Enhancing Traffic Safety in Unpredicted Environments with Integration of ADAS Features with Sensor Fusion in Intelligent Electric Vehicle Platform with Implementation of Environmental Mapping Technology

David S. Obando Ortegon

Follow this and additional works at: https://digitalcommons.georgiasouthern.edu/etd

Part of the Controls and Control Theory Commons, Electrical and Electronics Commons, Electro-Mechanical Systems Commons, Hardware Systems Commons, Navigation, Guidance, Control, and Dynamics Commons, Power and Energy Commons, and the Signal Processing Commons

Recommended Citation
https://digitalcommons.georgiasouthern.edu/etd/2542

This thesis (open access) is brought to you for free and open access by the Jack N. Averitt College of Graduate Studies at Digital Commons@Georgia Southern. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.
ENHANCING TRAFFIC SAFETY IN UNPREDICTED ENVIRONMENTS WITH INTEGRATION OF ADAS FEATURES WITH SENSOR FUSION IN INTELLIGENT ELECTRIC VEHICLE PLATFORM WITH IMPLEMENTATION OF ENVIRONMENTAL MAPPING TECHNOLOGY

by

DAVID S. OBANDO ORTEGON

(Under the Direction of Valentin Soloiu)

ABSTRACT

A major objective on society is to reduce the number of accidents and fatalities on the road for drivers, and pedestrians. Therefore, the automotive engineering field is working on this problem through the development and integration of safety technologies such as advanced driving assistance systems. For this reason, this work was intended to develop and evaluate the performance of different ADAS features and IV technologies under unexpected scenarios. This by the development of safety algorithms applied to the intelligent electric vehicle designed and built in this work, through the use of ADAS sensors based on sensor fusion. Evaluation of AEB, PA, steering by wire, and machine learning based distance predictions, has been studied in this work bringing a contribution to driver safety and the well-being of pedestrians. Based on this work, the enhancement of distance precision of ADAS features with a percentage error of 3.89% compared to average of raw sensors data was found as well as an study of impact of color in LiDAR data quality.

INDEX WORDS: Sensor fusion, Mapping, ADAS, Integrated safety, Intelligent vehicles
ENHANCING TRAFFIC SAFETY IN UNPREDICTED ENVIRONMENTS WITH INTEGRATION OF ADAS FEATURES WITH SENSOR FUSION IN INTELLIGENT ELECTRIC VEHICLE PLATFORM WITH IMPLEMENTATION OF ENVIRONMENTAL MAPPING TECHNOLOGY

by

DAVID S. OBANDO ORTEGON

B.S., Georgia Southern University, 2021

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

COLLEGE OF ENGINEERING AND COMPUTING
ENHANCING TRAFFIC SAFETY IN UNPREDICTED ENVIRONMENTS WITH INTEGRATION OF ADAS FEATURES WITH SENSOR FUSION IN INTELLIGENT ELECTRIC VEHICLE PLATFORM WITH IMPLEMENTATION OF ENVIRONMENTAL MAPPING TECHNOLOGY

by

DAVID S. OBANDO ORTEGON

Major Professor: Valentin Soloiu
Committee: Marcel Ilie
           Jinki Kim

Electronic Version Approved:
May 2023
DEDICATION

This work is dedicated to my parents Eliseo and Angela, my sister Natalia and my fiancée Emily.

They are my biggest motivation and constant support.
ACKNOWLEDGMENTS

Six years have passed since I started my path in Mechanical Engineering far away from home. It is with much gratitude and appreciation that I want to thank the support of my mentor and professor Dr. Valentin Soloiu during all these years. Since the beginning of my experience working in his electrical/hybrid intelligent vehicles research laboratory, he has continuously guided me to become a better engineer, team player, leader and a reliable person. Motivating me to challenge myself into scenarios that I could have not ever dreamed of. Encouraging me to work towards valuable goals that have given worth to my professional experience and personal dreams such as joining Georgia Southern Honors College, participating in a study abroad opportunity with a state of the art research institute such as CARISSMA of THI in Germany, and finally helping me to direct my professional career into exceptional destinations. His ambition for me to succeed, work ethics and determination are teachings that I will always attempt to apply in my life. I would like to acknowledge the love and support of my family that have always been there for me, encouraging me to give my best for my dreams and never give up. It is because of them that I have been able to grow so much since I left home, I will always keep them as my biggest motivation to give my best in everything I do. My fiancée, Emily, has also been a fundamental part of my success, I am so thankful for the love and support she has always given me. To all my laboratory teammates throughout the years: Jonathan Randall, David Mothershed, Levi McKinney, Kody Pierce, Shaen Mehrzed, Aidan Rowell, Luke Kroeger, Tim Sutton, Austin Brant, Tyler Wiley, Cesar Carapia, Drake Grall, Richard Smith III, Amanda Weaver, Brad Willis, John Mcafee, Nicholas Dillon, James O’Hara, Tyler Strickland, Zach Davis, Eric Pernell, Cameron Perry, Mario Machado, Seth Nowak, Dr. Robert Lugner, Dr. Gerald Sequeira, Dr. Kilian Schneider, Mr. Maximilian Inderst, I want to thank them for their constant support, shared knowledge, and lifetime friendships. They have been a fundamental part of the engineer and person that I am today. To all the students that during my Master studies contributed to this work, I thank you for your support. To all my professors along my years in GSU, thank you for your contribution to my education and your expertise.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>9</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ACRONYMS</td>
<td>13</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>17</td>
</tr>
<tr>
<td>1.1 Research Motivation</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Hypothesis Statement</td>
<td>20</td>
</tr>
<tr>
<td>1.3 Criteria for success</td>
<td>20</td>
</tr>
<tr>
<td>2 LITERATURE REVIEW</td>
<td>22</td>
</tr>
<tr>
<td>2.1 Transportation systems and sources of energy</td>
<td>22</td>
</tr>
<tr>
<td>2.1.1 Electrical energy as power source</td>
<td>24</td>
</tr>
<tr>
<td>2.1.2 Liquid fuels from biomass</td>
<td>24</td>
</tr>
<tr>
<td>2.1.3 Hydrogen powered vehicles</td>
<td>25</td>
</tr>
<tr>
<td>2.2 Intelligent vehicles introduction</td>
<td>26</td>
</tr>
<tr>
<td>2.3 Advanced Driver Assistance Systems (ADAS)</td>
<td>27</td>
</tr>
<tr>
<td>2.3.1 ADAS sensors overview</td>
<td>28</td>
</tr>
<tr>
<td>LiDAR automotive technology</td>
<td>28</td>
</tr>
<tr>
<td>Cameras applied in ADAS</td>
<td>34</td>
</tr>
<tr>
<td>Radar sensors overview automotive safety</td>
<td>36</td>
</tr>
</tbody>
</table>
Ultrasonic sensors applied in ADAS ........................................... 38

2.3.2 ADAS features integrated in intelligent vehicles ....................... 40
    Blind spot detection (BSD) ................................................... 40
    Adaptive cruise control (ACC) ............................................. 41
    Parking assistance (PA) ..................................................... 42
    Automatic emergency braking ............................................. 43

2.4 Intelligent vehicles’ systems ................................................... 44
    2.4.1 Sensor fusion for mitigation of environmental impact in ADAS sensors . 44
    2.4.2 Electronic Power Steering ........................................... 46
    2.4.3 Platooning technologies applied for smart clean and effective traffic . 47

2.5 State of the art in intelligent vehicle technologies ........................ 49

3 METHODOLOGY ..................................................................... 54
    3.1 Mechanical development of electric smart navigation platform ........ 54
    3.2 Implementation & wiring of the sensors and power sources ............. 56
        3.2.1 Sensor mounts for installation of ADAS sensors .................. 56
    3.3 Integration and performance evaluation of ADAS features & IV technologies 60
        3.3.1 Automatic Emergency Braking .................................... 60
        3.3.2 Parking Assistance .................................................. 62
        3.3.3 LSTM machine learning to enhance precision on ADAS features ... 65
        3.3.4 Steering by wire ...................................................... 67
        3.3.5 Drive by wire ......................................................... 68
        3.3.6 Environmental Data Acquisition for mapping integration .......... 68
3.3.7 LiDAR testing ................................................. 69
3.3.8 Impact of color in LiDAR data ............................... 71
3.3.9 LiDAR mapping ............................................. 72
4 DATA AND RESULTS ........................................... 74
  4.1 ADAS features performance evaluation ......................... 74
    4.1.1 Automatic Emergency Braking evaluation ............... 74
    4.1.2 Parking Assistance evaluation .......................... 78
    4.1.3 LSTM Machine Learning Distance Prediction ........... 82
  4.2 Environmental impact on LiDAR performance .................. 86
    4.2.1 Color target impact in LiDAR data ..................... 88
5 CONCLUSIONS .................................................. 92
  5.1 Validation of Proposed Hypothesis ........................... 93
  5.2 Future work ................................................ 94
REFERENCES ...................................................... 96
APPENDICES ....................................................... 104
A Sensors specifications and mechanical drawings .................. 105
  A.1 Ultrasonic sensor HC-SR04 .................................. 105
  A.1.1 Wire connected direct as following ....................... 105
  A.1.2 Electric parameter ....................................... 105
  A.1.3 Ultrasonic sensor HC-SR04 mechanical drawing ........... 105
  A.2 RPLiDAR A3M1-R3 .......................................... 107
  A.2.1 Specifications ........................................... 107
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2.2</td>
<td>RPLiDAR A3M1-R3 mechanical drawing</td>
<td>107</td>
</tr>
<tr>
<td>A.3</td>
<td>MicroLiDAR SF11-C</td>
<td>109</td>
</tr>
<tr>
<td>A.3.1</td>
<td>Specifications</td>
<td>109</td>
</tr>
<tr>
<td>A.3.2</td>
<td>Mechanical drawing SF11-C</td>
<td>109</td>
</tr>
<tr>
<td>A.4</td>
<td>Velodyne VLP-16 LiDAR Sensor</td>
<td>110</td>
</tr>
<tr>
<td>A.4.1</td>
<td>Sensor Specifications</td>
<td>110</td>
</tr>
<tr>
<td>A.4.2</td>
<td>Mechanical Drawings VLP-16 LiDAR sensor</td>
<td>110</td>
</tr>
<tr>
<td>B</td>
<td>Intelligent navigation platform codes</td>
<td>115</td>
</tr>
<tr>
<td>B.1</td>
<td>Driving codes</td>
<td>115</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Forward and reverse control</td>
<td>115</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Steering by wire code</td>
<td>115</td>
</tr>
<tr>
<td>B.2</td>
<td>ADAS Codes</td>
<td>116</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Automatic Emergency Braking</td>
<td>116</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Parking Assistance</td>
<td>119</td>
</tr>
<tr>
<td>B.2.3</td>
<td>Ultrasonic sensors Arduino test</td>
<td>121</td>
</tr>
<tr>
<td>B.3</td>
<td>Live data visualization</td>
<td>122</td>
</tr>
<tr>
<td>B.3.1</td>
<td>Ultrasonic sensors live data visualization</td>
<td>122</td>
</tr>
<tr>
<td>B.4</td>
<td>Data Processing Algorithms</td>
<td>124</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>AEB Performance observations for electric platform</td>
<td>77</td>
</tr>
<tr>
<td>4.2</td>
<td>Time to collision calculations for automatic emergency braking performance analysis</td>
<td>77</td>
</tr>
<tr>
<td>4.3</td>
<td>PA Performance observations for electric platform</td>
<td>80</td>
</tr>
<tr>
<td>4.4</td>
<td>Time to collision calculations for Parking Assistance performance testing</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Percentage error comparison</td>
<td>85</td>
</tr>
<tr>
<td>4.6</td>
<td>Results LiDAR analysis</td>
<td>88</td>
</tr>
<tr>
<td>A.1</td>
<td>Specifications ultrasonic sensor HC-SR04</td>
<td>105</td>
</tr>
<tr>
<td>A.2</td>
<td>Specifications ultrasonic sensor HC-SR04</td>
<td>107</td>
</tr>
<tr>
<td>A.3</td>
<td>Specifications microLiDAR SF11-C</td>
<td>109</td>
</tr>
<tr>
<td>A.4</td>
<td>Specifications VLP-16 LiDAR sensor</td>
<td>110</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Driving fatalities in the US 2020 (J.d. 2023)</td>
<td>17</td>
</tr>
<tr>
<td>1.2 ADAS features represented in past, present and future technologies (What’s next for Advanced Driver Assistance Systems? 2023)</td>
<td>18</td>
</tr>
<tr>
<td>1.3 SAE levels of driving automation categories(SAE J3016 automated-driving graphic 2023)</td>
<td>19</td>
</tr>
<tr>
<td>2.1 U.S. Greenhouse Emissions (Peak CO2 &amp; Heat-trapping Emissions — Climate Central 2023)</td>
<td>22</td>
</tr>
<tr>
<td>2.2 Expected energy sources of transportation based on the Zero Carbon Action Plan (ITS 2020)</td>
<td>23</td>
</tr>
<tr>
<td>2.3 Components of a lithium-ion cell (What Are Lithium-Ion Batteries? — UL Research Institutes 2023)</td>
<td>24</td>
</tr>
<tr>
<td>2.4 Sources of biomass (Biomass Fuels Quality Testing - Lifeasible 2023)</td>
<td>25</td>
</tr>
<tr>
<td>2.5 Hydrogen powered vehicle (Moore-coyler 2022)</td>
<td>25</td>
</tr>
<tr>
<td>2.6 Intelligent vehicles schematic (Smart Cities World 2023)</td>
<td>26</td>
</tr>
<tr>
<td>2.7 ADAS features implemented in vehicles (Spiceworks 2022)</td>
<td>27</td>
</tr>
<tr>
<td>2.8 First LiDAR sensor developed for Apollo’s 15 mission (Abshire, n.d.)</td>
<td>29</td>
</tr>
<tr>
<td>2.9 Schematic of LiDAR functioning for a single laser pulse (Wandinger 2005)</td>
<td>30</td>
</tr>
<tr>
<td>2.10 LiDAR sensor components (Goa, Electronics and Robotics Club BITS 2021)</td>
<td>33</td>
</tr>
<tr>
<td>2.11 Environmental object detection through RGB camera (Ralph 2019)</td>
<td>34</td>
</tr>
<tr>
<td>2.12 Camera performance under harsh weather conditions (Marshall 2017)</td>
<td>35</td>
</tr>
<tr>
<td>2.13 Environmental object detection through radar sensors (Marias 2021)</td>
<td>36</td>
</tr>
</tbody>
</table>
2.14 Radar waves in vehicles (Vortex 2017) .............................................. 37
2.15 Radar implemented in vehicles (Murata Manufacturing Co., Ltd. 2021) .......... 38
2.16 Ultrasonic sensor functioning principle (javatpoint 2023) .......................... 39
2.17 Blind spot detection schematic (Blind spot detection 2023) ......................... 40
2.18 Adaptive cruise control feature (Ogbac 2020) ........................................ 41
2.19 Parking Assistance feature (KiA Press 2020) .......................................... 42
2.20 Parking Assistance feature (Automatic emergency braking 2023) ................. 43
2.21 LiDAR sensors in AV operating under harsh weather conditions (“Self-driving cars succumb to snow blindness as driving lanes disappear” 2016) ............. 44
2.22 Sensor fusion object recognition (NOVELIC 2022) .................................. 45
2.23 Steering by wire model (“Steer-by-Wire System Wins CES Innovation Award” 2021) 46
2.24 Platooning systems (Milford 2017) ....................................................... 47
2.25 Platooning systems (“Truck Platooning: The Band of Semi-Trailers” 2017) .... 48
2.26 Camera and LiDAR vision (Templeton 2019) ........................................... 50
2.27 Ford Co-Pilot 360 (Garber Automall 2018) ............................................ 51
2.28 Audi environment awareness (Inside Audi — Audi Driver Assistance — Audi USA 2023) ................................................................. 52

3.1 Complete assembly of smart navigation platform developed by the author ........ 54
3.2 Frame base and extension, (a) shows the initial frame of the vehicle platform, (b) is the point at which the vehicle was extended with a 0.3 m rod, (c) is the final frame extension used as the platform body. ........................................... 55
3.3 Ultrasonic sensor mounts 3D-Model ......................................................... 56
3.4 Ultrasonic mounts assembly in front side of the vehicle platform ................... 57
3.5 LiDAR VLP-16 sensor mount

3.6 LSTM sensor mount. a) Ultrasonic sensor HC-SR04, b) RPLiDAR sensor, c) microLiDAR SF11/C

3.7 Wiring diagram vehicle platform

3.8 Parking assistance sensors integrated in navigation platform

3.9 Automatic emergency braking range of action

3.10 Parking assistance sensors integrated in navigation platform

3.11 Parking assistance range of action

3.12 LED warning system for AEB and PA. a) Green LEDs light up if distance to obstacles is safe and do not endanger the driver, and speed keeps constant. b) Blue LEDs light up if distance to obstacles is in the warning zone. c) Red LEDs light up if distance to obstacles endangers safety of driver and pedestrian. d) RGB LED that changes color with respect to the zone detected by the ultrasonic sensors

3.13 LSTM machine learning test configuration

3.14 LSTM Machine Learning Model configuration

3.15 Electronic steering set up

3.16 Point cloud acquired

3.17 Environment point cloud acquired

3.18 Measurement range for LiDAR testing

3.19 Region of interest point cloud

4.1 Velocity vs time Automatic Emergency Braking

4.2 Distance vs time Automatic Emergency Braking

4.3 Distance and velocity vs time Automatic Emergency Braking

4.4 Time to collision comparison AEB testing runs
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>velocity vs time Parking Assistance</td>
<td>78</td>
</tr>
<tr>
<td>4.6</td>
<td>Distance vs time Parking Assistance</td>
<td>79</td>
</tr>
<tr>
<td>4.7</td>
<td>Parking Assistance distance and velocity vs time results</td>
<td>80</td>
</tr>
<tr>
<td>4.8</td>
<td>Time to collision comparison PA testing runs</td>
<td>81</td>
</tr>
<tr>
<td>4.9</td>
<td>Training results for LSTM network</td>
<td>82</td>
</tr>
<tr>
<td>4.10</td>
<td>Tester algorithm results for 2 meters</td>
<td>83</td>
</tr>
<tr>
<td>4.11</td>
<td>LSTM sensor fusion evaluation for all distances</td>
<td>84</td>
</tr>
<tr>
<td>4.12</td>
<td>LSTM sensor fusion evaluation for 2 meters</td>
<td>85</td>
</tr>
<tr>
<td>4.13</td>
<td>Percentage error and standard deviation comparison</td>
<td>86</td>
</tr>
<tr>
<td>4.14</td>
<td>Visual representation of point clouds extracted from vehicles recorded. a) BMW Series 3, b) Ford Escape, c) Honda Civic, d) Chevrolet Malibu, e) Ford Taurus</td>
<td>87</td>
</tr>
<tr>
<td>4.15</td>
<td>Number of points comparison results between vehicles</td>
<td>89</td>
</tr>
<tr>
<td>4.16</td>
<td>Intensity values comparison results between vehicles</td>
<td>90</td>
</tr>
<tr>
<td>4.17</td>
<td>LiDAR point cloud acquired from 3D environment scanning</td>
<td>91</td>
</tr>
<tr>
<td>A.1</td>
<td>Dimension drawings of the SF11</td>
<td>109</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>Performance of LiDAR system</td>
<td></td>
</tr>
<tr>
<td>$G(R)$</td>
<td>Range-dependent measurement geometry</td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>Distance</td>
<td></td>
</tr>
<tr>
<td>$\beta(R)$</td>
<td>Back-scatter coefficient</td>
<td></td>
</tr>
<tr>
<td>$T(R)$</td>
<td>Transmission factor</td>
<td></td>
</tr>
<tr>
<td>$P_o$</td>
<td>Average initial power</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>Speed of light</td>
<td></td>
</tr>
<tr>
<td>$\tau$</td>
<td>Temporal pulse length of the sensor</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>Area of primary receiver optics</td>
<td></td>
</tr>
<tr>
<td>$\eta$</td>
<td>Overall efficiency</td>
<td></td>
</tr>
<tr>
<td>$O(R)$</td>
<td>Laser beam receiver field-of-view</td>
<td></td>
</tr>
<tr>
<td>$N_j$</td>
<td>Concentration of scattering particles type j</td>
<td></td>
</tr>
<tr>
<td>$\frac{d\sigma_{j,\text{sca}}}{d\Omega}$</td>
<td>Particle’s differential scattering cross section</td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength</td>
<td></td>
</tr>
<tr>
<td>$T(R, \lambda)$</td>
<td>Transmission</td>
<td></td>
</tr>
<tr>
<td>$\alpha(R, \lambda)$</td>
<td>Extinction coefficient</td>
<td></td>
</tr>
<tr>
<td>$P_t$</td>
<td>Radar power</td>
<td></td>
</tr>
<tr>
<td>$G_t$</td>
<td>Antenna Gain</td>
<td></td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Target cross-section</td>
<td></td>
</tr>
<tr>
<td>$A_e$</td>
<td>Antenna of effective aperture area</td>
<td></td>
</tr>
</tbody>
</table>
$S$  Distance calculated
$v$  Speed of sound
$t$  Ultrasonic pulse time
$v_r$  Ego-vehicle speed
$a_{TV}$  Acceleration of vehicle
$v_f$  Final velocity of ego-vehicle
$v_o$  Initial velocity of ego-vehicle
$\delta t$  Time difference
$Z$  Normalized data
$I$  Intensity values
$\mu$  Mean
$\sigma$  Standard deviation
ACRONYMS

**ADAS**  Advanced Driver Assistance Systems  
**AEB**  Automatic Emergency Braking  
**ACC**  Adaptive Cruise Control  
**PA**  Parking Assistance  
**BSD**  Blind Spot Detection  
**LKA**  Lane Keeping Assistance  
**LiDAR**  Light Detection and Ranging  
**Radar**  Radio Detection and Ranging  
**SLAM**  Simultaneous Localization and Mapping  
**V2X**  Vehicle to Everything  
**V2V**  Vehicle to Vehicle  
**GHG**  Greenhouse Gas  
**LDVs**  Light-Duty Vehicles  
**EPA**  Environmental Protection Agency  
**ASIRT**  Association for Safe International Road Travel  
**EC**  European Commission  
**NASA**  National Aeronautics and Space Administration  
**CC**  Cruise Control  
**CACC**  Cooperative Cruise Control  
**GPS**  Global Positioning System
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>MPC</td>
<td>Model Predictive Control</td>
</tr>
<tr>
<td>LQR</td>
<td>Linear Quadratic Regulator</td>
</tr>
<tr>
<td>TTC</td>
<td>Time to Collision</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Research Motivation

In the last years, growth of development and production of electrical and intelligent vehicles has been fundamental in the improvement of safety and autonomous technologies worldwide. The creation of these technologies originated due to the high number of fatalities and accidents on the road in different countries. Just in the US during 2020, a total amount of 35,766 crashes occurred resulting in 38,824 victims (J.d. 2023). Two main divisions for intelligent vehicles have been defined throughout time, safety and autonomy.

Figure 1.1: Driving fatalities in the US 2020 (J.d. 2023)
Safety is a must at the time of developing any vehicle, for this reason Advanced Driver Assistance Systems (ADAS) features have been introduced. Technologies such as Automatic Emergency Braking (AEB), Adaptive Cruise Control (ACC), Parking Assistance, Blind Spot Detection, Lane Keeping Assistance, etc., rely in multiple exteroceptive sensors that allow the ego-vehicle to accurately scan the environment and recognize features in the driving environment. LiDAR, Radar, cameras, and ultrasonic sensors are the principal sensors integrated in vehicles nowadays. Most of these sensors work under different physical principles including sound, light and radio waves under the time of flight theory. Furthermore, sensor fusion allows to fuse the signals coming from these sensors in order to have a more reliable environment recognition enhancing object recognition in the road such as pedestrians, other vehicles, traffic signs, etc. Using different sensors is imperative to maintain a safety level at all times neglecting exterior conditions that may impact sensor performance, therefore, functionality of ADAS features mentioned.

Figure 1.2: ADAS features represented in past, present and future technologies (What’s next for Advanced Driver Assistance Systems? 2023)
Autonomy relies in the introduction of novel technologies that while using sensor data, is able to perform path planning, road recognition, platooning under multiple conditions based on technologies such as Simultaneous Localization and Mapping (SLAM), Machine Learning, Sensor Fusion, etc. For the functioning of such systems, development of technologies to communicate between vehicles have been developed such as V2X and V2V; allowing vehicles to share sensor data and environment recognition to enhance safety between them. Technologies integrated in these vehicles rely on ADAS features making sure that while driving the car autonomously, the drive will always be safe for both driver and pedestrians at all times. The levels of automation in vehicles are divided into 6 different categories depending on the technologies implemented in the vehicle.

![SAE J3016™ Levels of Driving Automation](image)

**Figure 1.3:** SAE levels of driving automation categories (SAE J3016 automated-driving graphic 2023)
1.2 Hypothesis Statement

If vehicle safety has been constantly improved by development and integration of different safety technologies such as Advanced Driver Assistance Systems (ADAS), then it is possible to develop a series of intelligent vehicle technologies and ADAS features based on sensor fusion in one smart electric platform studying sensor performance for their integration in these technologies. Three different parameters will be taken into consideration in order to validate this hypothesis:

- Performance evaluation of parking assistance and automatic emergency braking based on methodology performed by the American Automobile Association (AAA) will reflect a constant performance on these safety features with a efficiency performance of around 90%.

- Machine learning LSTM model performance aimed to be used for enhancement in precision of distance measurement based on three different exteroceptive sensors, will reflect a better performance in comparison with average of raw data recorded.

- Influence of size and color of targets in the environment in LiDAR data will reflect how the performance of the LiDAR is optimum with light colors and compromised with darker colors, validation studied by relating the general LiDAR equation and the data recorded in terms of intensity and number of points.

1.3 Criteria for success

Integration and performance evaluation of multiple novel safety and autonomous technologies including ADAS features such as automatic emergency braking, parking assistance, in a scale electric vehicle navigation platform enabling ADAS features to enhance safety for driver, passenger and pedestrians in the environment.

Enhancement of these ADAS features mentioned with integration of sensor fusion with the combination of three different exteroceptive sensors (microLiDAR, ultrasonic sensor, RPLi-
introduction to accurate 3D-mapping on the road while driving in combination with camera for sensor fusion (video of the environment and 3D LiDAR mapping). Additionally, study of the performance of the LiDAR sensor with respect of the environmental conditions and targets.

This with the aim of enhancing safety on the road and proof autonomous driving scenarios in real time through intelligent vehicles technologies including steering and driving by wire, live data visualization and sensor fusion.
2.1 Transportation systems and sources of energy

Transportation has a major impact in economy worldwide. Distribution of goods and services nationally and internationally requires high amounts of resources. Even though mobility technologies have been growing up in the last years, great impact has been done in relation with congestion, pollution and accidents (Jimenez, n.d.). Modern transportation is dependent on oil to a 61.4%, with growth in transportation systems concerns about peak oil, oil price volatility and energy security have started. Considering all these issues from transportation systems, the most concerning one is the GHG (Greenhouse Gas) emissions (Banister et al. 2011). Light-duty vehicles (LDVs), that includes passenger cars and light trucks, covers three fifths of the total energy use and GHG emissions in the U.S. Currently, fuel economy is 26 mpg based on Environmental Protection Agency (EPA) vehicle certification test. By 2035, new standards will be set for 50 mpg for midsize cars and 75 mpg on road if cars have a hybrid-electric drive train.

![Graph showing U.S. Greenhouse Gas Emissions](image)

**Figure 2.1:** U.S. Greenhouse Emissions (*Peak CO2 & Heat-trapping Emissions — Climate Central 2023*)
Alternative fuels are also an important option in order to accomplish reductions in GHG emissions as well as oil use. Some of these options could be electricity, liquid fuels from biomass and hydrogen. Alternative sources of energy for transportation systems is a must for the transition into new clean energies that can guarantee a cleaner environment reducing pollution and emissions worldwide.

The transition of energy sources also come with the integration and development of safety technologies that help to reduce traffic collisions and fatalities on the road. Intelligent vehicles’ technologies are being constantly developed, improved, and integrated in new vehicles produced in industry. Subsequently, the automotive industry has the challenge in the coming years of making transitions of not just the energy sources to power transportation systems but also to make them safer for drivers and pedestrians.

![Energy Sources for Transportation](image)

**Figure 2.2:** Expected energy sources of transportation based on the Zero Carbon Action Plan (ITS 2020)
2.1.1 Electrical energy as power source

Thanks to the production of lithium-ion batteries as well as hybrid vehicles, a range between 10 and 40 mpg is incremented from normal vehicles in mileage consumption. Even though there is an improvement in the mile consumption of these vehicles, the battery cost and life time are issues that are still present. Electric cars will have an important impact in the GHG emission in the future. Nevertheless, the infrastructure required to accomplish such technology nowadays covering long distances between destinations is a work in progress that will take a good number of years. For this reason, the transition into this technology could be slow.

![Figure 2.3: Components of a lithium-ion cell (What Are Lithium-Ion Batteries? — UL Research Institutes 2023)](image)

2.1.2 Liquid fuels from biomass

These kind of fuels offer reduction of GHG emissions from the vehicles in which it could be applied. However, costs for these kind of fuels is high and there is an option to impact the use of land in the country. If well administered, the use of this fuel can bring 20 to 30 billions of gallons of gas from biomass replacing 19 to 25% of gas use by 2035. It is a very interesting option in terms of environmental impact.
2.1.3 Hydrogen powered vehicles

Hydrogen as a power source for vehicles do not produce GHG emissions. The use of hydrogen in vehicles also shows ranges of 300 miles in order to refuel (similar to actual cars). Nevertheless, use of hydrogen is very expensive and a new fuel system and infrastructure for the vehicles is required. Bringing new challenges for the adaptability of such fuel systems (Greene, Baker, and Plotkin 2010).
2.2 Intelligent vehicles introduction

Safety is one of the most important parameters taken into consideration in intelligent vehicle systems production. Every year, there is a considerable number of accidents and fatalities on the road being most of them caused due to human errors. Such accidents enclose wrong perception, decision-making, lack of attention on the road and in other cases late reaction time. For this reason, intelligent vehicles are based on a series of sensors that allow the vehicles to perceive the environment and as a result prevent accidents. Some advantages for these vehicles include more efficient travel, making the insurance fees cheaper since accidents will decrease eventually with the use of these technologies. Furthermore, the environment is also impacted positively with the integration of such technologies, taking into consideration platooning technologies where by vehicles driving with a constant velocity and a close distance between them, gas consumption can decrease improving gas mileage in overall.

![Intelligent vehicles schematic](image)

*Figure 2.6: Intelligent vehicles schematic (Smart Cities World 2023)*
Communication between vehicles will also be accomplished through technologies such as v2v sharing information of the road between vehicles enhancing safety and driving environment resulting in more optimal decisions of the vehicles. Once the data set recorded from all vehicles is reached, the network with all these information will create a more efficient driving environment and furthermore, issues like congestion could be solved. With the introduction of such intelligent vehicles, great advantages to society are offered such as elderly citizens being able to travel through an autonomous vehicle without relying on other drivers. Furthermore, transportation of goods and service in an autonomous systems is an advantage that benefits society in overall (Kala 2016).

2.3 Advanced Driver Assistance Systems (ADAS)

Based on the Association for Safe International Road Travel (ASIRT), approximately 1.35 million people die in road accidents worldwide yearly. Just in the United States, 38,000 people are part of this group and 23,000 in Europe. With this global concern, the U.S. government and the European Commission (EC) have created a road map to reduce accidents. One of these initiatives is the project Vision Zero which vision is to achieve zero deaths in road accidents by 2050.

How ADAS Works

![Figure 2.7: ADAS features implemented in vehicles (Spiceworks 2022)](image)
Based on this background, using ADAS technologies will safe lives in the future. ADAS sensors are more robust and have more efficiency compared to human response time. Furthermore, humans in some cases suffer drowsiness from driving after a long day or driving long distances. Furthermore, humans are also prone to emotions that can impact driving performance such as tiredness, anger, etc. Sensors, compared to humans will have a constant time response and will always react to possible accidents on the road. Time response in humans on possible crashing scenarios is variable and not efficient (Pathrose, Plato 2022).

2.3.1 ADAS sensors overview

Safety technologies applied in autonomous vehicles rely on sensors that scan and recognize the environment based on different physical principles such as light, sound, electromagnetic field, etc. These sensors play an important role in the vehicles used nowadays. Not only they scan the environment at all times during driving, but also commanding actions over different scenarios taking control of the vehicle if necessary or desired. Industry and researchers keep developing safety and autonomous technologies based on these exteroceptive sensors allowing a safer driving environment for drivers, passengers, and pedestrians. Some of the principal sensors used in vehicles and their functionalities are as follows:

LiDAR automotive technology

Light detection and Ranging (LiDAR) is a technology that uses the time of flight principle to scan the environment and create 2D or 3D map of the environment called point cloud. LiDAR uses the time of flight principle through infrared light laser beams to map the environment and record the distance of all surroundings within its range. The first appearance of LiDAR was back in 1971 when NASA used this technology with the purpose of scanning the lunar surface during the Apollo 15’s mission (J. Liu et al. 2018). With time, LiDAR sensors evolved with more lifetime efficiency, resolution and mass of lasers, allowing NASA to scan the surface topography of Mars.
with more than 600 million measurements. Subsequently, this technology has been used to scan Earth’s topography and different asteroids in space (Abshire 2009).

These sensors have a very precise functioning system since it relies on the speed of light to record data. Furthermore, it is able to record the exact position of every obstacle around of the ego-vehicle. The basic set up of a LiDAR sensor is made up by a transmitter and a receiver. Pulses of infrared light are sent to the environment, a telescope within the LiDAR sensor receives all the backscattered photons received from the environment. This system is then complemented by an optical analyzing system that selects specific wavelengths out of the recorded light from the medium. In consequence, this recorded light is sent to a detector where is converted into an electrical signal. From these points it is possible to record the distance represented in X, Y and Z coordinates, intensity values and depending on the sensor reflectivity values (Wandinger 2005). There are multiple factors present in the environment that can affect the signal recorded
Figure 2.9: Schematic of LiDAR functioning for a single laser pulse (Wandinger 2005)

The Lidar equation is divided into four main components that represent all the impact factors on the received signal of the LiDAR sensor. The first component is represented by the letter $K$ and considers the performance of the LiDAR system; the second factor is $G(R)$ which describes the range-dependent measurement geometry. It is important to consider that these two first components are able to be changed and determined by users of the sensors; the third factor is the term $\beta(R)$ which represents the back-scatter coefficient at a distance $R$. This factor is determined by the capability of a given environment to return the signal back to its origin (LiDAR sensor); the last factor is the transmission $T(R)$ which is referred to the amount of lost light in its path being emitted and received by the sensor. These last two factors are to be studied and determined depending on the operation medium of the sensor. Given these factors, the general form of the LiDAR equation
is given by:

\[ P(R) = KG(R)\beta(R)T(R) \]  

(2.1)

The system factor \( K \) of the LiDAR equation is denoted as:

\[ K = P_0 \frac{c\tau}{2} A\eta \]  

(2.2)

Where \( P_0 \) is the average initial power of a single laser pulse of the sensor, \( c \) is the speed of light, \( \tau \) is the temporal pulse length of the sensor, \( A \) is the area of the primary receiver optics that collects the back-scattered light from the environment and \( \eta \) is the overall efficiency of the system. The factor 1/2 is present in the equation as a representation of the time of flight behavior of the signal. Based on this equation, there is a proportional relation between the power of the transmitted signal of the sensor to the environment and the received signal. This received signal can also be represented as the resultant intensity recorded by the sensor. Thus, it is understood as the quality of the point cloud recorded.

The geometric factor of the LiDAR equation is represented as:

\[ G(R) = \frac{O(R)}{R^2} \]  

(2.3)

Where \( O(R) \) is the laser beam receiver-field-of-view overlap function and it describes the fraction of laser beam cross section within the receiver field of view with respect of the range \( R \) (Hey et al. 2011). As seen in the third equation, the inverse relation is related between the range of the sensor and the received signal, where it is shown how the distance between the target object and the sensor has an important influence in the back-scattered signal. The higher the distance between the sensor and the target, the less will be the amount of signal received back. \( R^2 \) is related to the receiver telescope area which is represented with a sphere surface with radius \( R \) enclosing
The atmospheric parameter that determines the strength of the LiDAR signal is the back-scatter coefficient $\beta(R, \alpha)$. This parameter defines what amount of light is being back-scattered to the sensor. This coefficient is specifically the value of the scattering coefficient for the angle $\theta=180^\circ$. This parameter is represented as follows:

$$\beta(R, \alpha) = \sum_j N_j(R) \frac{d\sigma_{j,\text{sca}}}{d\Omega}(\pi, \alpha)$$  \hspace{1cm} (2.4)

In the equation above, $N_j$ is the concentration of scattering particles of type $j$ illuminated by the laser, $d\sigma_{\pi,\lambda}/d\Omega$ is the representation of the particle’s differential scattering cross section for the backward direction with respect to the wavelength $\lambda$. Different environments could present multiple scatterers such as pollution particles of sulfates, organic compounds, pollen, etc. All of these different scatterers have an impact in the performance of the back-scattered signal tending to decrease it since the environment would not be ideally clean.

The last parameter that makes up the LiDAR equation is the transmission term $T(R)$ which has a value that ranges between 0 and 1 is represented by:

$$T(R, \lambda) = \exp[-2 \int_0^R \alpha(r, \lambda)dr]$$  \hspace{1cm} (2.5)

This equation is based on the Lambert-Beer-Bouger law for LiDAR. This equation covers the complete travel that the infrared light follows from leaving the sensor until coming back to its origin. The factor 2 represents this time of flight behavior. This equation represents the sum of all transmission losses and is denominated light extinction where $\alpha(R, \lambda)$ is the extinction coefficient.

Combining all these equations mentioned above, the general LiDAR equation can be represented as (Wandinger 2005):

$$P(R, \lambda) = P_0 \frac{cT}{2} A\eta \frac{O(R)}{R^2} \beta(R, \lambda) \exp[-2 \int_0^R \alpha(r, \lambda)dr]$$  \hspace{1cm} (2.6)
Equation 6 summarizes all the parameters that affect the back-scatter signal of a LiDAR sensor. These parameters that impact the signal are as mentioned above, the initial average power emitted from the sensor to the environment, the geometry dependency of the composition of the LiDAR sensor, the range of the targets in the environment, environmental factors such as different particles that can be found in the environment, and transmission losses that depend on the extinction light coefficient, among others.

Disadvantages of LiDAR sensors include the effect that colors have on the quality of the point cloud recorded (Sequeira et al. 2021). The backscattered signal recorded from the environment is variable depending on the colors and materials of the surroundings of the ego-vehicle. Darker colors absorb most of the infrared energy transmitted from the sensor and tend to reduce the backscattered signal to the sensor. Causing a reduction on quality (intensity values) of the point cloud and possibly causing to reduce the perception of vehicles with dark colors, putting in danger the driver and pedestrians around. Furthermore, harsh weather conditions also have a big impact on the sensor performance reducing visibility and environment perception.
LiDAR data can be used to develop intelligent vehicle technologies. For instance, authors in (Alzu’bi et al. 2020) developed a predictive cruise control feature that detects objects surrounding the vehicle, calculating the grade of the road, and path planning the route in drivable areas. With the use of this information recorded, optimal speed is set in the vehicle as well as path planning. The combination of detecting obstacles and estimating the road grade allowed the vehicle to accelerate and decelerate based on the scenario and the conditions at which the ego-vehicle is under.

**Cameras applied in ADAS**

Cameras are devices that scan the environment through a lens that records the light reflected in objects and then processes it to turn it into a color picture. Cameras work in a very similar way as the human eye by filtering light through color filters. Light is processed and then filter into red, green and blue pixels creating the colors perceived on the screen. Cameras are used for different algorithms in ADAS features for object recognition. Some of the applications for cameras in ADAS are autonomous emergency braking (AEB), parking assistance, blind spot detection, pedestrian detection, lane keeping assist, etc. One of the biggest advantages of using cameras in ADAS features is that there is capability of object recognition compared to sensors like Radar and LiDAR that do not have this capability.

![Image of Environmental object detection through RGB camera](Figure 2.11: Environmental object detection through RGB camera (Ralph 2019))
Cameras are also cheaper compared to the high cost that could have a LiDAR sensor, and for this reason are the sensors that has been implemented the most in the modern automotive field. Disadvantages on cameras include the lack of depth perception that other sensors have. Moreover, weather has a big impact on camera performance and reduces object recognition under harsh conditions such as rain, fog and snow.

![Camera performance under harsh weather conditions](image)

**Figure 2.12:** Camera performance under harsh weather conditions (Marshall 2017)

Multiple applications of cameras in the automotive field have been invented and applied nowadays. The reason of why cameras are so important and fundamental in intelligent vehicles systems is that they output a high accuracy graphical representation of the environment, allowing object recognition through AI algorithms that can determine obstacles in the environment and classify them for specific driving tasks (Roriz, Cabral, and Gomes 2022). Object’s speed and location was calculated in (Simacek et al. 2021) from a technique called videogrammetry in order to recreate accident collisions of vehicles understanding vehicle dynamics attempting to minimize risk of a probable collision.

Different lightning conditions can affect the object recognition on the road, impacting the performance of autonomy or safety systems. For this reason, Authors in (Huber et al. 2022)
presented an approach to recognize and track critical light conditions on cameras by meanwhile extracting disturbing parameters improving the performance of object recognition under this critical conditions. Different methods for color and contrast were used in this object recognition algorithm by using different color measurement theories and contrast processing. Authors in (Soloiu et al. 2017) developed a method based on color detecting vision that was developed to optimize lane following for a small scale model of an autonomous vehicle. As a result, it was possible to enable driving in the small platform by navigating a simulated road despite variations of lightning conditions and starting positions.

**Radar sensors overview automotive safety**

Radar sensors operate through radio signals. One of the biggest advantages of these sensors is vehicle tracking and velocity calculation of vehicles around the ego-vehicle. Radar sensors are not affected by weather conditions which makes them optimal to use under multiple driving scenarios without impacting their performance. Radar sensors have a lower range compared to LiDAR sensors and is not able to classify objects in the environment. Nevertheless, applications of Radar sensors in ADAS features in fundamental in technologies such as AEB, Adaptive Cruise Control (ACC), rear collision warning, etc.

![Figure 2.13: Environmental object detection through radar sensors (Marias 2021)](image)
First radar technologies were created early in the 70s. Companies and research institutes around the globe starting using these devices to develop avoidance collision systems. Due to the time, size of the apparatus and cost, no sensor made it into the automotive market until 1998. The introduction of this sensor in the automotive field came with the Distronic as an extra option in Daimler S class. Other companies such as Jaguar, Nissan, and BMW continued with the implementation of such sensors in their vehicles. Starting in 2003, most of the car manufacturers had radar sensors integrated (Hasch et al. 2012). There are multiple factors that can impact the performance of radar sensors, for this reason, (Skolnik 2001) represents radar behavior on a equation that describes radar performance and the parameters that affect its functioning.

\[ P_r = \frac{P_t G_t \sigma}{4\pi R^2 \cdot 4\pi R^2 A_e} \]  

(2.7)

Where the first factor of the equation represents the power density of the sensor at a specific distance \( R \) with a radar radiating a power \( P_t \) watts by the use of an antenna with gain \( G_t \). The \( \sigma \) variable in the second numerator represents the target cross-section in square meters. The denominator represents the divergence on path of the electromagnetic radiation. The first denominator represents the divergence of the outward path while the second denominator accounts for the divergence of the inward path of backscattered signal traveling back to the sensor. The antenna of effective aperture area represented as \( A_e \) intercepts a portion of the power with an amount that depends of the product of the three factors in this equation.

Figure 2.14: Radar waves in vehicles (Vortex 2017)
Figure 2.15: Radar implemented in vehicles (Murata Manufacturing Co., Ltd. 2021)

Ultrasonic sensors applied in ADAS

Ultrasonic sensors are widely used in the automotive industry in close-range applications due to their detection distance (max range of 4 meters) and time. These sensors work under the time-of-flight principle, where ultrasonic waves at 40kHz travel from the trigger sensor transducer and comes back into the echo transducer, measuring the time it takes for the backscattered signal to come back. This frequency of the ultrasonic waves is so high that humans are not capable to perceive them. Applications of these sensors are possible due to the versatility under harsh environmental conditions such as rain, fog, transparent obstacles, among others. Furthermore, these sensors are well known for their low cost and reliability (Tsujii et al. 2022). The equation used in ultrasonic sensors in order to find the distance to obstacles in the environment is:

$$S = \frac{vt}{2}$$  \hspace{1cm} (2.8)

Where $S$ is the distance calculated, $v$ is the speed of sound at which the ultrasonic wave
is traveling (343 m/s), and \( t \) is the time that the ultrasonic wave takes to travel to environment and come back. This formula is divided by two since the distance measured is exactly half of the trajectory traveled by the ultrasonic wave.

![Ultrasonic sensor functioning principle](javatpoint_2023)

**Figure 2.16:** Ultrasonic sensor functioning principle (javatpoint 2023)

Different safety applications have been developed to reduce collisions and keep drivers, passengers, and pedestrians safe under all driving circumstances including adaptive cruise control, blind spot detection, and automatic emergency braking. Radar has different applications in ADAS systems, and with the evolution of such sensors, more safety technologies keep being developed worldwide. Automotive safety research centers around the world focus on testing and development of safety features, estimating sensor performance, limitations, and functionalities of features developed. Authors in (Sezgin et al. 2022) developed a deep learning approach to estimate pedestrian behavior with point clouds of radar data. Data sets of radar data were recorded in the CARISSMA research institute and were trained based on the PointNet++ deep learning architecture. A result of this study was the development of a safety feature that could recognize the walking motion of a pedestrian classifying whether the pedestrian was walking on a straight direction or changing it by just analyzing one frame of radar data.
2.3.2 ADAS features integrated in intelligent vehicles

Based on the overview shown above, researchers and industry around the world have developed multiple technologies that are integrated in vehicles enhancing their safety for multiple driving scenarios including assistance for changing lanes, detection of bad positioning within the lane limits while driving, detecting obstacles in the blind spot of the vehicles, measuring rear distance for parking assistance, keeping a constant velocity while driving keep a safe distance to obstacles in front, etc. This section will give an overview of the development and integration of such systems in vehicles. Sensor fusion also has a fundamental role in these functions due to the performance enhancing resulted from combining different signals of sensors into one main function, allowing ADAS features to work efficiently under different environmental conditions.

**Blind spot detection (BSD)**

![Blind spot detection schematic](Blind spot detection 2023)

This feature has been developed to detect vehicles approaching the ego-vehicle from angles that the driver is not allowed to see. The principal objective of this feature is to warn the driver when there is a vehicle in the blind spot. Different automotive companies rely on just radar sensors for BSD (Blind spot detection). However, radar sensor performance is impacted by snowy weather, therefore, the performance of the sensor and ADAS feature is impacted considerably. Different methodologies have been adapted to different ADAS sensors in order to make the function reliable and effective. (S. S. Gale Bagi et al. 2019) provided five different main components
that take part in the development of BSD, being software, sensors, object detection, data fusion and control. Furthermore, is analyzed the performance of this feature with different sensors being camera, LiDAR, radar and ultrasonic sensors. (A. Naik et al. 2020) shows a LiDAR based BSD where the sensor has a region of interest (ROI) programmed to detect obstacles in the blind spots of the vehicle. Furthermore, safety zones are determined based on distance. This blind spot detection algorithm informs the driver once an obstacle has crossed the safety zone within the blind spot ROI.

**Adaptive cruise control (ACC)**

![Figure 2.18: Adaptive cruise control feature (Ogbac 2020)](image)

Adaptive cruise control (ACC) is a system that allows the ego-vehicle to drive at a constant velocity chosen by the driver while keeping a safe distance from vehicles in front. It reduces driving load trying to keep the vehicle at a constant speed. The base of this technology is cruise control (CC) which just keeps the vehicle running at a constant speed. ACC adapts to the environment instead of needing to monitor vehicles in front the ego-vehicle reducing speed while cruise control is activated. This technology improves comfort and safety of drivers and pedestrians. (J. Wu
et al. 2022) considers a variable headway time distance methodology that considers the acceleration of the front of the vehicle in order to determine the vehicle-to-vehicle safe distance model. (U.S. Federal Highway Administration 2016) presents a new methodology for more efficient CC technologies named cooperative adaptive cruise control (CACC) where velocity information of the vehicles is shared in a V2V communication system. With velocity data shared between vehicles, it is easier for the system to detect and track different vehicles in the environment. In consequence, the application of CACC enhances the traffic flow allowing vehicles to be closer with each other while driving which produces taking more advantage of the road space while reducing traffic and making driving safer and efficient.

Parking assistance (PA)

![Figure 2.19: Parking Assistance feature (KiA Press 2020)](image)
Parking assistance is an ADAS technology developed to give assistance to the driver avoiding possible collisions while reversing the vehicle. It is based on ultrasonic sensors due to their excellent distance measurement at short range applications. (Wada, Yoon, and Hashimoto 2003) proposes a parking driver assistance system that based on human interface module, where the system warns the driver for a possible collision but also creates a path to follow. (Park et al. 2008) described an ultrasonic sensor functionality where multiple echoes are being acquired from the ultrasonic waves propagated to the system with the aim of detect parking space more accurately.

Automatic emergency braking

![Figure 2.20: Parking Assistance feature](Automatic emergency braking 2023)

Automatic emergency braking is a safety feature where the sensors (combination of radar and cameras, other cases LiDAR), detect the ego-vehicle is approaching at a dangerous speed an obstacle in front in a steady position. Once the detection is done, the system proceeds to apply breaks for the driver. It is faster and more consistent than human reaction time, for this reason
it reduces the chances of a crash or fatality. This technology uses time to collision (TTC) as a trigger to know at what exact moment AEB needs to be engaged. Usually, higher TTC represent a safer system since there is more time to reduce the speed of the vehicle and therefore, avoid the crash. Nevertheless, lower numbers of TTC allow less risk of false positive braking in unnecessary scenarios (Sidorenko et al. 2022).

2.4 Intelligent vehicles’ systems

2.4.1 Sensor fusion for mitigation of environmental impact in ADAS sensors

![Figure 2.21: LiDAR sensors in AV operating under harsh weather conditions ("Self-driving cars succumb to snow blindness as driving lanes disappear" 2016)](image)

ADAS sensors used in the automotive field operate in multiple environmental conditions such as rain, hail, snow, and fog; impacting the sensor performance and limiting the field of view for each sensor and therefore, impacting the performance of safety and autonomy features implemented in the vehicle. Environmental harsh conditions have a considerable impact in the performance of the sensors, once the precipitation particles in the environment are smaller than 6mm,
sensors signals are subjected to Mie Scattering. Mie Scattering produces the signal of the sensors to produce false signals that can result in masked object recognition affecting performance of ADAS features. For instance, the pulsed signal from the sensors is absorbed by the water droplets in the environment which results in attenuation. This behavior affects the back-scattered signal of sensors such as LiDAR and radars creating false readings and distorting the sensors’ back-scattered signal (Vargas et al. 2021).

Precipitation affects environmental perception in cameras reducing visibility of objects in the environment impacting object recognition putting in risk drivers, passengers and pedestrians creating possible collision scenarios. On the other hand, ultrasonic sensors still work under raining conditions since the ultrasonic waves sent to the environment are not impacted by precipitation conditions. Nevertheless, the speed of sound is dependent on relative humidity and temperature of the environment. Therefore, readings of the obstacles can reduce accuracy based on environmental changes. For this reason, sensor fusion is a very useful technology that fuses the signals of the sensors used in the vehicle and gives a reliable output of sensor data.

Figure 2.22: Sensor fusion object recognition (NOVELIC 2022)
Sensor fusion is an approach to record data from different sensors and combine it aiming to have more reliable data from scanning the environment rather than relying from just one sensor. For instance, cameras are the only sensors that can perform object recognition from the environment. However, a single camera is not capable of calculating the distance to these obstacles recognized. For this reason, sensor fusion of cameras and LiDAR sensors allows the ego-vehicle to recognize obstacles and the environment but also calculate their distance with respect to the vehicle (Kocić, Jovičić, and Drndarević 2018). This complementary technologies enhance the safety of the drivers and pedestrians in the driving environment. It is imperative to show the importance of this technology, in cases where distance sensors are being used implementation of short range and long range distance sensors will allow to measure obstacles from all necessary ranges.

2.4.2 Electronic Power Steering

![Steering by wire model](image)

*Figure 2.23: Steering by wire model (“Steer-by-Wire System Wins CES Innovation Award” 2021)*

Electric power steering has been introduced in intelligent vehicles as a novel technology for steering. The importance of this technology involves the simplicity of its operation. Being all
based on the readings of a steering angle sensor that reflects the angular motion on the steering wheel and then it is replicated in the motor for the front wheels. Authors in (J. Svensson 2015) explain different parameters that need to be considered at the time of designing such a system. This technology relies in different sensors that can track the position of the steering wheel and then replicate it in the wheels. IR sensors, hall sensors, optical readings, potentiometers and rate sensors are the options to chose to create the power steering.

2.4.3 Platooning technologies applied for smart clean and effective traffic

![Platooning systems (Milford 2017)](image)

Platooning systems is the formation of a series of autonomous vehicles that follow each other with constant speed as well as keeping a safe distance between them. The first platooning system developed comes from the 1980s, with progress around the world with other creations of platoon-
ing systems developed such as GDC, SARTRE and Energy-ITS. Some advantages of platooning systems include enhanced fuel efficiency as a result of reduced air drag, improvement of highways capacity since the distance between vehicles is reduced and enhanced safety on drivers since there is a reduction of distraction factors in the environment. There is a constant communication of all the group of vehicles during the travel. The lead car shares with the group the different conditions on which its performing including speed and direction and all the vehicles on the group respond the the lead’s vehicle movement (Matching breaking and acceleration).

![Diagram](image)

**Figure 2.25:** Platooning systems (“Truck Platooning: The Band of Semi-Trailers” 2017)

These communication system use different sensors capable of sharing information from one vehicle to another, being Bluetooth and wireless GPS and radar-sensing systems the most common in this technology. 5G communications assist in terms of the volume of data processed and shared within the group that is using platooning as a safety option. Some advantages of this technology are improving of the aerodynamic effectiveness and performance increasing capacity of roads and also providing a more steady-state traffic flow; while vehicles are ”drafting” each
other, fuel consumption is reduced and with all the cars traveling at close distances to each other, efficiencies can be gained when vehicles are moving away from traffic lights at the same time and speed; in the future, drivers would be able to join a platoon system that may function like long trains of vehicles on highways where cars would become driverless and drivers would be free to do work or rest while the drives by itself; and lastly, this modern technology can reduce considerably the number of crashes in highways. However, no computer system is perfect and can commit mistakes (Driving Tests Resources 2014).

(Ge et al. 2022) Explains that one of the primary control factors in platooning control is developing a cooperative control strategy where all the vehicles within the platooning formation are capable to create and maintain a string formation with constant distance between drivers and constant speed. Such technology described relies on ACC which works based on sensors such as Radar and LiDAR. For this reason, the distance between vehicles is not close enough between vehicles and therefore, highway capacity utilization is not used as its best performance capability. (Li, Wang, and Zheng 2022) investigates different formations of autonomous vehicles and their impact in traffic performance based on ACC and optimization for efficient platooning formation in mixed traffic flow.

2.5 State of the art in intelligent vehicle technologies

Autonomous vehicles are considered a promising automotive field with a special introduction to new technologies that will improve in different perspectives mobility worldwide including innovative electrical/hybrid power systems, safety features such as ADAS with implementation of sensor fusion systems, communication between vehicles, among others (Zhao et al. 2021).

Application of novel technologies including artificial intelligence and data processing of big data sets have made an impact in autonomous navigation technologies. Application of these technologies resulted in the development of control technologies that influence factors such as environment perception and positioning, decision planning and execution control. These technolo-
gies are a reflection of how drivers perceive the surroundings, take decisions based on this data and then take action at driving. Environment perception and positioning are related to human senses, the way in which the environment is perceived including traffic lights, pedestrians, traffic signs, surrounding vehicles, obstacles, etc; decision planning is related to the human brain, analyzing and processing all data acquired allowing the driver to plan maneuvers during driving; and lastly execution control is related to hands and feet of the driver, taking action based on the environment perceived and processed. Some of the methods developed and applied in autonomous vehicles nowadays rely on control technologies based on pure pursuit, Stanley, PID, model predictive control (MPC), and linear quadratic regulator (LQR) (Yao et al. 2020).

Figure 2.26: Camera and LiDAR vision (Templeton 2019)

Advanced driving assistance systems (ADAS) features have been improved by multiple researchers and industry bringing an important contribution to drivers and pedestrians safety reducing numbers of accidents and fatalities on the road. Researchers from FEV North America Inc., (Alzu’bi et al. 2021) developed a lane changing assistance algorithm based on sensor fusion through the use of camera lane changing approach and LiDAR based object recognition. The al-
gorithm developed in this work allowed the vehicle to recognize the lanes in the road and driving
the vehicle exactly at the middle of the lane while controlling the speed of the vehicle based on
the obstacles detected in the environment. The sensor fusion is represented in this system as the
camera takes care of recognizing parameters in the road such as objects, lane lines, curvature of the
road, etc. The LiDAR sensor tracks objects in the environment and focuses just on the obstacles
so that the vehicle is capable of recognizing them and adjust its velocity based on this information
collected.

Ford is well known for its important contribution to safety on the road with innovative
technologies integrated in their vehicles. Co-Pilot 360 is a suite of safety features integrated in
their vehicles that includes technologies such as intelligent adaptive cruise control that combines
features such as speed sign recognition, stop-and-go, and lane centering. This feature adjusts
the speed of the vehicle depending on the speed limit of the highway driven, it also manages to
adjust the speed of the car depending on the vehicles in front of the ego-vehicle. Therefore, in a
situation where the vehicle in front slows down or completely stops, the vehicle will reduce the
speed or completely stop thanks to this feature. Furthermore, it has the advantage of keeping the car centered between the lines at all times, making sure the vehicle is well located while driving; the second feature included in this safety suite is the evasive steering assist, where the vehicle detects an abrupt traffic interruption ahead where the driver is in danger of crashing.

Once this scenario is detected, the vehicle gives an additional steering support to the driver attempting to evade the collision (Ford Motor Company 2023a); the third safety technology applied is the rear camera, where the vehicle provides the driver a better field of view of what is behind the vehicle when backing up, helping the driver to be aware of the obstacles surrounding the vehicle and avoid probable collisions; active park assist 2.0 gives support to the driver for parallel parking assistance, where the vehicle detects the spot at which the driver wants to part and on its own start the parking motion. Once the location of the vehicle is successful, the driver just need to engage the break of the vehicle; auto high-beam headlamps is a technology that detects poor lightning conditions in the environment and then, automatically switch on the high-beam headlamps. Furthermore, this system also detects once a vehicle is approaching driving on the opposite direction and dims automatically avoiding blinding other drivers in the road (Ford Motor Company 2023b).
Another important automotive company such as Audi focus on safety features to prevent crashes for drivers. Some of the technologies integrated in their vehicles include pedestrian and stationary vehicle detection and preparation, based on a forward-facing camera and a radar sensor, this feature recognizes potential accidents and starts preventive measurements. Rear detection integrated to pretension the belts in the vehicle in case of an imminent crash is accomplished by the rear collision detection and preparation. Audi relies on rear radar sensors that sense a possible collision from the rear side of the vehicle, in case that a crash is likely to happen, the system adjusts the occupant seats into an optimal position to reduce injuries. Other technologies also integrated in Audi vehicles include active lane assist, adaptive cruise assist, adaptive cruise control with traffic jam assist, night vision assistant, among others. Audi does a great job applying sensor fusion in the vehicle to enhance their safety systems. For instance, the adaptive cruise control with traffic jam assist combines radar, ultrasonic sensors, and front camera in order to recognize vehicles in the environment. Ultrasonic sensors as known are very good in close range applications, and using them in combination with radar sensors ensures the well functioning of tracking vehicles in the road for this assistance system during a high traffic scenario.

Platooning systems have also been in development and enhanced in multiple works. Advantages of platooning systems include improvements in fuel consumption, safety, and traffic efficiency. The fuel consumption has been improved by driving at a constant speed in the highway, allowing the vehicles to have a more efficient fuel consumption in every trip performed. Furthermore, traffic efficiency is improved by creating a driving environment where the vehicles are following each other with a safe determined distance, resulting in a reduction of space between vehicles and creating more space for vehicles in the road to drive. Traffic will be more fluent due to the implementation of this technology on the road. Researchers in (Alzu’bi and Tasky 2020) have performed a LiDAR based simulation of platooning technology between two vehicles by using Gazebo simulator, ROS middleware, and Matlab. This algorithm was capable of tracking objects in the scene by using the LiDAR data and adjust the distance between the lead and follower vehicle.
CHAPTER 3
METHODOLOGY

This chapter will show in detail the process to build the electric smart navigation platform. This methodology was divided between mechanical development, implementation & wiring of sensors and power sources, sensors programming, ADAS implementation, data acquisition process and evaluation methods.

![Figure 3.1: Complete assembly of smart navigation platform developed by the author](image)

3.1 Mechanical development of electric smart navigation platform

The mechanical frame of this navigation platform started with a small frame from a Go-kart. Extension of this mechanical frame was required in order to have more space in front for legs
space, stepper motor for steering by wire system and sensors assembly. This frame extension process started with removing old components in the vehicle and then cutting pipes in front. Once the frame was divided in half, a 0.30 m long pipe was added to the frame and then plug welded into the frame by drilling holes on both ends of the pipe on the platform with an overlap of 50 mm on both side of the connections. This welding technique contributed on giving more support to the extension preventing bending and reducing stress in the middle of the extension and focusing it on both sides of the structure. Extra pipes were added to the body to enhance the structure of the vehicle. As a final step for the extension of the frame, the body was sanded and then painted previous to assembly with mechanical and electronic components. Figure 3.2 shows the extension process for the frame of the platform on its initial stages.

Figure 3.2: Frame base and extension, (a) shows the initial frame of the vehicle platform, (b) is the point at which the vehicle was extended with a 0.3 m rod, (c) is the final frame extension used as the platform body.
3.2 Implementation & wiring of the sensors and power sources

3.2.1 Sensor mounts for installation of ADAS sensors

Different sensor mounts were designed in SolidWorks and 3D-printed to be assembled in multiple locations in the vehicle. The sensor mounts required for the vehicle were mounts for the ultrasonic sensor HC-SR04, Velodyne VLP-16 LiDAR sensor, RPLiDAR sensors and microLiDAR time of flight sensor. Through these mounts, it was possible to implement the sensors in the desired locations for the integration of ADAS features in the platform.

The first sensor mount created was for the ultrasonic HC-SR04 sensor. This sensor was mounted in both the front and back profiles of the vehicle, two different sensor mounts were designed and assembled in the vehicle frame. Figure 3.3 shows the design for the mounts in the front part of the frame.

![Figure 3.3: Ultrasonic sensor mounts 3D-Model](image)

Five different ultrasonic sensors have been assembled in the front of the vehicle using the sensor mounts mentioned above, the direction of the sensors point into all directions of the structure, creating a good field of view for object recognition on the road.
A mount for the LiDAR VLP-16 sensor was designed in order to assemble it at the top of the vehicle platform. A bracket was added to the mount in order to attach it to the aluminum profile bar in the car. This sensor was installed in the top of the vehicle so that it was possible to scan the environment in a 360° field of view for mapping technologies.
Another mount platform integrated in the vehicle was installed in the front with an ultrasonic sensor, a microLiDAR sensor and a RPLiDAR sensor. This mount was designed so that all the sensors were installed from the same origin in order to take consistent distance data for sensor fusion purposes. Furthermore, the principal objective of having the three sensor mounted from the same origin, was to perform data acquisition of different targets in the environment and then use an LSTM machine learning algorithm to enhance the precision of the distance measurements in the environment.

![Figure 3.6: LSTM sensor mount. a) Ultrasonic sensor HC-SR04, b) RPLiDAR sensor, c) microLiDAR SF11/C](image)

The sensors and components used in the vehicle platform include, ultrasonic sensors HC-SR04, potentiometers, DC motors, DC motor controllers, stepper motor, stepper motor controller, rotary encoders, LEDs, arduino microcontrollers, and raspberry pi microcomputer. Arduino microcontrollers were in charged of controlling the actuators in the vehicle and power the systems of steering by wire, drive, reverse. Furthermore, the Arduinos installed allowed the integration of the ADAS features developed in the vehicle platform. Automatic emergency braking and parking assistance were written in C++ code to be run in the Arduino microcontroller. The raspberry Pi was in charge of live data visualization of the sensors instrumented in the vehicle to see the performance of the sensors in real time and data acquisition to analyze the performance of the ADAS features integrated in the navigation platform. Data acquired from this data acquisition system includes time, speed (rotary encoders), and distance (ultrasonic sensors). Figure 3.7 shows the
wiring diagram created for the vehicle platform including all the components mentioned before. This system is a complete overview for the combination between functionality, analysis and data acquisition of the vehicle created.

Figure 3.7: Wiring diagram vehicle platform
3.3 Integration and performance evaluation of ADAS features & IV technologies

Different ADAS were developed by using the different sensors chosen for this project.

3.3.1 Automatic Emergency Braking

Automatic emergency braking in the vehicle was developed based on ultrasonic sensors, five different HC-SR04 sensors were installed in the front bumper of the smart platform in order to detect any obstacles that would put in danger the safety of the driver and pedestrians. The algorithm developed for this safety feature divided the distance recorded in three different scenarios. The first scenario was defined as safe zone, where the ego-vehicle was traveling at a constant speed and does not perceive any obstacle in at least 2.5m of distance. While being in the safe zone, the vehicle lights up a green LED symbolizing that the road is free; the second scenario is the warning zone, where the vehicle detects an obstacle within 0.7m - 2.5m of distance with respect to the ego-vehicle. Then, the speed decreases to prepare for braking. Meanwhile, a blue LED lights up every time the ego-vehicle detects an obstacle within this range; the third scenario happens when the distance is less or equal to 0.7m. At this point, the ego-vehicle intends to stop and lights up a red LED warning the driver a possible collision is going to happen.

![Parking assistance sensors integrated in navigation platform](image)

*Figure 3.8: Parking assistance sensors integrated in navigation platform*

Data was acquired from a series of 10 runs to test the AEB feature performance inte-
grated in the smart navigation platform. The test consisted on moving an obstacle in front of the vehicle 2 meters away. The vehicle was raised in a static position with at a constant speed while the ADAS system was evaluated. Every 10 runs recorded for the test, a different speed was recorded to test the performance of the safety feature.

![Figure 3.9: Automatic emergency braking range of action](image)

Time, speed, and distance were acquired from the sensors to test the performance of the safety feature once it was activated with the change in distance recorded from the ultrasonic sensors. The algorithm for this feature was run through the Arduino reading distance from the environment, changing the speed of the wheels based on the distance recorded from the ultrasonic sensors, and warning the driver with LEDs in the dashboard that an obstacle was in a dangerous position in front. Meanwhile, the data acquisition code was run through the Raspberry Pi 4 that acquired data from the rotary encoders, in the DC motors, distance from the frontal ultrasonic sensors, and record time for every pulse of data recorded.
3.3.2 Parking Assistance

Parking assistance feature was developed by relying on ultrasonic sensors. The precise short-range of these sensors made them ideal for this application. In the dashboard, LEDs would turn on in different patterns depending on how close the vehicle was to the wall. Indicating the best position for parking with respect to the distance to a wall.

![Parking assistance sensors integrated in navigation platform](image)

**Figure 3.10:** Parking assistance sensors integrated in navigation platform

A similar methodology to the AEB was used to evaluate the performance of the parking assistance feature. A series of 10 different runs were taken into consideration to test the performance of the parking assistance. Distance over time as well as velocity were recorded to evaluate the performance of the safety feature over different instances. The distance configuration for the parking assistance is depicted in Figure 3.11 where the safe zone is for any wall which distance with respect of the ego-vehicle is greater than 2.00 m, in which case the vehicle is traveling at a speed of 2.80 m/s. The warning zone is for any wall that is found by the vehicle when the distance with respect to the ego vehicle is between 0.4 m and 2.00 m, the velocity in this zone is reduced to 1.4 m/s. Finally, the critical zone is below 0.4 m where the vehicle is at a complete stop (0 m/s). At this point, the car will be complete parked and a safe distance will be kept between the ego-vehicle and the wall.
The evaluation method for both AEB and PA was based on (Gross 2022), where the AEB system was evaluated by performing multiple runs of the vehicles at variable speeds. The efficiency of alerts provided based on distances recorded, velocity reduction, and possible impact were factors to be evaluated. Furthermore, a time to collision formula was used in order to calculate the time to collision of the ego-vehicle against the mobile targets directed to the vehicle for the tests. The time to collision formula (TTC) for this study was depicted as follows:

\[ TTCa = \frac{-v_r - \sqrt{v_r^2 - 2a_{TV}r}}{a_{TV}} \]  

(3.1)

Where \( v_r \) represents the ego vehicle speed, \( a_{TV} \) is the acceleration of the vehicle at this instant, and \( r \) is the distance between the ego-vehicle and the target in the environment. The
derivation of this formula comes from finding the roots at which the velocity is equal to 0 m/s. This way is possible to calculate how much time is left for the collision to happen between the ego-vehicle and the obstacle.

The acceleration of the vehicle was calculated by using the following formula:

$$a_{TV} = \frac{(v_f - v_o)}{\Delta t}$$  \hfill (3.2)

Where \(v_f\) is the final velocity of the ego-vehicle, \(v_o\) is the initial velocity of the ego vehicle and \(\Delta t\) is the difference in time between these two velocities. The derivation of this acceleration formula comes from the slope of the velocity vs time data, where is possible to find the acceleration from the slope of the velocity graph.

By using this equation it was possible to calculate and analyze the time to collision of the vehicle at the time velocity was reduced for braking before impacting the obstacle (AEB) or just reducing the velocity to park the vehicle (PA). This analysis was taken into consideration in order to evaluate the performance of these to ADAS.

**Figure 3.12:** LED warning system for AEB and PA. a) Green LEDs light up if distance to obstacles is safe and do not endanger the driver, and speed keeps constant. b) Blue LEDs light up if distance to obstacles is in the warning zone. c) Red LEDs light up if distance to obstacles endangers safety of driver and pedestrian. d) RGB LED that changes color with respect to the zone detected by the ultrasonic sensors
3.3.3 LSTM machine learning to enhance precision on ADAS features

The LSTM machine learning model developed in this work considers the recording of distance for five different ranges (1m - 5m). Three different sensors were used for this machine learning model including ultrasonic sensor HC-SR04, microLiDAR SF11-C and RPLiDAR. This machine learning model was created based on sensor fusion in order to enhance the precision of sensor distance calculation that is used in ADAS features such as AEB, PA, ACC, BSD, etc.

An obstacle in the environment was set in front of the vehicle and the field of view of these three different sensors in intervals of 1 meter for each run. Data was taken overnight from
the three sensors pointing to the mobile obstacle in order to record enough data to subsequently be fed into the machine learning training model.

A data acquisition algorithm was developed in Matlab to record the distance calculations of the sensors. A total of number of points recorded from the machine learning model was 50,000 for each run. Considering the number of times the DAQ code was run for the 5 distances analyzed, gives as a result a total number of points recorded of 250,000 data points. this number of points was divided in two sections so that there would be data for both training and testing procedures for the LSTM model developed.

A region of interest (ROI) algorithm was defined for the RPLiDAR sensor reducing the data needed to be fed into the machine learning algorithm, since the field of view of this sensor was 360°, it was reduced to 10° to just focus on the object placed in front of the ego-vehicle. In this algorithm, the ROI of interest was set as an input and the data-set was split in half for training/testing purposes. Furthermore, the data was stored into a table having the four inputs for the trainer algorithm.

![Figure 3.14: LSTM Machine Learning Model configuration](image)

The table that was set in the ROI algorithm was fed into the trainer algorithm. In this code, the number of input features was defined for the training of the model, being RPLiDAR, ultrasonic, microLiDAR, and time between ROI recordings the inputs for this machine learning model. The structure for this LSTM machine learning algorithm is depicted in Figure 3.14 where a sequence input layer starts the structure which represents the number of inputs. subsequently, three LSTM layers combined with dropout layers are set in the configuration, a fully connected layer is connected and finally, a regression layer is connected for distance classification.
3.3.4 Steering by wire

The rack and pinion mechanism was assembled with the stepper motor receiving signals from the steering wheel using a potentiometer to allow power steering. The assembly of this system includes belt, pulley, rack and pinion, stepper motor, rod connectors, spacers, and wheels. Two brackets were created for the assembly of the stepper motor in the frame of the electric car. Pieces of metal were grinded and welded together for the assembly. New connector rods were machined by using an aluminum rod. Normal and left-handed threads are required for assembly to rack and pinion mechanism.

The steering by wire system was developed in C++ and executed by using an Arduino Mega 2560 that, through serial communication, pulses were sent to the stepper driver where it had communication with the motor, mirroring the motion executed in the steering wheel of the vehicle. The mirror-like behavior between steering wheel and stepper motor was based on the use of a potentiometer. The algorithm used for this system mapped the rotation of the potentiometer with a total number of steps of 1024 with the number of steps to take in the stepper motor. This way, it was possible to synchronize the stepper motor with the steering wheel. The potentiometer was connected mechanically to the shaft of the steering wheel, allowing the driver to interact with
the potentiometer rotation and then, as a result, being able to steer the vehicle into the desired directions.

### 3.3.5 Drive by wire

The drive control of the vehicle was a connection between the DC motor controller and the Arduino reading an analog signal of a potentiometer in the console. The Arduino had a constant serial communication with the DC motor controller, allowing to send bits of information in a scale from 1 - 255 controlling the speed and direction of the vehicle through a interpolation with the number of steps that the potentiometer had with a total of 1024. 255 was max speed in the forward direction related to max position of the potentiometer 1024. 1 was the max speed in the reverse position related to the starting position of the potentiometer at 1. All this information was sent to the DC motor controller through the Tx pin in the arduino connected to the serial terminal in the DC motor controller.

### 3.3.6 Environmental Data Acquisition for mapping integration

A data acquisition algorithm was developed in MATLAB in order to acquire data from 4 different sensors simultaneously. The sensors chosen for this task were RPLiDAR, Velodyne VLP-16, ultrasonic sensors HC-SR04 and a GO-Pro 10 camera. The synchronization of such sensors at the time of recording data allowed to map the environment from different sensor perspectives and functioning principles. The algorithm was developed in MATLAB and started by detecting the sensors connected to the computer via serial (ultrasonic, RPLiDAR) and ethernet (VLP-16). The sample frequency used for the DAQ code was 10Hz. The time taken to record the data depended on the time taken to drive up to Rusell Union and come back resulting in X number of frames recorded for mapping and analysis.
Figure 3.16: Point cloud acquired

The structure of the data was organized as a structure in MATLAB. Where the point clouds recorded from the VLP-16 sensor in MATLAB were organized in frames including location (X,Y,Z coordinates), count (Number of points), and intensity values; RPLiDAR data included polar coordinates including angle and radius for both sensors; ultrasonic sensors included distance; time between each frame recorded was also included in this structure.

3.3.7 LiDAR testing

LiDAR testing was performed to estimate the quality of the point clouds based on the analysis for both the intensity values and the number of points recorded from the backscatter signal. The importance of the experiments presented in this section is to understand and quantify the limitations
of LiDAR sensors during real life driving conditions and how the environmental conditions can impact on the safety or autonomous technologies integrated in the ego-vehicle.

Figure 3.17: Environment point cloud acquired

A range with 9 different distance points in the X and Y axes were defined in a space outside of the laboratory (2, 2.5, 5, 5.5, 7, 7.5, 10, 15, and 20 meters). It was possible to take LiDAR measurements for different positions of the vehicle and its variation based on distance.
3.3.8 Impact of color in LiDAR data

Targets of different colors were set in the driving environment used for testing for both static and dynamic conditions. Data was recorded from the LiDAR sensor acquiring the objects set in the testing environment.
The intensity values were evaluated by using Z-score normalization. Since the range of values for intensity was considerably distant, the normalization process helped to minimize the range of results and then, properly compare the results between the colors acquired. The equation used for the Z-score normalization process was as follows:

\[ Z = \frac{(I - \mu)}{\sigma} \]  

Criteria for success for this experiment was to analyze different data sets of LiDAR data and recognize performance variability based on color of the targets, weather conditions, and dark/bright lightning scenarios. Performance based on color should reflect a high quality point cloud in terms of intensity for bright colors, whereas darker colors are expected to have a low value of intensity data due to its physical properties.

3.3.9 LiDAR mapping

![Figure 3.19: Region of interest point cloud](image)

The methodology performed to map the environment of the Georgia Southern campus was based on recording LiDAR data of the campus while driving the smart navigation platform. The
sensors used to record the environment data were the LiDAR VLP-16 sensor and a GP-Pro 10 camera. The test consisted on driving along Forest Dr from the Engineering Building up to Russell Union and come back. With the data recorded, a 3D map synchronized with the Go-Pro was created. The objective of this test was to recreate a portion of the Georgia Southern campus with LiDAR data for LiDAR mapping and environment visualization.

Recording the data from the targets, pre-processing and processing methods were necessary. From the pre-processing phase, a region of interest (ROI) was defined in order to extract the points from the targets and then analyze them. The region of interest for the targets would result as follows:
4.1 ADAS features performance evaluation

Velocity from the rotary encoders, as well as the distance of the ultrasonic sensors was recorded in order to test the performance of the automatic emergency breaking and parking assistance features. Graphs below show the data recorded for velocity vs time, distance vs time, and velocity and distance vs time. Ten different test were executed to record the data from these sensors and then evaluate the performance of these safety features under different test configurations as explained in the methodology chapter.

4.1.1 Automatic Emergency Braking evaluation

![Speed vs Time Graph for AEB](image)

**Figure 4.1:** Velocity vs time Automatic Emergency Braking

For the automatic emergency braking test, the distances set for safety response of the vehicle were set as safe distance < 2.5 m with a constant velocity of 2.87 m/s. warning distance < 0.7
m and > 2.5 m for the first test configuration, and < 0.2 m and > 2.5 m for the second test configuration with a constant velocity of 1.42 m/s. Finally, the critical distance was define as > 0.7 m for the first test configuration and > 0.2 m for the second configuration with the vehicle coming to a stop at 0 m/s.

Figure 4.1 shows the change in velocity of the vehicle recorded from the rotary encoder with respect of time. This variation of velocity changed based on the distance recorded from the ultrasonic sensors to the mobile target that was moved towards the ego-vehicle in a stationary position. This test simulated the scenario of the ego-vehicle driving at a constant velocity, and then unexpectedly an obstacle showed up in the way of the ego-vehicle or a sudden brake from the vehicle in front was to take place. As a result, the vehicle took notice of the dangerous reduction of safe distance with respect to the vehicle in front and proceeded to take corrective actions preventing an accident or fatality for the driver, passengers and pedestrians.

![Distance vs Time Graph for AEB](image)

**Figure 4.2:** Distance vs time Automatic Emergency Braking

Change in distance was recorded from the ultrasonic sensors instrumented in the vehicle
platform. A variation in distance as shown in Figure 4.2 happened with respect of time and changed with the motion of the mobile target heading towards the ego-vehicle. The graph shows the reduction of distance with respect of time during the testing of the automatic emergency braking safety feature during the ten runs performed.

![Distance and Speed vs Time AEB](image)

**Figure 4.3:** Distance and velocity vs time Automatic Emergency Braking

Figure 4.3 shows the synchronized reaction of reduction of distance recorded and velocity with respect of time for the automatic emergency braking safety feature. It is shown how the velocity is reduced successively from 2.8 m/s to 1.4 m/s and then to 0 m/s. Furthermore, the distance was reduced from 2.5 m, to 0.7 m. The warning system based on the LEDs reacted based on the distance recorded from the ultrasonic sensors as well. Working as a warning system to display the distance warnings to the driver.

Table 4.1 shows the performance observations for the different runs performed on the AEB ADAS test. For every run of the test, the performance of the safety system was successful. An alert and velocity reduction took place every time the distance was in between a warning zone for the driver. Furthermore, the velocity reduction and alert was displayed at the correct distance range reaction.
Table 4.1: AEB Performance observations for electric platform

<table>
<thead>
<tr>
<th>Test</th>
<th>Provide an Alert</th>
<th>Speed reduced in correct distance range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 1</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 3</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 4</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 5</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 6</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 7</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 8</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Run 10</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

As shown in table 4.2 it was possible to calculate the time to collision (TTC) for the AEB safety system, this table depicts the constant performance for the AEB feature in terms of TTC with an average TTC of 0.879s and a standard deviation of 0.277. These results demonstrate the performance of a robust safety system integrated in this vehicle platform. The percentage difference average in terms of proper distance calculation for speed reduction and LED alert was 8.38%, giving a good response on the system based on the distance calculated from the ultrasonic sensors.

Table 4.2: Time to collision calculations for automatic emergency braking performance analysis

<table>
<thead>
<tr>
<th>Test</th>
<th>Braking Distance (m)</th>
<th>$V_0$ (m/s)</th>
<th>$V_f$ (m/s)</th>
<th>$T_o$ (s)</th>
<th>$T_f$ (s)</th>
<th>$\Delta T$ (s)</th>
<th>GT Braking Distance (m)</th>
<th>Braking acceleration (m/s²)</th>
<th>Braking TTC (s)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 1</td>
<td>0.613</td>
<td>1.512</td>
<td>0.000</td>
<td>6.380</td>
<td>6.633</td>
<td>0.252</td>
<td>0.700</td>
<td>-5.991</td>
<td>0.771</td>
<td>12.422</td>
</tr>
<tr>
<td>Run 2</td>
<td>0.664</td>
<td>1.517</td>
<td>0.000</td>
<td>6.387</td>
<td>6.890</td>
<td>0.503</td>
<td>0.700</td>
<td>-3.017</td>
<td>1.335</td>
<td>5.179</td>
</tr>
<tr>
<td>Run 3</td>
<td>0.621</td>
<td>1.515</td>
<td>0.000</td>
<td>5.125</td>
<td>5.628</td>
<td>0.503</td>
<td>0.700</td>
<td>-3.009</td>
<td>1.320</td>
<td>11.336</td>
</tr>
<tr>
<td>Run 4</td>
<td>0.613</td>
<td>0.379</td>
<td>0.000</td>
<td>6.380</td>
<td>6.633</td>
<td>0.252</td>
<td>0.700</td>
<td>-1.503</td>
<td>1.190</td>
<td>12.422</td>
</tr>
<tr>
<td>Run 5</td>
<td>0.690</td>
<td>1.128</td>
<td>0.000</td>
<td>5.884</td>
<td>6.137</td>
<td>0.252</td>
<td>0.700</td>
<td>-4.469</td>
<td>0.863</td>
<td>1.400</td>
</tr>
<tr>
<td>Run 6</td>
<td>0.176</td>
<td>0.379</td>
<td>0.000</td>
<td>4.678</td>
<td>4.929</td>
<td>0.251</td>
<td>0.200</td>
<td>-1.510</td>
<td>0.795</td>
<td>12.000</td>
</tr>
<tr>
<td>Run 7</td>
<td>0.196</td>
<td>1.139</td>
<td>0.000</td>
<td>5.705</td>
<td>5.956</td>
<td>0.251</td>
<td>0.200</td>
<td>-4.538</td>
<td>0.638</td>
<td>2.000</td>
</tr>
<tr>
<td>Run 8</td>
<td>0.173</td>
<td>1.511</td>
<td>0.000</td>
<td>4.174</td>
<td>4.426</td>
<td>0.252</td>
<td>0.200</td>
<td>-5.996</td>
<td>0.600</td>
<td>13.500</td>
</tr>
<tr>
<td>Run 9</td>
<td>0.210</td>
<td>1.133</td>
<td>0.000</td>
<td>4.173</td>
<td>4.425</td>
<td>0.252</td>
<td>0.200</td>
<td>-4.496</td>
<td>0.648</td>
<td>5.000</td>
</tr>
<tr>
<td>Run 10</td>
<td>0.183</td>
<td>1.138</td>
<td>0.000</td>
<td>6.459</td>
<td>6.710</td>
<td>0.251</td>
<td>0.200</td>
<td>-4.534</td>
<td>0.630</td>
<td>8.500</td>
</tr>
</tbody>
</table>

Average: -3.906, Standard Deviation: 1.526, 0.879, 8.376
4.1.2 Parking Assistance evaluation

Very similarly to AEB, parking assistance testing consisted in acquiring data from rotary encoders and ultrasonic sensors analyzing the performance of the safety features depending on the distance recorded from the ego-vehicle to the target in the environment. Ten runs were performed with different distance range configurations testing the well-functioning of the safety feature.

Figure 4.4: Time to collision comparison AEB testing runs

Figure 4.5: velocity vs time Parking Assistance
As shown in figure 4.4 velocity decreased with respect of time. This velocity reduction was related to the distance recorded from the ultrasonic sensors at the back of the vehicle. Velocity had two reductions during the testing, the first one was reduced from 2.8 m/s to 1.4 m/s. Subsequently, the velocity dropped from 1.4 m/s to a total stop of the vehicle at 0 m/s.

Figure 4.6 shows the variation of distance with respect of time which was acquired by the use of the ultrasonic sensors installed in the back bumper of the smart navigation platform. With the mobile target approaching to the back side of the vehicle, the distance recorded was reduced with respect of time.

![Distance vs Time Graph for PA](image)

**Figure 4.6:** Distance vs time Parking Assistance

Velocity and speed were acquired for the 10 runs of the PA testing. As shown, the reduction of velocity and distance were directly proportional to each other. Velocity was reduced with the reduction of distance acquired from the ultrasonic sensors. Once the distance recorded from the ultrasonic sensors to the target was in between 0.4 m - 1.50 m for the first configuration and 0.1 m - 2.00 m for the second configuration, the velocity was also reduced to 1.4 m/s from a starting constant velocity of 2.8 m/s. Similarly, once the distance recorded from the ultrasonic sensors went below 0.4 m and 0.1 m, the velocity dropped down to 0 m/s for the vehicle platform.
Table 4.3 shows the performance of the parking assistance and how both the warning system indicator (LEDs color based on distance) and the reduction of velocity happened once the distance from the ego-vehicle to the target was within the range of action. A proper warning activation as well as reduction of velocity happened in all the tests run for the PA testing.
Table 4.4: Time to collision calculations for Parking Assistance performance testing

<table>
<thead>
<tr>
<th>Test</th>
<th>Braking Distance (m)</th>
<th>$V_r$ (m/s)</th>
<th>$V_i$ (m/s)</th>
<th>$T_o$ (s)</th>
<th>$T_f$ (s)</th>
<th>$\Delta T$ (s)</th>
<th>GT Braking Distance (m)</th>
<th>Braking acceleration ($m/s^2$)</th>
<th>Braking TTC (s)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run 1</td>
<td>0.386</td>
<td>1.129</td>
<td>0.000</td>
<td>3.872</td>
<td>4.124</td>
<td>0.252</td>
<td>0.400</td>
<td>-4.479</td>
<td>0.737</td>
<td>3.533</td>
</tr>
<tr>
<td>Run 2</td>
<td>0.361</td>
<td>1.131</td>
<td>0.000</td>
<td>4.385</td>
<td>4.636</td>
<td>0.252</td>
<td>0.400</td>
<td>-4.495</td>
<td>0.725</td>
<td>9.856</td>
</tr>
<tr>
<td>Run 3</td>
<td>0.389</td>
<td>1.131</td>
<td>0.000</td>
<td>4.382</td>
<td>4.634</td>
<td>0.252</td>
<td>0.400</td>
<td>-4.483</td>
<td>0.739</td>
<td>2.685</td>
</tr>
<tr>
<td>Run 4</td>
<td>0.351</td>
<td>0.754</td>
<td>0.000</td>
<td>4.644</td>
<td>5.146</td>
<td>0.502</td>
<td>0.400</td>
<td>-1.502</td>
<td>1.350</td>
<td>12.309</td>
</tr>
<tr>
<td>Run 5</td>
<td>0.361</td>
<td>1.131</td>
<td>0.000</td>
<td>4.385</td>
<td>4.636</td>
<td>0.252</td>
<td>0.400</td>
<td>-4.495</td>
<td>0.725</td>
<td>9.856</td>
</tr>
<tr>
<td>Run 6</td>
<td>0.106</td>
<td>1.858</td>
<td>0.000</td>
<td>0.768</td>
<td>3.066</td>
<td>2.298</td>
<td>0.100</td>
<td>-0.809</td>
<td>4.652</td>
<td>6.000</td>
</tr>
<tr>
<td>Run 7</td>
<td>0.121</td>
<td>1.872</td>
<td>0.000</td>
<td>1.279</td>
<td>3.115</td>
<td>1.836</td>
<td>0.100</td>
<td>-1.020</td>
<td>3.736</td>
<td>21.000</td>
</tr>
<tr>
<td>Run 8</td>
<td>0.113</td>
<td>2.246</td>
<td>0.000</td>
<td>1.280</td>
<td>3.060</td>
<td>1.780</td>
<td>0.100</td>
<td>-1.262</td>
<td>3.610</td>
<td>13.000</td>
</tr>
<tr>
<td>Run 9</td>
<td>0.113</td>
<td>1.122</td>
<td>0.000</td>
<td>2.259</td>
<td>3.321</td>
<td>1.062</td>
<td>0.100</td>
<td>-1.056</td>
<td>2.220</td>
<td>13.000</td>
</tr>
<tr>
<td>Run 10</td>
<td>0.106</td>
<td>1.501</td>
<td>0.000</td>
<td>2.555</td>
<td>3.314</td>
<td>0.759</td>
<td>0.100</td>
<td>-1.978</td>
<td>1.586</td>
<td>6.000</td>
</tr>
</tbody>
</table>

Table 4.4: Time to collision calculations for Parking Assistance performance testing

Time to collision calculation for PA was performed and the results are depicted in table 4.8. As shown, the average time to collision calculated for the 10 runs performed was 0.472s with a standard deviation of 2.161. Furthermore, the percentage error average calculated comparing the ground truth braking distance with the braking distance recorded was 9.724%. The results shown in the table are proof of a parking system with an accurate distance precision and consistent performance.

Figure 4.8: Time to collision comparison PA testing runs
4.1.3 LSTM Machine Learning Distance Prediction

Figure 4.9 shows the training results for 1 meter distance with respect to the ego vehicle. Half of the data in the dataset for the 1 meter distance was fed into the trainer in order to get this result. The top side of the graph depicts the root mean square error (RMSE) as a way to measure the error of the training model for the LSTM model. Loss represents the quality of the machine learning training, once it dropped to zero it meant that the prediction based on the data was perfect. Every training for every distance had an output similar to the one shown in figure 4.9, with the expected training output, the rest of the results were analyzed as well.

![Figure 4.9: Training results for LSTM network](image)

As shown in figure 4.10, 2D LiDAR, ultrasonic and microLiDAR data were represented in the graphs as data acquired from the sensors for the LSTM training. The LSTM sensor fusion graph represents the fusion between the signal of the three sensors just mentioned. Giving a successful prediction of the data fed in the tester algorithm the graph shows the prediction of the model by feeding data corresponding to measuring distance from the environment to objects at \{1, 2, 3, 4, and 5\} meters.
Analyzing the LSTM performance for the 2 meters distance as shown in figure 4.12, it is demonstrated a good performance of the prediction model with respect to the ground truth for this distance and the average sensor measurement calculated. All distances demonstrated a good prediction performance for every distance performed. Figure 4.12 shows the average raw measured distance with a value of 2.127 meters (6.35% difference to GT). Whereas, the average LSTM distance value was 1.916 meters (4.2% difference to GT). Both of these results demonstrating a greater precision based on prediction compared to average of raw data.
Compiled results of all LSTM training models are shown in figure 4.11. It is depicted a constant efficient prediction for the distances trained where performance was variable depending on distances, where all of the predictions performed with the LSTM network were precise with respect to the ground truth distance reference. Furthermore, comparison between percentage errors and standard deviations between raw data and LSTM data was accomplished. It was demonstrated based on this comparison that the LSTM algorithm presented a more stable percentage error for each ground truth distance with respect to the raw data percentage error that decreased with a distance increment. Showing that the LSTM performance has a better distance calculation efficiency compared to raw sensor data. Furthermore, standard deviation also had a more stable behavior in term of the LSTM model compared to the raw data of the sensor, giving a more consistent output of distance calculation with respect to the target in the environment. This results are shown in table 4.5.
Figure 4.12: LSTM sensor fusion evaluation for 2 meters

Table 4.5: Percentage error comparison

<table>
<thead>
<tr>
<th>Distance (m)</th>
<th>Percentage Error Raw (%)</th>
<th>Percentage Error LSTM (%)</th>
<th>Standard Deviation Raw</th>
<th>Standard Deviation LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.310</td>
<td>4.190</td>
<td>0.080</td>
<td>0.030</td>
</tr>
<tr>
<td>2</td>
<td>6.360</td>
<td>4.210</td>
<td>0.090</td>
<td>0.060</td>
</tr>
<tr>
<td>3</td>
<td>2.150</td>
<td>5.740</td>
<td>0.046</td>
<td>0.122</td>
</tr>
<tr>
<td>4</td>
<td>1.680</td>
<td>3.890</td>
<td>0.048</td>
<td>0.101</td>
</tr>
<tr>
<td>5</td>
<td>0.020</td>
<td>1.430</td>
<td>0.001</td>
<td>0.050</td>
</tr>
<tr>
<td>Average</td>
<td>4.304</td>
<td>3.892</td>
<td>0.053</td>
<td>0.073</td>
</tr>
</tbody>
</table>
As shown in figure 4.13, the percentage error for the LSTM performance had an average value of 3.89%, compared to 4.30% from raw data of the sensors demonstrating a more accurate distance measurement by using machine learning prediction. Furthermore, the standard deviation from the LSTM machine learning model had a final average value of 0.073 for all distances compared to 0.053.

4.2 Environmental impact on LiDAR performance

Impact on LiDAR performance for mapping the environment under variable environmental conditions was described in the previous chapter. LiDAR data was acquired using the intelligent electric vehicle platform, mapping the environment by the use of the VLP-16 sensor and a Go-Pro camera was performed. Furthermore, analysis of LiDAR data based on the color of the vehicle was performed by analyzing the intensity values of the data recorded.
Figure 4.14: Visual representation of point clouds extracted from vehicles recorded. a) BMW Series 3, b) Ford Escape, c) Honda Civic, d) Chevrolet Malibu, e) Ford Taurus
4.2.1 Color target impact in LiDAR data

Based on the fundamental principle of light and its relation between infrared light and the electromagnetic spectrum, it was expected to acquire high quality point clouds on light colors in comparison to darker colors from the targets analyzed. Having vehicles with colors such as white, grey, dark blue and light blue, it was expected to have a higher value of intensity from the white vehicle with respect to the dark blue vehicle which could have absorbed more LiDAR energy. Furthermore, it was shown that the number of points was related to the size of the vehicle and it was also affected by the distance at which the data was analyzed.

Figure 4.14 shows a point cloud visual representation of the vehicles at the point (1, 3) that was decided to study due to the good visual perspective at which the vehicles were located. It is possible to see the different perception performance of the LiDAR sensor for each of the vehicles analyzed in this work. Difference of color and size of each vehicle had an impact in the performance of the point cloud recorded. Table 4.6 shows the performance for the LiDAR test performed and the values for both LiDAR and number of points recorded for each of the vehicles studied.

Table 4.6: Results LiDAR analysis

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Color</th>
<th>Intensity</th>
<th>Percentage Reduction Intensity (%)</th>
<th>Number of Points</th>
<th>Percentage Reduction Number of Points (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW Series 3</td>
<td>Light Blue</td>
<td>5.211</td>
<td>61.085</td>
<td>881</td>
<td>20.773</td>
</tr>
<tr>
<td>Honda Civic</td>
<td>Dark Blue</td>
<td>2.433</td>
<td>81.834</td>
<td>319</td>
<td>71.313</td>
</tr>
<tr>
<td>Ford Escape</td>
<td>White</td>
<td>8.263</td>
<td>38.291</td>
<td>1112</td>
<td>0.000</td>
</tr>
<tr>
<td>Chevrolet Malibu</td>
<td>Grey</td>
<td>13.391</td>
<td>0.000</td>
<td>908</td>
<td>18.345</td>
</tr>
<tr>
<td>Ford Taurus</td>
<td>Silver</td>
<td>2.103</td>
<td>84.298</td>
<td>458</td>
<td>58.813</td>
</tr>
</tbody>
</table>
Number of points analysis was performed as shown in figure 4.15, demonstrating that the vehicle with the greatest number of points recorded from the experiment was the Ford Scape with a total number of points of 1112 points, followed by the Chevrolet Malibu with a number of points reduction of (18.35%), the third place was achieved by the BMW series 3 with a reduction with respect to the first place of (20.77%). Last two places where for Ford Taurus and Honda Civic with reductions of (58.81%) and (71.31%) respectively.

The number of points analysis had a variation related to color and size of the vehicles studied. As expected, the Ford Scape being the biggest car analyzed and having a white color, had the best performance of number of points recorded compared to rest of the vehicles. The Honda Civic having a dark blue color, absorbed most of the infrared light sent from the LiDAR sensor and the number of points acquired were lower compared to the rest of the vehicles that had lighter colors and therefore, a higher number of points acquired.
Intensity values comparison was done between the vehicles studied as shown in figure 4.16. The vehicle with the best quality of intensity out of the 5 samples was the Chevrolet Malibu with a total maximum intensity value of 13.39, followed by the Ford Escape (38.29%), BMW Series 3 (61.08%), Honda Civic (81.83%) and Ford Taurus (84.30%).

The intensity values recorded from each vehicle varied significantly with respect to each other. The reason behind this behavior is the variation in color of the targets. As shown in the bar graphs, the best intensity value was achieved by the Chevy Malibu that had a very light grey color. Due to the color of this vehicle, the intensity value recorded was high compared to the rest of the vehicles. Light colors such as light grey, white, light blue, etc., tend to reflect light in a better way than darker colors. Darker colors such as black have lack of color in their surfaces, for this reason, any source of light directed to this colors tend to be absorbed rather than being reflected back to
A scanning of the Georgia Southern campus environment was performed with the VLP-16 LiDAR sensor. 436 frames of point clouds (LiDAR data) were acquired with the data acquisition code developed. As a result, a scan of the environment while driving the intelligent electric vehicle platform was created, being capable to record in live time the environment by using the VLP-16 LiDAR sensor. With this, it was possible to create a live data visualization of the LiDAR data acquired by scanning a test drive of the vehicle giving a 360° field of view to the driver of the environment.
CHAPTER 5
CONCLUSIONS

• **Vehicle development:** The development of a scale electric smart vehicle platform with integration of ADAS safety technologies such as parking assistance, automatic emergency braking, environmental mapping, and intelligent vehicle technologies based on sensor fusion was presented in this work by the use of multiple exteroceptive sensors and industry methodology evaluation. All of this, with the aim of reducing fatalities and collision in real life driving scenarios due to the alarming rates of automotive accidents worldwide, while understanding the environmental impact factors that have an effect on sensor performance.

• **ADAS performance evaluation:** It was possible to evaluate through industry accepted methodologies the performance of automatic emergency braking and parking assistance safety features by analyzing the response of the vehicle under risky scenarios where the distance was reduced suddenly in the environment. This by giving a warning to the driver once the distance was unsafe for the driver in the safety zones defined. Furthermore, time to collision data was calculated demonstrating a robust and reliable safety system where the vehicle reduced its velocity and finally stopped depending on the distance configured for both AEB and PA in a efficient performance.

• **LSTM Distance calculation performance enhancement:** It was determined the performance of prediction of distance calculation based on a LSTM machine learning model where the percentage error of distance prediction had an average of 3.89%, while the percentage error for raw data coming from the sensors was 4.30%, demonstrating an improvement in distance prediction by the use of fuse data by using ultrasonic sensors, microLiDAR sensor and 2D LiDAR sensor. This distance prediction algorithm helped to enhance the precision of distance of obstacles around the ego-vehicle allowing a safer response for implementation in ADAS features compared on just relying in one distance sensor.
• **LiDAR performance analysis and mapping technology** LiDAR sensors quality of data depending on the color of the targets scanned in the environment was studied in this work by classifying the intensity data and number of points recorded for the vehicles scanned. It was found that bright colors have result in a better performance for LiDAR data in terms of intensity, darker colors show a reduction in intensity data since they tend to absorb the infrared light emitted from the LiDAR sensor. Since white colors are a combination of all colors, they tend to reflect infrared energy better in comparison to darker colors that absorb the infrared energy of the LiDAR sensors. For this reason, once LiDAR sensors scan dark objects, the point clouds have less number of points, the intensity values are lowers and the shape represented in the point cloud has a lower resolution compared to bright colors.

Intensity value for the Chevrolet Malibu with a light grey color was 13.39, compared to a dark blue Honda Civic with a dark blue with a value of 2.43. Resulting in a percentage reduction for intensity values of 81.83%. Furthermore, the maximum number of points was achieved by the white Ford Scape with a total number of points of 1112, compared with the dark blue Honda Civic with 319. Resulting in a percentage reduction of 71.31%. These results show the difference in performance in LiDAR depending on the color and size of targets in the environment. LiDAR environment perception will always be better when the targets are bright color.

5.1 **Validation of Proposed Hypothesis**

Taking into consideration the results from the development of the smart vehicle platform and the analysis evaluation for both ADAS features and sensor performance under different environmental scenarios, it was possible to state that the hypothesis of this work has been accomplished. Accuracy and precision of the features integrated in the vehicle platform have been evaluated under the unexpected conditions established in this work, demonstrating a robust and reliable performance in the ADAS features discussed in this work, precise and accurate performance on dis-
distance prediction based on a LSTM machine learning model, scale performance in LiDAR mapping depending on the color of the targets in the environment, and LiDAR mapping of the Georgia Southern campus for live data visualization and environmental mapping purposes.

- AEB average braking TTC was 0.711s, the percentage difference of the system for the braking distance configuration was 10.065%, giving a efficiency performance of 89.34%. On the other hand, the average TTC calculated for the parking assistance feature, resulted in a 2.421s and the percentage difference for the braking configuration was 13.038%. Furthermore, the efficiency performance of the PA safety feature was 86.96%.

- Introduction of the machine learning LSTM model enhanced the distance recording in the environment by the use of the three exterocpetive sensors integrated in the vehicle platform. Demonstrating a better performance compared to average raw data of the sensors by a margin of 0.412%.

- Based on the results on the LiDAR performance for intensity values and number of points, a relation was determined between those factors and the depicted LiDAR equation in chapter 2. The performance component is related to the amount of backscattered signal traveling back to the sensor and its impact on the intensity values of the targets. The brighter the color, the more intense the backscattered signal \( P_o \) is going to be. Furthermore, the geometric component relates to the distance of the targets with respect to the sensor resulting in less number of points with distance increase \( R \).

5.2 Future work

The development of this vehicle platform was performed with the aim of bringing a contribution to safety, design and development of smart vehicles in the automotive industry. This first vehicle developed in the Georgia Southern electric/hybrid intelligent vehicles laboratory opens the
opportunity for the development of a smart platform fleet of vehicles that will allow the development and testing of further intelligent and autonomous technologies research.

Sensor fusion algorithms for enhancement of distance prediction for ADAS features can be introduced by using the LSTM machine learning model developed in this work. AEB, PA, ACC, safety features can be benefited by the introduction of this distance prediction technology in further work. Environment recognition can be developed with the integration of different sensors such as GPS, microLiDAR, thermal cameras, rotary encoders, IMU units, etc.

Communication technologies such as V2X and V2V through socket communication can be introduced with the development of a second smart vehicle platform allowing vehicles to communicate between each other and share valuable environmental information between them. Platooning can also be introduced by the use of these vehicles.

For an easier method to access and enable ADAS features in the vehicle, a steering wheel with different buttons could be designed and built in order to give the driver access to all different technologies embedded in the vehicle.
REFERENCES


Driving Tests Resources. 2014. What is Vehicle Platooning?


Appendix A

SENSORS SPECIFICATIONS AND MECHANICAL DRAWINGS

A.1 Ultrasonic sensor HC-SR04

A.1.1 Wire connected direct as following

- 5V supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

A.1.2 Electric parameter

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Voltage DC</td>
<td>5V</td>
</tr>
<tr>
<td>working Current DC</td>
<td>15mA</td>
</tr>
<tr>
<td>Working Frequency</td>
<td>40Hz</td>
</tr>
<tr>
<td>Max Range</td>
<td>4m</td>
</tr>
<tr>
<td>Min Range</td>
<td>0.02m</td>
</tr>
<tr>
<td>Measuring Angle</td>
<td>15°</td>
</tr>
<tr>
<td>Trigger Input Signal</td>
<td>10μS TTL pulse</td>
</tr>
<tr>
<td>Echo Output Signal</td>
<td>Input TTL lever signal and the range in proportion</td>
</tr>
<tr>
<td>Dimension</td>
<td>40x20x15mm</td>
</tr>
</tbody>
</table>

A.1.3 Ultrasonic sensor HC-SR04 mechanical drawing
UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN MILLIMETERS

SURFACE FINISH: DEBURR AND BREAK SHARP EDGES

NAME SIGNATURE DATE

MATERIAL: HCSR04

DO NOT SCALE DRAWING REVISION

TITLE:

DWG NO. A3

SCALE: 2:1 SHEET 1 OF 1

WEIGHT:
A.2  RPLiDAR A3M1-R3

A.2.1 Specifications

Table A.2: Specifications ultrasonic sensor HC-SR04

<table>
<thead>
<tr>
<th>Item</th>
<th>Enhanced Mode</th>
<th>Outdoor Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Scenarios</td>
<td>Extreme performance ideal for indoor environments with maximum ranging distance and sampled frequency</td>
<td>Extreme reliability ideal for both outdoor and indoor environments with reliable resistance to daylight</td>
</tr>
<tr>
<td>Distance Range</td>
<td>White object: 25 meters</td>
<td>White object: 20 meters</td>
</tr>
<tr>
<td></td>
<td>Dark object: 10 meters</td>
<td>Dark object: TBD</td>
</tr>
<tr>
<td>Minimum Operating Ranging</td>
<td>0.2m</td>
<td>0.2m</td>
</tr>
<tr>
<td>Sample Rate</td>
<td>16000 times per second</td>
<td>10000 times per second</td>
</tr>
<tr>
<td>Scan Rate</td>
<td>Typically value: 15Hz (adjustable between 10Hz - 20Hz)</td>
<td></td>
</tr>
<tr>
<td>Angular Resolution</td>
<td>0.225°</td>
<td>0.225°</td>
</tr>
<tr>
<td>Communication Interface</td>
<td>TTL UART</td>
<td></td>
</tr>
<tr>
<td>Communication Speed</td>
<td>256000 bps</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Support former SDK protocols</td>
<td></td>
</tr>
<tr>
<td>System Voltage</td>
<td>5V</td>
<td></td>
</tr>
<tr>
<td>System Current</td>
<td>450mA-600mA</td>
<td></td>
</tr>
<tr>
<td>Power Consumption</td>
<td>2.25W-3W</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>UART Serial (3.3 voltage level)</td>
<td></td>
</tr>
<tr>
<td>Temperature Range</td>
<td>0°C-40°C</td>
<td></td>
</tr>
<tr>
<td>Angular Range</td>
<td>360°</td>
<td></td>
</tr>
<tr>
<td>Range Resolution</td>
<td>≤1% of the range (≤12m)</td>
<td>≤2% of the range (12m - 25m)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>1% of the range (≤3m)</td>
<td>2% of the range (3-5m)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.5% of the range (5-25m)</td>
</tr>
</tbody>
</table>

A.2.2 RPLiDAR A3M1-R3 mechanical drawing

Mechanical drawing has been created by SLAMTEC
Technical Specification:

1. The valid working area of the laser is 3mm above and below the horizontal line based on the laser transmitter center and the lens center, 6mm in total.

2. The RPLIDAR A2 uses the XH2.54-5P plug as the external plug. With the touch points side upwards, the block terminals of the plug from left to right are: VCC (red), TX (yellow), RX (green), GND (black), PWM (blue). 3. The RPLIDAR A2 is fixed by four M3x(4+L) mechanical screws, and L is the third party mounting plate thickness.
A.3 MicroLiDAR SF11-C

Table A.3: Specifications microLiDAR SF11-C

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0.2 - 100 m (natural targets), 2-40 m (moving water)</td>
</tr>
<tr>
<td>Resolution</td>
<td>1cm</td>
</tr>
<tr>
<td>Update rate</td>
<td>20 readings per second</td>
</tr>
<tr>
<td>Accuracy</td>
<td>±0.1m (70% reflective target @ 20°C)</td>
</tr>
<tr>
<td>Power supply voltage</td>
<td>5.0V ± 0.5V DC</td>
</tr>
<tr>
<td>Power supply current</td>
<td>200mA (maximum)</td>
</tr>
<tr>
<td>Outputs &amp; interfaces</td>
<td>Serial, I2C (up to 400kHz) &amp; analog with maximum latency of 65 ms</td>
</tr>
<tr>
<td>Dimensions</td>
<td>30x56.5x50 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>35 grams</td>
</tr>
<tr>
<td>Connections</td>
<td>plug/socket, micro USB</td>
</tr>
<tr>
<td>Laser power</td>
<td>20W (peak), &lt;15mW (average), Class 1M</td>
</tr>
<tr>
<td>Optical aperture</td>
<td>51mm</td>
</tr>
<tr>
<td>Beam divergence</td>
<td>0.2°</td>
</tr>
<tr>
<td>Operating Temp.</td>
<td>0 - 40°C</td>
</tr>
<tr>
<td>Approvals</td>
<td>FDA: 1410968-002 (2020/09)</td>
</tr>
</tbody>
</table>

A.3.1 Specifications

A.3.2 Mechanical drawing SF11-C

Figure A.1: Dimension drawings of the SF11

Units in millimeter (inch)
### A.4 Velodyne VLP-16 LiDAR Sensor

#### A.4.1 Sensor Specifications

<table>
<thead>
<tr>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time of flight distance measurement with calibrated reflectivities</strong></td>
</tr>
<tr>
<td>16 channels</td>
</tr>
<tr>
<td>Measurement range up to 100m</td>
</tr>
<tr>
<td>Accuracy: +/- 3cm (typical)</td>
</tr>
<tr>
<td>Dual returns</td>
</tr>
<tr>
<td>Field of view (vertical): 30° (-15° to 15°)</td>
</tr>
<tr>
<td>Angular resolution (vertical): 2°</td>
</tr>
<tr>
<td>Field of view (horizontal/azimuth): 360°</td>
</tr>
<tr>
<td>Angular resolution (horizontal/azimuth): 0.1° - 0.4°</td>
</tr>
<tr>
<td>Rotation rate: 5 - 20 Hz</td>
</tr>
<tr>
<td><strong>Integrated web server for easy monitoring and configuration</strong></td>
</tr>
</tbody>
</table>

#### Laser

- Class 1 - eye safety
- 905 nm wavelength
- Power consumption: 8W (Typical)
- Operating voltage: 9 - 32VDC (with interface box and regulated power supply)
- Weight: 830 grams
- Dimensions: 103mm diameter x 72mm height
- Shock: 500 m/sec² amplitude, 11 msec duration
- Vibration: 5 Hz to 2000 Hz, 3G rms
- Environmental Protection: IP67
- Operating temperature: -10°C to +60°C
- Storage temperature: -40°C to +105°C

#### Mechanical/Electrical/Operational

- Up to 0.3 million points/second
- 100 Mbps Ethernet connection
- UDP packets containing: Distances, calibrated reflectivities, rotation angles, synchronized time stamps (μs resolution)
- $GPRMC$ NMEA sentence from GPS receiver

#### A.4.2 Mechanical Drawings VLP-16 LiDAR sensor
APPROXIMATE BEAM CENTERS
DOES NOT REPRESENT BEAM WIDTHS
BEAMS ARE APPROXIMATELY 0.75 TALL AT FOCAL POINT
BEAM DIVERGENCE IS APPROXIMATELY 3 MILLIRADIANS HORIZONTALLY AND 1.5 MILLIRADIANS VERTICALLY
APPLY VERTICAL CORRECTION FOR GREATEST ACCURACY
Appendix B
INTELLIGENT NAVIGATION PLATFORM CODES

B.1 Driving codes

B.1.1 Forward and reverse control

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(0,1);
//Defining pins
int P = A1;
int valP = 0;
int scale = 0;
void setup() {
  mySerial.begin(9600);
}
void loop() {
  valP = analogRead(P);
  Serial.print(valP);
  scale = map((valP),0,1023,1,255);
  mySerial.write(scale);
}
```

B.1.2 Steering by wire code

```
// Define stepper motor connections and steps per revolution:
#define dirPin 12
#define stepPin 13
int p=A1;
int valp=0;
int nextval = 0;
int scale =0;
int bufferval =20;
//this is the buffer to the motor. it changes the minimum number of
steps that the motor needs to take before it activates
//lowering this could make it significantly jumpier. lowering it will
also allow it to react to much smaller changes
int bufferval2=10;
//this buffer changes how the system reacts when it gets told to steer
far one way and then quickly told to steer far the other way before
it has had time to complete the first action
int startp=0;
int motorpulses =278;
//motorpulses is how many pulses the motor takes to go from all the
way right to all the way left
int currentval=motorpulses/2;

void setup() {
  // Declare pins as output:
```
pinMode(stepPin, OUTPUT);
pinMode(dirPin, OUTPUT);
digitalWrite(stepPin, LOW);
Serial.begin (9600);
}

void loop() {
start:
valp=analogRead(p);
Serial.println(valp);
nextval = map(valp,0,1023,0,motorpulses);
if(nextval > (currentval+bufferval)){
startp=analogRead(p);
while(nextval > (currentval)){
valp=analogRead(p);
if(valp<(startp-bufferval2)){
goto start;
}
digitalWrite(dirPin, HIGH);
digitalWrite(stepPin, HIGH);
delayMicroseconds(3000);
digitalWrite(stepPin, LOW);
delayMicroseconds(3000);
currentval=currentval+1;
}
}
if(nextval < (currentval-bufferval)){
startp=analogRead(p);
while(nextval < (currentval)){
valp=analogRead(p);
if(valp>(startp+bufferval2)){
goto start;
}
digitalWrite(dirPin, LOW);
digitalWrite(stepPin, HIGH);
delayMicroseconds(3000);
digitalWrite(stepPin, LOW);
delayMicroseconds(3000);
currentval=currentval-1;
}newline newline newline

B.2 ADAS Codes

B.2.1 Automatic Emergency Braking

#include <SoftwareSerial.h>
SoftwareSerial mySerial(0,1);

    // Ultrasonic Sensor
const int trigPin = 2;
const int echoPin1 = 22;
const int echoPin2 = 24;
const int echoPin3 = 26;

    // RGB LED
const int red = 5;
const int green = 6;
const int blue = 7;
    // LEDs
const int LED1 = 10;
const int LED2 = 11;
const int LED3 = 12;
const int LED4 = 13;
const int LED5 = 8;
const int LED6 = 9;

    int fullsend = 255;
int chill = 191.5;
int save = 127;

    float duration1, duration2, duration3, distance1, distance2, distance3, distance;

    void setup() {
        mySerial.begin(9600);

        pinMode(trigPin, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(echoPin2, INPUT);
pinMode(echoPin3, INPUT);

        pinMode(red, OUTPUT);
pinMode(green, OUTPUT);
pinMode(blue, OUTPUT);

        pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(LED5, OUTPUT);
pinMode(LED6, OUTPUT);
    }

    void loop() {

        digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = (duration1*.0343)/2;
delay(100);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2 = (duration2*.0343)/2;
delay(100);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration3 = pulseIn(echoPin3, HIGH);
distance3 = (duration3*.0343)/2;
delay(100);

distance = (distance1 + distance2 + distance3)/3;

  if (distance > 100){
RGB_color(255,255,0); // Green
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
digitalWrite(LED6, LOW);
mySerial.write(fullsend);
  } else if (distance >= 35 && distance < 100){
RGB_color(255,0,255); // Blue
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, HIGH);
digitalWrite(LED4, HIGH);
digitalWrite(LED5, LOW);
digitalWrite(LED6, LOW);
mySerial.write(chill);
  } else if (distance < 35){
RGB_color(0,255,255); // Red
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, HIGH);
digitalWrite(LED6, HIGH);
mySerial.write(save);
}

void RGB_color(int red_value, int green_value, int blue_value) {
analogWrite(red, red_value);
analogWrite(green, green_value);
analogWrite(blue, blue_value);
}
B.2.2 Parking Assistance

```cpp
#include <SoftwareSerial.h>
SoftwareSerial mySerial(0, 1);

    // Ultrasonic Sensor
const int trigPin = 2;
const int echoPin1 = 3;
const int echoPin2 = 4;

    // RGB LED
const int red = 5;
const int green = 6;
const int blue = 7;

    // LEDs
const int LED1 = 10;
const int LED2 = 11;
const int LED3 = 12;
const int LED4 = 13;
const int LED5 = 8;
const int LED6 = 9;

    int fullsend = 220;
int chill = 180;
int park = 126;

    float duration1, duration2, distance1, distance2, distance;

    void setup() {
    
        mySerial.begin(9600);

        pinMode(trigPin, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(echoPin2, INPUT);

        pinMode(red, OUTPUT);
pinMode(green, OUTPUT);
pinMode(blue, OUTPUT);

        pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(LED5, OUTPUT);
pinMode(LED6, OUTPUT);
    }

    void loop() {
    
        digitalWrite(trigPin, LOW);
```
```c
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = (duration1*.0343)/2;
delay(100);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2 = (duration2*.0343)/2;
delay(100);

distance = (distance1 + distance2)/2;

if (distance > 100)
{
    RGB_color(255,255,0);  // Green
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
    mySerial.write(fullsend);
}
else if (distance >= 20 && (distance < 100))
{
    RGB_color(255,0,255);  // Blue
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
    mySerial.write(chill);
}
else if (distance < 20)
{
    RGB_color(0,255,255);  // Red
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, HIGH);
    digitalWrite(LED6, HIGH);
    mySerial.write(park);
}

void RGB_color(int red_value, int green_value, int blue_value)
{
analogWrite(red, red_value);
analogWrite(green, green_value);
analogWrite(blue, blue_value);
}
```
B.2.3 Ultrasonic sensors Arduino test

```python
import RPi.GPIO as GPIO
from itertools import count
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import time
from datetime import datetime
GPIO.setwarnings(False)

x_vals = []
y_vals = []
distList = []
x_ = []
US = []
index = count()
startTime = time.time()

def UltraAct():
    GPIO.setmode(GPIO.BCM)
    TRIG = 23
    ECHOUS = 21
    ECHO_LIST = [ECHOUS]

    print("Distance Measurement In Progress @ ",datetime.now().strftime("%H:%M:%S.%f"))
    GPIO.setup(TRIG,GPIO.OUT)
    GPIO.setup(ECHO_LIST,GPIO.IN)
    GPIO.output(TRIG, False)
    time.sleep(0.06)
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    pulse_startUS = time.time()
    while GPIO.input(ECHOUS) == 0:
        pulse_startUS = time.time()
    while GPIO.input(ECHOUS) == 1:
        pulse_endUS = time.time()
    pulse_durationUS = pulse_endUS - pulse_startUS
    distanceUS = round(pulse_durationUS*17150,2)

    print("Sensor Distance:",distanceUS,"cm")
    time.sleep(0.1)
    GPIO.cleanup()
    distList.append([distanceUS])
    UltraAct()
```

B.3 Live data visualization

B.3.1 Ultrasonic sensors live data visualization

```python
import RPi.GPIO as GPIO
from itertools import count
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import time
from datetime import datetime
GPIO.setwarnings(False)
plt.style.use('bmh')

def ultraAct():
x_vals = []
y_vals = []
distList = []
x = []
US1 = []
US2 = []
USB1 = []
USB2 = []

    index = count()
    startTime = time.time()

    def UltrasonicPing():
        GPIO.setmode(GPIO.BCM)
        TRIG = 23 #16
        ECHOUS1 = 24 #18
        ECHOUS2 = 17 #11
        ECHOUSB1 = 20 #38
        ECHOUSB2 = 21 #40
        ECHO_LIST = [ECHOUS1, ECHOUS2, ECHOUSB1, ECHOUSB2]

        print ("Distance Measurement In Progress @
"+datetime.now().strftime("%H:%M:%S.%f"))
        GPIO.setup(TRIG,GPIO.OUT)
        GPIO.setup(ECHO_LIST,GPIO.IN)
        GPIO.output(TRIG, False)
        time.sleep(0.06)
        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

        # Sensor 1

        pulse_startUS1 = time.time()
        while GPIO.input(ECHOUS1) == 0:
```
pulse_startUS1 = time.time()
while GPIO.input(ECHOUS1) == 1:
pulse_endUS1 = time.time()
pulse_durationUS1 = pulse_endUS1 - pulse_startUS1
distanceUS1 = round(pulse_durationUS1*17150,2)

time.sleep(0.06)
GPIO.output(TRIG, False)
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

# Sensor 2

pulse_startUS2 = time.time()
while GPIO.input(ECHOUS2) == 0:
pulse_startUS2 = time.time()
while GPIO.input(ECHOUS2) == 1:
pulse_endUS2 = time.time()
pulse_durationUS2 = pulse_endUS2 - pulse_startUS2
distanceUS2 = round(pulse_durationUS2*17150,2)

time.sleep(0.06)
GPIO.output(TRIG, False)
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

# Sensor Back 1

pulse_startUSB1 = time.time()
while GPIO.input(ECHOUSB1) == 0:
pulse_startUSB1 = time.time()
while GPIO.input(ECHOUSB1) == 1:
pulse_endUSB1 = time.time()
pulse_durationUSB1 = pulse_endUSB1 - pulse_startUSB1
distanceUSB1 = round(pulse_durationUSB1*17150,2)

time.sleep(0.06)
GPIO.output(TRIG, False)
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

# Sensor Back 2

pulse_startUSB2 = time.time()
while GPIO.input(ECHOUSB2) == 0:
pulse_startUSB2 = time.time()
while GPIO.input(ECHOUSB2) == 1:
pulse_endUSB2 = time.time()
pulse_durationUSB2 = pulse_endUSB2 - pulse_startUSB2
distanceUSB2 = round(pulse_durationUSB2*17150,2)
print("Sensor 1 Distance:", distanceUS1, "cm")
print("Sensor 2 Distance:", distanceUS2, "cm")
print("Sensor Back 1 Distance:", distanceUSB1, "cm")
print("Sensor Back 2 Distance:", distanceUSB2, "cm")
time.sleep(0.1)
GPIO.cleanup()
distList.append([distanceUS1, distanceUS2, distanceUSB1, distanceUSB2])

    def UltrasonicGraph(i):
        UltrasonicPing()
        x_.append((time.time() - startTime))

        US1.append(distList[i][0])
        US2.append(distList[i][1])
        USB1.append(distList[i][2])
        USB2.append(distList[i][3])
        plt.cla()

        plt.plot(x_, US1, label=f'Sensor 1 distance: distList[i][0] cm')
        plt.plot(x_, US2, label=f'Sensor 2 Distance: distList[i][1] cm')
        plt.plot(x_, USB1, label=f'Sensor Back 1 Distance: distList[i][2] cm')
        plt.plot(x_, USB2, label=f'Sensor Back 2 Distance: distList[i][3] cm')
        plt.title('Distance Ultrasonic Sensors', fontsize = 40)
        plt.xlabel('Time (s)', fontsize = 20)
        plt.ylabel('Distance (cm)', fontsize = 20)
        plt.ylim((0, 600))

        # plt.legend(loc='upper right', fontsize = 20) plt.tight_layout()

        ani = FuncAnimation(plt.gcf(), UltrasonicGraph, interval=1)
    plt.tight_layout()
    plt.show()
    test = 2
    return test

    ultraAct()
close all; clear; clc;

path = pwd;

Analysis = menu('Select to process the data from','Automatic Emergency Braking',
    'Parking Assistance',...
    'Blind Spot Detection','LSTM','LiDAR 3D mapping of campus','ROI LiDAR', 'LiDAR
Intensity Analysis');

if Analysis == 1

    % Automatic Emergency Braking Analysis

    fprintf('Automatic Emergency Braking Analysis Started:  

    prompt_run = 'Input test number:  ';
    run = input(prompt_run, 's');

    AEB = load(strcat(path,'\ADAS\Data\AEB\AEB\AEB_run',run,'.csv'));

    % AEB Analysis
    AEB(1,:) = [];
    Time_AEB = AEB(:,1);
    MPH1_AEB = AEB(:,6);
    MPH2_AEB = AEB(:,6);
    d1_AEB = AEB(:,7);
    d2_AEB = AEB(:,8);

    figure;
    plot(Time_AEB, MPH1_AEB,'r--','LineWidth',3);
    hold on
    plot(Time_AEB, MPH2_AEB,'r--','LineWidth',5);
    xlabel('Time (s)', 'FontSize',22, 'FontWeight', 'bold')
    ylabel('Speed (m/s)', 'FontSize',22, 'FontWeight', 'bold')
    title('Speed vs Time Graph for AEB', 'FontSize',26)
    grid on
    grid minor
    legend('Rotary Encoder', 'FontSize',22)

    figure;
    plot(Time_AEB, d1_AEB,'b--','LineWidth',3);
    hold on
    plot(Time_AEB, d2_AEB,'b--','LineWidth',5);
    xlabel('Time (s)', 'FontSize',22, 'FontWeight', 'bold')
    ylabel('Distance (m)', 'FontSize',22, 'FontWeight', 'bold')
    title('Distance vs Time Graph for AEB', 'FontSize',26)
    grid on
    grid minor
    legend('Ultrasonic Sensor', 'FontSize',22)

    figure;
    colororder({'r','b'})
    yyaxis left
    plot(Time_AEB, MPH2_AEB,'r--','LineWidth',5)
yyaxis right
plot(Time_AEB, d2_AEB, 'b--', 'LineWidth', 5)

yyaxis left
title('Distance and Speed vs Time AEB', 'FontSize', 26)
xlabel('Time (s)', 'FontSize', 22, 'FontWeight', 'bold')

yyaxis right
ylabel('Distance (m)', 'FontSize', 22, 'FontWeight', 'bold')
legend('Speed Rotary Encoder', 'Distance Ultrasonic Sensor', 'FontSize', 22)
grid on
grid minor

%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
% Workbook: F:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Results\AEB.xlsx
% Worksheet: AEB
% Auto-generated by MATLAB on 19-Mar-2023 20:49:25

%% Set up the Import Options and import the data
opts = spreadsheetImportOptions("NumVariables", 7);

% Specify sheet and range
opts.Sheet = "AEB";
opts.DataRange = "J2:P9";

% Specify column names and types
opts.VariableNames = ["Test1", "AlertDistance m", "AlertVelocity m/s", "BrakingDistance m", "BrakingVelocity m/s", "BrakingAcceleration m/s^2", "BrakingTTCs"];
opts.VariableTypes = ["string", "double", "double", "double", "double", "double", "double", "double"];

% Specify variable properties
opts = setvaropts(opts, "Test1", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Test1", "EmptyFieldRule", "auto");

% Import the data
AEBAnalysis = readtable("G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Results\AEB.xlsx", opts, "UseExcel", false);

%% Clear temporary variables
clear opts

%% Clear temporary variables
clear opts

figure;
X = categorical({'Run 1', 'Run 2', 'Run 3', 'Run 4', 'Run 5'});
TTC = AEBAnalysis(2:6,7);
TTC = table2array(TTC);
bar(X,TTC,"FaceColor",[0 0.7 0.7])
elseif Analysis == 2

% Parking Assistance

fprintf('Parking Assistance Analysis Started: \n\n')

prompt_run = 'Input test number:    ';
run = input(prompt_run,'s');

PAD = load(strcat(path,'\ADAS\Data\PA\David',run,'.csv'));

% PA David run
PAD(1,:) = []; Time_PAD = PAD(:,6);
MPH1_PAD = PAD(:,7);
MPH2_PAD = PAD(:,7);
d1_PAD = PAD(:,8);
d2_PAD = PAD(:,9);

figure;
plot(Time_PAD, MPH1_PAD,'r--','LineWidth',5);
xlabel('Time (s)', 'FontSize',22, 'FontWeight','bold')
ylabel('Speed (m/s)', 'FontSize',22, 'FontWeight','bold')
title('Speed vs Time Graph for PA', 'FontSize',26)
grid on
grid minor
legend('Rotary Encoder', 'FontSize',22)

figure;
plot(Time_PAD, d1_PAD,'b--','LineWidth',2);
hold on
plot(Time_PAD, MPH2_PAD,'b--','LineWidth',2);
xlabel('Time (s)', 'FontSize',22, 'FontWeight','bold')
ylabel('Distance (m)', 'FontSize',22, 'FontWeight','bold')
title('Distance vs Time Graph for PA', 'FontSize',26)
grid on
grid minor
legend('Ultrasonic Sensor', 'FontSize',22)

figure;
colororder({'r','b'})
yyaxis left
plot(Time_PAD, MPH1_PAD,'r--','LineWidth',5)
yyaxis right
plot(Time_PAD, d2_PAD,'b--','LineWidth',5)
yyaxis left
title('Distance and Speed vs Time PA', 'FontSize',26)
xlabel('Time (s)', 'FontSize', 22, 'FontWeight', 'bold')
ylabel('Speed (m/s)', 'FontSize', 22, 'FontWeight', 'bold')
yyaxis right
ylabel('Distance (m)', 'FontSize', 22, 'FontWeight', 'bold')
legend('Speed Rotary Encoder', 'Distance Ultrasonic Sensor', 'FontSize', 22)
grid on
grid minor

%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
% Workbook: F:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Results\PA.xlsx
% Worksheet: PA
% Auto-generated by MATLAB on 19-Mar-2023 20:15:05

%% Set up the Import Options and import the data
opts = spreadsheetImportOptions("NumVariables", 7);

% Specify sheet and range
opts.Sheet = "PA";
opts.DataRange = "M4:S11";

% Specify column names and types
opts.VariableTypes = ["string", "double", "double", "double", "double", "double", "double", "double"];

% Specify variable properties
opts = setvaropts(opts, "Test", "WhitespaceRule", "preserve");
opts = setvaropts(opts, "Test", "EmptyFieldRule", "auto");

% Import the data
PAAnalysis = readtable("G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Results\PA.xlsx", opts, "UseExcel", false);

%% Clear temporary variables
clear opts

figure;
X = categorical({'Run 1', 'Run 2', 'Run 3', 'Run 4', 'Run 5'});
TTC = PAAnalysis(2:6,7);
TTC = table2array(TTC);
bar(X,TTC, 'FaceColor', [0 0.7 0.7])
title('Time to Collision PA', 'FontSize', 26)
ylabel('Time to collision (s)', 'FontSize', 22)
set(gca, 'FontSize', 14);
grid on
grid minor

elseif Analysis == 3

else

elseif Analysis == 3

end
% Blind spot detection
fprintf('Blind Spot Analysis Started:  

')
elseif Analysis == 4
    % LSTM Analysis
fprintf('LSTM Machine Learning Model Analysis Started  

')

%% Figures
load('G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Codes\LSTM\Results\1m_results.mat')
load('G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Codes\LSTM\Results\2m_results.mat')
% load('G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Codes\LSTM\Results\3m_results.mat')
% load('G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Codes\LSTM\Results\4m_results.mat')
% load('G:\David\GSU\Research\Masters\Projects\Intelligent Vehicles\Thesis Project\Codes\LSTM\Results\5m_results.mat')

figure(1)
title('LSTM Sensor Fusion Evaluation')
line([timeplot(1),timeplot(end)], [truthROI,truthROI], 'LineWidth', 2.5, 'Color', '[0, 1, 0, 0.8]', 'LineStyle', '-.')
hold on
plot(timeplot,avgmeasurements, ':o', 'LineWidth', 2, 'MarkerSize', 1) %avgsensor measurements
ylim([.50 4.5]);
xlim([950 500000])
hold on
plot(timeplot,YPred, ':.', 'LineWidth',1.5, 'MarkerSize', 8) %lstm

x = (200, 18.5, {sprintf('Median Measured Distance %0.2f m',avg2d), sprintf('Std. Dev. Measured Distance %0.2f m',std2d)}, 'EdgeColor', 'k', 'BackgroundColor', [1-eps, 1, 1])
plot(nan, nan, 'Color', 'White');
plot(nan, nan, 'Color', 'White');
plot(nan, nan, 'Color', 'White');
plot(nan, nan, 'Color', 'White');
plot(nan, nan, 'Color', 'White');

legend('Ground Truth Distance', 'Average Sensor Measurement', ...
'LSTM Fusion', ...
sprintf('Ground Truth %0.3f m',truthROI), ...
sprintf('Avg. Measured Distance %0.3f m',mean(avgmeasurements)), ...
sprintf('Avg. LSTM Fusion Distance %0.3f m',mean(YPred)), ...
sprintf('Std. Dev. Measured Distance %0.3f m',std(avgmeasurements)), ...
sprintf('Std. Dev. LSTM Fusion Distance %0.3f m',std(YPred)));
xlabel('Time (ms)')
ylabel('Distance Measured (m)')
ylim([4 5.8])
grid on
grid minor
%mean

figure(2)
subplot(2,2,1)
scatter((1:1:length(ROItest)),ROItest(1, :), 10, 'k')
title('2D LiDAR')
xlabel('ROI Datapoints')
ylabel('Distance Measured (m)')
text(200, 18.5, {sprintf('Median Measured Distance %0.2f m',avg2d), sprintf('Std. Dev. Measured Distance %0.2f m',std2d)}, 'EdgeColor', 'k', 'BackgroundColor', [1-eps, 1, 1])
grid on
grid minor

subplot(2,2,2)
scatter((1:1:length(ROItest)),ROItest(2, :), 10, 'k')
title('Ultrasonic')
xlabel('ROI Datapoints')
ylabel('Distance Measured (m)')
text(200, .5812, {sprintf('Avg. Measured Distance %0.2f m',avgus), sprintf('Std. Dev. Measured Distance %0.2f m',stdus)}, 'EdgeColor', 'k', 'BackgroundColor', [1-eps, 1, 1])
grid on
grid minor

subplot(2,2,3)
scatter((1:1:length(ROItest)),ROItest(3, :), 10, 'k')
title('Solid State LiDAR')
xlabel('ROI Datapoints')
ylabel('Distance Measured (m)')
% text(200, .645, {sprintf('Avg. Measured Distance %0.2f m',avgssl), sprintf('Std. Dev. Measured Distance %0.2f m',stdssl)}, 'EdgeColor', 'k', 'BackgroundColor', [1-eps, 1, 1])
grid on
grid minor

subplot(2,2,4)
scatter((1:1:length(YPred)),YPred, 10, 'k')
yline(2);
title('LSTM Sensor Fusion')
xlabel('ROI Datapoints')
ylabel('Distance Measured (m)')
text(200, .66, {sprintf('Avg. Fusion Distance %0.2f m',avglstm), sprintf('Std. Dev. Fusion Distance %0.2f m',stdlstm)}, 'EdgeColor', 'k', 'BackgroundColor', [1-eps, 1, 1])
ylim([0 2.8])
grid on
grid minor

% LSTM data
% 1m
load(strcat(path,'\LSTM\Results\1m_results.mat'));
oneLSTM = [];
oneLSTM = [oneLSTM;YPred];
clear('YPred')
% 2m
load(strcat(path,'\LSTM\Results\2m_results.mat'));
twoLSTM = [];
twoLSTM = [twoLSTM;YPred];
clear('YPred')
% 3m
load(strcat(path,'\LSTM\Results\3m_results.mat'));
threeLSTM = [];
threeLSTM = [threeLSTM;YPred];
clear('YPred')
% 4m
load(strcat(path,'\LSTM\Results\4m_results.mat'));
fourLSTM = [];
fourLSTM = [fourLSTM;YPred];
clear('YPred')
% 5m
load(strcat(path,'\LSTM\Results\5m_results.mat'));
fiveLSTM = [];
fiveLSTM = [fiveLSTM;YPred];
clear('YPred')

% Raw data
% 1m
load(strcat(path,'\LSTM\Results\1m_results.mat'));
oneRaw = [];
oneRaw = [oneRaw; ROI];
clear('ROI')
% 2m
load(strcat(path,'\LSTM\Results\2m_results.mat'));
twoRaw = [];
twoRaw = [twoRaw; ROI];
clear('ROI')
% 3m
load(strcat(path,'\LSTM\Results\3m_results.mat'));
threeRaw = [];
threeRaw = [threeRaw; ROI];
clear('ROI')
% 4m
load(strcat(path,'\LSTM\Results\4m_results.mat'));
fourRaw = [];
fourRaw = [fourRaw; ROI];
clear('ROI')
% 5m
load(strcat(path,'\LSTM\Results\5m_results.mat'));
fiveRaw = [];
fiveRaw = [fiveRaw; ROI];
clear('ROI')

%% Standard Deviation

% LSTM
% LSTM 1m
avg_1m_LSTM = mean(oneLSTM);
one_LSTM = [avg_1m_LSTM 1];
std_1m = std(one_LSTM);
fprintf('Standard deviation 1 m = %d
', std_1m)
% LSTM 2m
avg_2m_LSTM = mean(twoLSTM);
two_LSTM = [avg_2m_LSTM 2];
std_2m = std(two_LSTM);
fprintf('Standard deviation 2 m = %d\n', std_2m)

% LSTM 3m
avg_3m_LSTM = mean(threeLSTM);
three_LSTM = [avg_3m_LSTM 3];
std_3m = std(three_LSTM);
fprintf('Standard deviation 3 m = %d\n', std_3m)

% LSTM 4m
avg_4m_LSTM = mean(fourLSTM);
four_LSTM = [avg_4m_LSTM 4];
std_4m = std(four_LSTM);
fprintf('Standard deviation 4 m = %d\n', std_4m)

% LSTM 5m
avg_5m_LSTM = mean(fiveLSTM);
five_LSTM = [avg_5m_LSTM 5];
std_5m = std(five_LSTM);
fprintf('Standard deviation 5 m = %d\n', std_5m)

% Raw Data
% Raw 1m
avg_1m_RAW = mean(oneRaw(1:3,:));
avg_1m_RAW = mean(avg_1m_RAW);
one_RAW = [avg_1m_RAW 1];
std_1m_RAW = std(one_RAW);
fprintf('Standard deviation Raw data 1 m = %d\n', std_1m_RAW)

% Raw 2m
avg_2m_RAW = mean(twoRaw(1:3,:));
avg_2m_RAW = mean(avg_2m_RAW);
two_RAW = [avg_2m_RAW 2];
std_2m_RAW = std(two_RAW);
fprintf('Standard deviation Raw data 2 m = %d\n', std_2m_RAW)

% Raw 3m
avg_3m_RAW = mean(threeRaw(1:3,:));
avg_3m_RAW = mean(avg_3m_RAW);
three_RAW = [avg_3m_RAW 3];
std_3m_RAW = std(three_RAW);
fprintf('Standard deviation Raw data 3 m = %d\n', std_3m_RAW)

% Raw 4m
avg_4m_RAW = mean(fourRaw(1:3,:));
avg_4m_RAW = mean(avg_4m_RAW);
four_RAW = [std_3m_RAW 4];
std_4m_RAW = std(four_RAW);
fprintf('Standard deviation Raw data 4 m = %d\n', std_4m_RAW)

% Raw 5m
avg_5m_RAW = mean(fiveRaw(:,1:3));
avg_5m_RAW = mean(avg_5m_RAW);
five_RAW = [avg_5m_RAW 5];
std_5m_RAW = std(five_RAW);
fprintf('Standard deviation Raw data 5 m = %d\n', std_5m_RAW)

avg_STD_LSTM = [std_1m, std_2m, std_3m, std_4m, std_5m];
avg_STD_LSTM = mean(avg_STD_LSTM);
avg_STD_Raw = [std_1m_RAW, std_2m_RAW, std_3m_RAW, std_4m_RAW, std_5m_RAW];
avg_STD_Raw = mean(avg_STD_Raw);

%% Percentage Error

% LSTM
PE1m = abs(((avg_1m_LSTM-1)/1)*100);
fprintf('Percentage Error 1 m = %4.2f\n', PE1m)
PE2m = abs(((avg_2m_LSTM-2)/2)*100);
fprintf('Percentage Error 2 m = %4.2f\n', PE2m)
PE3m = abs(((avg_3m_LSTM-3)/3)*100);
fprintf('Percentage Error 3 m = %4.2f\n', PE3m)
PE4m = abs(((avg_4m_LSTM-4)/4)*100);
fprintf('Percentage Error 4 m = %4.2f\n', PE4m)
PE5m = abs(((avg_5m_LSTM-5)/5)*100);
fprintf('Percentage Error 5 m = %4.2f\n', PE5m)

% RAW
PE1m_Raw = abs(((avg_1m_RAW-1)/1)*100);
fprintf('Percentage Error Raw 1 m = %4.2f\n', PE1m_Raw)
PE2m_Raw = abs(((avg_2m_RAW-2)/2)*100);
fprintf('Percentage Error Raw 2 m = %4.2f\n', PE2m_Raw)
PE3m_Raw = abs(((avg_3m_RAW-3)/3)*100);
fprintf('Percentage Error Raw 3 m = %4.2f\n', PE3m_Raw)
PE4m_Raw = abs(((avg_4m_RAW-4)/4)*100);
fprintf('Percentage Error Raw 4 m = %4.2f\n', PE4m_Raw)
PE5m_Raw = abs(((avg_5m_RAW-5)/5)*100);
fprintf('Percentage Error Raw 5 m = %4.2f\n', PE5m_Raw)

avg_PE_LSTM = [PE1m, PE2m, PE3m, PE4m, PE5m];
avg_PE_LSTM = mean(avg_PE_LSTM);
avg_PE_RAW = [PE1m_Raw, PE2m_Raw, PE3m_Raw, PE4m_Raw, PE5m_Raw];
avg_PE_RAW = mean(avg_PE_RAW);

%% Graphs

XLimit = length(oneLSTM);

% Test Graph
figure;
plot((1:1:length(oneLSTM)),oneLSTM, '--k', 'LineWidth',1, "MarkerSize", 1)
xlim([0 XLimit])
ylim([0 1.2])
label = ['GT 1 m'];
yline(1, '-.b', 'LineWidth', 3)
xlabel('Data-points')
ylabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 1 m')
legend('LSTM 1 m')

% Distances Analysis
figure;
plot((1:1:length(oneLSTM)), oneLSTM, ':', 'Color', '#c21e0c', 'LineWidth', 2)
hold on
plot((1:1:length(twoLSTM)), twoLSTM, ':', 'Color', '#79c20c', 'LineWidth', 2)
plot((1:1:length(threeLSTM)), threeLSTM, ':', 'Color', '#0caac2', 'LineWidth', 2)
plot((1:1:length(fourLSTM)), fourLSTM, ':', 'Color', '#6d0cc2', 'LineWidth', 2)
plot((1:1:length(fiveLSTM)), fiveLSTM, ':', 'Color', '#d6cc11', 'LineWidth', 2)
yline(1, '-.b', 'GT 1 m')
yline(2, '-.b', 'GT 2 m')
yline(3, '-.b', 'GT 3 m')
yline(4, '-.b', 'GT 4 m')
yline(5, '-.b', 'GT 5 m')
xlim([0 XLimit])
ylim([0 XLimit])
l = {'1 m GT', '2 m GT', '3 m GT', '4 m GT', '5 m GT'};
legend(l, 'Location', 'BestOutside', 'FontSize', 13)
xlabel('Data-points')
ylabel('LSTM Precision (m)')
grid on
grid minor
title('Performance comparison LSTM fusion')

% Individual Analysis Critical Zone
figure;
% 1m
subplot(5,1,1)
plot((1:1:length(oneLSTM)), oneLSTM, '--', 'Color', '#141be0', 'LineWidth', 2,
'MarkerSize', 1)
xlim([0 XLimit])
ylim([0 1.2])
%label = {'GT 0.05 m'};
yline(0.05, '-.b', label)
xlabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 1 m')
% 2m
subplot(5,1,2)
plot((1:1:length(twoLSTM)), twoLSTM, '--', 'Color', '#141be0', 'LineWidth', 2,
'MarkerSize', 1)
xlim([0 XLimit])
ylim([0 2.2])
%label = {'GT 0.10 m'};
yline(0.10, '-.b', label)
xlabel('Data-points')
ylabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 2 m')

% 3m
subplot(5,1,3)
plot((1:1:length(threeLSTM)),threeLSTM, '--', 'Color', '#141be0', 'LineWidth', 2, 'MarkerSize', 1)
xlim([0 XLimit])
ylim([0 3.2])
%label = {'GT 0.15 m'};
%yline(0.15, '-.b', label)
xlabel('Data-points')
ylabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 3 m')

% 4m
subplot(5,1,4)
plot((1:1:length(fourLSTM)),fourLSTM, '--', 'Color', '#141be0', 'LineWidth', 2, 'MarkerSize', 1)
xlim([0 XLimit])
ylim([0 4.2])
%label = {'GT 0.20 m'};
%yline(0.20, '-.b', label)
xlabel('Data-points')
ylabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 4 m')

% 25 cm
subplot(5,1,5)
plot((1:1:length(fiveLSTM)),fiveLSTM, '--', 'Color', '#141be0', 'LineWidth', 2, 'MarkerSize', 1)
xlim([0 XLimit])
ylim([0 5.2])
%label = {'GT 0.25 m'};
%yline(0.25, '-.b', label)
xlabel('Data-points')
ylabel('ROI Distance (m)')
grid on
grid minor
title('Performance LSTM fusion 5 m')

figure;
X = categorical({'Raw Performance', 'LSTM Performance'});
bar_PE = [avg_PE_LSTM avg_PE_RAW];
a = bar(X, bar_PE);
a.FaceColor = 'flat';
a.CData(1,:) = [64/255, 227/255, 224/255];
a.CData(2,:) = [28/255, 9/255, 235/255];
ylabel('Percentage Error (%)')

title('Percentage Error Comparison of Raw Data Relative to Fusion Model', 'fontweight', 'bold', 'fontsize', 12)
grid on
grid minor

figure;
bar_STD = [avg_STD_LSTM avg_STD_Raw];
b = bar(X,bar_STD);
b.FaceColor = 'flat';
b.CData(1,:) = [64/255, 227/255, 224/255];
b.CData(2,:) = [28/255, 9/255, 235/255];
ylabel('Standard Deviation (m)')
title('Standard Deviation Comparison of Raw Data Relative to Fusion Model',
'fontweight','bold','fontsize',12)
grid on
grid minor

%% Percentage error and STD Graphs

PERAW = [PE1m_Raw, PE2m_Raw, PE3m_Raw, PE4m_Raw, PE5m_Raw];
PELSTM = [PE1m, PE2m, PE3m, PE4m, PE5m];
STDRAW = [std_1m_RAW, std_2m_RAW, std_3m_RAW, std_4m_RAW, std_5m_RAW];
STDLSTM = [std_1m, std_2m, std_3m, std_4m, std_5m];
Distance = [1 2 3 4 5];

figure;
plot(Distance, PERAW,'b--','LineWidth',5);
hold on
plot(Distance, PELSTM,'r--','LineWidth',5);
grid on
grid minor
legend('Percentage Error Raw Data', 'Percentage Error LSTM')
title('Percentage Error Comparison', 'FontSize',26,'FontWeight','bold');
ylabel('Percentage Error (%)','FontSize',22,'FontWeight','bold');

figure;
plot(Distance, STDRAW,'b--','LineWidth',5);
hold on
plot(Distance, STDLSTM,'r--','LineWidth',5);
grid on
grid minor
legend('Standard Deviation Raw Data', 'Standard Deviation LSTM')
title('Standard Deviation Comparison', 'FontSize',26,'FontWeight','bold');
ylabel('Standard Deviation','FontSize',22,'FontWeight','bold');

elseif Analysis == 5

% LiDAR 3D mapping of campus

fprintf('LiDAR 3D Mapping Analysis Started: \n\n')

MappingFile = load(strcat(path,'\AV\Mapping\matlab.mat'));
v = VideoWriter('CampusMapping', 'Uncompressed AVI');
v.FrameRate = 15;
open(v)
for i = 1:436
    figure = pcshow(MappingFile.lidar_data(i).data.Location);
    title('Point Cloud Mapping Analysis')
    xlabel('X(m)')
    ylabel('Y(m)')
    zlabel('Z(m)')
    set(gcf, 'InvertHardCopy', 'off')
    box on
    xlim([-20 20])
    ylim([-20 20])
    zlim([-0.70 2.0])
    saveas(figure, sprintf('Mapping_%d.png',i));
    frame= getframe(gcf);
    writeVideo(v,frame)
end

elseif Analysis == 6

    % Region of Interest LiDAR Analysis
    fprintf('ROI LiDAR Analysis has been selected
')

    cp1_BMW = load(strcat(path, 'LiDAR_Point_Clouds_David\BMW_S3_1_3.mat'));
    cp1_MAL = load(strcat(path, 'LiDAR_Point_Clouds_David\Chevy_Malibu_1_3.mat'));
    cp1_ESC1 = load(strcat(path, 'LiDAR_Point_Clouds_David\Ford_Escape_1_3.mat'));
    cp1_ESC2 = load(strcat(path, 'LiDAR_Point_Clouds_David\Ford_Escape_1_3.mat'));
    cp1_TAU = load(strcat(path, 'LiDAR_Point_Clouds_David\Ford_Taurus_1_3.mat'));
    cp1_CIV = load(strcat(path, 'LiDAR_Point_Clouds_David\Honda_Civic_1_3.mat'));

    % Short Analysis
    xlimits_BMW = [-1.28, 0.48];
    ylimits_BMW = [2.8, 5.4];
    zlimits_BMW = [-0.48, 0.6];
    player_BMW = pcplayer(xlimits_BMW,ylimits_BMW,zlimits_BMW);

    % ROI Preview
    cp_BMW = cp1_BMW.BMW_S3_1_3;
    view(player_BMW,cp_BMW)

    % Point Cloud Raw
    figure_2 = figure(2);
    pcshow(cp_BMW)

    % ROI visualization
    ROIL_BMW = [xlimits_BMW(1) xlimits_BMW(2) ylimits_BMW(1) ylimits_BMW(2) zlimits_BMW(1) zlimits_BMW(2)];
    Visualization_BMW = findPointsInROI(cp_BMW,ROIL_BMW);
    SPROI_BMW = select(cp_BMW,Visualization_BMW);
    raw_data_BMW = cp_BMW.Location;
    ROI_data_BMW = SPROI_BMW.Location;
    figure_3 = figure(3);
    pcshow(raw_data_BMW, [0.5 0.5 0.5])
    hold on
    pcshow(ROI_data_BMW, [0.27 0.51 0.71])
legend('Point Cloud Raw','ROI Points','Location','southoutside','Color',[1 1 1])
hold off
save('SPROI_BMW.mat','SPROI_BMW')

figure_4 = figure(4);
pcshow(SPROI_BMW)

%%% Chevrolet Malibu

% Short Analysis
xlimits_MAL = [-1.8, 0.3];
ylimits_MAL = [2.8, 5.4];
zlimits_MAL = [-0.44, 0.6];
player_MAL = pcplayer(xlimits_MAL,ylimits_MAL,zlimits_MAL);

% ROI Preview
cp_MAL = cp1_MAL.Chevy_Malibu_1_3;
view(player_MAL,cp_MAL)

% Point Cloud Raw
figure_5 = figure(5);
pcshow(cp_MAL)

% ROI visualization
ROIL_MAL = [xlimits_MAL(1) xlimits_MAL(2) ylimits_MAL(1) ylimits_MAL(2) zlimits_MAL(1) zlimits_MAL(2)];
Visualization_MAL = findPointsInROI(cp_MAL,ROIL_MAL);
SPROI_MAL = select(cp_MAL,Visualization_MAL);
raw_data_MAL = cp_MAL.Location;
ROI_data_MAL = SPROI_MAL.Location;
figure_6 = figure(6);
hold on
pcshow(ROI_data_MAL, [0.95 0.95 0.95])
legend('Point Cloud Raw','ROI Points','Location','southoutside','Color',[1 1 1])
hold off
save('SPROI_MAL.mat','SPROI_MAL')

figure_7 = figure(7);
pcshow(SPROI_MAL)

%%% Ford Escape 1

% Short Analysis
xlimits_ESC1 = [-1.25, 0.9];
ylimits_ESC1 = [2.7, 5.4];
zlimits_ESC1 = [-0.44, 0.92];
player_ESC1 = pcplayer(xlimits_ESC1,ylimits_ESC1,zlimits_ESC1);

% ROI Preview
cp_ESC1 = cp1_ESC1.Ford_Escape_1_3;
view(player_ESC1,cp_ESC1)

% Point Cloud Raw
figure_8 = figure(8);
pcshow(cp_ESC1)

% ROI visualization
ROIL_ESC1 = [xlimits_ESC1(1) xlimits_ESC1(2) ylimits_ESC1(1) ylimits_ESC1(2) zlimits_ESC1(1) zlimits_ESC1(2)];
Visualization_ESC1 = findPointsInROI(cp_ESC1,ROIL_ESC1);
SPROI_ESC1 = select(cp_ESC1,Visualization_ESC1);
raw_data_ESC1 = cp_ESC1.Location;
ROI_data_ESC1 = SPROI_ESC1.Location;
figure_9 = figure(9);
pcshow(raw_data_ESC1, [0.5 0.5 0.5])
hold on
pcshow(ROI_data_ESC1, [0.27 0.51 0.71])
legend('Point Cloud Raw','ROI Points','Location','southoutside','Color',[1 1 1])
hold off
save('SPROI_ESC1.mat','SPROI_ESC1')

figure_10 = figure(10);
pcshow(SPROI_ESC1)

%% Ford Escape 2

% Short Analysis
xlimits_ESC2 = [-1.25, 0.9];
ylimits_ESC2 = [2.7, 5.4];
zlimits_ESC2 = [-0.44, 0.92];
player_ESC2 = pcplayer(xlimits_ESC2,ylimits_ESC2,zlimits_ESC2);

% ROI Preview
cp_ESC2 = cp1_ESC2.Ford_Escape_1_3;
view(player_ESC2,cp_ESC2)

% Point Cloud Raw
figure_11 = figure(11);
pcshow(cp_ESC2)

% ROI visualization
ROIL_ESC2 = [xlimits_ESC2(1) xlimits_ESC2(2) ylimits_ESC2(1) ylimits_ESC2(2) zlimits_ESC2(1) zlimits_ESC2(2)];
Visualization_ESC2 = findPointsInROI(cp_ESC2,ROIL_ESC2);
SPROI_ESC2 = select(cp_ESC2,Visualization_ESC2);
raw_data_ESC2 = cp_ESC2.Location;
ROI_data_ESC2 = SPROI_ESC2.Location;
figure_12 = figure(12);
pcshow(raw_data_ESC2, [0.5 0.5 0.5])
hold on
pcshow(ROI_data_ESC2, [0.27 0.51 0.71])
legend('Point Cloud Raw','ROI Points','Location','southoutside','Color',[1 1 1])
hold off
save('SPROI_ESC2.mat','SPROI_ESC2')

figure_13 = figure(13);
pcshow(SPROI_ESC2)
%% Ford Taurus

% Short Analysis
xlimits_TAU = [-2.25, -.3];
ylimits_TAU = [2.7, 5.4];
zlimits_TAU = [-0.7, 0.92];
player_TAU = pcplayer(xlimits_TAU,ylimits_TAU,zlimits_TAU);

% ROI Preview
cp_TAU = cp1_TAU.Ford_Taurus_1_3;
view(player_TAU,cp_TAU)

% Point Cloud Raw
figure_14 = figure(14);
pcshow(cp_TAU)

% ROI visualization
ROIL_TAU = [xlimits_TAU(1) xlimits_TAU(2) ylimits_TAU(1) ylimits_TAU(2) zlimits_TAU(1) zlimits_TAU(2)];
Visualization_TAU = findPointsInROI(cp_TAU,ROIL_TAU);
SPROI_TAU = select(cp_TAU,Visualization_TAU);
raw_data_TAU = cp_TAU.Location;
ROI_data_TAU = SPROI_TAU.Location;
figure_15 = figure(15);
pcshow(raw_data_TAU, [0.5 0.5 0.5])
hold on
pcshow(ROI_data_TAU, [0.83 0.68 0.22])
legend('Point Cloud Raw','ROI Points','Location','southoutside','Color',[1 1 1])
hold off
save('SPROI_TAU.mat','SPROI_TAU')
figure_16 = figure(16);
pcshow(SPROI_TAU)

%% Honda Civic

% Short Analysis
xlimits_CIV = [-2.45, -.3];
ylimits_CIV = [2.7, 5.4];
zlimits_CIV = [-0.7, 0.92];
player_CIV = pcplayer(xlimits_CIV,ylimits_CIV,zlimits_CIV);

% ROI Preview
cp_CIV = cp1_CIV.Honda_Civic_1_3;
view(player_CIV,cp_CIV)

% Point Cloud Raw
figure_17 = figure(17);
pcshow(cp_CIV)

% ROI visualization
ROIL_CIV = [xlimits_CIV(1) xlimits_CIV(2) ylimits_CIV(1) ylimits_CIV(2) zlimits_CIV(1) zlimits_CIV(2)];
Visualization_CIV = findPointsInROI(cp_CIV,ROIL_CIV);
SPROI_CIV = select(cp_CIV,Visualization_CIV);
raw_data_CIV = cp_CIV.Location;
ROI_data_CIV = SPROI_CIV.Location;
figure_18 = figure(18);
pcshow(raw_data_CIV, [0.5 0.5 0.5])
hold on
pcshow(ROI_data_CIV, [0 0.14 0.42])
legend('Point Cloud Raw', 'ROI Points', 'Location', 'southoutside', 'Color', [1 1 1])
hold off
save('SPROI_CIV.mat', 'SPROI_CIV')

figure_19 = figure(19);
pcshow(SPROI_CIV)

elseif Analysis == 7

% Intensity Values LiDAR Analysis

fprintf('LiDAR Intensity Analysis')

load(strcat(path, '\LiDAR\Data_VLP16\SPROI_BMW.mat'));
load(strcat(path, '\LiDAR\Data_VLP16\SPROI_CIV.mat'));
load(strcat(path, '\LiDAR\Data_VLP16\SPROI_ESC1.mat'));
load(strcat(path, '\LiDAR\Data_VLP16\SPROI_MAL.mat'));
load(strcat(path, '\LiDAR\Data_VLP16\SPROI_TAU.mat'));

% Data Analysis

VLP16_Total_Intensity = [];
VLP16_TotalNOP = [];

% BMW
VLP16_BMWIntensity = mean(SPROI_BMW.Intensity);
VLP16_Total_Intensity = [VLP16_Total_Intensity; VLP16_BMWIntensity];
VLP16_BMWNOP = mean(SPROI_BMW.Count);
VLP16_TotalNOP = [VLP16_TotalNOP; VLP16_BMWNOP];

% Honda Civic
VLP16_CIVIntensity = mean(SPROI_CIV.Intensity);
VLP16_Total_Intensity = [VLP16_Total_Intensity; VLP16_CIVIntensity];
VLP16_CIVNOP = mean(SPROI_CIV.Count);
VLP16_TotalNOP = [VLP16_TotalNOP; VLP16_CIVNOP];

% Ford Scape
VLP16_ESC1Intensity = mean(SPROI_ESC1.Intensity);
VLP16_Total_Intensity = [VLP16_Total_Intensity; VLP16_ESC1Intensity];
VLP16_ESC1NOP = mean(SPROI_ESC1.Count);
VLP16_TotalNOP = [VLP16_TotalNOP; VLP16_ESC1NOP];

% Malibu
VLP16_MALIntensity = mean(SPROI_MAL.Intensity);
VLP16_Total_Intensity = [VLP16_Total_Intensity; VLP16_MALIntensity];
VLP16_MALNOP = mean(SPROI_MAL.Count);
VLP16_TotalNOP = [VLP16_TotalNOP; VLP16_MALNOP];
% Taurus
VLP16_TAUIntensity = mean(SPROI_TAU.Intensity);
VLP16_Total_Intensity = [VLP16_Total_Intensity;VLP16_TAUIntensity];
VLP16_TAUNOP = mean(SPROI_TAU.Count);
VLP16_TotalNOP = [VLP16_TotalNOP;VLP16_TAUNOP];

% Graphing Intensity
figure(1);
car = categorical({'BMW', 'Civic', 'Ford Escape', 'Malibu', 'Taurus'});
I_Graph = bar(car,VLP16_Total_Intensity);
I_Graph.FaceColor = 'flat';
I_Graph.CData(1, :) = [(48/255) (133/255) (161/255)];
I_Graph.CData(2, :) = [(5/255) (2/255) (184/255)];
I_Graph.CData(3, :) = [(255/255) (254/255) (252/255)];
I_Graph.CData(4, :) = [(135/255) (134/255) (132/255)];
I_Graph.CData(5, :) = [0.83 0.68 0.22];
title('Intensity Value Comparison')
grid on
grid minor
set(gca, 'color',[217/255 217/255 217/255]);
xlabel('Car Type and Color');
ylabel('Average Intensity Values of Points Detected');

% Graphing NOP
figure(2);
car = categorical({'BMW', 'Civic', 'Ford Escape', 'Malibu', 'Taurus'});
NOP_Graph = bar(car,VLP16_TotalNOP);
NOP_Graph.FaceColor = 'flat';
NOP_Graph.CData(1, :) = [(48/255) (133/255) (161/255)];
NOP_Graph.CData(2, :) = [(5/255) (2/255) (184/255)];
NOP_Graph.CData(3, :) = [(255/255) (254/255) (252/255)];
NOP_Graph.CData(4, :) = [(135/255) (134/255) (132/255)];
NOP_Graph.CData(5, :) = [0.83 0.68 0.22];
title('Number of Points Comparison')
grid on
grid minor
set(gca, 'color',[217/255 217/255 217/255]);
xlabel('Car Type and Color');
ylabel('Number of Points Detected');
end