Relationship of Simulator and Emulator and Real Experiments on Intelligent Transportation Systems

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Güçhan Özbilgin, B.S.

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2016

Master’s Examination Committee:

Prof. Ümit Özgüner, Advisor

Prof. Keith A. Redmill
Abstract

This thesis focuses on the importance of early and continued testing for Intelligent Transportation System applications, utilizing simulation environments and scaled-down testbeds. By introducing complete end-to-end testing procedures and illustrating these on state-of-the-art ITS algorithms, the relationship between different test platforms and scales are described in detail. Results from different scales, and the corresponding quality metrics are presented and compared.

A low-cost and flexible supplement to full-scale ITS testing is presented through the use of small-scale testbeds, which reduces the time and effort spent on the testing stage of ITS system design and development. This allows the researchers to implement, compare, and assess different architectures for intelligent transportation by deploying hardware-in-the-loop (HIL) simulations and tests, and it gives strong indications on the performance and high-level behavior of such systems at full scale. A range of concepts were demonstrated at The Ohio State University Control and Intelligent Transpiration Research Laboratories. Detailed implementations of applications based on an autonomous parking system, stop sign precedence system, green light speed advisory system, and a collaborative vehicle tracking system are provided.

Finally, the design, development, and implementation details for a novel testing and evaluation methodology for Lane Departure Warning and Prevention systems
are discussed. Development and testing steps from computer simulations to full-scale vehicle experiments are presented.
Dedicated to my family...
Acknowledgments

First and foremost, I would like to thank my advisor Dr. Ümit Özgüner, for providing me the opportunity to work with him the past few years. His guidance, support, and patience leads me to have better understanding on autonomous vehicles and intelligent transportation. I extend my sincere thanks to Dr. Keith Redmill, for being on my committee and for his tremendous support on various projects during my time at The Ohio State University.

I would like to thank to Dr. Arda Kurt for his support, feedback, help, and the discussions we had over the years.

I thank everyone I have worked with in the Control and Intelligent Transportation Research Laboratory, past and present; especially Dr. Engin Özatay, Michael Vernier, Emrah Adamey, Dr. Scott Biddlestone, and others with whom I have shared so much throughout my graduate studies.

I would like to acknowledge the financial support from NSF (60046665), Toyota Motor Engineering and Manufacturing NA (60035783), U.S. DoT CrIS UTC (60040605), Leidos (60047684), and Toyota InfoTechnology Center (60052565).

Finally, my deepest gratitude goes to my parents Sıdıka Özgüner and Ahmet Özbilgin, my brother Gökhan Özbilgin, my beloved nephew Arsen and niece Melisa Özbilgin, and my friends especially Rüzgar Peker, without their support I would not be here.
Vita

June 2, 1990 .......................... Born - Ankara, TURKEY

2012 ................................. B.S. Mechatronics Engineering,
                            Sabanci University
2012-present ........................ Graduate Research Assistant,
                            The Ohio State University

Publications


Fields of Study

Major Field: Electrical and Computer Engineering
Table of Contents

Abstract ................................................................. ii
Dedication ............................................................... iv
Acknowledgments ......................................................... v
Vita ........................................................................ vi
List of Tables .............................................................. xi
List of Figures .............................................................. xii

1. Introduction ............................................................ 1
   1.1 Overview .............................................................. 1
   1.2 ITS and Importance of Testing ................................. 2
   1.3 Organization of the Thesis ....................................... 4

2. Scaled Down Testing on ITS Applications ...................... 7
   2.1 Introduction and Literature Review .......................... 7
   2.2 Testbed Architecture ............................................. 9
       2.2.1 SimVille: Indoor Testbed ................................. 10
       2.2.2 Real-Size Vehicle Testing ............................... 15
   2.3 Testing Scenarios ................................................ 18
       2.3.1 Autonomous Parking ..................................... 18
       2.3.2 Stop Sign Intersection Handling ....................... 29
       2.3.3 Green Light Speed Matching ............................. 34
   2.4 Conclusion .......................................................... 38
## 3. Collaborative Vehicle Tracking on SimVille

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction and Literature Review</td>
<td>40</td>
</tr>
<tr>
<td>3.2 Overall Architecture</td>
<td>42</td>
</tr>
<tr>
<td>3.2.1 V2V Standard Information Sharing</td>
<td>42</td>
</tr>
<tr>
<td>3.2.2 V2V Collaborative Tracking</td>
<td>43</td>
</tr>
<tr>
<td>3.3 Localization and Vehicle Detection</td>
<td>44</td>
</tr>
<tr>
<td>3.3.1 Grid Mapping and Scan Matching</td>
<td>45</td>
</tr>
<tr>
<td>3.3.2 Segmentation and Vehicle Detection</td>
<td>48</td>
</tr>
<tr>
<td>3.4 Multi-Vehicle Tracking</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1 Detect-to-Track Data Fusion</td>
<td>50</td>
</tr>
<tr>
<td>3.4.2 Track-to-Track Data Fusion</td>
<td>53</td>
</tr>
<tr>
<td>3.4.3 Entropy Calculation and Track Deletion</td>
<td>55</td>
</tr>
<tr>
<td>3.5 SimVille Testing</td>
<td>56</td>
</tr>
<tr>
<td>3.5.1 Example Scenario I</td>
<td>56</td>
</tr>
<tr>
<td>3.5.2 Example Scenario II</td>
<td>59</td>
</tr>
<tr>
<td>3.5.3 Example Scenario III</td>
<td>61</td>
</tr>
<tr>
<td>3.6 Conclusion</td>
<td>64</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction and Literature Review</td>
<td>66</td>
</tr>
<tr>
<td>4.2 Test Procedure Design</td>
<td>68</td>
</tr>
<tr>
<td>4.2.1 Test Variable Determination</td>
<td>69</td>
</tr>
<tr>
<td>4.2.2 Scenario Selection</td>
<td>70</td>
</tr>
<tr>
<td>4.2.3 Prioritization and Elimination Based on Crash Data Analysis</td>
<td>71</td>
</tr>
<tr>
<td>4.2.4 Prioritization and Elimination Based on Scenario Security and Logistics</td>
<td>72</td>
</tr>
<tr>
<td>4.2.5 Prioritization Based on Joint Probabilities</td>
<td>73</td>
</tr>
<tr>
<td>4.2.6 Comparable Procedures</td>
<td>75</td>
</tr>
<tr>
<td>4.3 Simulation Investigation</td>
<td>76</td>
</tr>
<tr>
<td>4.3.1 Driving Simulator</td>
<td>77</td>
</tr>
<tr>
<td>4.3.2 Simulated World</td>
<td>77</td>
</tr>
<tr>
<td>4.3.3 Simulated ADAS</td>
<td>78</td>
</tr>
<tr>
<td>4.3.4 Simulated Scenarios</td>
<td>80</td>
</tr>
<tr>
<td>4.3.5 Simulator Test Procedure</td>
<td>80</td>
</tr>
<tr>
<td>4.3.6 Simulator Results, Analysis and Initial Metrics</td>
<td>82</td>
</tr>
<tr>
<td>4.4 Vehicle Testing</td>
<td>87</td>
</tr>
<tr>
<td>4.4.1 Vehicle Instrumentation</td>
<td>87</td>
</tr>
<tr>
<td>4.4.2 Vehicle Test Procedure</td>
<td>89</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Classification of Vehicle Types for Collaborative Vehicle Tracking.</td>
<td>41</td>
</tr>
<tr>
<td>4.1 Portion of the Joint Probability Table.</td>
<td>74</td>
</tr>
<tr>
<td>4.2 NHTSA LDW Test Matrix</td>
<td>75</td>
</tr>
<tr>
<td>4.3 ISO 17361 Standard Table Showing Straight Road Tests.</td>
<td>76</td>
</tr>
<tr>
<td>4.4 Different Scenarios on Driving Simulator</td>
<td>80</td>
</tr>
<tr>
<td>4.5 An Example Table of Maximum Exceedence Analysis</td>
<td>85</td>
</tr>
<tr>
<td>4.6 Metrics for An Example Run</td>
<td>100</td>
</tr>
<tr>
<td>A.1 Test Scenarios Table for Individual Probabilities</td>
<td>107</td>
</tr>
<tr>
<td>A.2 Test Scenarios Table for Individual Probabilities Cont’d</td>
<td>108</td>
</tr>
<tr>
<td>A.3 Test Scenarios Table for Joint Probabilities</td>
<td>109</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Indoor Testbed (SimVille) at The Ohio State University Control and Intelligent Transportation Research Laboratory.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>System Modules and Connections on SimVille</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>iRobot Create Mobile Robots without (a) and with Positioning LiDAR Sensors (b).</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Traffic Light Intersection on SimVille</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>The Ohio State University Autonomous City Transport (OSU-ACT)</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>The Ohio State University Semi-Autonomous Vehicle</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>The Ohio State University - V2I Traffic Light</td>
<td>18</td>
</tr>
<tr>
<td>2.8</td>
<td>Hierarchical Structure of the Vehicle Controller, with Discrete-State Decision Making Control, Continuous-State Speed and Steering Control and Interfaces in Between</td>
<td>19</td>
</tr>
<tr>
<td>2.9</td>
<td>Vehicle Model with Ackerman Steering [1]</td>
<td>21</td>
</tr>
<tr>
<td>2.10</td>
<td>Local Coordinates and Goal Point [1]</td>
<td>22</td>
</tr>
<tr>
<td>2.11</td>
<td>Parking Procedure [1]</td>
<td>23</td>
</tr>
<tr>
<td>2.12</td>
<td>States of Parking Procedure</td>
<td>24</td>
</tr>
<tr>
<td>2.13</td>
<td>Finding Final Path After Second Maneuver [1]</td>
<td>26</td>
</tr>
</tbody>
</table>
2.14 Demonstration of the Same Parking Algorithms (a) Full-Scaled (b) Scaled-Down Testing. ................................. 27

2.15 Desired Arc Path and Followed Trajectory ................................. 27

2.16 Polygon Casting .......................................................... 28

2.17 Emulated LiDAR Stopping Distance Comparison ........................ 29

2.18 Stop Sing Intersection Handling Flowchart ................................. 31

2.19 Demonstration of the Intersection Handling ................................. 32

2.20 Intersection Handling Vehicle Demonstration ............................... 34

2.21 On-board GUI for V2X Applications at CITR Laboratory ............... 34

2.22 Hitting the Green Light Vehicle Testing ................................. 38

3.1 Mixed-Traffic Environment. Red, Blue, and Green Vehicles Represent Fully-Equipped, V2V-equipped, and Non-Equipped Vehicles Respectively ................................................................. 42

3.2 Illustration of Overall Collaborative Vehicle Tracking ................. 44

3.3 Illustration of Overall Vehicle Detection ................................. 45

3.4 Step-by-Step Grid-Mapping Implementation on SimVille. ............... 48

3.5 Segmented LiDAR Scan, with Walls and Vehicles Identified ............. 49

3.6 Explanation of the First Scenario. The Paths of Fully-Equipped (red), V2V-Equipped (blue), and Non-Equipped (green) Vehicles. ............... 57

3.7 Tracking Results of the Fully-Equipped Vehicle for First Scenario. .... 58

3.8 Tracking Results of the V2V-Equipped Vehicle for First Scenario. .... 58

3.9 Explanation of the Second Scenario. The Paths of Fully-Equipped (red), V2V-Equipped (blue), and Non-Equipped (green) Vehicles. .... 59
3.10 Tracking Results of the Fully-Equipped Vehicle for Second Scenario. 60
3.11 Tracking Results of the V2V-Equipped Vehicle for Second Scenario. 61
3.12 Explanation of the Third Scenario. The Paths of Fully-Equipped (red), V2V-Equipped (blue), and Non-Equipped (green) Vehicles. 62
3.13 Tracking Results of the Fully-Equipped Vehicle for Third Scenario. 63
3.14 Tracking Results of the V2V-Equipped Vehicle for Third Scenario. 63
3.15 Screen Shot from Real-Time Monitoring Confidence Ellipsoids 65
4.1 Driving Simulator at CITR Laboratories 67
4.2 TRC Test Track - Skid Pad 68
4.3 Tile Development for Simulation Studies 78
4.4 Block Diagram of LDP Correction Signal 79
4.5 Simulated LDW/LDP Error Signals and the System Zone Descriptions 79
4.6 Test Starting Point with Cones 81
4.7 Marks on Steering Wheel 83
4.8 Sample Simulated Run 1: 30mph, Straight Road, Right Departure, Steep Departure, LDP-1 84
4.9 An Example Plot for Correlation Between Multiple Metrics 86
4.10 Data Collection Instruments for LDP/LDW Testings 88
4.11 Wheel Camera Snapshots for Right Departure 90
4.12 Dashboard of the Test Vehicle with the LDW system Status Display and the Synchronization LED Visible. 90
4.13 Start Gate Configuration for a Lane Departure to the Right 91
Chapter 1: Introduction

1.1 Overview

In recent years, Intelligent Transportation Systems (ITS), especially autonomous vehicles and advanced driving assistance systems (ADAS), have attracted and received great amount of interest from many research communities and the automotive industry. Innovations in these systems, over the last decades, aim to make vehicles easier to drive and control, safer, more efficient, and ultimate goal is to build autonomous vehicles that can operate without human interaction [2].

Academia has taken the lead for the research and development of autonomous driving. Defence Advance Research Project Administration (DARPA) helped to accelerate development of intelligent vehicles. DARPA Grand Challenge and Urban Challenge were the first significant steps in raising public awareness and expectations for the future cars [3, 4]. This has resulted on the introduction of various perception sensors usage in traffic situations, and state-of-the-art technologies that are advancing from simple control to the full autonomy of the vehicle.

Moreover, over past decades, vehicle manufacturers have been introducing at a rapid pace ADAS, and recently, they have started to introduce partially autonomous driving capable systems [5]. Therefore, autonomous vehicles have no longer will be in
the realm of science fiction, and driver-less cars will be on our streets in the relatively near future.

There are many established programs and working prototypes as a result of these research activities on ITS. However, there are, still, many challenges in conducting full-scale research. Hence, sufficient testing becomes an essential part of the research and development activities for ITS. The main focus of this thesis is to describe the relationships between simulator and emulator and real experiments by introducing the end-to-end testing procedures, implementing state-of-the-art algorithms, and comparing from different scales.

1.2 ITS and Importance of Testing

ITS refers to integration of various advanced communication, sensors, and computing technologies to infrastructure and vehicles in order to improve efficiency, safety, convenience and comfort of travel [6]. Intelligent vehicles are important components of ITS, which are the systems use various sensors, and intelligent decision and control algorithms to perceive nearby environment, either helping driver for in vehicle operations, e.g advance/cruise control, lane departure, forward collusion warning/prevention systems or controlling the vehicle with fully without human interaction. Taking the advantage of cooperative elements and enhancing system performance is the general philosophy for the autonomous systems. Different definitions and statements have been made and still can be made for the ITS and the automation level of the intelligent vehicles; however, in 2013, The U.S. Department of Transportation’s National Highway Traffic Safety Administration (NHTSA) announced new policies and regulations regarding to vehicle automation and research related
safety issues and automation levels are defined formally[7]. **No-Automation (Level 0):** Driver is the only authority and full responsible for controlling over steering, braking, or throttle. Vehicle can have some driver support and convenience systems, such as forward collision warning and lane departure warning systems. **Function-specific Automation (Level 1):** The vehicle can have moderate automation that helps driver for certain actions. Driver is responsible for safe operation but vehicle’s automated system can have the control for either steering or braking/throttle, such as adaptive cruise control, automatic breaking for crash avoidance and lane departure prevention systems. **Combined Function Automation (Level 2):** Contrast to level 1, in this level, the vehicle has the ability to involve control of at least two primary functions in order to assist to driver, such as combination of adaptive cruise control with lane keeping system. The driver shall expect to take control of the vehicle, and is still responsible for monitoring the system and roadway for safe operations. **Limited Self-Driving Automation (Level 3):** Vehicle can have the full control of the vehicle and driver does not need to monitor the the road constantly. However, driver shall be expected to take control, but with sufficiently in advance warning. **Full Self-Driving Automation (Level 4):** Last level of the automation, and the vehicle can accomplish all driving tasks and be monitored by itself. The driver is only required to set the destination.

Multidisciplinary research topic like ITS, require testing of multiple components working together, including low level actuator and sensors interacting with the physical world, computational decision making, higher-level group dynamics, and networked systems. As introduced above, these systems are mainly results from research activities of different areas such as electronics, control, communication and robotics.
Testing of these multifaceted systems is of paramount importance. In ITS, the testing generally includes preliminary testing in simulations and further tests in physical environment. However, conducting full-scale ITS tests presents numerous challenges, such as outdoor testing having a high cost in terms of logistics and scheduling. Also testing in full-scale environments is not always readily accessible for all and it can be dangerous when multiple parties and traffic are involved. Therefore, above concerns of prototyping these systems in real traffic have led us to use pure simulation in the research community [8]. Simulations are the most important stage for the testing and development. With increased computational power, new generation simulators support a subset of ITS technologies. In this thesis, importance of the simulation is presented with different examples of ITS applications.

On other hand, problems related to simulations include a lack of appropriate measurement and possible low-fidelity representation of real world dynamics such as communication, sensor interactions and vehicle dynamics. Therefore, this study also presents small scaled, low-cost, flexible support to outdoor testing without compromising safety and feasibility [9]. This scaled down testing platform enables us to implement, compare, and assess different architectures for intelligent transportation by deploying hardware-in-the-loop (HIL) simulation and testing.

1.3 Organization of the Thesis

This thesis presents relationship of simulator and emulator and real experiments which are implemented in the Control and Intelligent Transportation Research Laboratory (CITR) at The Ohio State University. Testing phases and implementation of some ITS applications are described. The thesis is organized as follows:
Chapter 1 begins by giving overview about the recent research and intelligent transportation systems and autonomous vehicles. It is followed by describing the importance of the simulation and testing, which explains the motivation and the main focus on this thesis.

Chapter 2 starts with literature review on testing and small-scale testbeds for intelligent transportation systems. Then, the multi-agent, small-scale testbed that is called SimVille, at the CITR Laboratory is described. It is followed by the explanation and implementation of some test scenarios, such as automated parking, stop sign precedence, green light speed advisory on SimVille and real vehicles with results and comparisons on different scales.

Chapter 3 begins by giving theoretical background and details of sensor fusion especially in collaborative vehicle tracking. The approach and its implementation on SimVille is presented for mixed-traffic environments. The solutions to practical problems come with real-time implementation, such as segmentation for vehicle detection, grid mapping and scan matching for localization, inter-vehicular communication for track-list sharing on scaled-down platform, are also presented in this chapter.

Chapter 4 describes the series of research activities conducted as part of the Lane Departure Warning and Prevention (LDW/LDP) systems testing and evaluation research project between Toyota Technical Center (TTC) and CITR Laboratory and Center for Automotive Research (CAR). It starts with overview of the LDW and LDP systems and testing procedures. Then details of test variable determination, scenario selection, test procedure design with the driving simulator at CITR Laboratory are given. Finally, vehicle instrumentation, vehicle testing, and analysis and evaluation of results are presented.
Chapter 5 summarizes the work, results and importance of the testing, concludes this thesis, and discussed the possible future work may be required.
Chapter 2: Scaled Down Testing on ITS Applications

2.1 Introduction and Literature Review

Experimenting ITS tests, especially with multiple vehicles, on full-scale can presents different challenges and brings multiple concerns, such as costs in terms of money and time, safety. Researchers have attempted to use pure simulation [10, 11, 12]. Problems related to simulations include a lack of appropriate measurement and possible low-fidelity representation of real world dynamics such as communication, sensor interactions and vehicle dynamics.

To test and evaluate the appropriate system interactions, Woo and Lee [13] described a proving ground facility and an evaluation system to verify the performance of ITS applications. Although a full-scale ITS testbed is crucial for extensive research, it generally has to be limited due to the fact that testing with full-scale vehicles can be dangerous and costly as mentioned above. Therefore, small-scaled platforms are proposed to bridge the gap between pure computer simulations and real-size vehicles as in [14, 15]. In [16], in order to investigate and develop unmanned vehicle technologies for multi-vehicle platforms, testbed called Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) is introduced. Moreover, with similar purpose, multi-vehicle indoor tesbeds are presented in [17, 18] for different ITS applications as attractive
supplements for consistent and repeatable testing. However, those are mostly built to study higher-level decision making and situations assessments, which lack any kind of low-level actuator and sensor interaction evaluation.

One of the unique aspects of our indoor testbed at The Ohio State University CITR Laboratory, as seen in Figure 2.1, is having the high-level decision making, and low-level system dynamics and control processes decouple [19]. The decoupling allows for the high-level code to run unchanged in the simulated robots, the real robots and full size outdoor vehicles.

The complete end-to-end testing environment includes simulated robots, the real robots and the full-size outdoor vehicles, which is designed for a complete ITS development cycle including the model verification, software verification, scalability, sensing and actuation. This architecture allows for the high-level code to run unchanged in the all different environments. The simulated robots and indoor testbed are the first step towards testing at full-scale. However, to achieve same real-time performances and reliability on the outdoor vehicles and scaled testing, these different scales are also compared at sensor and actuator levels.

High level situation making and situation assessments have been illustrated in earlier studies [20]. In this chapter, it is illustrated that the high-level behaviour of the small-scale testing is observed in full-scale testing. Along with the same high-level behaviour of two scales, this study provides guidelines to compare performances in different platforms based on sensor/actuator interactions, and low level control. This has shown the direct relation between the dynamics of two scales in ITS testing scenarios. The approach decreases amount of testing requires on the full-scale
dramatically. The performance of the scaled down testing is illustrated using a specific example based on autonomous parking developed in . Moreover, in this chapter, simple stop sign precedence and green light speed advisory applications are presented.

2.2 Testbed Architecture

This section gives overall working principles and the layout of the system architecture for a scaled-down testbed, descriptions on how to scale down the measurements,
and guidelines on how one matches the components of the full-scale outdoor testing with the components of the indoor testing. The complete architecture gives the idea of how the testing on scaled environments present indicators on the full-scale experiment performance.

2.2.1 SimVille: Indoor Testbed

The progression in ITS development according to the proposed approach starts with computer software simulation that can emulate physical sensors, and contains code to run on full-scale testing. This simulation is deployed with vehicle mathematical model and shorten the development time of the algorithms significantly. The first actual physical tests involving mobile agents are performed on the SimVille. The generic indoor testbed defined in the proposed system consists of several discrete modules in the software and hardware domains, as illustrated in Figure 2.2.

In the example of SimVille, a number of mobile agents are deployed in parallel. iRobot Creates are used as vehicles in the laboratory scale as seen in Figure 2.3. The Gumstix Overo Fire computers are deployed for the control of the robots and connected to Creates Open Interface [21]. This setup also provides 802.11b wireless modules to establish direct communication links with the other agents, infrastructures and the central computer for monitoring. The control algorithm that is running on Gumstixs uses the Player [22] server that run on this module to encapsulate the specifics of the serial control interface and provide access to the platforms sensors and actuators which can communicate with Creates motors, and through additional USB connections with add-on devices like web-cams and LiDAR.
Since actual GPS does not work in indoor, each mobile agent has a visual tag that can be identified via a stationary camera system. This, as called Virtual GPS system, uses image processing techniques with ARTKPlus Toolkit [23] to generate the real-time position, orientation and velocity of each agents. This example of indoor localization can be replaced with various means of achieving the same outcome, yet the end result of dependable localization needs to be replicated in any indoor testbed, as most full-scale ITS applications make use of GPS localization. The data from Virtual GPS is fused with data from wheel encoders to create position, orientation and
velocity estimates. In order to estimate own state, robots fuse the information from the positioning system and their wheel odometry through a Kalman filter. This localization technique, used in SimVille, gives considerably high accuracy for position and orientation of the vehicles (error < 1 cm). Then the data is transmitted over Unicast UDP packets to IP address associated each tag ID, via wireless 802.11b interfaces at ~ 10Hz.

Broadcasting each agents own pose estimates through vehicle-to-vehicle (V2V) communications through virtual GPS and V2V messaging is analogous to real vehicles equipped with GPS/IMU dead-reckoning systems and the Dedicated Short Range Communications (DSRC) radios.
As important part of urban test scenarios, SimVille has its own traffic light as shown in Figure 2.4. The traffic light transmits the information through a vehicle-to-infrastructure (V2I) interface which is based on the Signal Phase and Timing messages defined in [24]. This message is broadcast and it contains information about the position of the traffic light, stop lines for the lanes that intersects, and light status.

As for the physical dimensions and the scale of the testbed, the mobile robots for this particular indoor testbed are chosen roughly 33-centimeters wide and the testing area is constructed to be 1/7 scale with 0.45 meter wide lanes, which corresponds to scaled down version of 3.15 meter lane size in full-scale. As seen in Figure 2.1, road network design focused on urban scenarios. This environment allow us to create and test different scenarios. For example, several intersection are exist with and without traffic lights where stop sign precedence algorithm can be tested.
as buildings and an overpass are used to test GPS dropout situations and evaluate the dead reckoning performance of the system. Parking spaces are used for testing different parking algorithms. Flexible load lanes can be used to change direction of travel, simulate a highway scenarios, such as lane change and merging maneuvers and convoying applications.

The robots in the testbed have differential drive mechanisms, which do not match the kinematics of the full-scale cars, however, limitations are added to the software that is executed on the mobile agents. For example, in the parking zones, linear speed of the agents assumed to be constant and steering rate is bounded to have minimum turning circle with diameter of 1.5 meters (which is 1/7 scale of the full-size vehicles turning circle diameter). These limitations are implemented via the Player interface, allowing us to conduct experiment with limited turning radii as in the full-size vehicles.

In order to give LiDAR functionality for sensing environment such as obstacles, in scaled down testing, some robots are equipped with Hokuyo URG-04LX-UG01 laser range finder, as seen in Figure 2.3a. The laser range finder has 0.36 degrees in angular and 1 mm in distance resolution. However, even for small-scale testing, it could be too expensive to install every sensors to robots.

Various sensor modalities can also be replicated on the robots without physical sensors, through software emulation and virtualization. In order to emulate the LiDAR data, preconfigured static models and Virtual GPS can be used to pass real-time data to Stage [25]. Then robot mounted LiDAR information can be simulated and passed to Player server for transmission to robots [26].
2.2.2 Real-Size Vehicle Testing

The OSU CITR Laboratory has been involved in the research and implementation of various types of automated vehicle since at least 1995. Successful development and implementation of control and sensing systems for autonomous lateral and longitudinal maneuvers and functions have been demonstrated. In this chapter, fully automated and semi-autonomous vehicles that are presented.

Figure 2.5: The Ohio State University Autonomous City Transport (OSU-ACT)
The OSU participation for DARPA Urban Challenge 2007, Autonomous City Transport (OSU-ACT) is a full-size automated 2006 Toyota Highlander Hybrid SUV developed by a team of faculty, staff and students at OSU in order to solve the autonomous urban driving scenarios, as shown in Figure 2.5.

The vehicle equipped with Novatel Propak-LB-L1L2 GPS using Omnistar HP differential correction service for localization. This data is also augmented with measurements from gyroscopes, accelerometers, independent wheel speeds, transmission, and steering wheel angles. High accuracy of position and orientation estimates (error <10 cm) is provided through Kalman filters.

OSU-ACT has been equipped with a number of different scanning laser sensors such as Ibeo Alasca XT and Sick LMS 221-30206. Four vertical scanning planes, an angular resolution 0.25 degrees and distance resolution of 1 cm (values up to 100 m) are provided by Ibeo Alasca XT. The other sensors, including MaCom Short Range Sensor and a Mobileeye, have relatively low resolutions, so it can expected to have valid comparison between sensors that are used in indoor and outdoor environments.

Yet another vehicle at OSU CITR Laboratory, is designed for partial-autonomy demonstrations which reflects near-future technologies that can be implemented as Advanced Driver Assistance Systems (ADAS), befitting from Dedicated Short Range Communication (DSRC) systems. 2008 Honda Accord, as shown in Figure 2.6 is modified to control speed via throttle and break automation.

Similar to indoor traffic light on SimVille Figure2.4, we have designed a traffic light unit for full-scale testing as shown in Figure 2.7. This traffic light can control light state and timing and broadcast SPaT messages with DSRC Wireless Safety Unit (WSU) and antenna. A GUI has been designed to control traffic light for different
Figure 2.6: The Ohio State University Semi-Autonomous Vehicle

types of demonstrations. For example, vehicle can change it from regular traffic light to stop sign intersection. Details of the experiments and scenarios are presented in next sections.

Another advantage of the proposed testing architecture is that the decision making and control processes, implemented as a Hybrid-State System (HSS) controller [27], can be easily developed to accomplish control of the both scaled-down and full-size robots. HSS structure separates Finite State Machine (FSM) decision making process (High-level control) and PID-based continuous controller (Low-level control),
as illustrated in Figure 2.8. This enables testing to evaluate high-level algorithm performance and limits the time spent on debugging at full-size experiment level. Moreover, different algorithms and control strategies with various hardware selections can be compared by deploying different HSS controllers.

2.3 Testing Scenarios

2.3.1 Autonomous Parking

The parking problems are recognized as one of the most difficult autonomous driving function involving different tasks such as path generation, localization, control of the vehicle and obstacle avoidance. Therefore this particular example is selected as
Figure 2.8: Hierarchical Structure of the Vehicle Controller, with Discrete-State Decision Making Control, Continuous-State Speed and Steering Control and Interfaces in Between

test scenario to validate robustness of the proposed scaled-down testing methodology.

In [28] and [29] several algorithms and methods are proposed for the both parallel and perpendicular parking.
Example autonomous parking scenarios were demonstrated in 2007 DARPA Urban Challenge [20], where the parking lots are defined as the “zones”. In this section, the arc path parking algorithm that has been developed in [1] is used. The algorithm is responsible to generate a circular arc path that connects the car position to the goal point. The autonomous vehicle is tasked to follow this path to park; however, due to possible angle differences between the vehicle yaw angle and the path angle, the vehicle may not be able to follow this path initially. In this case, the vehicle moves forward or backward with maximum steering angle (maximum yaw change) to generate a new path in real-time. When final path is generated, the vehicle follows the final path. If an obstacle is detected during this task, again a new path is generated and required maneuvers are taken. The details of the algorithm is described below.

Car Model

In order to generate a parking path that can be followed by the vehicle, the kinematic model is used. Nonholonomic, car-like kinematic model is shown in Figure 2.9

Ackerman steering theory and kinematic model of the car are used to obtain following equations:

\[ \theta_{\text{steer}} = \frac{\theta_l + \theta_r}{2}, \quad R = \frac{L_{fr}}{\tan\theta_{\text{steer}}} \quad (2.1) \]

where \( \theta_l \) and \( \theta_r \) are the turning angle of right and left front wheel respectively. \( L_{fr} \) is the distance from the front bumper to the rear axle, \( \theta_{\text{steer}} \) is the average steering angle and \( R \) is the vehicle turning radius. The car yaw angle is described as the velocity...
direction of the rear axle and the vehicle position is also defined to be at the center point of the rear axle.

**The Arc Path Generation**

Finding a circular arc for parking path is reasonable due to its simplicity of calculations and the non-holonomic constraints of the vehicle. It also has significant advantages for following the path. In order to generate the arc path, a number of constraints such as minimum and maximum turning radius are taken into account and new local coordinates are defined in Equation 2.2 and 2.3, as illustrated in Figure 2.10.
Figure 2.10: Local Coordinates and Goal Point [1]

![Diagram of local coordinates and goal point](image)

\[
R_{\text{min}} = R_{\text{turn}} - \frac{W_c}{2}
\]

\[
R_{\text{max}} = \sqrt{\left( R_{\text{turn}} + \frac{W_c}{2} \right)^2 + L_c^2} + \frac{W_c}{2}
\]

\[
X_0 = \frac{W_p}{2}
\]

\[
Y_0 = \begin{cases} 
\sqrt{R_{\text{max}}^2 - \left( R_{\text{min}} + \left( \frac{W_c}{2} \right)^2 \right)} t^2, & \text{if } R_{\text{max}}^2 - \left( R_{\text{min}} + \left( \frac{W_c}{2} \right)^2 \right) \geq 0 \\
-\sqrt{\left( \frac{W_c W_p}{2} \right) - \left( \frac{W_c}{4} \right)^2}, & \text{otherwise}
\end{cases}
\]

\[ (2.2) \]

\[ (2.3) \]

\[ Y_0 = \begin{cases} 
\sqrt{R_{\text{max}}^2 - \left( R_{\text{min}} + \left( \frac{W_c}{2} \right)^2 \right)} t^2, & \text{if } R_{\text{max}}^2 - \left( R_{\text{min}} + \left( \frac{W_c}{2} \right)^2 \right) \geq 0 \\
-\sqrt{\left( \frac{W_c W_p}{2} \right) - \left( \frac{W_c}{4} \right)^2}, & \text{otherwise}
\end{cases}
\]

\[ W_c \text{ and } W_p \text{ are car and parking spot width respectively. } L_c \text{ denotes the length of the car, } R_{\text{min}} \text{ and } R_{\text{max}} \text{ are minimum and maximum radius of the vehicle body and } R_{\text{turn}} \text{ is the vehicle minimum turning radius.} \]

22
The arc path is generated to connect the current vehicle reference point to the goal point. However, due to the difference between the tangent angle of the path and the initial yaw angle of the car, path may not always be followed at first, as shown in Figure 2.11.

![Diagram of parking procedure](image)

**Figure 2.11: Parking Procedure [1]**

The Parking Procedure

The procedure starts with the vehicle entering the zone, where the parking spots are, and navigate to empty parking spot. Next, the vehicle starts parking with the necessary maneuvers. This parking procedure is divided into different sub-states and these state transitions can be illustrated in Figure 2.12.
In order to find final parking path vehicle drives forward or backward with maximum steering angle. Vehicle steering direction ($\theta_{\text{steer}}$ direction) is obtained from Equation 2.4. During this search, the circular arc path is updated in real time until the car’s yaw angle equals to the tangent angle of the circle. If found circular arc path’s radius is greater than minimum turning radius of the car, path can be followed directly.
\[ R = \left| \frac{(x^2 + y^2)}{2x} \right| \]

\[ \theta_{FinalPath} = -\sin^{-1}(y/R) + \frac{\pi}{2} \]

\[ \theta_{FinalPath} - \theta_{yaw} = \theta_r \quad (2.4) \]

\[ \theta_{steer} = \begin{cases} 
  \text{left}, & \text{if } \theta_{FinalPath} \geq \theta_{yaw} \\
  \text{right}, & \text{otherwise}
\end{cases} \]

However, if the circular arc path’s radius is less than the minimum turning radius, the car drives with maximum steering angle until it reaches the boundary defined in the local coordinates. A new path is introduced during this stage and car can follow this path with a second maneuver as shown in Figure 2.13. The final path is found when the tangent angle of the path and the yaw angle of the vehicle are equal. \( \theta_{FinalPath} = \theta_{yaw} \). The details and proof of the existence of the parking path can be found in the Appendix of earlier work [1].

**Test Comparison and Evaluation**

Algorithm that is developed for DARPA Urban Challenge and implemented on vehicle OSU-ACT’s results are used. In order to compare results from different scales, high-level performance quality metrics such as the number of maneuvers for exact same parking scenarios, are measured.

The performance of the scaled down testing is illustrated in Figure 2.14. It has been shown that, full-scale and small-scale vehicles has same corresponding maneuvers in both environments.

The indoor localization and the control performance in terms of trajectory following is one of the important factors for verification of the testing. The arc path
following control of the scaled robots with steering rate input can be difficult compared to the full-size vehicle, where steering angle is used as one of the control inputs. In both cases, squared-integral-derivate control laws are applied to follow circular arc path for parking. Tracking performance of the controller for arc parking path is shown in Figure 2.15.

The localization and path following performances that have been achieved in the testbed creates a baseline for the scaled-up settings. The implemented system architecture provides flexibility to incorporate the additional data by software. For instance, the GPS, odometry and V2V communication data might be integrated with noise to have same performance on different sensors and scales. Same approach has been applied to emulate various sensors. These disturbances help predict the algorithms performances in the outdoor vehicles.
Figure 2.14: Demonstration of the Same Parking Algorithms (a) Full-Scaled (b) Scaled-Down Testing.

Figure 2.15: Desired Arc Path and Followed Trajectory
In order to assess the reliability and verify the sufficiency of the emulated sensors, feasibility tests have been performed. In these tests, position data for the obstacles are transmitted through V2I communication to the mobile agents. A polygon is cast in front of the agents, so that the obstacles can be detected without using actual LIDAR. Illustration of the obstacles and polygon casting can be seen in Figure 2.16. Hokuyo laser scanner is also installed as reference to evaluate robustness of the emulated sensor with the polygon casting by measuring the closest distance to the moving obstacles. 2 cm safety distance is set for obstacle avoidance and closest distances are measured. The results from 20 different experiments, as given in Figure 2.17, shows the reliability of the emulated sensors.

Figure 2.16: Polygon Casting
2.3.2 Stop Sign Intersection Handling

Complexity of the driving environment makes intersection handling and collision avoidance difficult problem in ITS. Detection of the vehicles and their movements accurately at the intersection was one of the hardest tasks in DARPA Urban Challenge.

Several researchers proposed centralized system by utilizing the V2I communication in which vehicles are supposed to talk intersection manager and reservations are made accordingly [30]. Since the integration of centralized management is not very practical for each intersections, V2V communication based solutions and new protocols are proposed in [31].

An all-way stop intersection system is implemented in different scales at CITR Laboratories. The vehicle approaching to an intersection will identify the intersection from Road Network Definition File (RNDF) [32]. Utilizing the V2V communication
systems with DSRC, vehicles are able to share knowledge regarding to stop sign precedence.

**Algorithm Overview**

The overall flowchart is depicted in Figure 2.18. While vehicles check points to be traversed, they look for intersection from the road network structure. If vehicle is within some distance of stop sign intersection, it checks whether this intersection is to be traversed by detecting the right target points for the intersection using the Algorithm 1.

After stop sign intersection is determined, priorities can be assigned for precedence. In contrast to determining priorities right after stop, this structure is added to handle different types of intersections by assigning initial priorities before stop. After vehicle comes to stop with precise stopping, it updates priority list by communication with other vehicles, and decide to take right action depends on its precedence. With this integration 2 way stops or vehicles with higher priorities, such as fire truck, ambulance can also be handled. On the other hand, this allows us to detect vehicles at uncontrolled intersection even if the other vehicles are not visible on approach, which enhance safety and can avoid possible crashes by actively (breaking automatically) or passively (warning driver).

**Demonstrations on Different Platforms**

After the simulation on Stage, the intersection handling procedure is firstly implemented on SimVille, as seen in Figure 2.19. Checkpoints of the vehicles are given such that they come from different direction to traverse all-way stop sign intersection. In this specific scenario, Robot 9 (red) arrived to intersection before Robot 10, and
it updates its priority list. Then, it obeys to regular stop sign and waited at the stop sign intersection. However, Robot 10 waited until Robot 9 clears to intersection to go through intersection.

The same algorithm is used to demonstrate an intersection access ADAS, longitudinally controlling a vehicle in order to, again, automatically obey stop-sign precedence rules and handle intersections. The semi-automated test vehicle used in this
Algorithm 1 Stop Line Detection

Require: $dist_{center} < d_{threshold}$ meters

for $validstop_{index} = 0$ to $validstop_{index} < allstoplines_{validstops}$ do

$dist_{center} = \sqrt{(x - c_x)^2 + (y - c_y)^2}$

$stoppoint = allstoplines_{validstop}$

$dist_{stopcenter} = \sqrt{(stoppoint_x - c_x)^2 + (stoppoint_y - c_y)^2}$

$dist = \sqrt{(x - stoppoint_x)^2 + (y - stoppoint_y)^2}$

if $dist < mindist$ AND $dist_{center} > dist_{stopcenter}$ AND $dist_{center} > dist$ then

$stoppoint = alltrafficlight_{validstop}$

$mindist = dist$

end if

if $dist > maxdist$ then

$maxstoppoint = allstoplines_{validstop}$

$maxdist = dist$

end if

end for

(a) Stop Sign Precedence at Stage

(b) Stop Sign Precedence on SimVille

Figure 2.19: Demonstration of the Intersection Handling
work is modified Honda Accord (Figure 2.6). Modifications include control computer running a Linux operating; Novatel GPS receiver with Real Time Kinematics (RTK) correction for absolute positioning; Denso WSU capable of V2V communication; connection to the Control Area Network (CAN) to control throttle and read vehicle sensors, such as velocity, acceleration, break pedal pressure; linear actuator to control brake actuation. The V2V communication and control GUI (Figure 2.21) is carried out by Denso WSUs. Modified Basic Safety Message (BSM) structure is broadcast at 5.9 GHz and transmitted by vehicle with unique IDs to be received by any listener in range.

As seen from picture of real demonstration Figure 2.20. Semi-automated vehicle, deployed with human steering and computer-controller speed automation, stops at the stop sign. The fully manual driven vehicle (with WSU) is already at the other stop line is detected by V2V communication, and semi-automated vehicle checks for higher precedence vehicle locations over the message sets. Once semi-automated vehicle detects that other vehicle has the precedence it waits until the other vehicle clears the intersection.

On-board GUI is developed to show the controlled vehicle speed, distance to the intersection and most importantly current precedence situation. GUI is designed to monitor entire system and communication between vehicles and infrastructure. From the connected vehicle technologies perspective, GUI also present a warning system that informs drivers status of the intersection which can prevents collusion especially when vehicles are not visible on approach.
2.3.3 Green Light Speed Matching

Over the last years, energy efficient driving attracts governments and researchers, since reducing fuel consumption and related gas emission is one of the critical benefits from ITS technologies [33].
The main causes of energy consumption is given as the traffic congestion and idling time at signalized intersection. It has been proved that information of the traffic lights signal status has important effect on energy efficiency in urban traffic. More specifically, green light speed matching reduces the fuel consumption and average idling time [34].

Discriminating the hard accelerations, caused by sudden stops especially at the intersections, could increase the energy efficiency. In order to show effects of minimizing the acceleration while driving through signalized intersections, simulation studies [35] have been presented to demonstrate reduction of fuel consumption, emission and travel time.

In this section, implementation of speed generation for catching a green light at signalized intersection is presented. Utilization of the V2I communication and SPaT messages are depicted for different scales at CITR Laboratories.

**Algorithm Overview**

Algorithm start with finding closest traffic light stop point by fetching the information from SPaT messages. In order to do so, we have used similar algorithm that is presented above Algorithm 1. Detection of different phases of lights is important to fetch complete cycle timings. Then phase cycles can be calculated by simply adding timer counts depending on the current traffic light status.

First goal is to find out the green phase of the traffic light in an intersection, and to drive the vehicle through intersection within the green phase. In order to do so, an advisory speed needs to be generated passing through the intersection. In this demonstration, advisory speed is calculated based on the distance between the vehicle
to the intersection, and the traffic light change logic. A constant mean speed is then computed based on the distance and time using the following equation:

\[
\begin{align*}
\min v_a &= \frac{d}{T_w} \\
\max v_a &= \frac{d}{T_g}
\end{align*}
\]

where

\(T_w\) : time before traffic light turn into green
\(T_g\) : time to the end of the following green phase

with \(T_w\) and \(T_g\), we could get the maximum and minimum possible speed guiding the vehicle. This means if the controller can perform speed control within the advisory speed range, it can go through the intersection within the green traffic light phase. However, there are some cases that the vehicle is far away from the intersection and \(T_g\) is not long enough to let the vehicle go through the intersection with an available speed, namely

\[
max v_a > v_{max}
\]

where \(v_{max}\) is the largest speed with which the set vehicle speed. In order to make the maximum advisory speed within the available speed range, next cycles of green phase of the traffic light is also considered in modulo.

\[
\max v_a = \begin{cases} 
\frac{d}{T_w} & \text{max } v_a \leq v_{max} \\
\frac{d}{K*(T_w+T_c)} & \text{otherwise}
\end{cases}
\]

where \(T_c\) is the whole traffic light circle time.

\[
T_c = 2(T_{green} + T_{yellow} + T_{allred})
\]
The correlated $\text{min} v_a$ will then be updated using the same time value.

$$\text{min} v_a = \begin{cases} \frac{d}{T_g} & \text{max} v_a \leq v_{\text{max}} \\ \frac{d}{K_s(T_g + T_c)} & \text{otherwise} \end{cases}$$ (2.8)

**Demonstrations on Different Platforms**

Preliminary simulation tests and scaled down testing has been done on SimVille, to test the algorithms that have been develop to detect traffic light, stop light and to control speed of the vehicle to hit green light phase.

After reasonable simulation results, the same algorithm is used to demonstrate applicability on real-size vehicles. (Figure 2.22). In this implementation, semi-automated vehicle start $\sim$500ft north of the intersection and heading south through the signalized intersection. SPaT messages are generated and broadcast from the WSU on the traffic light (Figure 2.7). The vehicle receives this information and calculates the required speed to reach the intersection when the light is green, using the algorithm above. The calculated speed command is passed to the vehicle control computer if the vehicle is not able to catch green phases it stops precisely by stop light.

Speed advisory GUI (Figure 2.21b) is switched to V2I mode, and informs the driver about the distance to the traffic light stop line with current status of the traffic light is shown. The maximum and minimum advisory speed that is described in the algorithm section is visualized with red needles on the speedometer which cab be helpful to lead driver (without automation) to hit green light phase.
2.4 Conclusion

In this section, a small-scale, low-cost and flexible supplement to the intelligent transportation system test procedure is discussed. It is proposed that not only the higher-level behavior can be tested at these physical, smaller-scale testbeds, but also certain insights into dynamic performance of full-scale counterparts can be gained through careful investigation and analysis of individual subsystem behavior.

The results shows that, scaled-down testing platforms can be used to implement and assess different architectures for intelligent transportation. The HIL simulation and tests were used to measure performance of the each components as well as overall system, and it has been shown that algorithms can be applied interchangeably on different scales.
The concepts were demonstrated on intelligent transportation system testbed SimVille at The Ohio State University Control and Intelligent Transportation Research Laboratory, with a specific example based on an autonomous parking, stop sign precedence and green light speed advisory applications. The results were validated and tested in comparison with full scale tests performed with an autonomous and semi-autonomous vehicles.
Chapter 3: Collaborative Vehicle Tracking on SimVille

3.1 Introduction and Literature Review

In intelligent vehicles research, vehicle tracking one of the significant problems among the basic components of any situation awareness technology. In mixed-traffic environments, namely vehicles with different degrees of sensing and communication capabilities, the vehicle-tracking is particular demanding area of research. In this chapter, a collaborative vehicle tracking approach that is developed in [36] is presented with modifications, where on board sensing and inter-vehicular communication resources are utilized in an efficient manner to provide track list to all participating vehicles in a mixed-traffic environment. Most importantly, the implementation of the approach on SimVille with practical solutions are described. The performance of the approach is evaluated and presented using entropy values of vehicle tracks.

The state of art methods for solving the vehicle-tracking problem can roughly be categorized as on-board sensor-based [3, 37] V2V-based [38, 39], and hybrid [40, 41] solutions. The joint work presented here is of the latter type, building upon Adamey’s work [36], directed towards mixed traffic environments where vehicles with different sensing and communication capabilities.
The participating vehicles are classified according to their capabilities, as in Table 3.1: fully-equipped, V2V-equipped, and non-equipped vehicles. According to this taxonomy, fully-equipped vehicles have both on-board sensing and V2V communication capabilities; V2V-equipped vehicles have V2V communication capabilities alone; and non-equipped vehicles do not possess communication capabilities. The problem is to provide accurate and reliable vehicle tracking results to each participating vehicle, which can be done most effectively through wireless collaboration of fully-equipped vehicles with V2V-equipped vehicles. A particular challenge that arises in mixed-traffic situations is making V2V-equipped vehicles privy to the state information of the nearby, situation-awareness-wise relevant, non-equipped vehicles.

The idea of mixed traffic environment is depicted in Figure 3.1. Implementation of the collaborative vehicle tracking revised according to DSRC and Wireless Access in Vehicular Environments (WAVE) standards and real-time requirements such as asynchronous data sources, communication delays, and bandwidth limitations. Additional data entries need to be exchanged, are appended to DSRC basic safety messages (BSMs). A multi-sensor data fusion algorithm based on Kalman filtering
and covariance intersection [42] methods are used. As explained in previous chapter, SimVille is used to implement proposed approach under multiple traffic scenarios. In this chapter, results are also demonstrated for real-time applicability and effectiveness of the approach.

Figure 3.1: Mixed-Traffic Environment. Red, Blue, and Green Vehicles Represent Fully-Equipped, V2V-equipped, and Non-Equipped Vehicles Respectively

3.2 Overall Architecture

3.2.1 V2V Standard Information Sharing

In the collaborative vehicle tracking, focus is on two sources of information: V2V communication and on-board sensing (e.g. LiDAR/vision). In V2V communication, BSM messages are exchange between fully-equipped and V2V-equipped vehicles to broadcast their state information (e.g. unique identifier, position, orientation, and
velocity of the ego vehicles). Using BSMs, therefore, vehicle tracking can be carried out simply by listening to the inter-vehicular network and collecting the BSMs in a track list, in vehicular environments where all vehicles are equipped with DSRC radios. However, in mixed-traffic environments, which may contain non-equipped vehicles that do not participate in the inter-vehicular network, the problem becomes more challenging.

3.2.2 V2V Collaborative Tracking

In case of fully-equipped vehicles, tracking non-equipped vehicles is a matter of detecting them using on-board sensing and then fusing such vehicle detections into vehicle tracks. In case of V2V-equipped vehicles, however, there needs to be an information sharing mechanism to ensure that they have access to information regarding non-equipped vehicles. (Recall that V2V-equipped vehicles do not possess the sensing capabilities to detect non-equipped vehicles, and that non-equipped vehicles do not have the V2V communication capabilities to broadcast BSMs.) However, fully-equipped vehicles can broadcast the vehicle tracks corresponding to the non-equipped vehicles in their track lists in a form similar to BSM. In the paper [41] we named data packets: quasi basic safety messages, or shortly QBSM. Upon receiving these QBSMs, each V2V-equipped vehicle, then, by fusing them into local tracks, can also keep track of non-equipped vehicles.

The overall architecture of the collaborative vehicle tracking approach is depicted in Figure 3.2. The main components are shown as: scan-matching and vehicle detection, detect-to-track data fusion, and track-to-track data fusion, which are explained in following sections.
3.3 Localization and Vehicle Detection

Localization of the ego vehicle and detecting non-equipped vehicles by fully-equipped vehicles is an important phase for the collaborative tracking. As explained in previous chapter, vehicles on SimVille have intermittent access to GPS measurements; however even if there is small errors in ego pose estimate it result in large errors in vehicle detections, during their transformation from ego-relative to global coordinates especially for ray casting. In order to eliminate effect of the erroneous localization scan matching is implemented before detection of the vehicles. In order to do so, static likelihood map is generated via grid-mapping approach. It provides satisfactory results; and therefore the implementation of a SLAM algorithm is not needed. Figure 3.3 illustrates the overall scheme of the localization improvement (scan matching) and vehicle detection approach. Details of the each part is given in this section.
3.3.1 Grid Mapping and Scan Matching

Before detecting vehicles using on-board sensors using LiDAR scans, agents have to correct localization errors for correct detection results. Therefore, scan matching is a good approach for the pose correction. The rationale behind scan matching is that small errors in ego pose can be corrected by comparing previous scans of the environment.

There are several ways to implement scan matching, two most commons are optimization over maps and set of point clouds, namely Iterative Closest Points (ICP) [43]. In order to reduce computational complexity, likelihood field model is implemented on SimVille. Since this algorithm does not consider the unexplored areas and it requires to likelihood field, grid map of the (static) environment is obtained.
Grid-mapping algorithms often rely on sensor data coming from LiDAR, RADAR, vision, or any combination of them. In most of its implementations, as in our scaled-down setup, the primary choice is LiDAR data. LiDAR scan, consists of an array of values each one corresponding to a distance measurement obtained by a specific beam.

The probabilistic way of formulating grid mapping with the Bayesian approach helps us to implement a grid map in a recursive way. Assuming that occupancy states of individual grid cells are independent, it is also called inverse sensor model [44], the objective of the mapping algorithm is to estimate the posterior probability of occupancy:

\[
P(m|1:t, z_{1:t}) = \sum_c (m_c|x_{1:t}, z_{1:t})
\]

for each cell \( m_c \) of the grid, given observations and corresponding poses \( z \) and \( x \).

The total grid map (likelihood field in this case) can be estimated by estimating the states of the individual grid cells using the assumption of mutual independence of cell occupancies:

\[
P(m_c|x_{1:t}, z_{1:t}) = \frac{P(z_t|x_{1:t}, z_{1:t-1}, m_c)P(m_c|x_{1:t}, z_{1:t-1})}{P(z_t|x_{1:t}, z_{1:t-1})}
\]

using independence assumption from the current pose and previous measurements and applying Bayes theorem:

\[
P(z_t|x_{1:t}, z_{1:t-1}, m_c) = P(z_t|x_t, m_c)
P(m_c|x_{1:t}, z_{1:t}) = \frac{P(z_t|x_t, m_c)P(z_t|x_t)P(m_c|x_{1:t}, z_{1:t-1})}{P(m_c)P(z_t|x_{1:t}, z_{1:t-1})}
\]
Here, occupancy probabilities get values within the range of (0,1) where 0 corresponds to being certain that the cell is not occupied, 1 corresponds to being certain that the cell is occupied, and 0.5 corresponds to the unknown initial state (with 50% probability of being occupied.) For numerical stability and easiness of the recursive implantation, relative probability \(odds = \frac{P(m_c)}{P(\bar{m}_c)} = \frac{P(m_c)}{1-P(m_c)}\) of the cells and its logarithm is used \(\log(odds)\):

\[
\text{odds}(m_c|x_{1:t}, z_{1:t}) = \frac{\text{odds}(m_c|x_t, z_t)\text{odds}(m_c|x_{1:t-1}, z_{1:t-1})}{\text{odds}(m_c)} \quad (3.4)
\]

taking the logarithms changes between \((-\infty, \infty)\); where 0 corresponds to initial unknown state, and where positive and negative values correspond to probabilities of being occupied and non-occupied correspondingly. The larger the absolute value, the greater the certainty:

\[
\log(odds)(m_c|x_{1:t}, z_{1:t}) = \log(odds)(m_c|z_t, x_t) - \log(odds)(m_c) + \log(odds)(m_c|x_{1:t-1}, z_{1:t-1}) \quad (3.5)
\]

Results of the likelihood field (grid map) of SimVille with number of consecutive steps of a mobile agent shown in Figure 3.4. Here, white, black, and gray regions correspond to traversable, occupied, and unknown grid cells. Due to the probabilistic nature of the grid-mapping algorithm, each of these categories of grid cells are identified from their occupancy probabilities.
After grid map is obtained with high accuracy, this likelihood map is used on each agent for scan matching. Identifying segments corresponding to walls, improving ego vehicle pose by matching wall segments to the likelihood map.

3.3.2 Segmentation and Vehicle Detection

After correction of the poses, sensor measurements are transformed to global coordinates in order to cluster returns. In scan segmentation step, first of all Random Sample Consensus (RANSAC) [45] is used to detect objects in the environment. However, in order to fasten the detection with limited computation on robots and
eliminate the cons of the single model representation of RANSAC, a simple algorithm is used which finds the differences between consecutive LiDAR beam returns, compares them with a predefined threshold, and then divides the scan into segments at the points where the threshold is exceeded. Each segment is then defined based on the indexes of the beams at its beginning and at its end. Example of a LiDAR scan with identification is illustrated in Figure 3.5.

Figure 3.5: Segmented LiDAR Scan, with Walls and Vehicles Identified

Once the scan is divided into segments, the lengths of the surface corresponding to each scan segment are computed. If the surface length of a given segment is above a certain threshold, the corresponding scan segment is labeled as a wall.
After wall segments are identified, the corresponding returns are matched to the grid map (likelihood map) of the environment. This is done by searching for an improved ego pose around the estimate obtained from dead reckoning. For any chosen ego vehicle pose hypothesis, the endpoints of the beams corresponding to wall segments fall into different locations on the grid map. The best hypothesis is chosen by evaluating the collective probabilistic distances of beam endpoints to the closest obstacles defined in the grid map. This step is important to detect moving obstacles such as other vehicles.

Vehicle detections are then found by first generating hypothesis from scan segments (by comparing surface length to upper and lower thresholds), checking if the segment is occluded or not (since occluded vehicles are hard to differentiate from occluded walls), and then checking if the corresponding grid cells are traversable. If all the conditions are met, then the vehicle detection is sent to the vehicle-tracking algorithm described in next sections.

3.4 Multi-Vehicle Tracking

Once the fully-equipped vehicles perform vehicle detection through their local sensors, integration of the information to track list follows with detect-to-track data fusion. Then it is followed by fusing additional V2V information in to local tracks with track-to-track data fusion.

3.4.1 Detect-to-Track Data Fusion

After vehicle detections are acquired, integration of the information to track list is needed to perform. The approach presented in [41] is based on Kalman filtering coupled with data association and initialization steps. It can be seen as consisting
of a time update which uses motion model to generate predictions from tracks, a
data association step which finds correspondences between detections and predictions,
a data initialization step which initializes unmatched detections as new tracks, a
measurement update which fuses detections to their corresponding tracks, and a track
list manager which manages the contents of the track list.

For the time update step, vehicle motion model is used to project the tracks
forward in time. In other words, the prediction step for vehicle track are represented
as follows:

\[
\begin{align*}
x_{k|k-1}^{(i,j)} &= F_k x_{k-1|k-1}^{(i,j)} \\
P_{k|k-1}^{(i,j)} &= F_k P_{k-1|k-1}^{(i,j)} F_k^T + Q_k
\end{align*}
\] (3.6)

where the vectors \(x_{k|k-1}^{(i,j)}\) and \(x_{k-1|k-1}^{(i,j)}\) represent the (priori) prediction and the (poste-
riori) estimate of track \(j\) by vehicle \(i\) at time-step \(k\) and time-step \(k - 1\) respectively;
the matrices \(P_{k|k-1}^{(i,j)}\) and \(P_{k-1|k-1}^{(i,j)}\) represent the corresponding covariance matrices;
and the matrices \(F_k\) and \(Q_k\) represent the state transition matrix and process noise
covariance of the vehicle motion model. The state vector consists of position and
velocity in global (x,y) coordinates.

After the time update step of the Kalman filter, data association is performed.
First, the innovation terms and their covariance matrices, \(v_k^{(i,j,p)}\) and \(S_k^{(i,j,p)}\) are com-
puted for all combinations of detections and tracks available to vehicle \(i\). Here, \(z_k^{(i,p)}\)
represents the vehicle detection \(p\) performed by fully-equipped vehicle \(i\) at time-step
\(k\). The matrices \(H_k\) and \(R_k\) represent the measurement model and the covariance of
the measurement noise respectively.
\begin{align}
    v_k^{(i,j,p)} &= z_k^{(i,p)} - H_k x_{k|k-1}^{(i,j)} \\
    S_k^{(i,j,p)} &= H_k P_{k|k-1}^{(i,j)} H_k^T + R_k 
\end{align} (3.7)

For data association and finding the correspondences between any given vehicle track and vehicle detection Equation 3.8 is used. Here, \( p_j \) is the index of the track which is (probabilistically) closest to detection \( p \). The term inside the \( \text{argmin} \) function is the Mahalanobis distance, a distance metric for probabilistic variables, calculated using the innovation vector and covariance.

\[
p_j = \text{argmin} \sqrt{\left(v_k^{(i,j,p)}\right)^T \left(S_k^{(i,j,p)}\right)^{-1} v_k^{(i,j,p)}} \quad (3.8)
\]

Finding closest correspondence has been checked for the same threshold value, such as detect-to-track correspondence threshold (DTT). If the condition is met, then the correspondence is established and the Kalman filter integrates the two in the correction step; if the condition is unmet, however, then the detection is passed to the data initialization step.

For the detect-to-track data integration Kalman filter correction step is used (Equation 3.9) in order to fuse all corresponding detection \( p_j \) and track \( j \) with the matrix \( K_k^{(i,j)} \) Kalman gain.

\[
    K_k^{(i,j)} = P_{k|k-1}^{(i,j)} H_k^T \left(S_k^{(i,j,p)}\right)^{-1} \\
    x_k^{(i,j)} = x_{k-1|k-1}^{(i,j)} + K_k^{(i,j)} v_k^{(i,j,p)} \\
    P_{k|k}^{(i,j)} = (I - K_k^{(i,j)} H_k) P_{k|k-1}^{(i,j)} 
\] (3.9)
For the vehicle detections with unassociated vehicle track, a new track is created. In order to initialize a vehicle track, consecutive detections from the current and previous time steps are required. To associate the detections reliably, hidden state information is acquired through basic geometric calculations. Similar to previous data association step, one can use the Mahalanobis distance; however, for the first time threshold for Euclidean distance is also implemented.

\[
q_p = \arg \min \sqrt{(z_k^{(i,p)} - z_k^{(i,q)})^T (z_k^{(i,p)} - z_k^{(i,q)})} \tag{3.10}
\]

After correspondences are found between consecutive detections, geometric calculations are used to infer states. Since the simple motion model is used for moving objects, decoupled position and speed calculations for the state and corresponding covariance matrix can be made as:

\[
x_{i,j}^{(i)}|_{k(k} = \begin{bmatrix} z_k^{(i,p)} \\ z_k^{(i,p)} - z_k^{(i,q)} \end{bmatrix} \tag{3.11}
\]

\[
P_{i,j}^{(i)}|_{k(k} = \begin{bmatrix} R_k & 0 \\ 0 & \frac{R_k + R_{k-1}}{dt^2} \end{bmatrix}
\]

### 3.4.2 Track-to-Track Data Fusion

Effect of V2V communication is used in this section in order to integrate additional information to detections gathered by only on-board sensors. In addition to BSMs, data packets containing state information (as result of on-board sensing and tracking) related to non-equipped vehicle tracks is added by fully-equipped vehicles.
By incorporating such data packets in the communication strategy, V2V-equipped vehicles also have access to information regarding to non-equipped vehicles.

The track list received via V2V communication is also subject to data association, initialization and integration step as explained in previous section and depicted in Figure 3.2. In data association, correspondences between received tracks and local tracks are determined. For the QBSMs, since the vehicle ID’s are unique in communication, data association is carried out using vehicle identifiers. However, for the non-equipped vehicles, similar approach as Mahalanobis distance calculation is used.

Data association follows up with fusion of corresponding vehicle tracks using a covariance intersection algorithm (Equation 3.12). An important reason to use covariance intersection method is that the resulting fusion is consistent even when the fused signals are highly correlated. For example, relative locations of vehicle tracks obtained from Kalman filtering is one factory that correlation occurs.

\[
P^{(i,j)}_{k|k} = \left( \omega^{(i,j)}_{k} (P^{(i,j)}_{k|k})^{-1} + (1 - \omega^{(i,j)}_{k}) (P^{(i,j)}_{k|k})^{-1} \right)^{-1}
\]

(3.12)

\[
x^{(i,j)}_{k|k} = \omega^{(i,j)}_{k} (P^{(i,j)}_{k|k})^{-1} x^{(i,j)}_{k|k} + (1 - \omega^{(i,j)}_{k}) (P^{(i,j)}_{k|k})^{-1} x^{(j,j)}_{k|k}
\]

where \( \omega^{(i,j)}_{k} \) denotes the weightings of the data sources such as V2V and local-sensors in this specific case. The weightings are calculated as in Equation 3.13.

\[
\omega^{(i,j)}_{k} = \frac{\text{trace}(P^{(j,j)}_{k|k})}{\text{trace}(P^{(i,j)}_{k|k}) + \text{trace}(P^{(j,j)}_{k|k})}
\]

(3.13)
Finally, the data initialization for the received tracks without corresponding local tracks is carried out by the track list manager, by simply copying its contents into a new entry in the track list.

### 3.4.3 Entropy Calculation and Track Deletion

The concept of entropy appears when discussing the predictability of the information content shared in the message and it a general measure for the uncertainty of a belief [46]. Messages with low entropy have a higher predictability, while higher entropy is associated with less predictable content.

As in our example, assumption of a multivariate Gaussian distribution as tracked with Kalman filter has special representation of an entropy formulation which is proportional to the logarithm of the determinant of its covariance matrix. The following entropy $H_k^{(i,j)}$ calculation (Equation 3.14) describes the entropy of any given track, $i$ of any given vehicle, $j$ as an information-theoretic measure of uncertainty.

$$H_k^{(i,j)} = 2 (1 + \ln(2\pi)) + \frac{1}{2}(\ln|P_k^{(i,j)}|)$$  \hspace{1cm} (3.14)

Whenever the entropy value exceeds a given threshold for any given vehicle track, it means that its uncertainty exceeds an upper bound, the track list manager then deletes it from the track list. Alternatively, drawing covariance ellipsoids over estimated positions throughout estimated vehicle trajectories gives the idea of uncertainty, which is presented with example implementation.
3.5 SimVille Testing

In order to illustrate effectiveness of the proposed approach, three different scenarios are devised. Although a number of different scenarios can be demonstrated on SimVille, the following three are selected to demonstrate proof-of-concept in a clear and efficient manner. Vehicles with different types started test from various positions in order to cover possible situation in mixed traffic scenarios.

From the starting position, vehicles path are drawn with different colors that represents vehicle types such that red, green and yellow corresponds for fully-equipped, v2v-equipped, and non-equipped vehicle.

The performance of the approach is evaluated using entropy values, a metric for evaluating uncertainties contained in track lists. Each vehicles track list entropies are illustrated for scenarios.

3.5.1 Example Scenario I

As the first scenario is depicted in Figure 3.6, the vehicles start from a configuration where the fully-equipped vehicle cannot detect any vehicles in the beginning. This is observed from the fact that the tracks corresponding to the non-equipped vehicle are not initialized in the first 7 seconds of the experiment, as shown in Figures 3.7 and 3.11. During this time however the equipped vehicles know each others pose, by listening to each others basic safety messages.

Around 8th second, the fully-equipped vehicle detects the non-equipped vehicle, initializes the track, and broadcasts it. Upon receiving the track data, the V2V-equipped vehicle also starts tracking the non-equipped vehicle, as shown in Figures 3.7 and 3.8.
At the 15th second, the fully-equipped vehicle passes the non-equipped vehicle, and since its LiDAR is forward-looking it cannot detect the non-equipped vehicle anymore. Here, we see that the entropy of the non-equipped vehicle track increases rapidly, as it is updated using purely the motion model prediction step. As its entropy becomes greater than a threshold, it is deleted from the track list which happens around 22nd second.
Figure 3.7: Tracking Results of the Fully-Equipped Vehicle for First Scenario.

Figure 3.8: Tracking Results of the V2V-Equipped Vehicle for First Scenario.
As it can be seen from Figure 3.8, after the 20th second, the fully-equipped vehicle starts detecting the V2V-equipped vehicle (albeit imperfectly) and reduces the entropy of its track of the V2V-equipped vehicle.

3.5.2 Example Scenario II

Figure 3.9: Explanation of the Second Scenario. The Paths of Fully-Equipped (red), V2V-Equipped (blue), and Non-Equipped (green) Vehicles.
The second scenario is also similar to the first one. It can be seen from Figure 3.9 that the fully-equipped vehicle is not aware of the state information of the non-equipped vehicle, until the fully-equipped vehicle starts detecting it around 30th second.

![Fully-Equipped Vehicle's Tracks](image)

Figure 3.10: Tracking Results of the Fully-Equipped Vehicle for Second Scenario.

It is obvious from Figures 3.10 and 3.11, that the entropy of the track corresponding to the V2V-equipped vehicle is reduced as the fully-equipped vehicle starts detecting it around 36th seconds.

Finally, it is observed that, once the fully-equipped passes the non-equipped vehicle, and thereby stops detecting it, around 35-40th seconds, its entropy starts increasing, as expected, until the track is deleted from the track list as explained in previous sections.
3.5.3 Example Scenario III

In the third scenario, as depicted in Figure 3.12, two fully-equipped, one V2V-equipped, and one non-equipped vehicle are presented. Here the idea is to show the effects of detecting non-equipped vehicles with more than one fully-equipped vehicles on track entropy.

It can be seen that in Figure 3.13 first fully-equipped vehicle starts detecting and tracking the non-equipped vehicle around the 6th second. During this period, its entropy converges to -1. Around the 11th second, the second fully-equipped vehicle also starts detecting the non-equipped vehicle.

With the new source of information, the entropy for the track corresponding to the non-equipped vehicle further reduces to -2. Therefore, one can conclude that as the
number of fully-equipped vehicles increase in a traffic environment, the uncertainties of the vehicle tracks corresponding to non-equipped vehicles are reduced for both fully-equipped and non-equipped vehicles, as observed from Figures 3.13 and 3.14.
Figure 3.13: Tracking Results of the Fully-Equipped Vehicle for Third Scenario.

Figure 3.14: Tracking Results of the V2V-Equipped Vehicle for Third Scenario.
3.6 Conclusion

In this chapter, a collaborative vehicle tracking approach and its implementation on SimVille is presented for mixed-traffic environments, where vehicles with varying degrees of sensing and communication capabilities coexist.

The collaborative vehicle tracking approach with local track list update using various sensor fusion and data association techniques, such as Kalman filtering, and covariance intersect section methods are presented.

Contrast to the pure simulations on MATLAB, the real-time implementation brings different challenges, namely as reliable localization, online on-board vehicle detection and inter-vehicular communication. In this chapter, solutions to these problems are also presented, such as segmentation, grid mapping and scan matching on scaled-down urban driving testbed SimVille.

The approach is tested under the three different scenarios and results demonstrate the effectiveness of the approach and implementation. As an alternative method of showing the uncertainties of vehicles, estimated positions throughout estimated vehicle trajectories with %97 confidence ellipsoids are illustrated in Figure 3.15. These promising results can be used to test algorithms with real cars, as future work.
Figure 3.15: Screen Shot from Real-Time Monitoring Confidence Ellipsoids
Chapter 4: Test Scenarios, Equipment and Testing Process for LDW/LDP Performance Evaluation

4.1 Introduction and Literature Review

In this chapter, design, development, and implementation details for testing and evaluation of Lane Departure Warning and Prevention (LDP and LDW) systems are presented. The approach taken to generate a set of repeatable and relevant test scenarios and to formulate the test procedures to ensure the fidelity of the collected data includes initial statistical analysis of applicable statistics; growth and probabilistic pruning of a test matrix; simulation studies to support procedure design; and vehicle instrumentation for data collection.

Functions related to active safety systems and advanced driver assistance systems (ADAS) are relatively new additions to passenger vehicles [47]; the effects, performance requirements and evaluation methods for these systems are not finalized and are currently being investigated by both government agencies and the automotive industry. It is imperative to develop proper testing and evaluation methods for these systems, holding them to a standard level of performance similar to more conventional safety systems such as anti-lock brake or stability control systems.
The main goal is to generate a test procedure that results in repeatable and representative tests, by collecting a larger set of test data to analyze and identify most significant scenarios that indicate performance. There are a number of reports and documents on existing test efforts from federal agencies in United States, and international standardization organizations (ISO). For example, ISO standard 11270 defines the basic control strategy, minimum functionality requirements, basic driver interface elements, minimum requirements for diagnostics and reaction to failure, and performance test procedures for Lane Keeping Assistance Systems (LKAS) [48]. Similarly in [49], the U.S. National Highway Traffic Safety Administration (NHTSA) provides the specifications for confirming the existence of LDW hardware on light vehicles.

![Driving Simulator at CITR Laboratories](image)

(a) Outside View  
(b) Driver Perspective

Figure 4.1: Driving Simulator at CITR Laboratories

As similar to philosophy developed in this thesis, first step is to perform simulation studies. In the literature, the role of the driving simulator and its “close” relationship between real testing have been studied [50, 51]. Since the procedure development is
the important phase of this research, applicability of the proposed driving maneuvers for testing and test driver, first, evaluated on the driving simulator at CITR Laboratory (Figure 4.1). Then the tests are being conducted at the Transportation Research Center (TRC) in East Liberty, OH, (Figure 4.2).

![TRC Test Track - Skid Pad](image)

Figure 4.2: TRC Test Track - Skid Pad

In this chapter general guidelines for future LDW/LDP testing is presented starting from simulator studies to vehicle testing it also offer directions for automotive active-safety system tests in general.

### 4.2 Test Procedure Design

In order to develop a procedure for detailed testing and performance assessment matrix for LDW/LDP systems, various variables has been condensed to represent important scenarios.
The focus on this thesis is specific to lane-departure applications; however the overall methodology is generic enough to be applicable to a different variations of active/passive safety systems.

4.2.1 Test Variable Determination

To cover all possible cases and combinations for lane departure, one needs to consider more than thousands scenarios, such that a vehicle can depart its lane on roads of different geometry (straight or curved at different radii), to either direction, at different longitudinal and lateral speeds, over different lane markers, starting from different lateral positions on the original lane, etc. The first step towards developing the test suite and procedures for LDW/LDP systems included identifying all possible variable which is listed below:

- **Vehicle Initial Departure Angle:** Steep, medium, shallow
- **Vehicle Initial Position and Alignment:** Centered and aligned with lane, centered and not aligned with lane, off center and aligned with lane, off center and not aligned with lane
- **Vehicle Speed:** Slow urban (30 mph), fast urban (45 mph), average highway (60 mph), fast highway (75 mph)
- **Vehicle Departure Direction:** Left, right
- **Road Geometry:** Straight, mild curve, sharper curve, straight to curve transition, curve to straight transition
- **Road Marking:** Solid, dashed, raised pavement or reflectors (Botts dots), no marker; color; single side or both sides; intersections or incidental road or lane boundary markings; road surface type and quality; lane marker quality
• **Environmental Road conditions:** Dry, wet, crosswinds

• **Environmental Lighting conditions:** Day, night, sunset/sunrise glare

• **Driver Steering Reaction:** Hands off (free), fixed steering, reaction preceding LDW event, reaction after LDW event, reaction after LDP autonomous correction.

• **LDP/LDW Status:** On, off

• **LDW Method:** Audio, haptic, both

Simulator test and experiments are conducted based on elimination of these variables. This is mostly determined by what is physically available at the test track. On the other hand, some combinations of speed and road geometry are unfeasible for the safe testing, therefore these combinations are needed to be eliminated from the test scenarios.

### 4.2.2 Scenario Selection

In order to find the high-impact scenarios among the thousands of combinations, statistical studies on various crash databases are investigated. The most likely scenarios and configurations of lane departures that resulted in greatest loss of life and property is analyzed so that the test variable combinations identified in the previous step can be prioritized. Statistical results presented in [52, 53, 54] are used to calculate highest occurrence probabilities among the different combinations.
4.2.3 Prioritization and Elimination Based on Crash Data Analysis

According to the National Automotive Sampling System, Crashworthiness Data System (NASS/CDS) [55] evaluation for 2006-2010, 26% of all crashes were departure-related scenarios, and 65% of all departure-related crashes involved a single vehicle. In the sub-categories of road departure crashes, single vehicle, opposite direction and parked car crashes form 82% of road departure crashes, and in these three sub-categories the dominant (70%-90%) pre-crash movement is lane keeping. Only the same direction crashes, which form 15% of lane-departure cases, show a different dominant pre-crash maneuver (lane change), followed by lane keeping. Since the same direction, lane change crashes are within the domain of blind spot monitoring systems; it is safe to remove any possible lane changing or turning type of pre-departure maneuvers from the test matrix. On the other hand, LDW/LDP applicable crash scenarios are as illustrated and focused on the crash-data analysis aspects for these systems such as [56], 72% were single vehicle cases. Factoring this dominance and safety/coordination concerns for actual physical tests, the final test matrix can be trimmed to contain no multi-vehicle scenarios.

As 54% of all single-vehicle crashes were on undivided two-lane roads (one lane in each direction), compared to at best 12% of cases in any other lane/direction configuration, it is reasonable to limit the right-side departures to solid lane boundaries and left-side departures to dashed lane boundaries. Similarly, since the vast majority of all LDW/LDP applicable crash scenarios occur on non-intersection areas it is possible to remove intersections or incidental road or lane boundary markings from the test matrix. Road conditions for lane-departure related crashes are shown to be
mostly dry, at 83% of all single vehicle crashes, compared to the 15% of wet-road crashes according to the NASS/CDS data. For both this statistical dominance and safety/logistics concerns, the vast majority of the test scenarios should be conducted on dry surface, with only a few straight and relatively slow scenarios on wet surface.

If the road curvature and vehicle alignment data presented in NASS/CDS is utilized, 55% of road departures happening on straight roads should be proportionally represented in the test matrix. As already discussed above, departures towards both sides (left and right) can be prioritized to be tested at more common speeds, and the data further prioritizes the curved-road tests to depart the road in the opposite direction of the curve.

4.2.4 Prioritization and Elimination Based on Scenario Security and Logistics

In addition to the trimming and prioritization concepts discussed in the previous section, a number of test variable combinations had to be removed due to inherently unsafe characteristics such as steep departures at high speeds. In this context, the term “unsafe” refers to the safety of the test driver. Also, the final test matrix and list of test scenarios need to be physically implementable at the test site, selected to be TRC for LDW/LDP tests, and the availability of adequate room to reach certain speeds is a factor that may limit the possible scenarios.

As a first step towards ensuring the safety of the test driver and the testing equipment, only shallow departure angles are tested at higher speeds. At the comparatively slower speeds, shallow and medium departure angles can be accommodated at 50mph; and the slowest speed selected (40mph) can be used to test three departure angles, shallow, medium and steep.
Road geometry and light conditions are two of the numerous factors that will be affected by the availability of TRC test tracks. Once the set of available test tracks/courses was determined, a simulation world was built to check if certain speeds and departure rates are physically possible and safe, as further discussed in the next subsection. Moreover, the final test matrix included most, if not all tests conducted at daylight, with a potentially safer few conducted at early morning or late afternoon to test for the effects of glare and long shadows. The reasoning behind heavily favoring the daytime conditions is twofold. First of all, the majority of LDW-applicable crashes happen under daylight (38%) or dark but lighted (36%) conditions, based on NASS/CDS 2012 data [56]. At TRC, which hosted the current generation of NHTSA LDW tests, it is not possible to artificially light most of the test track except with that light provided by the test vehicles headlights. Therefore, the only part of the 76% lighted (daylight or artificial) scenarios that can be captured in practical NCAP-style testing is the daylight portion. On the other hand running unlighted scenarios, which correspond to 21% of LDW-applicable scenarios, with precision gate positioning and road departures at highway speeds would be difficult to implement and potentially unsafe for the test driver and researchers on board. Hence, in the final matrix, daylight conditions are selected as main test scenarios.

4.2.5 Prioritization Based on Joint Probabilities

After prioritization of the combination of scenarios based on crash data, joint distributes over the entire test variables are calculated to prune the scenarios in the final matrix.
Combination of different variable presented above is used to calculate joint probabilities; where the statistical frequencies are used for probabilities and independence assumption is taken. From purely statistical point of view, more advance analysis might be required to investigate the independence assumption; however, in this research, we have multiplied joint probabilities are used to prioritize the combination of scenarios. The color coding of the cells indicate a higher probability for darker red cells.

The following variables are represented in the probability table by using the statistics that are available in [57] dataset ASS/CDS and the 100-car naturalistic study [58]:

- **Lane Marker**: Solid, dashed
- **Road Shape**: Straight, curved
- **Departure Direction**: Left, right
- **Departure Angle**: Shallow, medium, steep
- **Vehicle Speed**: 30,45,60,70mph
- **Surface Condition**: Dry, wet

The condensed set of variables and their independent probabilities, a table of 288 joint probabilities was generated. The entire table is presented in the Appendix A, in Table 4.1 portion of this table is given for solid marker, straight road shape.

In total, LDW-only system performance was tested in 29 combinations of speeds, departure directions, lane marker types and curvatures, each for a range of steering angles, covering $29 \times 3 = 87$ cells of the probability table. Similarly, 20 combinations were tested in the LKA-on state, covering $20 \times 3 = 60$ cells of the joint probability table.

### 4.2.6 Comparable Procedures

Existing test by federal and international agencies are investigated in order to verify steps for the test scenarios and have guidelines for practical considerations of LDW/LDP performance tests.

**Table 4.2: NHTSA LDW Test Matrix**

<table>
<thead>
<tr>
<th>Lane Geometry</th>
<th>Lateral Velocity</th>
<th>Line Type</th>
<th>Departure Direction</th>
<th>Number of Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight</td>
<td>Low (0.5m/s)</td>
<td>Solid</td>
<td>L</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dashed</td>
<td>L</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Botts Dots</td>
<td>L</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 4.3: ISO 17361 Standard Table Showing Straight Road Tests.

<table>
<thead>
<tr>
<th>Rate of Departure</th>
<th>Departure Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left</td>
</tr>
<tr>
<td>0.1 &lt; X1 ± 0.05 &lt; 0.3 m/s</td>
<td>Group 1</td>
</tr>
<tr>
<td></td>
<td>Four Trials</td>
</tr>
<tr>
<td>0.6 &lt; X2 ± 0.05 &lt; 0.8 m/s</td>
<td>Group 3</td>
</tr>
<tr>
<td></td>
<td>Four Trials</td>
</tr>
</tbody>
</table>

The tests conducted by National Highway Traffic Safety Administration (NHTSA) New Car Assessment Program (NCAP) for LDW confirmation [49] use the test matrix shown in Table 4.2. The speed selected for these tests is 45mph and only low lateral departure velocities are tested. The tests conducted in this NCAP matrix is almost entirely covered by the consolidated test matrix of the previous section, with the exception of Botts Dots, because of the unavailability of the test track.

On the other hand, ISO 17361 standard [48] uses the Table 4.3 as the test matrix for the straight-road testing and it is similar to the prioritization concepts discussed above (where X1 and X2 are selected by the manufacturer).

4.3 Simulation Investigation

In order to achieve comprehensive testing procedure and reduce the time spend on actual physical testing, driving-simulation studies are performed. To better reflect the conditions at TRC, the location of the actual physical tests with real LDW/LDP-equipped vehicles.
4.3.1 Driving Simulator

Fixed based with vehicle cab, high-fidelity simulator located at OSU CITR Laboratories, as seen in Figure 4.1, is used in this research. Realtime Technologies (RTI)’s SimCreator [59] is used to create control base of the simulation environment. The graphical simulation and modeling system allow us to integrate new functions and modify the existing dynamics of the vehicles.

Along with the vehicle cab, 180-degrees, field-of-view across three 48 inch LCD displays are used to provide realistic feeling to driver. SimVista [60] is the main software for the tile-based scenario visualization. The simulation world is created through Internet Scene Assembler (ISE) using VRML files. The visuals operate at a 60 Hz update rate and latency of less than 50 ms. To create course, as explained in next sections for specific case of TRC, various road segments, such as two-lane highways, buildings, cones combined together.

4.3.2 Simulated World

The simulated world contains two straight road sections, each 600 meters long, and two curved sections with half circles with 230-meter and 730-meter radii of curvature. These curvatures are selected in order to match the curvatures of the TRC test tracks. The roads are built to have three lanes in the same direction, two of which are separated with dashed lane markers and two separated with solid lane markers. The VRML file generation using ISE and the simulated roads can be seen in Figure 4.3.
4.3.3 Simulated ADAS

Since actual details of the algorithms implemented on the commercial and test vehicles are unavailable (confidential), we have developed simple LDW/LDP systems to facilitate the simulation activities.

The LDW/LDP systems implemented in the simulated vehicle with SimCreator blocks, use two main error signals. The distance from the vehicle center to the lane center, which is denoted as the lane offset, and the angle between the lane direction and the vehicle yaw, which is denoted as the heading error.

On the other hand, the lane offset is used to drive a PD controller and the heading error drives a proportional controller (as seen in Figure 4.4. If LDP is enabled, the outputs of these controllers are added to the steering input coming from the driver.
before the steering command is sent to the simulated vehicle. Error calculations and warning zones are depicted in Figure 4.5.

The warning for LDW is activated when the vehicle exceeds a lateral certain distance from the lane center. If the LDP system is enabled, the correction system is designed to be active at a greater distance than the warning line, as shown below. These two deadzones around the lane center are selected to warn the driver before correcting the departure. Since we don’t have vibration mechanism for the steering wheel, in the simulation environment, the warnings are auditory.
Table 4.4: Different Scenarios on Driving Simulator

<table>
<thead>
<tr>
<th>Run no</th>
<th>Speed (mph)</th>
<th>Depart Dir</th>
<th>Road</th>
<th>Depart Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>R</td>
<td>straight</td>
<td>shallow</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>R</td>
<td>straight</td>
<td>steep</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>L</td>
<td>moderate cw curve</td>
<td>shallow</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>L</td>
<td>moderate cw curve</td>
<td>steep</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>L</td>
<td>mild cw curve</td>
<td>shallow</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>L</td>
<td>mild cw curve</td>
<td>steep</td>
</tr>
<tr>
<td>7</td>
<td>45</td>
<td>R</td>
<td>straight</td>
<td>shallow</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>R</td>
<td>straight</td>
<td>steep</td>
</tr>
<tr>
<td>9</td>
<td>45</td>
<td>L</td>
<td>straight</td>
<td>steep</td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>L</td>
<td>moderate cw curve</td>
<td>shallow</td>
</tr>
<tr>
<td>11</td>
<td>45</td>
<td>L</td>
<td>moderate cw curve</td>
<td>steep</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>L</td>
<td>mild cw curve</td>
<td>shallow</td>
</tr>
<tr>
<td>13</td>
<td>45</td>
<td>L</td>
<td>mild cw curve</td>
<td>steep</td>
</tr>
<tr>
<td>14</td>
<td>60</td>
<td>R</td>
<td>straight</td>
<td>shallow</td>
</tr>
<tr>
<td>15</td>
<td>60</td>
<td>L</td>
<td>mild cw curve</td>
<td>shallow</td>
</tr>
</tbody>
</table>

4.3.4 Simulated Scenarios

A sample list of test scenarios, containing 15 combinations of speed, departure direction, road geometry and departure rate/angle are used to collect data from the simulated system. The list of simulated scenarios, which can be seen below on Table 4.4, is run three times, with each scenario tested once with LDW only and the driver manually correcting the departure, once with an LDP implementation that is tuned to have weaker and later response to a departure (LDP-1), and once with an LDP implementation with stronger and earlier response (LDP-2).

4.3.5 Simulator Test Procedure

In order to have repeatable tests for better comparison and evaluation, a test procedure was developed for the simulated runs. Steering rates for different departure
angles were defined in terms of steering wheel positions (±5, ±10 degrees). Timing for the departure-generating maneuvers were defined in terms of metronome tick-tocks (emulated by a smartphone metronome app). Longitudinal position of departure for the steering maneuvers was defined by cones in the simulated world.

The complete procedure is listed below:

1. The simulator started with right input files in order to load the relevant test world, such as straight, curve with different radii.
2. Metronome is set to 60 beats per minute.
3. Once the driver arrives at the cone marking, as seen in Figure 4.6, depending on the departure rate, perform either the shallow or speed departure using the markers on the steering wheel (as seen in Figure 4.7 and the timing of the metronome:

Figure 4.6: Test Starting Point with Cones
• On straight road departures, a shallow departure is steering to the 30-degrees marker for one second (the period between two beats of the metronome) and zeroing the wheel in the next second.

• Similarly, a steep departure on the straight road is steering to the 60-degrees marker for one second (the period between two beats of the metronome) and zeroing the wheel in the next second.

• On curved roads, a shallow departure is performed by letting go of the steering wheel.

• On curved roads, a steep departure is performed by steering to the 30-degrees marker for one second (the period between two beats of the metronome) and zeroing the wheel in the next second.

4. Depending on the test run (LDP or LDW) suites driver lets the LDP to correct or wait for the LDW warning buzzer.

5. Stop simulation once the driver within the original lane for the corrected cases, and if the LDP fails to correct the departure, stop once the vehicle is fully in the wrong lane, and collect the data.

4.3.6 Simulator Results, Analysis and Initial Metrics

Five main metrics are picked and calculated for each initial simulation testings, these are listed as:

• Time spent outside original lane (in seconds)

• Maximum exceedance amount (in meters)

• Area covered in the wrong lane (in meters squared)

• Time of maximum exceedance (in seconds)
• Longitudinal distance covered outside the lane (in meters)

Simulator data are processed and various scripts are written in MATLAB in order to fasten the process. The plots and calculated metrics are analyzed, as the scenarios became more varied, the calculation and analysis methods were redesigned to be more generic, working with all road geometries and departure directions.

In total, 45 simulated lane departures are processed and sample plots for lateral acceleration, lateral speeds, time-to-departure (TTD) estimates, the lane offset of the leading tire, vehicle speed and steering angle is presented in Figure 4.8.

The metrics listed above are calculated for each of the 45 runs (15 scenarios times three ADAS configurations). An example of the table including the results of the simulation analysis for maximum exceedence is depicted in Table 4.5.
Figure 4.8: Sample Simulated Run 1: 30mph, Straight Road, Right Departure, Steep Departure, LDP-1
Table 4.5: An Example Table of Maximum Exceedence Analysis

Once data for entire simulation study is collected and classified, next attempt is to identify a correlation between the metrics by plotting one against another in 2D plots or three metrics together on 3D plots, for example, correlation between multiple matrix is illustrated in Figure 4.9. These comparisons can be deduced that the better LDP implementation generates generally lower metrics. However, the distributions of metrics for different LDP implementations do not seem to be clustered enough to be classified through linear methods.

Correspondingly, the following issues are noted and lessons are learned that can be applied to improve tests as a result of the preliminary tests conducted at the OSU CITR driving simulator:

- Due to the relatively short lateral distances to be covered, following a specific line for lane departure is problematic. The driver does not always have time to settle on a specific line before a departure actually occurs.
- Marking longitudinal positions for departure maneuver start point and describing steering behavior provides better and more repeatable experiments.
Figure 4.9: An Example Plot for Correlation Between Multiple Metrics

- Departure rate/angle values can be tied to steering wheel positions. For example, on straight road departures, turning the steering wheel 30 degrees can be a mild departure, while a 60 degree turn can be a steep departure. We just need to match the departure rate of the same run in different configurations (LDW-only, LDP-1, LDP-2). These steering-wheel turn arcs can be marked on the wheel.

- To have matching departure maneuver times between runs and run sets, a metronome (or in this case a free metronome app on a smartphone) is useful.

Finally, with the help of this simulation study, we are able to develop test procedures for repeatable test runs. These tests focuses on certain longitudinal speeds and
precisely time departure-generating maneuvers executed at surveyed locations on the test track, which explained in next sections.

4.4 Vehicle Testing

Experience gained from simulation studies is used to define initial test procedure for the vehicle testing. In this section details of the vehicle testing including the development and improvement stages of the test procedure design, updated metrics, vehicle instrumentation, and, finally, results are presented.

4.4.1 Vehicle Instrumentation

Sensor outputs from the simulator was almost perfect; however, for the selected test cases to be accurately captured for analysis and evaluation in real testing, the test vehicles are instrumented using cameras, high-precision GPS receivers, IMUs, and CAN-bus adapters. Relevant CAN-bus messages are identified. Using multiple independent data sources such as RTK-corrected GPS tracks of the vehicle compared to surveyed lane-marker locations and externally mounted cameras to identify lane departures, a high-fidelity analysis of the departure/drift events was possible. A camera aimed at the vehicle instrument cluster captured warning indications, and two side-mounted cameras capture the front vehicle wheels and road lane markers.

The specific instrumentation used to be used in tests is shown in Figure 4.10. The equipment consists of:

1. Novatel OEM-4 RTK capable dual channel GPS receiver and antenna- to provide 2cm level position, pose, and motion information at 10Hz.
2. Microhard Systems 1Watt 900 MHz data modem and antenna- to provide RTK correction data to the GPS receiver (compatible with existing OSU system)
3. Freewave FGR-115RC 900 MHz data modem and antenna (not shown)- to provide RTK correction data to the GPS receiver (compatible with existing TRC system)

4. Kvaser Leaf light HS CANbus to USB transceiver- to access data from the vehicles CAN bus

5. Vectornav VN100 AHRS- to provide calibrated and bias-corrected IMU and AHRS information

6. Gladiator Technologies LMRK50VG-175-06-100 (not shown)- potentially to be used as a near-tactical grade vertical gyroscope IMU in case the Vectornav device proves insufficiently accurate

7. ACTI MPEG 4 audio/video capture system- network attached video capture system
8. Color NTSC video cameras- to collect 30-fps dashboard and wheel/lane marker video

9. Labjack UE9 USB IO device (not shown)- to be used if needed for analog or digital inputs

10. Dashboard mounted LED (shown in Figure 4.12)- provides precise video/data collection synchronization

11. Data collection computer

12. 300 watt dc/ac power inverter (not shown)

The dashboard camera is mounted on top of the steering column to capture the visual indicators for LDW/LDP systems, and the wheel cameras are mounted on articulated clamp mounts clamped on the rear door handles of the vehicle and pointed towards the two front wheels, as seen in Figure 4.11

### 4.4.2 Vehicle Test Procedure

Similar gate setup as tested in simulation studies (Figure 4.6), is used to set cones for road testings. The gate, shown in Figure 4.13, is set 150 cm away from the lane marker on the side that will be crossed during departure. The width of the gate shown in the figure is set for a specific vehicle and can be changed depends on the vehicle base width, and it is set at 205cm, as the NHTSA LDW test procedure uses gates that are eight inches wider than the vehicle. The longitudinal distance between the first and second pairs of cones is set to 6 meters, which is also consistent with NHTSA procedures.

The driver is instructed to achieve the desired longitudinal speed (40, 50, or 60 mph) before reaching the start gate. For this, the leading portions of the straight
Figure 4.11: Wheel Camera Snapshots for Right Departure

Figure 4.12: Dashboard of the Test Vehicle with the LDW system Status Display and the Synchronization LED Visible.
track or the connecting curves are used to accelerate to speed and the portion of the scenario after the gate occurs at relatively constant longitudinal speed. Cruise control may be used if applicable.

Again, as consistent with simulator studies, the maneuver is timed using a metronome set to 60bpm. Following the metronome beats, the driver steers to the desired steering wheel angle over the one second interval and then releases the wheel. A laser pointer aimed at the steering wheel is used as a reference point to mark both the neutral steering point, and the desired steering amount.
Note that the amount of steering required for each road geometry is defined based on the neutral steering that is required to follow the lane. For straight roads, the neutral steering generally corresponds to the steering wheel being centered (zeroed). For each speed on each curve and banking configuration, the neutral steering that generates centered lane following is noted, and steering deviations are centered on this neutral steering value to generate the test matrix.

The test drivers were instructed not to interfere with the system for the both LDW (different from the simulation studies) and LDP tests. If the LKA system successfully kept the vehicle in the lane, or returned the vehicle back into the original lane after a departure maneuver, the driver keeps hands off the steering wheel until the system warns the driver to hold the wheel and turns itself off. If the departure is not corrected, the driver takes back the control of the steering wheel once the vehicle is fully in the next lane.

4.4.3 Vehicle Testing and Data Collection

The tests designed and refined in the steps described above have been conducted at TRC with two test vehicles. Among the variables measured are vehicle speed, steering wheel angle, absolute yaw angle, yaw rate, vehicle/wheel position with respect to the lane marker using RTK GPS/IMU measurements and a highly accurate site survey, electronic LDW/LDP activation indicators from vehicle CAN bus, and visual dashboard indicators for the system captured using a camera.

Examples of the recorded raw data are listed below:

- Timing and level of vehicle intervention steering or braking actions
- Driver reaction
– Steering angle over time (including driver delay)
– Steering rate over time
– Brake actuation over time

• Final state or position of vehicle
• State, trajectory, or position of vehicle profile over time
• LDW/LDP activation indicators
  – Visual from dashboard camera
  – Electrical from CAN-bus

• Front wheel positions relative to lane markers, visually captured from wheel cameras

4.4.4 Metrics for Final Analysis and Evaluation

Metrics that are provided from simulation are revised. From the measurements described above, a number of metrics can be extracted, such as the time spent and area covered outside the original lane, maximum exceedance into the next lane (or off-road), and the lateral positions and times of the warning and correction signals.

For a given departure scenario, a series of important points can be identified, as illustrated in Figure 4.14. The identification of these points, where possible, is useful in defining the performance metrics.

A list of the significant scenario points and short descriptions are given below. Note that the ordering and sometimes even existence of these points are based on the system performance. For instance, if the system does not detect and correct the lane departure, the activation time or peak exceedance points cannot be defined for that test run.
• **Point 1:** Departure-generating maneuver initiation (test driver starts steering). This point is identified using the recorded steering wheel position, which is included in the captured CAN-bus data.

• **Point 2:** LDW issues warning. This point can be before or after Point 3 depending on the system implementation. This point is identified using the recorded CAN-bus data. The LDW status and activation is available on the CAN bus. Dashboard camera recordings showing the visual LDW indicators are used to verify that the CAN-bus recordings are correct and timely.

• **Point 3:** Leading tire (the one closest to the departure side) crosses the lane marker. This is calculated using the high-precision GPS position of the vehicle and the high-precision survey data of the lane markers. The recordings from the external cameras showing the tire and lane marker are used to verify these calculations.

• **Point 4:** LKA/LDP control input is applied. This point can be before, between, or after points 2 and 3 depending on the system implementation. This point is identified using the recorded CAN-bus data. Dashboard camera recordings
showing the visual LDW/LKA indicators are used to verify that the CAN-bus recordings are correct.

- **Point 5:** Leading tire reaches maximum distance from the lane marker. This point is only identified if the departure is corrected or the vehicle settles on an off-lane but parallel trajectory. Similar to point 3, this is calculated using the high-precision GPS position of the vehicle and the high-precision survey data of the lane marker.

- **Point 6:** The tire that first left the lane is back in the original lane. Similar to points 3 and 5, this is calculated using the high-precision GPS position of the vehicle and the high-precision survey data of the lane marker. External camera images are used to verify.

- **Point 7:** LKA/LDP system stops the correction/control input. This point can be inside or outside the original lane depending on the success of the departure correction. This point is identified using the recorded CAN-bus data.

Using the test run data and the significant points described above, a range of metrics are calculated for each test run:

- **T1 T7:** The time when the vehicle is at Points 1-7. T1 is also used as the time origin for each run.

- **T8:** Time spent outside the lane. The difference between T6 and T3. Not calculated if the vehicle did not return to the lane.

- **T9:** Time to peak exceedance. The difference between T5 and T3. Not calculated if the trajectory has no maximum distance to the lane marker in the case of uncorrected departures.
- **D1-D7**: The distances between points 1-7 and the lane marker. D3 and D6 are zero, by definition. D5 is also referred to as maximum exceedence.
- **D8**: Distance to peak exceedence. Longitudinal distance between points 5 and 1.
- **D9**: Distance covered before re-entry. Longitudinal distance between points 6 and 1.
- **D10**: Distance before system gives up. Longitudinal distance between point 1 and point 7.
- **A**: Area covered (swept) by the vehicle outside the original lane. This metric can only be calculated if the departure is corrected completely.
- Maximum lateral acceleration and the timestamp of this point before system acts (generally between points 1 and 3)
- Maximum lateral velocity and the timestamp of this point before system acts (generally between points 1 and 3)
- Lowest lateral acceleration and the timestamp of this point after system acts (after Point 4)
- Maximum lateral velocity and the timestamp of this point after system acts (after Point 4)
- Maximum yaw angle and the timestamp of this point before system acts (generally between points 1 and 3)
- Lowest yaw angle and the timestamp of this point after system acts (after Point 4)
4.4.5 Analysis and Evaluation

In this section results of processed outputs and their analysis with example runs are presented. Since comparison of different vehicles is not in the scope of this thesis, details of the comparison results are not presented.

Example runs are selected from the straight section of TRC Skid Pad at 50mph, departing to the right over white solid lane markers. In the example run, 9.0 degrees of steering is used to generate the departure maneuver. In the Figure 4.15, the actual departure trajectory for the right front wheel of the vehicle can be seen, with the time instances of wheel crossings marked with vertical lines, and the lane marker position (normalized to 0) marked with the horizontal dotted line.

![Figure 4.15: Wheel Position Obtained High Accuracy GPS and Surveyed Data for Right Departure](image)

Figure 4.15: Wheel Position Obtained High Accuracy GPS and Surveyed Data for Right Departure
The timestamps for lane marker crossings are automatically calculated by checking the wheel position trajectory crossing the zero level. For this example, the wheel crosses the line towards the outside at t=24.2s, and comes back at t=27.1s, remaining outside for 2.9 seconds.

The area covered outside the lane is calculated by numerically integrating the trajectory of the wheel under the zero level on this plot, and the calculation gives 1.06 m² for this case.

The system reaction to the departure can be seen in the Figure 4.16, with the CAN bus data showing the amount of LKA steering generated, and the state of the LKA and LDW subsystems of the Lexus. The state changes in both LKA and LDW corresponding to the lane crossings in both directions are clearly visible, and the amount of steering applied by the LKA system is plotted in the units reported on the CAN bus.

In the Figure 4.17, we can see the steering maneuver of the driver and the resulting yaw, yaw rate, lateral velocity changes. The steering input can be seen to be 9.0 degrees, changing from 4.5 degrees reported by the CAN bus when driving straight to -4.5 degrees at the end of drivers maneuver.

Also visible on the Figure 4.17 are the maximum departure (heading error) angle induced by the steering maneuver (2.8 degrees), and the maximum lateral velocity resulting in the departure (2.88 m/s).

Using these figures and the corresponding MATLAB scripts generating them, the Table 4.6 illustrates the set of metrics are calculated.
Figure 4.16: LKA and LDW System States and Responses

Figure 4.17: Steering, Yaw Angle and Lateral Velocity
In order to validate the simulation studies and proposed approach, as a first step, the relationship between the departure maneuvers generated by fixed steering amounts and the resultant departure angles and lateral speeds are investigated.

As an example, at the lateral departure speeds obtained by the fixed steering maneuvers, as seen in Figure 4.18, it can be seen that the maximum lateral speeds obtained within a ±15 degree steering window reach ~1.5 m/s at 50mph and almost 2.0 m/s at 60mph. Given an average lane width of 4 meters, this corresponds to covering one lane width in 2 seconds, which is as fast as an intentional lane change. This range of lateral speeds also cover much more than the region prescribed in the ISO or NHTSA tests, as in simulator studies, which consider 0.8 m/s as the maximum lateral speed of interest.
Figure 4.18: Lateral Speeds for Different Steering Maneuvers at Two Different Speeds

On the other hand, correlations between different combinations are presented as similar to simulation studies. For an example, Figure 4.19 illustrated the LDW performance difference was observed when the behaviors of dashed-yellow line departures (on straight roads) on lanes with only one-sided markers and lanes with markers on both sides. On these departures, as shown below, the overall response time plot shows failed warnings mostly after 2 m/s lateral speed as usual, with a few failed warnings close to 1.6 m/s departure speed.

Complete analysis including the all combinations of the test matrix has been made, similar to examples and discussion presented in this section.
Figure 4.19: An Example LDW Response Distances for a Range of Lateral Speeds.

4.5 Conclusion

In this chapter, design, development, and implementation details for testing and evaluation of Lane Departure Warning and Prevention (LDP and LDW) systems are presented. Similar to our philosophy, high-fidelity simulations are performed first to generate a set of repeatable and relevant test scenarios and test matrices. Using these metrics as numerical representations for each run for a particular system, detailed analysis are presented.

Collected data and results of the analysis suggests that the developed test procedure is capable of repeatable test cases, with the selected scenarios providing a good range of different behavior profiles to assess system performance. The provided methodology, starting with simulation studies, comprehensive test variable selection,
growth and pruning of a complete test matrix down to significant scenarios, and the identification of repeatable test procedures spanning the test matrix of interest apply to a broader range of intelligent systems, and the same steps can certainly be adapted to a number of other active safety systems.

Most important result of the simulations studies were to help researchers to realize the potential problems at testing stage, such as setting steering angles, departure-initiating steering maneuver. These led researchers to redefine some of the scenarios and metrics.

Future work can include novel methods of comparative analysis on the collected test data using a number of direct and higher-order performance metrics, and possible applications of data classification tools to active-safety system evaluation, Certain grouping/clustering and classification tools such as k-means clustering or Gaussian mixture models can be utilized.
Chapter 5: Conclusion and Future Work

In this thesis, importance of the preliminary testing on the simulation and scaled-down environments for different intelligent transportation system applications, is presented. The relationship between various platforms and scales are described in detail; by introducing complete end-to-end testing procedures, implementing state-of-the-art algorithms. Results from different scales, quality metrics, are compared and illustrated.

A small-scale, low-cost and flexible supplement to the intelligent transportation system test procedure is discussed. Different concepts were demonstrated on scaled down testbed SimVille at OSU CITR Laboratories, with detailed implementation of applications based on an autonomous parking, stop sign precedence and green light speed advisory applications. The results were validated and tested in comparison with full scale tests performed with an autonomous and semi-autonomous vehicles.

A collaborative vehicle tracking approach on scaled down test platform is also presented. Various sensor fusion methods and algorithm implementation details for vehicle tracking, such as Kalman filtering, and covariance intersection, are given. Contrast to the pure simulations result on MATLAB, solutions to challenges, come
with the real-time implementation, such as vehicle detection, and localization are explained. Different mixed traffic scenarios are created to demonstrate the effectiveness of the approach and implementation.

Future work can include simulation studies for analyzing the large-scale effects of collaborative vehicle tracking on safety and experimental tests with real cars using additional sensing, such as vision-based vehicle detection in selected traffic scenarios.

Finally, design, development, and implementation details for testing and evaluation of LDP and LDW systems are presented. Development and testing steps starting from simulations to full experiments are presented. Test scenarios and metrics are developed, collected data are analysed to procedure repeatable test cases. Scenarios are pruned based on preliminary simulation study results and the selected scenarios provided a good range of different behaviors to assess system performance. Identification of repeatable test procedures spanning the test matrix of interest apply to a broader range of intelligent systems, and the same steps can certainly be adapted to a number of other active safety systems.

Novel methods of comparative analysis on the collected test data using a number of direct and higher-order performance metrics, and possible applications of data classification tools to active-safety system evaluation was left as future work.
Appendix A: Test Scenario Tables
<table>
<thead>
<tr>
<th>Lane Marker &amp; Road Shape &amp; Departure</th>
<th>Surface &amp; Speed &amp; Departure Dir</th>
<th>Rate</th>
<th>Solid</th>
<th>Dashed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left</td>
<td>30 Dry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.83 x</td>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td>45 Dry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.83 x</td>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>60 Dry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.83 x</td>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.26 x</td>
</tr>
<tr>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.26 x</td>
<td>.26 x</td>
</tr>
<tr>
<td>.62 x</td>
<td>.15 x</td>
<td>.26 x</td>
<td>.26 x</td>
<td>.26 x</td>
</tr>
<tr>
<td>.15 x</td>
<td>.26 x</td>
<td>.26 x</td>
<td>.26 x</td>
<td>.26 x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right</td>
<td>70 Dry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.83 x</td>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.37 x</td>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.62 x</td>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
<tr>
<td>.15 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
<td>.28 x</td>
</tr>
</tbody>
</table>

Table A.1: Test Scenarios Table for Individual Probabilities
<table>
<thead>
<tr>
<th>Wet</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>.15 x</td>
<td>.15 x</td>
</tr>
<tr>
<td></td>
<td>.17 x</td>
<td>.17 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td>45</td>
<td>.15 x</td>
<td>.15 x</td>
</tr>
<tr>
<td></td>
<td>.17 x</td>
<td>.17 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td>60</td>
<td>.15 x</td>
<td>.15 x</td>
</tr>
<tr>
<td></td>
<td>.17 x</td>
<td>.17 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td>70</td>
<td>.15 x</td>
<td>.15 x</td>
</tr>
<tr>
<td></td>
<td>.17 x</td>
<td>.17 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
<tr>
<td></td>
<td>.37 x</td>
<td>.37 x</td>
</tr>
</tbody>
</table>

Table A.2: Test Scenarios Table for Individual Probabilities Cont'd
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dry 30 Left</td>
<td>3.10E-03</td>
<td>1.81E-03</td>
<td>4.27E-03</td>
<td>1.86E-03</td>
<td>9.63E-04</td>
<td>4.27E-03</td>
<td>1.86E-03</td>
<td>9.63E-04</td>
<td>2.09E-03</td>
<td>7.74E-04</td>
<td>4.91E-04</td>
<td>1.07E-04</td>
<td>4.13E-04</td>
<td>2.41E-04</td>
<td>1.07E-04</td>
</tr>
<tr>
<td>Right</td>
<td>3.34E-02</td>
<td>1.29E-02</td>
<td>7.32E-03</td>
<td>1.31E-02</td>
<td>5.98E-03</td>
<td>2.97E-03</td>
<td>1.31E-02</td>
<td>5.98E-03</td>
<td>2.97E-03</td>
<td>1.41E-03</td>
<td>5.45E-04</td>
<td>3.18E-04</td>
<td>5.71E-04</td>
<td>2.21E-04</td>
<td>1.29E-04</td>
</tr>
<tr>
<td>Dry 45 Left</td>
<td>5.40E-03</td>
<td>2.09E-03</td>
<td>1.28E-03</td>
<td>2.88E-03</td>
<td>1.12E-03</td>
<td>6.51E-04</td>
<td>2.88E-03</td>
<td>1.12E-03</td>
<td>6.51E-04</td>
<td>1.38E-03</td>
<td>5.23E-04</td>
<td>3.06E-04</td>
<td>7.26E-04</td>
<td>2.79E-04</td>
<td>1.63E-04</td>
</tr>
<tr>
<td>Right</td>
<td>2.38E-02</td>
<td>8.48E-03</td>
<td>4.94E-03</td>
<td>8.88E-03</td>
<td>3.44E-03</td>
<td>2.05E-03</td>
<td>8.88E-03</td>
<td>3.44E-03</td>
<td>2.05E-03</td>
<td>9.52E-04</td>
<td>3.69E-04</td>
<td>2.15E-04</td>
<td>3.86E-04</td>
<td>1.49E-04</td>
<td>8.72E-05</td>
</tr>
<tr>
<td>Dry 60 Left</td>
<td>6.08E-03</td>
<td>2.34E-03</td>
<td>1.37E-03</td>
<td>3.33E-03</td>
<td>1.26E-03</td>
<td>7.29E-04</td>
<td>3.33E-03</td>
<td>1.26E-03</td>
<td>7.29E-04</td>
<td>3.23E-03</td>
<td>1.26E-03</td>
<td>7.29E-04</td>
<td>1.51E-03</td>
<td>5.84E-04</td>
<td>3.42E-04</td>
</tr>
<tr>
<td>Right</td>
<td>2.45E-02</td>
<td>9.48E-03</td>
<td>5.54E-03</td>
<td>9.94E-03</td>
<td>3.81E-03</td>
<td>2.25E-03</td>
<td>9.94E-03</td>
<td>3.81E-03</td>
<td>2.25E-03</td>
<td>1.07E-03</td>
<td>4.13E-04</td>
<td>2.41E-04</td>
<td>4.32E-04</td>
<td>1.67E-04</td>
<td>9.76E-05</td>
</tr>
<tr>
<td>Dry 70 Left</td>
<td>2.36E-03</td>
<td>8.37E-04</td>
<td>4.88E-04</td>
<td>1.15E-03</td>
<td>4.46E-04</td>
<td>2.80E-04</td>
<td>1.15E-03</td>
<td>4.46E-04</td>
<td>2.80E-04</td>
<td>5.40E-04</td>
<td>2.09E-04</td>
<td>1.22E-04</td>
<td>2.88E-04</td>
<td>1.12E-04</td>
<td>6.51E-05</td>
</tr>
<tr>
<td>Right</td>
<td>8.76E-03</td>
<td>3.39E-03</td>
<td>1.98E-03</td>
<td>3.55E-03</td>
<td>1.37E-03</td>
<td>8.02E-04</td>
<td>3.55E-03</td>
<td>1.37E-03</td>
<td>8.02E-04</td>
<td>3.81E-04</td>
<td>1.47E-04</td>
<td>8.00E-05</td>
<td>1.54E-04</td>
<td>5.98E-05</td>
<td>3.49E-05</td>
</tr>
<tr>
<td>Right</td>
<td>6.64E-03</td>
<td>2.57E-03</td>
<td>1.50E-03</td>
<td>2.69E-03</td>
<td>1.04E-03</td>
<td>6.08E-04</td>
<td>2.69E-03</td>
<td>1.04E-03</td>
<td>6.08E-04</td>
<td>2.89E-04</td>
<td>1.32E-04</td>
<td>6.52E-05</td>
<td>1.79E-04</td>
<td>4.53E-05</td>
<td>2.89E-05</td>
</tr>
<tr>
<td>Wet 45 Left</td>
<td>1.11E-03</td>
<td>4.28E-04</td>
<td>2.50E-04</td>
<td>5.90E-04</td>
<td>2.38E-04</td>
<td>1.33E-04</td>
<td>5.90E-04</td>
<td>2.38E-04</td>
<td>1.33E-04</td>
<td>2.77E-04</td>
<td>1.07E-04</td>
<td>6.25E-05</td>
<td>1.48E-04</td>
<td>5.71E-05</td>
<td>3.33E-05</td>
</tr>
<tr>
<td>Right</td>
<td>4.48E-03</td>
<td>1.74E-03</td>
<td>1.01E-03</td>
<td>1.82E-03</td>
<td>7.9E-04</td>
<td>4.11E-04</td>
<td>1.82E-03</td>
<td>7.9E-04</td>
<td>4.11E-04</td>
<td>1.99E-04</td>
<td>7.35E-05</td>
<td>4.40E-05</td>
<td>7.91E-05</td>
<td>3.09E-05</td>
<td>7.91E-05</td>
</tr>
<tr>
<td>Wet 60 Left</td>
<td>1.34E-03</td>
<td>4.80E-04</td>
<td>2.80E-04</td>
<td>6.16E-04</td>
<td>2.95E-04</td>
<td>1.49E-04</td>
<td>6.16E-04</td>
<td>2.95E-04</td>
<td>1.49E-04</td>
<td>3.10E-04</td>
<td>1.20E-04</td>
<td>7.05E-05</td>
<td>1.66E-04</td>
<td>6.40E-05</td>
<td>3.73E-05</td>
</tr>
<tr>
<td>Right</td>
<td>5.02E-03</td>
<td>1.94E-03</td>
<td>1.13E-03</td>
<td>2.04E-03</td>
<td>7.81E-04</td>
<td>4.08E-04</td>
<td>2.04E-03</td>
<td>7.81E-04</td>
<td>4.08E-04</td>
<td>2.18E-04</td>
<td>8.45E-05</td>
<td>4.85E-05</td>
<td>8.86E-05</td>
<td>3.43E-05</td>
<td>2.09E-05</td>
</tr>
<tr>
<td>Wet 70 Left</td>
<td>4.43E-04</td>
<td>1.76E-04</td>
<td>1.00E-04</td>
<td>2.36E-04</td>
<td>9.14E-05</td>
<td>5.33E-05</td>
<td>2.36E-04</td>
<td>9.14E-05</td>
<td>5.33E-05</td>
<td>1.11E-04</td>
<td>4.28E-05</td>
<td>2.90E-05</td>
<td>5.90E-05</td>
<td>2.28E-05</td>
<td>1.33E-05</td>
</tr>
<tr>
<td>Right</td>
<td>1.76E-03</td>
<td>6.94E-04</td>
<td>4.09E-04</td>
<td>7.27E-04</td>
<td>2.82E-04</td>
<td>1.54E-04</td>
<td>7.27E-04</td>
<td>2.82E-04</td>
<td>1.54E-04</td>
<td>7.98E-05</td>
<td>3.02E-05</td>
<td>1.76E-05</td>
<td>3.96E-05</td>
<td>1.76E-05</td>
<td>7.14E-05</td>
</tr>
</tbody>
</table>

Table A.3: Test Scenarios Table for Joint Probabilities
Bibliography


[60] Inc Realtime Technologies. SimVista (Version 2.38), 2012. [Computer software].