROL

Anup Deshpande and Kovid Mathur [4] performed autonomous vehicle control for an all-terrain ground vehicle in the DARPA Urban challenge. The autonomous vehicle control implementation lead the design towards energy efficiency, and eliminated the need of human input which could be helpful to people with physical limitations. The study contrasted the three methods of vehicle control – namely mechanical, electrical and hydraulic – on the basis of brake control and vehicle speed control. Figure 3 represents the block diagram of the computer controlled steering mechanism.

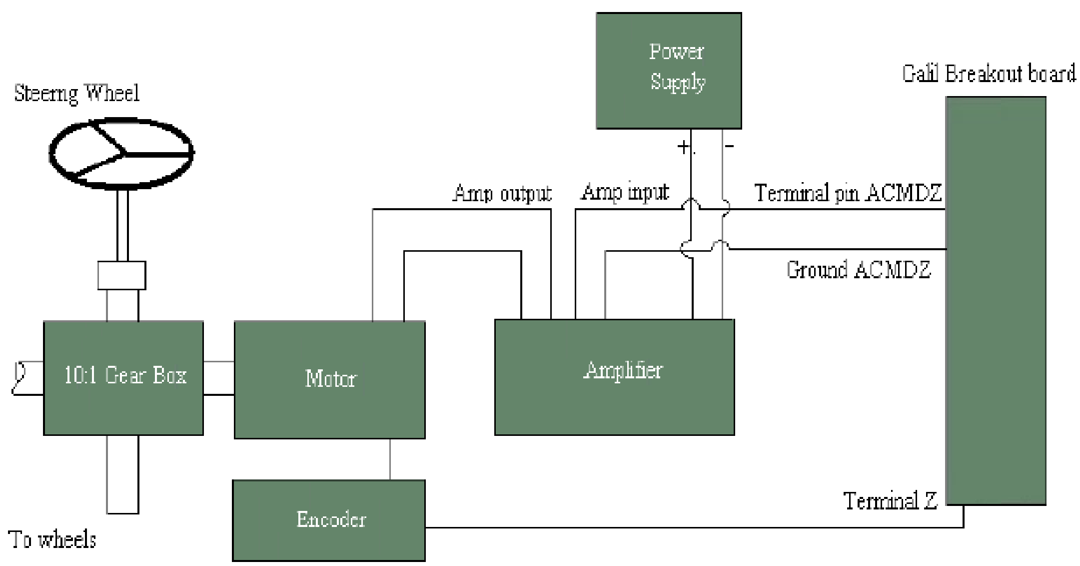

Figure 3: Block diagram of the computer controlled steering mechanism [4]

Sandor M Veres and Levente Molnar [2] presented the fundamentals of autonomous machinery, including generalized concepts of decision making, modelling of the environment and data abstractions for symbolic processing and logic based reasoning. The research is reviewed on autonomous vehicles on ground, air and under water and comments on the decision making methods they employ.

Nikolai Schlegel [5] implemented autonomous vehicle control on a small scaled model vehicle, designing a system capable of operating under both, a tele-robotic manual mode and an automated driving mode. The system consisted of an operational stationary remote control station consisting of a full-scale steering and gas pedal. The model vehicle was controlled by human input from the station in the tele-robotic manual mode. The vehicle, essentially a modified remote controlled car, was equipped with a video camera and a wireless radio modem, and was able to transmit the video footage to the remote control station, to be displayed on a monitor. The autonomous driving mode consisted of a lateral control algorithm to guide the vehicle's lateral movement and keep it in the center of the road. Two image processing algorithms were designed – one based on the pixel intensity and the other based on vanishing points in the frame – in order to determine the vehicle's position and orientation with respect to the road by identifying the white marker in the middle of the road. Two designs for control were introduced – a classical P-control and one based on H_∞.

Pan Zhao and Tao Mei [6] modeled an autonomous vehicle’s lateral dynamics using a simplified bicycle model with two degrees of freedom. A suitable system was designed for the vehicle named Intelligent Pioneer, which incorporated the use of an adaptive PID control strategy. Their work focuses on the development of a PID control system which is both adaptable and stable. The controller based on adaptive PID can automatically change parameters in the event of changing environments, and is hence proven more successful than a conventional PID which is not able to incorporate environmental adaptability and has to be assigned parametric values in a specific environment to optimize performance. Although the designed algorithm for motion planning was not as elaborate, the trajectory tracking was a considerably accurate.

Suresh Golconda [1] developed a PID control system based steering algorithm for a skid-steered vehicle, implementing the design on a 6-wheel skid-steered all-terrain vehicle named as the CajunBot. The control system was designed to be able to steer the vehicle at varying speeds, while also its safety by the prevention of sharp turns at high speeds. A prediction technique was implemented in order to prevent the over-steering of the vehicle during sharp turns, while focusing on aligning the vehicle into a straight line trajectory of the road tangent. The research focuses mainly on the steering control algorithm development in contrast to the obstacle detection and path planning, as an AGV unable to function without the steering control modules. The research makes an association of the PID control parameters with the heading error, and by correcting it, makes the parameters more dependent on the vehicle speed.

Khalid Bin Isa [7] demonstrated an autonomous vehicle control by the means of a simulated system, using image processing algorithm. The steering command for the lateral control of the simulated vehicle was determined by processing the images captured while the vehicle was in drive. The vehicle's dynamic performance was determined by combining the steering command with other parameters, and simulated and implemented in MATLAB. The image processing steps for the algorithm involved image segmentation using color cues, considering only the lower part of the image for the road at a certain look-ahead distance. This is followed by a canny-edge detection algorithm, where pixels with strong intensity of contrast are represented as white. The lane marker edges are determined, and followed by Hough Transform in order to detect the lane boundary for the image. The lane tracking algorithm was restricted to 30 frames per second, improving the computational speed and accuracy of the detection. The simulation results yielded an accurate portrayal of a vehicle's dynamic model, and the designed controllers were consistent with the approach.

Vito Cerone, Mario Milanese and Diego Regruto [8] addressed automatic lane keeping with driver's steering for obstacle avoidance, and implemented a closed loop control strategy. Their research was aimed to develop an autonomous system, where the lateral dynamics of the vehicle were to be controlled by the driver themselves, but automatic lane-keeping system would control the lateral dynamics when there was no driver input. The simulation performed on a 2 DOF system, and the corresponding experimental results on FIAT Brava 1600 ELX proved that the lane task was achieved smoothly and accurately. The autonomous system was comprised of a single CCD camera with related image-processing algorithms, and both the control and vision based processing was performed by an INTEL 486 microprocessor based PC, designed for industrial applications.

E. Onieva, V. Milanés, J. Pérez and T. De Pedro [9] designed, for a mass produced vehicle based on fuzzy logic control, an autonomous steering control. The system was designed to obtain a precision similar to human drivers, and provided relative smoothness in control actions. The ORBEX fuzzy development environment was utilized in the design and construction of the controller, which was dependent on the two inputs – lateral and angular errors – obtained from the GPS positioning. The image capture and processing lead to the determination of the vehicle's position and orientation, and sent to a low level control layer which consisted of a PID controller managing the motor attached to the steering bar. The controllers, when tested on a private asphalted test circuit, demonstrated a considerably efficient behavior on straight road segments and tracks but less precise on curved roads.

Bilin Aksun Güvenç and Lavent Güvenç [10] presented a robust, 2 DOF design of a controller for the automatic steering of a city bus. The research focuses on the formation analysis of the disturbance observer-based add-on compensator. A disturbance feed forward-based add-on compensator is also defined for reference trajectories.

2.2 – NEURAL NETWORKING

When using a simple neural network for training image datasets, the number of nodes of the input layers are defined by the resolution of the subsequent image, and the input corresponds to the individual pixel intensities collected in a numpy array [16]. The corresponding outputs for the image data is then paired together, to generate a training data set for the neural network. Error is calculated from the difference between the desired output and the one generated by the neural network prediction, and is used to adjust the weights based on the error. Figure 4 below shows the structure of a simple feed-forward neural network, with 2 nodes in the input layer, 3 in the hidden layer and 1 in the output layer [17].

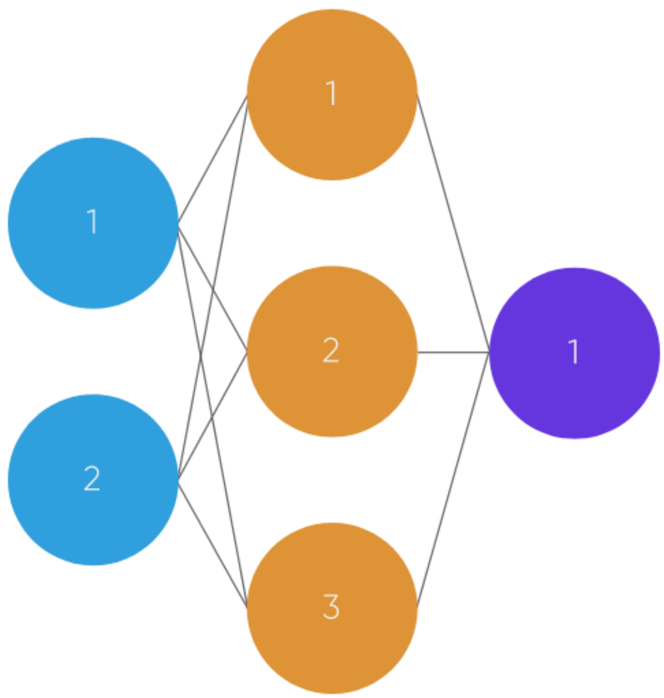

Figure 4: Simple feed-forward neural network structure with an input, hidden and output layer

The training of the neural network is based on backward propagation method, and the greater the amount of training data, the better the accuracy of the neural network in predicting the output.

CHAPTER 3: METHODOLOGY

3.1 – CONTROL MODEL FOR REAL CAR

As defined in Figure 2 in Chapter 1, an autonomous vehicle control methodology can be broken down into four functions. This sub-section focuses on the adopted methodology and its underlying mathematical modelling for each function of the autonomous control system implementation, namely the *Vehicle Control Mechanism*, the *Obstacle Detection Strategy*, the *Manipulation and Vehicle Control* and the *Optimization of Output*.

3.1.1 – Vehicle Control Mechanism

3.1.1.1 – Ackerman-Type Control Steering:

An Ackerman-type steering, as commonly used in vehicles, provides an easier use of wheel encoders. The precision of steering is also good, and a smooth turning response. The Ackerman-type Steering involves the turning of both wheels at a different angle in order to turn the vehicle. Figure 5 shows the diagram for an Ackerman-type steering control.

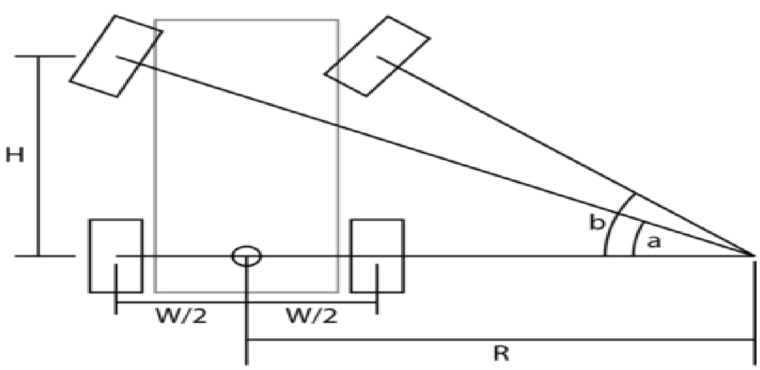

Figure 4: Ackerman steering principle and turn angles for both wheels

3.1.1.2 – Skid Steering Control (Differential Steering)

In a skid-steering strategy, the wheels do not turn. The wheels on each side of the vehicle are connected via chain or belt drive, and have the same output RPM. The difference in the RPM of the left and right side provides the turning direction and angle for the vehicle. Figure 6 shows the design of a skid-steered vehicle.

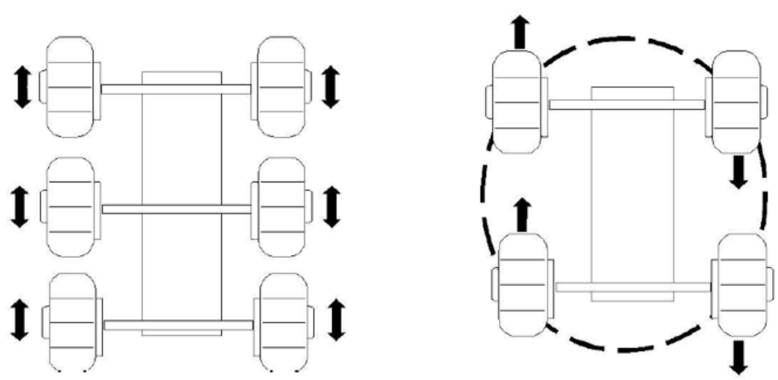

Figure 5: Skid steered vehicle block diagram and principle

The use of an Ackerman type steering control is recommended for real cars. The skid steering is not recommended, as although it provides a quicker response to the actuation, is not as stable and highly dependent on the terrain parameters involved.

On the other hand, an Ackerman-type steering control can be easily incorporated into the autonomous control using servo control. This control parameters, the steering angle and the velocity can be utilized directly in the controller algorithm and in output regulation.

3.1.2 – Obstacle Detection Strategy:

An autonomous vehicle control system can incorporate a number of strategies in order to sense and detect the obstacles from their surroundings. Among these methods are the usage of a proximity sensor to detect obstacles, LIDAR and RADAR and image processing using a camera as a sensor. Multi-sensor systems are usually recommended in real life situations, as feedback data from a single sensor is not sufficient enough for the vehicle to function autonomously.

Image processing algorithm for the lane tracking and detection as well as obstacle detection are widely used and recommended. In the lane tracking algorithm, Figure 7 shows the adopted methodology that is adopted by the controller in real time.

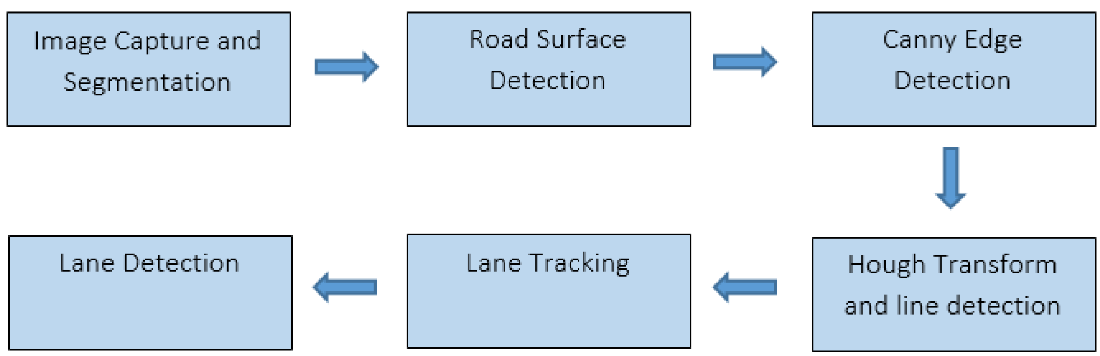

Figure 6: Image processing algorithm methodology in steps

The camera captures the image in real-time, and the image is processed and segmented for use in edge detection algorithm. The general steps followed for the processing of the image are as follows, represented in Figure 8:

Figure 7: General steps for pre-processing of the image

Since the response is measured in intensity variations, image is converted to grey scale and HSV color space. The image can then be blurred for the canny edge detection algorithm to be applied and detect the edges of the road. Figure 9 shows the original road image, Figure 10 displays the gray-scaled processed image and Figure 11 shows the blurred image.

The purpose of Canny Edge detection algorithm is to extract the edges from the image, by locating the area where significant pixels are concentrated. The maximum intensity of pixels is represented as white in the resulting processed image, while the minimum intensity will be depicted as black. Figure 12 shows the processed image after the edges have been extracted.

The lane tracking and lane modelling will be discussed in the next section, as the steering algorithm utilizes the obstacle and lane detection algorithms.

Figure 8: Original image as captured by the camera

Figure 9: Image conversion into grayscale and adjusted saturation

Figure 10: Grayscale image blurred to reduce computational load

Figure 11: Edges detected by the Canny Edge Detection Algorithm

3.1.3 – Manipulation and Vehicle Control:

The vehicle manipulation and control uses the steering control algorithm in conjunction to the detected image. First, the lane detection and tracking algorithm is used to determine the position and orientation of the vehicle with respect to the road frame. Using the edge detection algorithm, the right and left edges of the road path are identified. This is more suitable when there are road markings in the middle of the road, but in this case we will not consider them being present.

The algorithm developed makes use of the distance from the center of the road as the error reading. Then, for a certain look-ahead distance, the heading error is calculated with respect to the road tangent. The road tangent is the road path in case of a straight line path. Figure 13 represents the heading error on the road path.

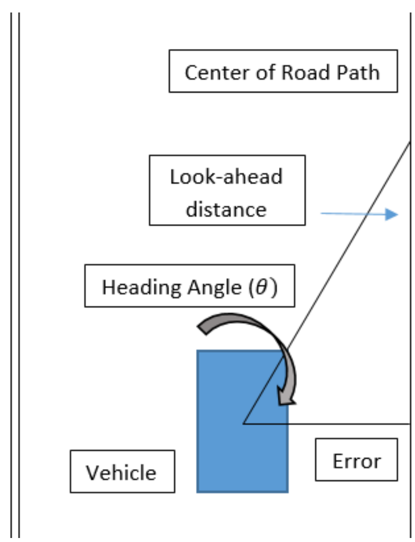

Figure 12: Heading angle and look-ahead distance based on the lateral error E

3.1.4 – Optimization of Output:

A classical PID control is recommended for use for the optimization of the output and the vehicle control. Although an adaptive PID control is a more efficient system which incorporates the variability of the system control parameters and does not require manual adjustments, it is complicated and requires extensive and thorough research [6].

We make use of a 2 DOF control analysis for the vehicle parameters [6]. Figure 14 shows the equivalent 2 DOF model (bicycle model) for the prototype car:

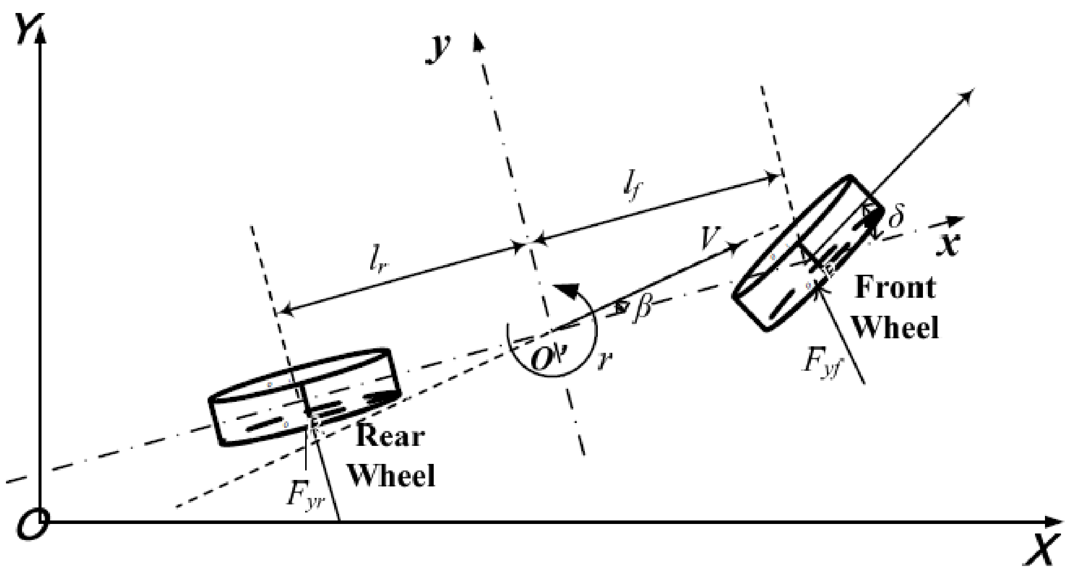

Figure 13: A 2 DOF model for the prototype of the model vehicle

Hence, a classical PIC control will be implemented on the prototype model. The state space system of equations is defined as follows:

$$\boldsymbol{X} = \boldsymbol{A}\dot{\boldsymbol{X}} + \boldsymbol{B}\boldsymbol{U}$$

Where:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} v \\ r \end{bmatrix} \text{and } \boldsymbol{U} = f(t)$$

Here, v and r represents the vehicle's lateral speed and the rotational speed about its center of mass respectively.

Now,

$$a_{11} = -\frac{C_f + C_r}{u_c m} \qquad a_{12} = -u_c - \frac{aC_f - bC_r}{u_c m}$$

$$a_{21} = -\frac{aC_f - bC_r}{u_c I} \qquad a_{22} = -\frac{a^2 C_f - b^2 C_r}{u_c I}$$

$$b_1 = \frac{C_f}{m} \qquad b_2 = \frac{aC_f}{I}$$

Here, C_f and C_r represent the coefficient of cornering friction of the front and rear wheels respectively, m represents the vehicle's mass, I is the yaw moment of inertia of the vehicle, and a represents the longitudinal position of the front wheel from the vehicle center.

As mentioned before, in the previous section, the quantity which has to be controlled is the lateral error E. The control system transfer function is hence the transformation from the steering input angle (or the heading angle calculated by the vehicle manipulation and control algorithm) to the lateral path error output.

$$G(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$$

This concludes the general modelling for the multi-sensor based autonomous vehicle control system for a real electric car. The specific numerical modelling makes use of the car parameters such as the weight, the coefficient of friction between the tires and the road etc. hence a general model for the real car is presented.

The following sub-section demonstrates the modelling and complete methodology for the prototype model car developed in the project.

3.2 – METHODOLOGY FOR THE PROTOTYPE MODEL

The prototype model was developed on a small-scaled RC car as a base model. The radio control unit was removed from the model, as it was no longer needed. Instead, the input from the multi-sensor system and the actuation signal from the processing unit based on those specific inputs was used to directly control the motors of the vehicle using PWM. Figure 15 below shows the data flow diagram of the system.

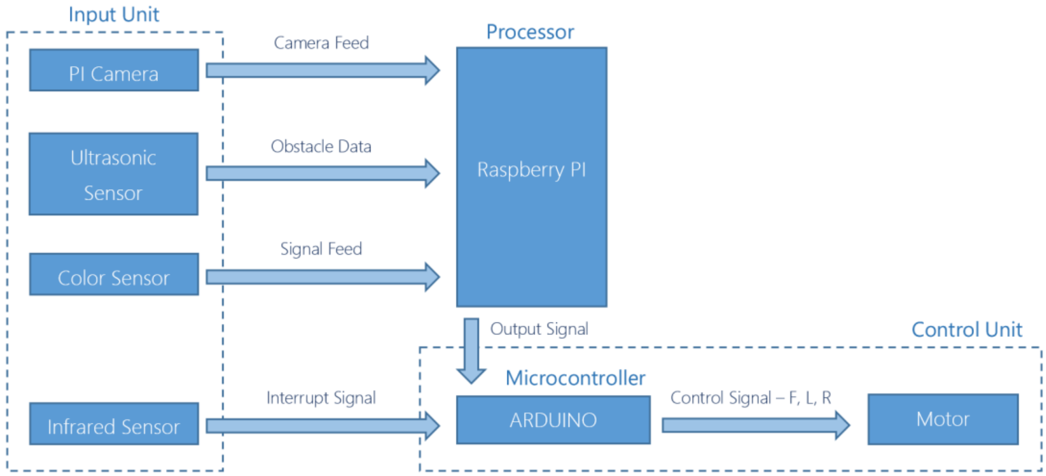

Figure 14: System data flow diagram for the prototype autonomous vehicle control system

It can be seen from Figure 15 that the system makes use of inputs from 3 different sensors (the camera functions as a sensory unit for visual input) to generate a single output signal from the processing unit, i.e. the Raspberry PI. Based on this output signal and an additional sensor feed from the infrared sensor which acts as a backup sensor, the microcontroller (ARDUINO UNO) generates an actuation or control signal to drive the motors. Figure 16 below shows the top and side views of the completed prototype model respectively.

Figure 15: Top and side views of the completed prototype autonomous vehicle

The following sub-sections will cover the detailed working methodology of the autonomous vehicle prototype with respect to the input, processing and control units.

3.2.1 – The Input (Sensor) Unit:

The autonomous vehicle control prototype system makes use of a multi-sensor technology to fulfill all the requirements, but the main objective (lane detection and lane tracking) is fulfilled through image processing using a simple neural network. There are a total of 4 sensors working in conjunction to deliver the required feedback to the processing unit and the control unit. Figure 17 below shows the placement of all the sensors on the prototype model.

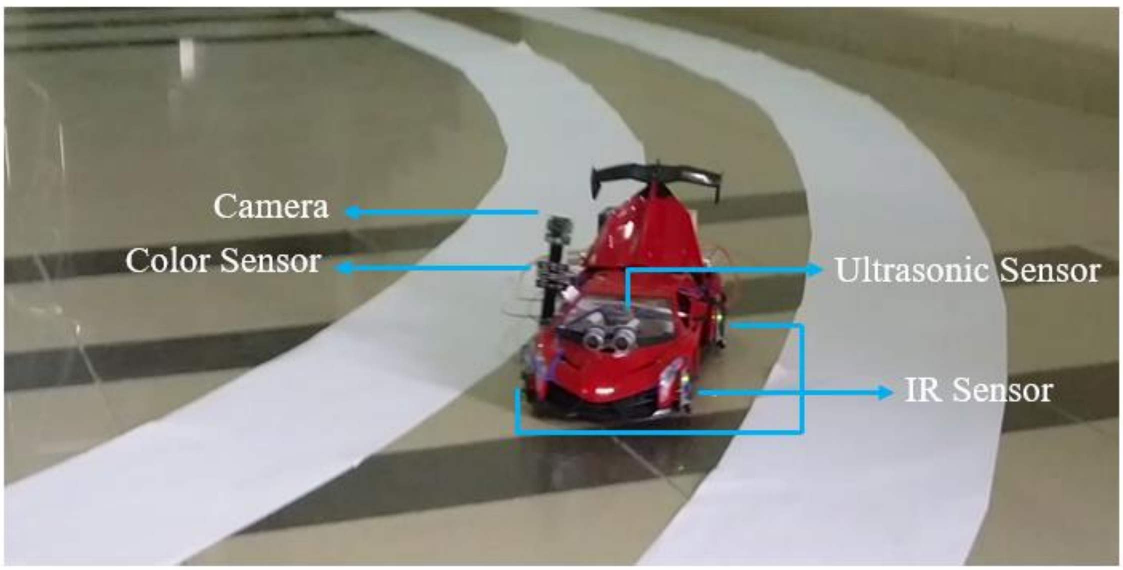

Figure 16: Sensor placement on the prototype system

3.2.1.1 – Camera:

PI-Camera module is used for capturing the visual feed for the neural network processed in the Raspberry PI microprocessor. The frame rate for the camera is set to 15 frames per second, with a resolution of 320×240 which is sufficient for the processing of the neural network.

3.2.1.2 – Ultrasonic Sensor:

A single ultrasonic sensor (HC-SR04) was used for front obstacle detection. Since the objective of the prototype model was to avoid collision with the obstacle directly in front of the car, a single sensor was sufficient for use. The feedback for the sensor was used directly in the microprocessor (Raspberry-PI) to determine the presence of an obstacle at a distance of 20 cm (greater than one car length, which is equivalent to the optimum safe driving distance in real life).

3.2.1.1 – Color Sensor:

A color sensor (TCS-3200) is used for the detection of traffic lights signal, detecting the colors red and green only. The color sensor feed is not sufficient alone in this case due to variation in light intensity, speed of the vehicle etc. hence the sensor feed is used in conjunction with Haar feature-based cascade classifier for the traffic signal (discussed in the processing unit section). At the optimum vehicle speed of 15km/h, the color sensor shows good functionality in the detection of traffic signal lights.

3.2.1.1 – Infrared Sensor:

Four infrared sensors are mounted just above each of the four wheels of the ground, facing the ground at a protruding angle of 12 degrees. The infrared sensors act as backup sensors, i.e. if the vehicle goes, or is about to go out of the lane boundary due to image processing errors or miscalculation, the four sensors act as backup to turn the vehicle back on the track as soon as possible. This method of self-correction is crucial in case of sharp turns, where the trajectory of the path changes. The camera vision is limited to the track in front and hence it is difficult to correct the position if the vehicle has already exited the lane boundary. The feed from the infrared sensor detects the change in color intensity (based on the infrared feed) to determine whether the lane boundary has been crossed. The angle of 12 degrees serves as the optimum angle to allow an early warning in case the vehicle is about to go out of the lane boundary, but has not yet exited it. A schematic of the working of infrared sensor is shown in Figure 18 below.

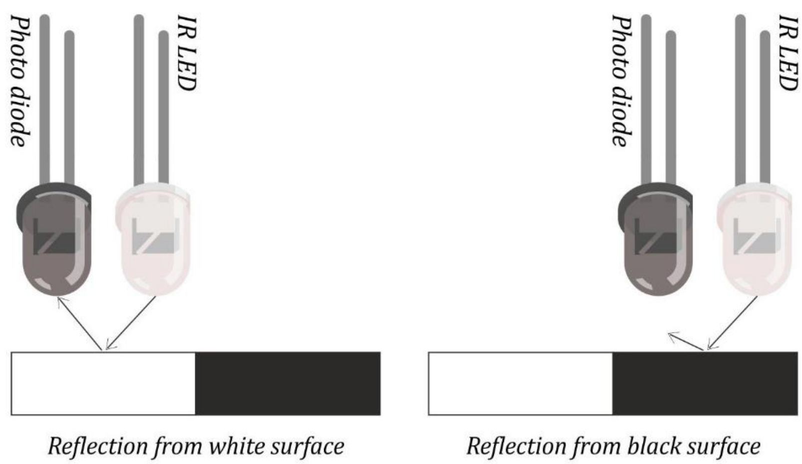

Figure 17: Working schematic for an IR sensor

3.2.2 – The Processing Unit:

The processing unit of the autonomous control system for the prototype model is the Raspberry-PI microprocessor, which runs the lane detection, lane tracking, obstacle detection and signal detection algorithms. Figure 19 shows the data flow diagram for the processing unit of the autonomous vehicle prototype model:

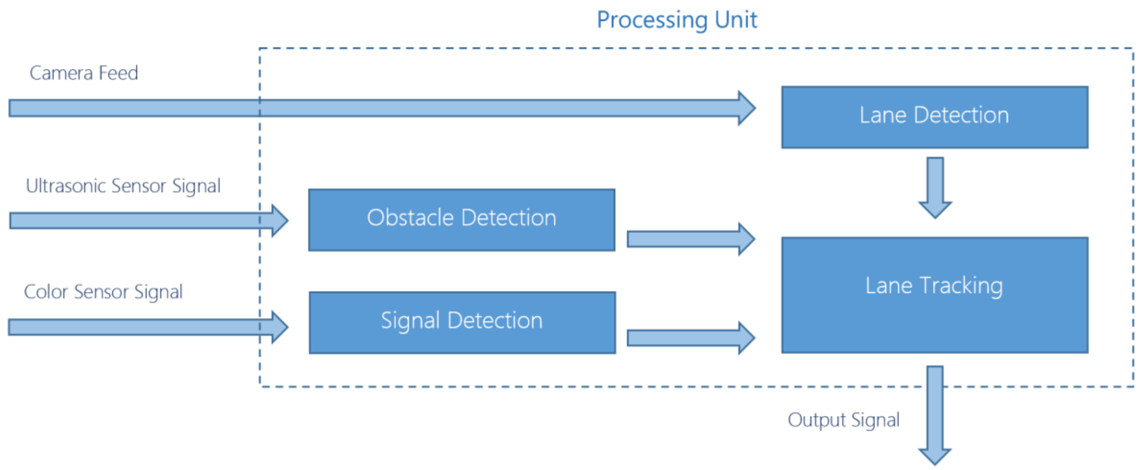

Figure 18: Data flow schematic for the processing unit of the autonomous vehicle prototype system

3.2.2.1 – Lane Detection and Tracking:

The main function of the Raspberry-PI microprocessor is to run the image processing algorithm, i.e. a simple neural network to allow the vehicle to perform lane detection and tracking on the defined track. The neural network was first trained after the data collection process.

The feed-forward neural network only uses the lower half of the image for lane tracking and detection. Hence the original image resolution of 320×240 is reduced to 320×120, translating into 38,400 nodes of the input layer. The hidden layer of the neural network consists of 32 nodes; the figure was arbitrarily chosen for optimum performance from literature [15]. The output layer of the neural network structure consists of four nodes, each corresponding to a single output (i.e. forward, left, right and reverse). The reverse output is not used in the prototype model, but the option is still available in the neural network. Figure 20 below shows the structure of the neural network.

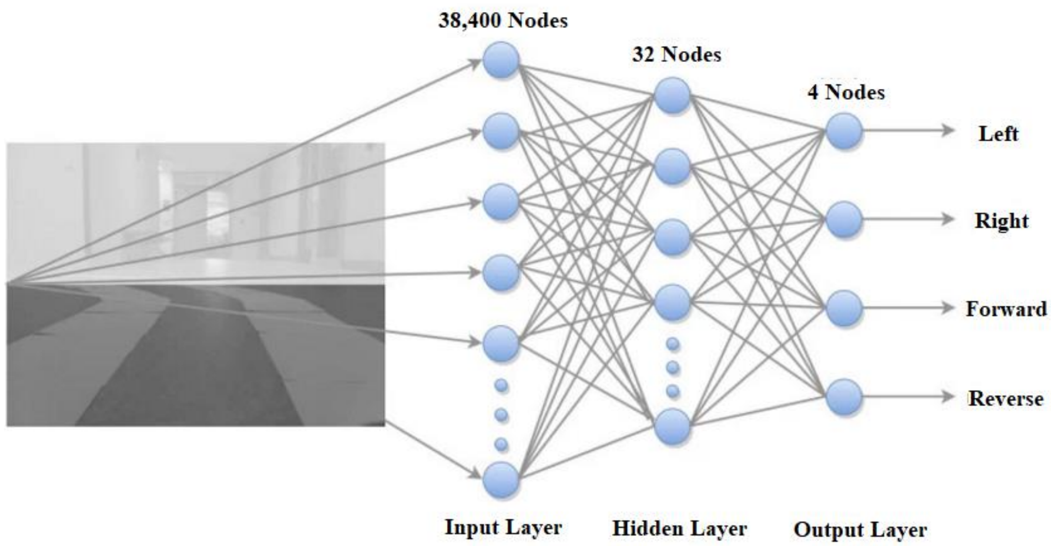

Figure 19: Schematic diagram for the neural network used in the prototype model

For training the neural network, data collection was performed with the PI-Camera and manual vehicle control using ARDUINO board and wireless keyboard. At a processing rate of 15 FPS, the training image frames and the human input data, i.e. train labels were obtained. For each frame, the image frame is first cropped and converted into a numpy array. After the image has been paired with the corresponding label, all the training data was processed and saved into an ".npz" file. The training of the neural network was performed using OpenCV Python. The backward propagation method was employed during the parametric training, to generate the weights for the neural network.

3.2.2.2 – Obstacle Detection:

Obstacle detection is performed using the input feed from the ultrasonic sensor. The control unit will stop the motor when the distance between the obstacle and the sensor is 20cm or less for frontal collision avoidance. The monocular vision method is employed to measure the distance. This was tested in real time. The results are displayed in Chapter 4.

3.2.2.3 – Traffic Signal Detection:

Traffic signal is detected via Haar feature-based cascade detection for the traffic signal object classifiers. This approach was relatively a good option as the object is stationary. Figure 21 below shows the steps in the traffic signal detection process.

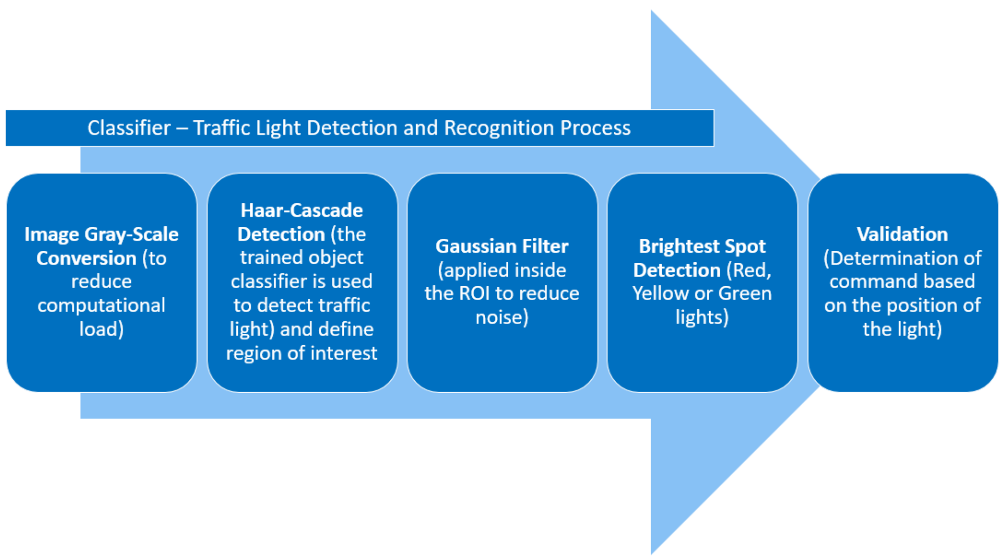

Figure 20: Image processing steps for traffic signal detection process

The ROI is defined as the bounding box of the object classifier, and noise reduction of the image was performed using Gaussian blur. The traffic light was detected with respect to the position of the light using the brightest spot detection.

Images for the signal classifiers were taken from a mobile phone and cropped, with only the desired object available in the frame. The positive samples included the object while the negative samples did not. The feature-based detection is used in conjunction with the input feed from the color sensor, which validated the traffic signal color. In the case when the traffic light was red, the color sensor returned the signal for the red light and the Haar cascade detection returned the validated output for red light, using both the outputs the processor returns an output for red signal light, which is later sent to the microcontroller which turns of the motors. Figure 22 below shows some of the positive image samples used for training in OpenCV.

Figure 21: Positive image samples for development of object classifiers

3.2.3 – The Control Unit:

The control unit of the autonomous system of the prototype model comprises of an ARDUINO UNO board. The output signal from the processor is a four bit output corresponding to each of the four states i.e. left [1, 0, 0, 0], right [0, 1, 0, 0], forward [0, 0, 1, 0] and reverse [0, 0, 0, 1] (although reverse is not used). At a single instant of output from the processor, only a single pin is high and the other 3 pins are low.

The interrupt light is the signal feed from the processor which determines the traffic light state (red [LOW] or green [HIGH]). The distance measured using ultrasonic sensor is used as a parameter for obstacle collision avoidance.

The interrupt light signal is high for when there is no traffic light, and also when the light is green. The only state in which it is low is when the light is red. The two bounding parameters for control are the distance of the obstacle from the vehicle (sensor) and the interrupt signal for the traffic lights. If the distance is 20cm or less or if the light is red, the actuating signal to the forward drive DC motor is LOW, i.e. the vehicle stops. If the distance is greater than 20cm (or in case of no obstacle) and if the traffic signal is red, the control unit takes the output signal from the processing unit. This output signal, as discussed before corresponds to the 4 states – forward, left, right and reverse. In the forward state, the actuating signal to the forward drive motion is high, i.e. the vehicle drives straight. In case of left and right signals, the forward drive DC motor remains switched on, and the steering motor also correspondingly moves to the left or the right. The turning is not based on the turning angle, but rather the wheels are fully turned in either direction. The controlling parameter is the duration – longer duration means greater turning and vice versa. Hence the lance tracking is based completely on image processing.

The camera input determines the control for both forward drive and turning, while the inputs from color sensor and ultrasonic sensor is used for only forward drive control. Lastly, the input feed from the infrared sensors is used only for steering control. As soon as a change in color intensity of a threshold equal to or greater than 20% is detected, from the sensors from either side, the vehicle turns to the opposite side until the color intensity is stabilized again. This makes sure that the vehicle gets back on track if it crosses, or is about to cross a lane boundary.

CHAPTER 4: RESULTS AND DISCUSSION

The results of the prototype model are evaluated on the based on the project objectives, i.e. obstacle detection, signal detection and lane tracking. For lane tracking, there are three cases corresponding to the type of track – straight, left curved, and right curved. The evaluation parameters for the obstacle detection is the distance from the object, and the rate of success. The lane tracking is based on the success rate, i.e. the ratio of the number of experiments where the vehicle tracks and follows the path correctly to the total number of experiments performed. Lastly, the traffic signal detection is also evaluated based on success rate of the vehicle in stopping correctly at the signal. All the experiments are performed in real time, and data was collected over 3 weeks, in both day and night times.

4.1 – RESULTS FOR OBSTACLE DETECTION

Results for the experimentation for the obstacle detection yielded the largest error of 3cm (15%) from the 20cm threshold. Results for a total of 40 experiments were collected for a 30m long straight track. The experiments were performed in a total of 4 sets, with 10 experiments per set. Experimentation of sets 1 and 2 were performed during day time and those of sets 3 and 4 during the night time. Figure 23 below shows the obstacle (solid box) used in the experimentation procedure.

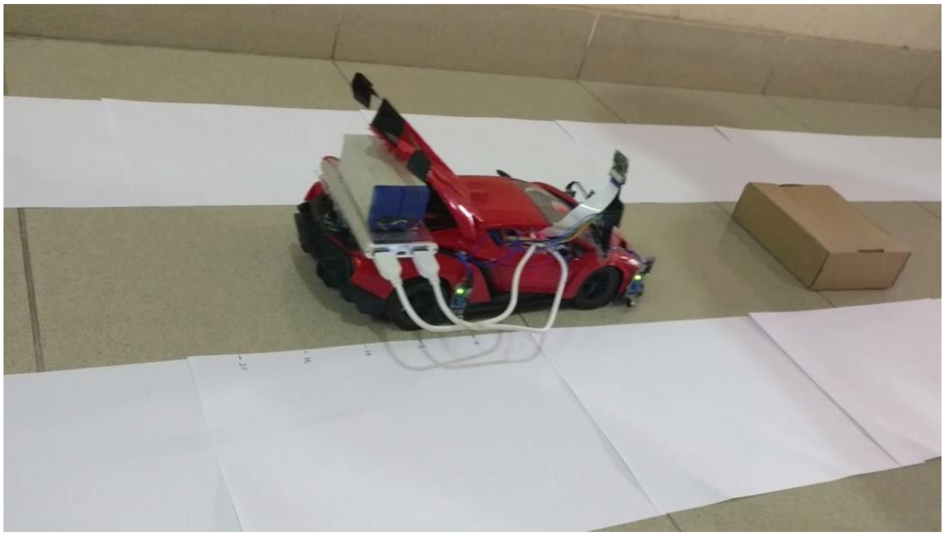

Figure 22: Obstacle detection experimentation on straight track

The experimentation results for obstacle detection are shown in Figure 24, and Table 1.

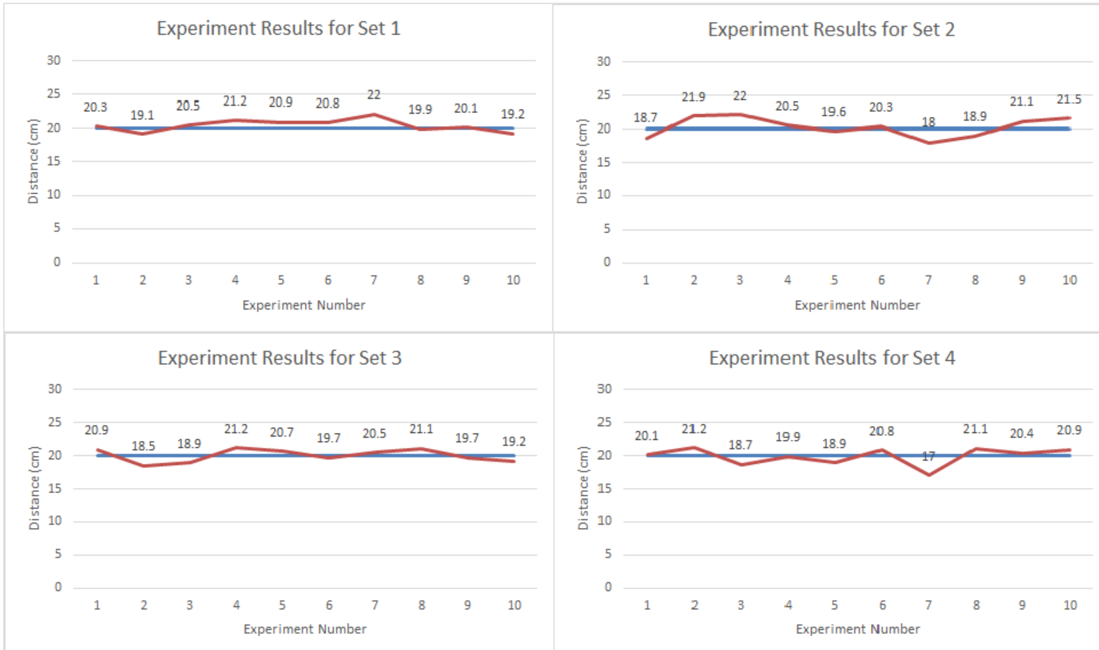

Figure 23: Experimentation results for obstacle detection during day (sets 1 and 2) and night (sets 3 and 4) time

Parameter	**Value**
Average Distance Set 1 (cm)	20.40
Average Distance Set 2 (cm)	20.25
Average Distance Set 3 (cm)	20.04
Average Distance Set 4 (cm)	20.00
Average Distance – Overall (cm)	20.17

Standard Deviation	1.153
Variance	1.329

Table 1 Statistical experimentation results for obstacle detection during day and night

4.2 – RESULTS FOR LANE TRACKING

4.2.1 – Case 1: Straight Track

The autonomous control system shows a very good performance on the straight track. A total of 60 experiments were performed to evaluate the lane tracking accuracy. The system showed a remarkable 95% accuracy in the day time, with a total of 57 out of 60 experiments being successful in proper lane tracking. The accuracy reduced to 90% in the night time with 54 out of 60 successful experiments, due to the contrast between the lane boundaries and the path being reduced. Overall results for lane tracking on straight track were promising. Figure 25 shows the lane tracking on the straight track, and Figure 26 shows the statistical results of the experimentation.

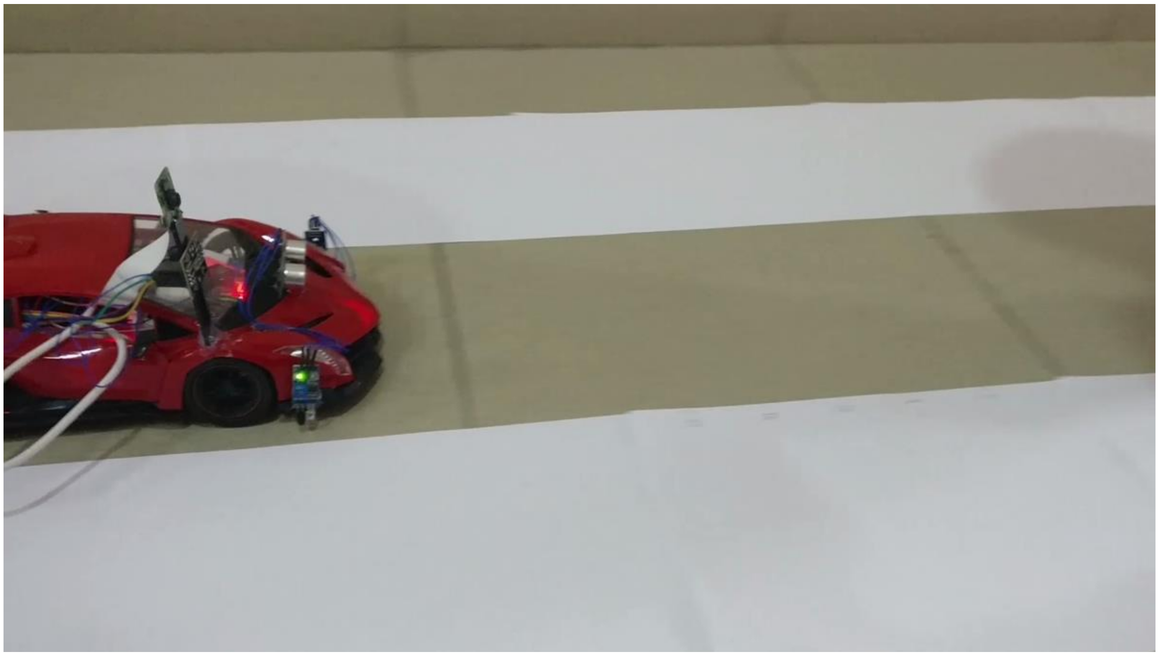

Figure 24: Lane tracking experimentation on straight track

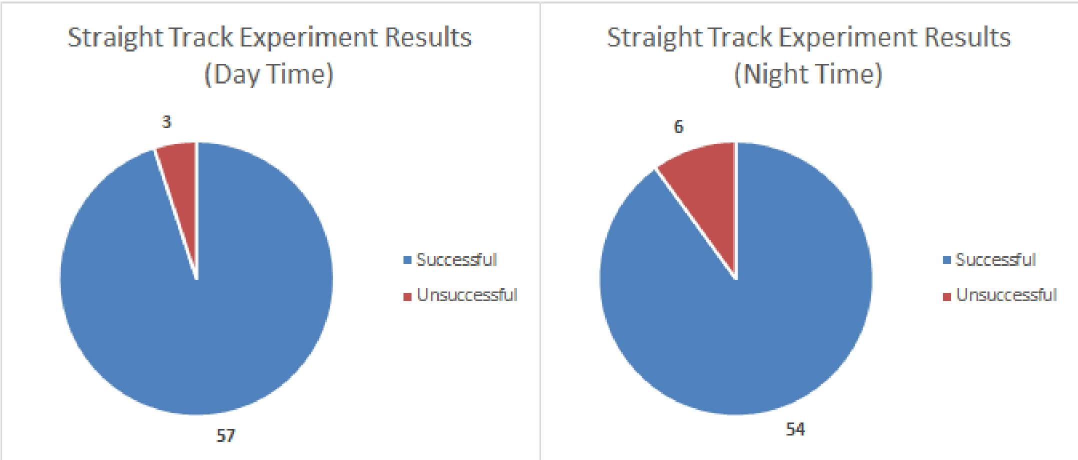

Figure 25: Statistical experimentation results for lane tracking on straight track during day and night times

4.2.2 – Case 2: Left-Curved Track

The 30m long arc length of a track of a turn radius of 28.6m was selected for experimentation. The total number of experiments were again 60, with 53 out of 60 (88.3%) successful experiments during the day time and 50 out of 60 (83.3%)during the night. Figure 27 shows the schematic for the track, while Figure 28 shows the lane tracking experiment for left-curved track during day and night time. The statistical results for the experiment are shown in Figure 29.

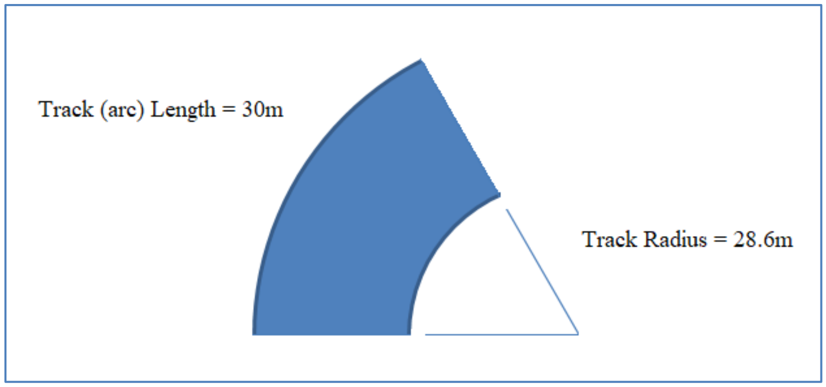

Figure 26: Schematic Diagram for the left curved track (anti-clockwise travel direction)

Figure 27: Lane tracking experiment for left-curved track during day and night times

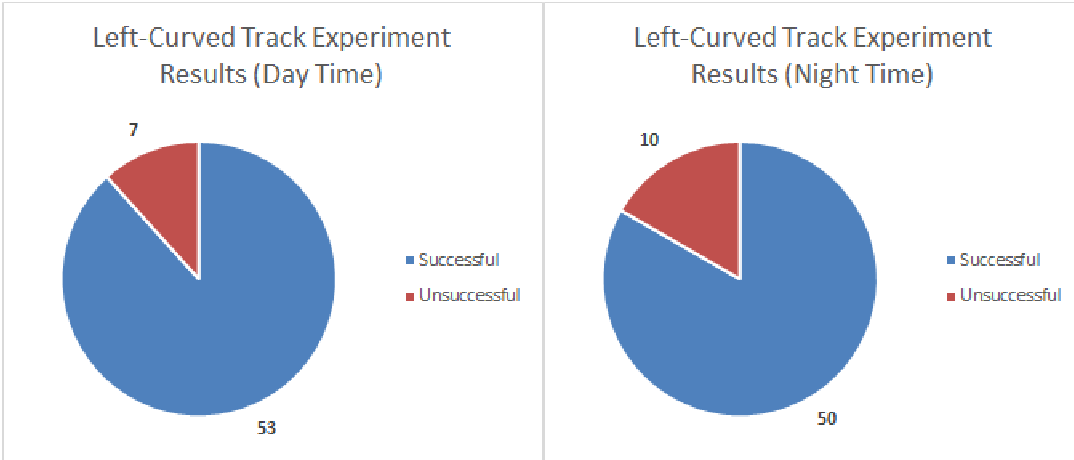

Figure 28: Lane tracking experiment results for left-curved track

<u>4.2.3 – Case 3: Right-Curved Track:</u>

The same track was used from the opposite side in order to set the same track parameters (track length and turning radius). Out of 60 experiments, day time data returned a success rate of 52 out of 60 experiments (86.7%), while that of night time was 51 out of 60 experiments (85.0%). Figure 30 below shows the experimentation during day and night times, while Figure 31 shows the experimentation results.

Figure 29: Lane tracking experiment results for right-curved track during day and night times

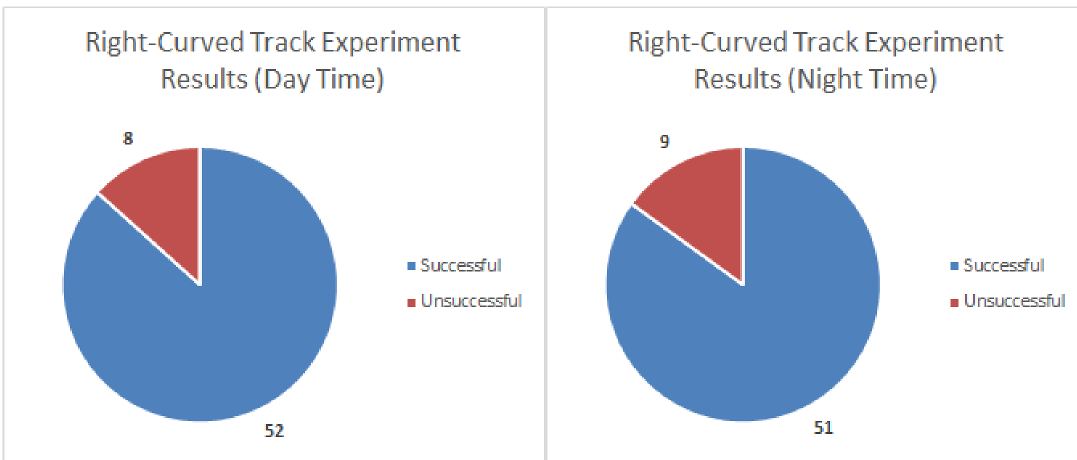

Figure 30: Lane tracking experiment results for left-curved track

4.3 – RESULTS FOR TRAFFIC SIGNAL DETECTION

The experimental verification for traffic signal detection was performed in 6 sets, with a 10 experiments per set. The first 3 show the results for day time, while the last 3 show results for the experiments during night time. The success rates during the day time came out to be 90%, 86.7% and 88.33%, while those at the night time corresponds to the figures of 95%, 96.7% and 95%. The overall success rate of signal detection is higher at night time as compared to day time. Figure 32 and Figure 33 show the experiment of signal detection from top and back view respectively. The statistical results for signal detection are shown in Figure 34.

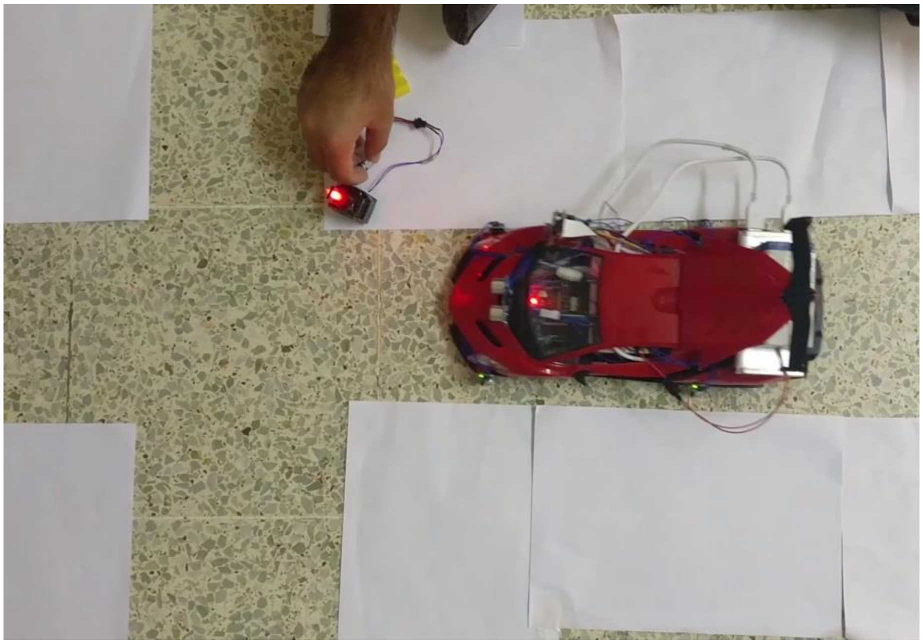

Figure 31: Signal detection experimentation (top view)

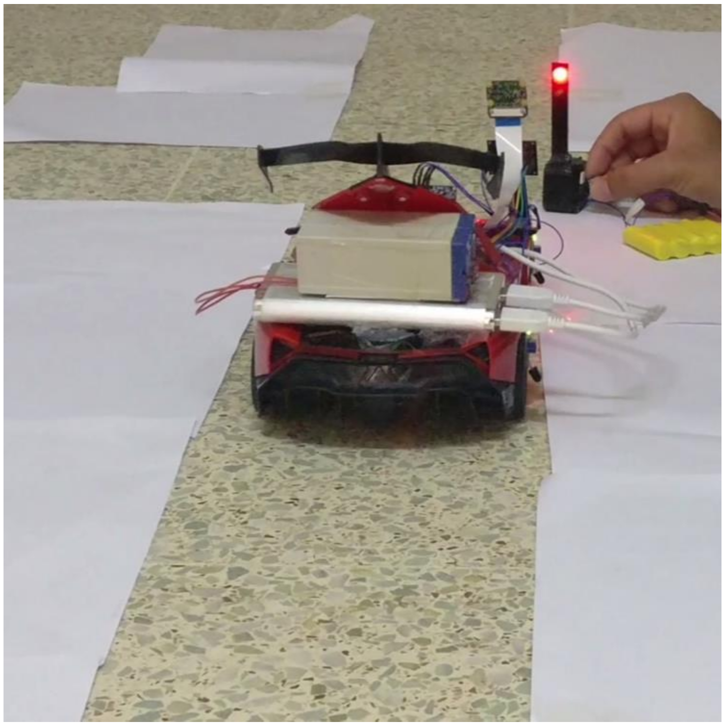

Figure 32: Signal detection experimentation (back view)

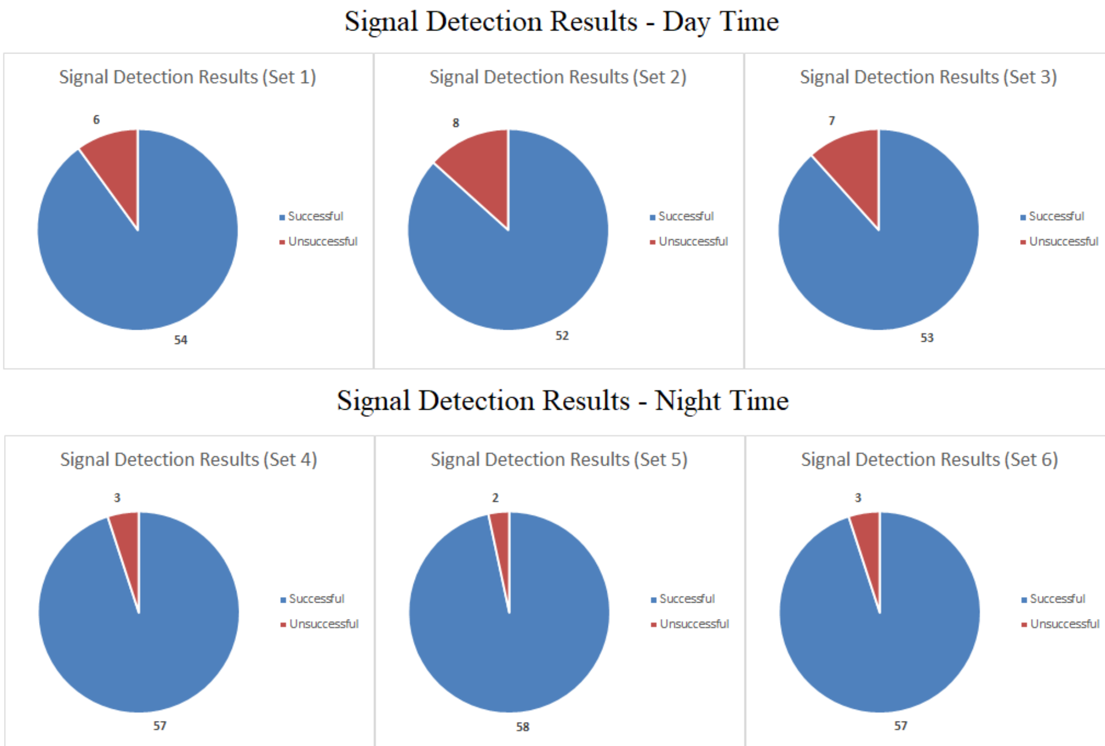

Figure 33: Statistical Results for signal detection experimentation

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 – CONCLUSION OF THE PROJECT

The aim of the project was two-fold, first to develop a numerical model for a multi-sensor autonomous vehicle control system utilizing vision-based lane tracking system for a real car and second, to make a prototype model of an autonomous vehicle utilizing image processing as the main lane tracking methodology. The objectives for the prototype model were obstacle detection (frontal collision avoidance), lane detection and tracking, and traffic signal detection.

For the first part, a processing unit based on OpenCV Python platform is recommended for use in monocular vision for lane tracking. Ackerman steering system is recommended with a classical PID output control regulation. The PID parameters are manually trained to achieve optimum drive and smooth regulation.

The prototype model consists of a simple neural network based vision control system, along with feed from various sensors (ultrasonic, color and infrared sensors) to control the longitudinal and lateral drive for the vehicle.

Obstacle Detection showed a maximum error of 3cm (15%) in 1 of the total of 4 experiments. The average error in the total experiments was below 8%, which is a promising result in case of obstacle detection at greater speeds. All collisions were avoided at the safe distance, and obstacle detection methodology was a success in the prototype system. The lane tracking results showed greater accuracy during the day time, which was generally 4-5% more than that in the night time. The reason for the decrease is the decrease in contrast between the path (road) and the boundary (lane markers). Also, the results for left-turning track showed slightly greater accuracy than those of right turning tracks. Possible reasons may include the positioning of the camera (right side of the vehicle), or limitations of training data. Nevertheless, the success of the prototype model was greater than the expected accuracy of similar neural networks (75 – 80%). Signal detection results were noticeably better at night time, due to prominence of the signal light detected and lesser noise for the color sensor to detect. Without the Haar feature-based cascade detection, by only relying on the color sensor to detect the traffic signals, the difference in success rate increases as much as 20-30%, hence showing that color sensor alone is not adequate for this task.

5.2 – FUTURE RECOMMENDATIONS

1. The accuracy of the neural network can be improved by 4-5% through a larger training data set.
2. The prototype model was based on an RC car, hence the braking option was not available. The car slowed to a stop when the motor was turned off, which was suitable for low speeds. In order to let the system achieve greater speeds, physical model with braking is recommended.
3. A second camera and an ultrasonic sensor can be used to detect cars and obstacles from behind, enabling speed control and greater autonomy of the vehicle.

4. GPS technology can be used in conjunction with the multi-sensor system in order to incorporate path planning and to avoid collision from the camera's "blind spots".
5. Haar feature-based cascade detection can be employed for stop signs as well, enabling a greater and flexible autonomous control over the vehicle.
6. Image processing in conjunction with other sensors (such as ultrasonic sensor for frontal collision avoidance) can be used for obstacle detection for greater accuracy. Figure 35 below shows an autonomous vehicle's multi-sensor concept.

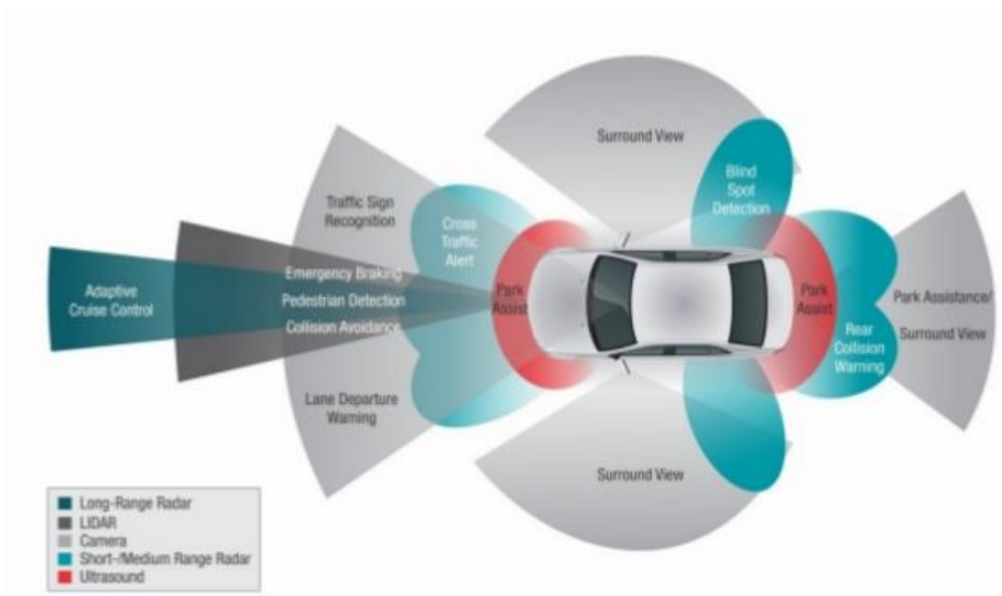

Figure 34: Multi-sensor based autonomous vehicle control system concept [18]

REFERENCES

[1] G. S, "Steering Control for a Skid-Steered Autonomous Ground Vehicle at Varying Speed," pp. 3-58, 2005.

[2] S. M, "Autonomous Vehicle Control Systems - A Review of Decision Making," pp. 3-20, 2015.

[3] "Pakistan Bureau of Statistics – Annual Road Accidents (2014 - 2016)," [Online]. Available: http://www.pbs.gov.pk/content/traffic-accidents-annual.

[4] D. A, "Comparison of Three Control Methods for an Autonomous Vehicle," pp. 1-12, 2010.

[5] S. N., "Autonomous Vehicle Control using Image Processing," pp. 3-130, 1997.

[6] Z. P, "Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID," pp. 1-11, 2012.

[7] K. B, "An Autonomous Vehicle Driving Control System," pp. 1-12, 2005.

[8] C. V, "Combined Automatic Lane-Keeping and Driver's Steering Through a 2-DOF Control Strategy," pp. 1-8, 2009.

[9] E. Onieva, "Genetic Fuzzy Based Steering Wheel Controller using a Mass-Produced Car," pp. 1-19, 2012.

[10] A. B, "Robust Two Degree of Freedom Vehicle Steering Controller Design," 2002.

[11] S. Emarose, "How to make a simple Autonomous Vehicle," 2013. [Online]. Available: https://www.robotshop.com/letsmakerobots/how-make-a-simple-autonomous-vehicle.

[12] N. E, "Choosing a vehicle," 2016. [Online]. Available: https://www.embeddedrelated.com/showarticle/964.php.

[13] S. P. K, "Neural Networks in Autonomous Vehicle Control," 1993.

[14] N. Kehtarnavaz, "Steering Control of Autonomous Vehicles by Neural Networks," 1991.

[15] S. K. Kenue, "Lanelok: Detection of lane boundaries and vehicle tracking using image processing techniques," 1989.

[16] A. Rosebrock, "A simple neural network with python and Keras," 2016. [Online]. Available: https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras/.

[17] S. Shamdasani, "Build a neural network," 2017. [Online]. Available: cited from: https://enlight.nyc/neural-network/ .

[18] V. Maru, "Autonomous Driving," 2016. [Online]. Available: cited from: https://www.slideshare.net/vivekmaru/autonomous-driving-tu-chemnitz.

[19] Z. Whang, "Self Driving RC Car," [Online]. Available:
https://zhengludwig.wordpress.com/projects/self-driving-rc-car/.